Technische Universität München
Lehrstuhl für Datenverarbeitung

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Robotik und Mechatronik

A Flexible Approach to Close-Range 3-D Modeling

Klaus H. Strobl Diestro

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades

Doktor–Ingenieur (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. Wolfgang Kellerer
Prüfer der Dissertation:		
	1.	Univ.-Prof. Dr.-Ing. Klaus Diepold
	2.	Hon.-Prof. Dr.-Ing. Gerd Hirzinger
	3.	Prof. Andrew J. Davison, Ph.D. (Imperial College London, UK)

Die Dissertation wurde am 16.09.2013 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 06.06.2014 angenommen.

Acknowledgement

It is an enduring mystery of human race that there are people who willingly give you more than is given by you. During the more than ten years of research summarized in this doctoral dissertation, I felt honored in this respect by many noble people to whom I owe a debt of gratitude.

Research was mainly conducted during my employment at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR) in Oberpfaffenhofen, Germany. The worldwide recognition of the institute during its last 20 years is primarily due to its longtime director Prof. Dr.-Ing. Gerd Hirzinger. It is, I think, all the more remarkable when a man of his renown actively helps out younger students in spite of difficulties. I shall never forget the numerous conference calls with him and his assistants back in 2002, leading to my employment at DLR (I was absent due to an internship in Norway). I remember being a stubborn, perhaps unabashed, youngster discussing my optimal position in his institute. Later on I learned that, at that time, the institute's financial position was anything but buoyant (a 2-year ban of recruitment followed my hire at DLR). I would like to sincerely thank Prof. Dr.-Ing. Gerd Hirzinger for his then generous disposition, as well as for his support during these years consisting of trustful encouragement, continuous challenge, and his visionary ideas.

I am grateful to Prof. Dr.-Ing. Klaus Diepold from the Technische Universität München for honoring the academic lead of my doctorate, for fruitful discussions, and for reminding me *to use a bigger hammer if it won't fit.*

Prof. Andrew J. Davison from Imperial College London has been kind enough to assume the third examiner role of this thesis. I want to sincerely thank him also for his support during my stay in his group, which considerably influenced my recent research work. During those four months of 2009 I probably experienced my steepest learning curve since childhood.

Unfortunately, students often leave university with their only experience being that of attending lectures—missing the other side of university that is scientific work in academia. In 2000 I was lucky enough to encounter one of the most renowned chairs of automatic control, the Institute of Automatic Control Engineering at the Technische Universität München, led by Prof. Dr.-Ing. Dr.-Ing. h.c. Günther Schmidt. I want to sincerely thank him and his distinguished student Dr.-Ing. Javier F. Seara for teaching me how to conduct scientific work of excellence. Furthermore, I want to thank Javier for being a loyal, decent friend ever since (despite all the ups and downs) and for his multi-sector savvy.

If there is a sole reason for the excellence of the Institute of Robotics and Mechatronics, it is for the widespread readiness for cooperation. I had the privilege of joining the 3-D modeling group formed by Eric Wahl, Michael Suppa, Tim Bodenmüller, and Wolfgang Sepp. Thank you guys for your support and for your early understanding that "*aller Anfang ist schwer*." I would like to gratefully acknowledge the fine work of Elmar Mair in the same context. Such a sympathetic working environment naturally facilitates stronger friendships; I would like to mention Thomas Wimböck and Paolo Robuffo Giordano, and thank them for their kindness at work and beyond. Special thanks go to my longstanding office colleague Friedrich Lange for being my tutor at DLR. A big thank you to my absolutely extraordinary students Cristian Paredes and Michal Smíšek. To all other friends and colleagues at DLR not yet mentioned, I owe you *one* coffee. Thank you as well to my proofreader Roger W. Ward. I would also like to give thanks to my fantastic lab mates back in Albion for the many happy memories (Margarita, Hatice, Emma, Adrien, Ankur, Antonis, Jeroen, Gerardo, Hauke, Kostas, Ogi, Richard, Stavros, Stefan, and Steve). I give profuse thanks to Sophie for her friendship and for the putto statue we removed from her father's mansion in the name of science (see Fig. 5.17 (a)). Very special thanks to my friends in Spain, *Champi* and *Rufas*, for reminding me where I belong.

It is your closest relatives that mark out your life and, more importantly, share the journey with you. I want to express my deepest gratitude to Julia for her love, guidance, and patience, particularly while I have been writing up. Thank you for being yourself! I would like to dedicate this dissertation to her, to my parents M^{a} Carmen and Wolf, who instilled the love of science in me (among other things), to my brother Wolfi, who led the way to go abroad, and to my grandparents *Yaya*, *Yayo*, *Oma*, and *Vati*, who taught me all that ordinary things that really matter. They have always looked out for me, setting the foundation of this achievement.

Oberpfaffenhofen, August 2013 Klaus Strobl

∼ *For my Family and the Love of my Life Julia* ∼

Abstract

Service robotics has the potential to become a major socio-technological and industrial achievement. An essential aspect of this technology is the degree of autonomy featured by the robotic agent, such as its capacity to make informed decisions. It is clear that isolated robots in unknown environments are highly dependent on perception to promote their degree of autonomy. This thesis focuses on visual perception of the geometry and the appearance of the scene.

Visual perception is the process by which visual sensory information about the environment is received and interpreted. This definition leaves aside the sort of sensory information used; for example, it does not necessarily mandate a geometric 3-D model of the scene. It is believed, however, that it is through the explicit formation of 3-D models that a considerable number of the remaining challenges on visual perception eventually will be solved.

When devising perception systems for service robotics, the consideration of cost, size, and weight of the sensors is of primary importance, as are their flexibility of use and the nature of the information provided. The development of sensors compliant with all these needs is, however, rare; more often than not, technical advances in isolated areas focus researchers on high performance, specialized sensors that may not observe all of the former requirements. Though promising at first, these systems face severe limitations when deployed in service robotics applications; hence they will not likely have long term success. In contrast to these efficient solutions, this thesis advocates effective perception systems that are inherently consistent with the requirements of service robotics.

This thesis presents the algorithms required for the production of an effective, multisensory hand-held 3-D modeling system, the DLR 3D-Modeler. Critically, it is not only the sensors within the perception system that have to comply with the guidelines, but also the methods required to arrange the sensors in the first place, and to make them work. In this spirit, lightweight, flexible, and highly-accurate sensor models, as well as their novel calibration methods, are presented. In addition, the robust and efficient processing of raw sensor data that might be compromised is also addressed.

Another contribution, to promote autonomy during its operation, turned the DLR 3D-Modeler into a worldwide novelty. Due to object self-occlusion, object size, or limited field of view, it is often impossible to acquire a complete 3-D model in a single measurement step. It is common for 3-D modeling devices to revert to external tracking systems in order to represent data in a common reference frame. This option is inconvenient as external systems are the largest and most expensive part of the system. In this work the DLR 3D-Modeler is extended to passive visual pose tracking, yielding the first hand-held 3-D modeling device for close-range applications that localizes itself passively from its own images in realtime, at a high data rate.

The system is applied to a number of scenarios in robotics and beyond. This low-cost system pushes traditional 3-D modeling forward to conquer new frontiers owing to its flexibility, passivity, and accuracy.

Zusammenfassung

Serviceroboter haben das Potenzial, große Bedeutung im sozialen, technologischen und industriellen Bereich zu erlangen. Ein wesentlicher Aspekt dieser Technologie ist der Grad an Autonomie, den der robotische Agent besitzt, wie auch seine Fähigkeit, fundierte Entscheidungen zu treffen. Es ist klar, dass die Autonomie isolierter Roboter in unbekannten Umgebungen sehr von der Perzeption abhängt. Diese Arbeit beschäftigt sich mit der visuellen Wahrnehmung der Geometrie und des Aussehens der Szene.

Visuelle Wahrnehmung ist der Prozess, durch den visuelle Information über die Umgebung erhalten und interpretiert wird. Diese Definition verzichtet auf die Art der sensoriellen Information. Beispielsweise setzt sie kein 3-D geometrisches Modell der Szene voraus. Es wird jedoch angenommen, dass eine wesentliche Zahl an Herausforderungen auf dem Gebiet der visuellen Perzeption schließlich durch die explizite Formulierung von 3-D Modellen gelöst wird.

Bei der Entwicklung von Perzeptionssystemen für Serviceroboter sind Kosten, Größe und Gewicht der Sensoren genauso wichtig wie ihre vielseitige Anwendbarkeit und die Art der gelieferten Information. Es werden allerdings kaum Sensoren entwickelt, die all diesen Anforderungen genügen. Meistens werden spezialisierte Sensoren verwendet, die aber oft nicht die anderen Anforderungen erfüllen. Obwohl solche Systeme am Anfang vielversprechend sind, erreichen sie ihre Grenzen, wenn sie in robotischen Serviceanwendungen eingesetzt werden. Somit werden sie keinen langfristigen Erfolg haben. Im Gegensatz zu effizienten Lösungen plädiert diese Arbeit für wirksame Wahrnehmungssysteme, die inhärent zu den Anforderungen der Servicerobotik passen.

Diese Arbeit zeigt die Algorithmen, die zum Bau von wirksamen multisensoriellen handgeführten 3D-Modellierungssystemen nötig sind, wie dem DLR 3D-Modellierer. Genau genommen müssen nicht nur die Sensoren in den Wahrnehmungssystemen den Richtlinien entsprechend, sondern auch die Methoden, die nötig sind, um die Sensoren geeignet anzuordnen und zu betreiben. In diesem Geist werden vielseitige und hochgenaue Modelle für Leichtbau-Sensoren präsentiert, zusammen mit ihren Kalibrierungsmethoden. Zusätzlich wird eine robuste und effiziente Verarbeitung von u. U. gestörten Rohdaten betrachtet.

Ein anderer Beitrag zur Förderung von Autonomie im Betrieb machte den DLR 3D-Modellierer zu einer weltweiten Neuheit. Wegen Selbstverdeckung, Objektgröße oder begrenztem Blickwinkel ist es oft nicht möglich, ein 3-D Modell innerhalb von einem Messschritt zu bekommen. Es ist bei Geräten zur 3-D Modellierung üblich, sich auf externe Trackingsysteme zu beziehen, um die Daten in einem gemeinsamen Referenzsystem darzustellen. Dies ist insofern unpraktisch, als externe Systeme die größten und teuersten Teile der Anlage sind. In dieser Arbeit wird der DLR 3D-Modellierer erweitert, um passiv die sichtbare Lage zu verfolgen, wodurch das erste handgeführte 3-D Modellierungsgerät für den Nahbereich entsteht, das sich selbst passiv und in Echtzeit mit hoher Datenrate aus den eigenen Bildern lokalisiert.

Das System wird u. a. in einigen Szenarien der Robotik angewendet. Aufgrund seiner Vielseitigkeit, Passivität und Genauigkeit, schiebt dieses low-cost System herkömmliche 3-D Modellierung an, neue Gebiete zu erobern.

Contents

List of Symbols

General

DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.
ESA	European Space Agency
CNES	Centre national d'ètudes spatiales
NASA	National Aeronautics and Space Administration

Robotics, computer science, and computer vision

1-D	One-dimensional
2-D	Two-dimensional
2.5-D	Depth image
3-D	Three-dimensional
6-D	Six-dimensional
AM	Active matching
AOV	Angular field of view
BA	Bundle adjustment
CCD	Charge-coupled device
CMOS	Complementary metal-oxide-semiconductor
CMM	Coordinate measuring machine
CRS	Compressed row storage
CT	Computed tomography
CPU	Central processing unit
DLT	Direct linear transformation
DoF	Degrees of freedom
DoG	Difference of Gaussians
EKF	Extended Kalman filter
FOV	Field of view
FPGA	Field programmable gate array
GPGPU	General-purpose GPU
GPU	Graphics processing unit
GUI	Graphical user interface
HazCam	Hazard avoidance camera system
HSL	Huew-saturation-lightness color space
ICP	Iterative closest point algorithm
IMU	Inertial measurement unit
IPP	Integrated performance primitives

IR	Infrared
kBA	Keyframe-based BA
KLT	Kanade-Lucas-Tomasi
LADAR	Laser detection and ranging
LBR	Leichtbauroboter
LIDAR	Light detection and ranging
LRS	Laser range scanner
LSP	Light stripe profiler
LUT	Look-up table
LWR	Lightweight robot
MI	Mutual information
ML	Maximum likelihood
MIS	Minimally invasive surgery
MoG	Mixture of Gaussians
MRI	Magnetic resonance imaging
NavCam	Navigation camera system
OdoCam	Odometry camera system
P3P	Three point perspective pose estimation problem
PanCam	Panoramic camera system
pdf	Probability density function
pel	Picture element
PF	Particle filter
PNG	Portable network graphics
PSD	Position sensitive device
PTAM	Parallel tracking and mapping
RADAR	Radio detection and ranging
RANSAC	Random sample consensus
RGB	Red-green-blue color space
RGB-D	Red-green-blue and depth
RMS	Root mean squares
RoMo	RoboMobil
RVGPS	Robust V-GPS
SAD	Sum of absolute differences
SCS	Stereo camera system
sel	Sensing element
SGM	Semiglobal matching
SLAM	Simultaneous localization and mapping
SVD	Singular value decomposition
SVGA	Super VGA
TCP	Tool center point
ToF	Time-of-flight
VGA	Video graphics array
V-GPS	Visual GPS
V-SLAM	Visual SLAM

Algebraic Conventions

Scalars are denoted by upper or lower case letters in italic type. Vectors are denoted by lower case letters in boldface type, as the vector $\boldsymbol{x}$ composed of scalar components x_i. Matrices are denoted by upper case letters in boldface type, as the matrix $\boldsymbol{M}$ composed of elements M_{ij} (ith row, jth column).

x	Scalar
$\boldsymbol{x}$	Vector
$(\cdot) \cdot (\cdot)$	Dot product or scalar product
$(\cdot) \times (\cdot)$	Cross product or vector product
$\boldsymbol{M}_{(i \times j)}$	Matrix (size $i \times j$)
$\det(\boldsymbol{M})$	Determinant of a square matrix $\boldsymbol{M}$
$\mathrm{trace}(\boldsymbol{M})$	Trace of a square matrix $\boldsymbol{M}$
$\boldsymbol{M}^{-1}$	Inverse of a matrix $\boldsymbol{M}$
$\cdot^{\mathsf{T}}$	Transposition operator
$f(\cdot)$	Scalar function
$\boldsymbol{f}(\cdot)$	Vector-valued function
$\sin(\cdot)$	Sine
$\cos(\cdot)$	Cosine
$\tan(\cdot)$	Tangent
$\cot(\cdot)$	Cotangent
$\arcsin(\cdot)$	Arcsine
$\arccos(\cdot)$	Arccosine
$\arctan(\cdot)$	Arctangent
$\arg\max(\cdot)$	Argument of maximum of an expression
$\arg\min(\cdot)$	Argument of minimum of an expression
$\exp(\cdot)$	Exponential function
$\lvert \cdot \rvert$	Norm of scalar
$\mathrm{norm}(\cdot)$ or $\lVert\cdot\rVert$	Norm of vector
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
$\mathbb{N}_1$	Set of natural numbers excluding 0
$\mathbb{Z}$	Set of integers
$\boldsymbol{q}$	Unit quaternion
$\bar{\boldsymbol{q}}$	Conjugate unit quaternion
$\check{\boldsymbol{q}}$	Unit dual quaternion
$\bar{\check{\boldsymbol{q}}}$	Conjugate unit dual quaternion
$(\cdot) * (\cdot)$	Quaternion multiplication
$[\boldsymbol{\omega}]_\times$	Skew-symmetric cross product matrix of vector $\boldsymbol{\omega}$

Reference Frames

S_0	3-D world reference frame
S_B	3-D robot base reference frame
S_C	3-D camera reference frame

S_{canvas}	2-D canvas reference frame
S_{I}	2-D image reference frame
S_{IMU}	3-D IMU reference frame
S_{LRS}	3-D LRS reference frame
S_{LSP}	3-D LSP reference frame
S_{M}	2-D computer memory frame
S_{RoMo}	3-D RoMo reference frame
S_{Rotor}	3-D rotor reference frame of the LRS
S_{T}	3-D robot TCP reference frame

Subcripts, Superscripts, and Mathematical Accent Marks

${}_{\mathrm{A}}\boldsymbol{p}$	Vector $\boldsymbol{p}$ in S_{A}
$\boldsymbol{p}_{\mathrm{d}}$	Distorted projection $\boldsymbol{p}$
$\boldsymbol{p}_{\mathrm{u}}$	Undistorted projection $\boldsymbol{p}$
$\bar{\boldsymbol{p}}$	Homogeneous representation of the vector $\boldsymbol{p}$
$\hat{\boldsymbol{p}}$	Estimated vector $\boldsymbol{p}$
$\tilde{\boldsymbol{p}}$	Measured vector $\boldsymbol{p}$
$\boldsymbol{\Omega}_{\star}$	Optimized $\boldsymbol{\Omega}$

Recurrent Symbols, Scalars, Vectors, Matrices, and Functions

α, β, γ	Intrinsic parameters of the pinhole camera model
δ	Image distortion
δ_{d}	Decentering image distortion
δ_{r}	Radial image distortion
δ_{t}	Thin prism image distortion
κ	Scaling factor of the calibration target
λ	Relative angle between ${}_{\mathrm{M}}x$ and ${}_{\mathrm{M}}y$
ν	Aspect ratio of the calibration target
$\boldsymbol{\Omega}$	Optimization parameters
$\boldsymbol{\omega}_{\infty}$	Absolute conic $\boldsymbol{\omega}_{\infty} = \boldsymbol{A}^{-\mathsf{T}}\boldsymbol{A}^{-1}$
ρ	Radial distance of a projection in S_{I}
$\sigma^2_{\angle}$	Flatness of the reconstructed plane
σ^2_{rot}, σ^2_{tra}	Second central moments in rotation and translation error
ϑ	In-plane angular direction
ϑ_0	In-plane angular direction of max. tangential decentering distortion
ϑ_1	In-plane angular direction of max. tangential thin prism distortion
$\Phi(\cdot, \kappa)$	Scaling function according to κ
$\boldsymbol{\Upsilon}$	Camera system model
$\mathbf{0}$	Void matrix
$\boldsymbol{A}$	Intrinsic matrix of the pinhole camera model
$\boldsymbol{AX} = \boldsymbol{XB}$	First formulation of the hand-eye problem
$\boldsymbol{AX} = \boldsymbol{ZB}$	Second formulation of the hand-eye problem

$\mathcal{C}_i$	Camera $\#i$
d→u	Distorted-to-undistorted
f	Focal length
$\boldsymbol{H}$	Homography
${}_c\boldsymbol{H}_\infty$	Infinite homography for $\mathcal{C}_c$ $({}_c\boldsymbol{H}_\infty = \boldsymbol{A}\,{}_\mathrm{C}\boldsymbol{R}^{\mathrm{C}_c}{}_c\boldsymbol{A}^{-1} \;\Leftrightarrow\; \boldsymbol{\omega}_\infty = {}_c\boldsymbol{H}_\infty^{\mathsf{T}}\,{}_c\boldsymbol{\omega}_\infty\,{}_c\boldsymbol{H}_\infty)$
$\boldsymbol{I}$	Identity matrix
$\mathcal{I}_i$	Image $\#i$
k_1, k_2, k_3	Coefficients of radial distortion
$\mathcal{O}^{\mathrm{rot}}$	Rotational error metric in the context of hand-eye calibration
$\mathcal{O}^{\mathrm{tra}}$	Translational error metric in the context of hand-eye calibration
$\boldsymbol{p}$	Point, feature, or projection—either in 2-D or in 3-D
$\check{\boldsymbol{p}}$	Erroneously measured control point $\boldsymbol{p}$
p_1, p_2, p_3, p_4	Coefficients of decentering distortion
ΔP, ΔY, ΔR	Differential rotations in *pitch*, *yaw*, and *roll* angles
$\boldsymbol{P}$	Perspective projection matrix
$\mathrm{proj}(\cdot)$	Projection function
$\boldsymbol{R}$	Rotation matrix
$\boldsymbol{R}_\mathrm{x}$, $\boldsymbol{R}_\mathrm{y}$, $\boldsymbol{R}_\mathrm{z}$	Rotation matrix around the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ axes
$\boldsymbol{R}_\mathrm{p}$, $\boldsymbol{R}_\mathrm{t}$, $\boldsymbol{R}_\mathrm{r}$	Rotation matrix around the *pan*, *tilt*, and *roll* axes
${}_\mathrm{A}\boldsymbol{R}^\mathrm{B}$	Rotation matrix between reference frames S_A and S_B
$\{\mathcal{R}, \mathcal{G}, \mathcal{B}\}$	Red, green, and blue image components
s_1, s_2	Coefficients of thinprism distortion
s_x, s_y	Side sizes of *sel*s
$SE(3)$	Euclidean group of rigid body transformations $\{\boldsymbol{R}, \boldsymbol{t}\}$ where $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{t} \in \mathbb{R}^3$
$SO(3)$	Rotation group for 3-D space or special orthogonal group of 3×3 matrices $\boldsymbol{R} \in \mathbb{R}^{3\times3}$: $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{R} = \boldsymbol{I}$, $\det(\boldsymbol{R}) = 1$
$\boldsymbol{t}$	Translation vector $\boldsymbol{t} \in \mathbb{R}^3$
${}_\mathrm{A}\boldsymbol{T}^\mathrm{B}$	Homogeneous transformation matrix from reference frame S_A to reference frame S_B
${}_0\boldsymbol{T}^\mathrm{B}$	World-to-base homogeneous transformation matrix
${}_0\boldsymbol{T}^\mathrm{C}$	World-to-camera homogeneous transformation matrix
${}_\mathrm{B}\boldsymbol{T}^\mathrm{T}$	Base-to-TCP homogeneous transformation matrix
${}_\mathrm{T}\boldsymbol{T}^\mathrm{C}$	TCP-to-camera homogeneous transformation matrix (or hand-eye transformation)
u_0, v_0	2-D coordinates of the principal point of a camera in S_M
u_x, u_y	Unit lengths of the calibration target
u→d	Undistorted-to-distorted
$\mathrm{undist}(\cdot)$	Undistortion function
${}_\mathrm{A}\boldsymbol{x}$, ${}_\mathrm{A}\boldsymbol{y}$, ${}_\mathrm{A}\boldsymbol{z}$	Main axes of the 3-D reference frame S_A

List of Figures

List of Tables

"First he will see the shadows best, next the reflections of men
and other objects in the water, and then the objects themselves;
then he will gaze upon the light of the moon and the stars
and the spangled heaven. Last of all he will be able to see the sun."

—Plato, The Republic (Book VII, 516-A to 516-C)

1 Introduction

Imagine being a prisoner held chained in a cave ever since your childhood. Your legs and neck are fixed to gaze at a wall, as are the ones of your fellow prisoners. A fire on your back casts shadows of passing men, objects, and animals unto that wall. Since you are unaware that they are shadows, you will certainly take them for reality. It is safe to say that a culture of shadow projections will develop within the prisoners' society, e.g. by predicting the size of the shadows, their chronological order, or their interactions. Whoever claims that he knows about the real nature of these shadows will surely go unregarded as distinguished from the fellow prisoners that are expert on shadow projections.

Imagine further that you are freed from your chains and permitted to explore your surroundings. You would not recognize real objects, for the only objects that you hold to be real are still the shadows on the wall. Furthermore, you naturally would be struck blind by a look at the fire, and prefer to gaze back at the familiar shadows. You are loath to leave the cave. Imagine being forced to leave the cave, to see the world and to look directly at the sun—your eyes would burn with searing pain. Of course, after a short time on the surface you acclimate and learn. You will understand the role of sunlight getting reflected on objects. Now you are a philosopher and would consider yourself lucky and your fellow prisoners pitiful. Conversely, the prisoners will find you stupid as you are no longer accustomed to the darkness—you will be bad at their silly game of shadows.

The above tale is Plato's allegory of the cave as regularly narrated in high school. It is alleged to explain that education ought to be the object of the human race (the sun is our illumination), and that it is the purpose of philosophers to achieve the best education and to explain the real world to us.

May I mention my own interpretation? I think that the whole allegory is pointless if it is only about the superiority of achieving wisdom about the ultimate truth. An impossibly smart prisoner, for instance, could come up with the idea that they were being played to only see part of the reality (as humans generally believe by their exercise of religion). I rather interpret the allegory as an early allusion to the basics of cybernetics. Cybernetics says that perception intercedes between reality and ourselves. Feedback helps us to know reality and obtain wisdom, and even to achieve goals. It furthermore claims that this allows us to take action[1] on our own behavior in order to reinforce the positive feedback process, *i.e.*, in order to actively enhance feedback. In other words, it is by the acquisition of **better, more explicit evidence** on the real world that the average prisoner will invariably improve their understanding of the world.

A similar trend is presently taking place in computer vision (visual perception as performed by computers). Visual perception is the process by which visual sensory information about the environment is received and interpreted. This definition leaves aside the sort of sensory information used, *i.e.*, it does not necessarily mandate to reveal the geometric 3-D model of the scene. It is believed, however, that it is through the **explicit formation of 3-D models** that a considerable number of the remaining challenges on visual perception will be eventually solved. Note that Plato's allegory plays into our own hands as he suggests explicit, 3-D geometry for eventual understanding of the world, as opposed to traditional 2-D projections of it (*i.e.*, 2-D image processing).

In the end, the computerization of perception (*i.e.*, computer vision), as well as its potential active guidance by a robot itself, ought to bring robots to markets outside industry, like service robotics and transportation systems where a high degree of autonomy is desired. Indeed, perception loops (*i.e.*, to actively adjust our own behavior to follow a purpose) can be considered the hallmark of human intelligence, which arguably is what makes us human. Note that perception loops in humans also extend in time (memories) and also embrace other humans (social intelligence).

In this thesis I will introduce novel key technologies for the development of more useful 3-D modeling devices and beyond. In fact, this thesis describes the main algorithms for the generation of accurate 3-D pointclouds using a novel 3-D modeling device devised at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR): the DLR 3D-Modeler.

1.1 Motivation

Scientists strive to maximize the *immediate* performance improvement in their particular fields of expertise. This **maximum efficiency paradigm** achieves significant improvements in a short period of time and leads to cutting-edge technologies and highly specialized devices. Ambitious technological goals, however, like those enabling groundbreaking new industries like service robotics, invariably call for a wide range of technologies—these often turn out to be mutually restricting. Furthermore those higher goals may impose fundamental constraints (in costs, size, weight, performance etc.) that may not be being observed by the technologies originated following that paradigm in the first place.

[1] The term cybernetics comes from the Greek word κυβερνέτες (kubernētēs) for "to steer" (eventually in Latin "gubernare" and in English "to govern").

In contrast to efficiency, the focus of *effectiveness* is the achievement as such. Effective research calls for *foresight*, possibly with the cost of reduced initial performance, but often resulting in a prolonged development. In this work I claim that reaching ambitious socio-technological or industrial goals, such as achieving critical mass in the service robotics industry, definitely requires a different approach to maximum efficiency, namely a purely maximum effectiveness approach to that goal. I term this the **maximum effectiveness paradigm**. I focus on visual perception for service robotics, specifically on the realization of perception systems in compliance with the maximum effectiveness paradigm.

Visual perception is the process by which visual sensory information about the environment is received and interpreted, and it is key to achieving truly autonomous robots. Visual perception does not necessarily reveal a geometric 3-D model of the scene; it is believed, however, that the formation of 3-D models is essential to solve a considerable number of the remaining challenges on visual perception. For many years, specialists have pushed diligently forward on 3-D modeling quality following the maximum *efficiency* paradigm, which has proved beneficial in many areas. This, however, critically misled roboticists into using the very same sensors. For instance, many robotics labs around the world invested in armies of tiny service robots carrying around massive laser range scanners. These may deliver accurate, robust data, but are clearly *not* the kind of sensors (and they do not provide the sensory information) that will ultimately push service robotics to overarching success. Furthermore, these approaches provably diverted research efforts away from more foresighted approaches. In recent years, research on vision-based, real-time simultaneous localization and mapping (visual SLAM) using, for example, only one camera, is a much-anticipated break from previous approaches—despite some initial loss in accuracy. The latter approach using monocular vision does comply with the maximum effectiveness paradigm and will be predominantly adopted throughout this work.

1.2 Problem Statement

Due to its intensive dependency on computer vision, I believe that service robotics will only achieve critical mass and become widespread if perception systems are realized following the maximum effectiveness paradigm. The following **guidelines** apply:

- Holistic design (diversity of data types and sensors)
- Large amounts of information
- Avoidance of moving parts
- Passive operation with respect to (w.r.t.) the environment
- Modularity
- Operation autonomy
- Flexibility

- Small size
- Light weight
- Low cost
- Low consumption
- High data rate
- “Softwarization” of hardware

The last guideline is inspired by Andreessen’s article in The Wall Street Journal (Andreessen, 2011). The author addresses the disruptive power of softwarization, focusing on present industrial developments that, when taken over by software, end up experiencing the typical exponential growth of software solutions. In our case of perception for e.g. service robotics it is not about exponentially growing the number of sensors (pervasive sensing) but about exponentially decreasing the computational cost of processing sensory information—compared to established hardware solutions that are physically limited to operate at certain rates. In addition, by decreasing the cost of sensor data computation, we will be able to process more data.

Due to the fact that the availability of computing power and digital storage is growing exponentially, while their cost and required power is decreasing, it follows that virtually any successful substitution of established hardware interfaces by software algorithms results in dramatic performance increases and, on top of that, dispenses with inconvenient hardware.

In addition, the potential paradigm shift into “softwarization” of hardware would suit robotics applications just fine due to the present shift of software into multithreaded applications. A massive relocation of hardware efforts into software would, for instance, allow for natural solutions to the problem of concurrency that is typical in robotic systems. Multisensory data, potentially at different data rates, have to be synchronized and fused. In addition, in the context of 3-D modeling, data potentially have to be reconstructed by stereo triangulation, meshed and textured online, while performing extensive feature matching as well as appearance-based recognition, and nonlinear optimizations can still be conducted in the background, on the same computer.

It is worth noting that the paradigm of “softwarization” of hardware readily supports compliance with other guidelines. Consider the following problem: Several factors, like object self-occlusion, object size, or limited field of view, make it impossible for a 3-D modeling system to acquire a complete model in a single measurement step; this is especially true at close range. Multiple views (or multiple sensors) are required to subsequently merge data to a single 3-D model. The prevalent approach is to measure the position and orientation (pose) of the sensor while acquiring range data, thereby registering multiple views, potentially in realtime. A range of tracking systems, robotic manipulators, passive arms, turntables, coordinate measuring machines (CMMs), or electromagnetic devices are commonly deployed for this purpose. These options are inconvenient for three reasons: First, they limit the user’s mobility;

second, they require accurate synchronization and extrinsic calibration, which are cumbersome, error-prone processes (Bodenmüller *et al.*, 2007; Strobl and Hirzinger, 2006), and, what is more, they cannot be rearranged; last, it turns out that external positioning systems almost always represent the largest and most expensive part of the 3-D modeling system. In this work I opt for the "softwarization" of pose tracking systems, presenting the required algorithms for robust and accurate pose tracking of close-range 3-D modeling devices at high data rate, by the use of the images captured by their own internal cameras; cameras are already present in most of these devices after all. In this way, the three limitations mentioned above are lifted.

Similarly, the "softwarization" paradigm led to unprecedented approaches, like the removal of optical filters from the cameras of the DLR 3D-Modeler. By doing so, the detection of laser light projections on images may have been aggravated, but it renders other tasks possible, like the abovementioned visual pose tracking of the sensor, live augmentation of 3-D models, or even texturing of these models.

A further example of "softwarization" is the online meshing algorithm for real-time representation of the 3-D model (potentially in augmented reality) by Tim Bodenmüller in (Bodenmüller, 2009). It is by this computationally expensive task that the user is able to timely finish the scanning procedure without occupying the expensive scanner for longer periods of time.

Last but not least, **a last guideline** to realize effective perception systems is to provide services to the research community, like the standardization and sharing of useful software as well as the publication of novel ideas, which are essential for timely and widespread distribution of technologies.

Going back to the abovementioned example on SLAM, the historic use of laser range scanners did not fulfill any of these criteria. Today, the use of digital cameras instead of scanners is the most attractive option as they meet all of the above criteria.

1.3 Selected Approach

Indeed, the abovementioned guidelines led to the development of the DLR 3D-Modeler:

- **Multisensory capability** enables holistic sensing thereby reducing costs. Modular design enables its flexible deployment in varied scenarios.

- **The extensive use of digital cameras** produces plenty of visual information, avoids moving parts, supports light-weight design with a small footprint, and allows for passive operation w.r.t. the scene. They are affordable and consume less energy, allowing for a very accurate parametrization of its simple operating model. They can gather a plethora of information (including radiometric and geometric) within a single, rapid measurement. In addition, all image-based sensing becomes inherently calibrated and synchronized.

- **Exact modeling of sensors**, together with their **accurate parametrization**, yield complete, non-redundant sensor models that will eventually enable highly accurate measurements.

- **Robust operation of the sensors** embodies the guideline of "softwarization" of sensors—software as the other half of sensors. For instance, inconvenient laser filters on the camera can be removed by improved image processing, which allows for additional features like model texturing and:

- **accurate, visual (passive) pose tracking estimation**, which is the most flexible option for data registration, providing flexibility and autonomy during operation, as well as dramatically driving down the price of the sensor.

- **Development of a congruent software suite** that has been (in part) freely released; I am the main author of the camera calibration toolbox DLR CalDe and DLR CalLab (Strobl *et al.*, 2005); we also **publicated all novel ideas** that we came up with during the development of the DLR 3D-Modeler.

In the end, we produced a self-contained, hand-guided 3-D modeling system for close- and medium-range applications that presents advantages when measuring complex objects compared to industrial scanners which are inconvenient at that. Hand-guided scanners allow for natural scanning of areas similar to using a spray can. Of course, the sensor has to be of low weight to allow a convenient and acceptable guidance by the user, see Fig. 1.1. The motivation for these types of systems are twofold: for close-range applications themselves, and as complementary devices for large-scale, extensive 3-D modeling that usually requires different devices to fulfill complex tasks. Both options call for flexible, low-priced platforms. This work is a significant step in this direction.

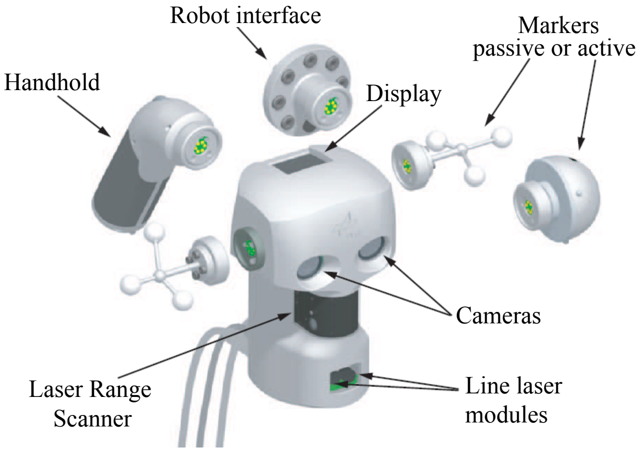


Figure 1.1: The DLR 3D-Modeler and its components.

The investigations and novelties undertaken within this project are similar to the recent developments in simultaneous localization and mapping (SLAM) in many respects. Even though the computational requirements of current SLAM approaches have increased, the advantages of using cameras instead of laser range scanners not only provides an unprecedented degree of flexibility and accuracy but also makes it possible to pursue new objectives like concurrent image understanding and augmented reality on live image streams.

Due to its flexibility, the DLR 3D-Modeler has been already deployed in **many applications**:

- Hand-guided modeling device
- Robot work cell autonomous exploration
- Perception system of the humanoid robot “Justin”
- 3-D modeling from aerial images
- Hazard avoidance camera system (HazCam) for the ExoMars rover of the European Space Agency (ESA)
- Sensing unit for automated mounting of car wheels (concepts and algorithms)
- Patient registration in medical preoperative planning
- Supporting system for the kinematic calibration of robots

These fields of direct application of the DLR 3D-Modeler will be revisited in Appendix B together with other indirect applications of individual methods developed in the context of this thesis.

1.4 State of the Art of 3-D Modeling Systems

1.4.1 A Taxonomy of 3-D Modeling Systems

A plethora of sensors exists that can obtain 3-D geometric information about the scene. Most options are listed in Table 1.1 with regard to their principles of operation. Other representations are possible e.g. regarding the sort or the amount of information delivered, but the division featured in Table 1.1 is more convenient to my purposes. The design guidelines mentioned in the above sections rule out both, contact sensors using single-point probes (because of their physical interaction with the scene and because of their slowness), and transmissive sensors (because of their spatial constraints, lack of accuracy, and potential harm to human operators).

During the design of the DLR 3D-Modeler we concentrated on optical, reflective methods for accurate geometric and photometric acquisition of surface-related information. Non-optical reflective methods like sonar and radar may

3-D sensors	contact	mechanical		passive arm + probe
				coordinate measuring machines (CMM)
				attached ultrasonic tracker
		inertial		attached inertial meas. unit (IMU)
		electromagnetic		attached electromagnetic tracker
	transmissive			industrial computed tomography (CT) using X-rays
				magnetic resonance imaging (MRI)
				ultrasound
	reflective	optical	active	structured light
				active stereo (interferometry)
				depth from defocus
				laser triangulation
				non-scanning LIDAR, ToF
			passive	stereopsis
				shape-from-X (shading, motion, focus, etc.)
		non-optical (active)		sonar
				radar

Table 1.1: Taxonomy of 3-D sensors by operation principle (adapted from (Curless, 1997)).

provide fair range information in an inexpensive way, but they are especially inaccurate in their bearings information, *i.e.*, in the direction of the measurements. On the other hand, optical sensors typically provide highly accurate bearings information; the estimated distance to the surface, *i.e.*, its range, can also be estimated with high accuracy depending on the computational ability of the implemented method as well as on their valid calibration, cf. (Chen *et al.*, 2000; Blais, 2004). Passive, optical reflective methods are especially convenient because they do not actively project light unto the object. Active, optical reflective methods, however, tend to be more accurate e.g. when using laser light because it is very well collimated, which allows it to define small details.

Coded structured light techniques project varying patterns, either by time-multiplexing, spatial neighborhood coding, or direct coding (Salvi *et al.*, 2004). The varying patterns readily enable pixel correspondence between a separate camera featuring a parallax to the beamer and the beamer so that depth information can be obtained by triangulation. Overall, this method allows for fair 2.5-D images, *i.e.*, 2-D images including range data for every 2-D coordinate.

Laser triangulation methods, like the laser stripe profiler (or slit scanner), are widely used because of their simplicity, potential accuracy, and lower cost. Their immunity to ambient light is poor, however, as the sensor captures a bigger part of the scene since the object geometry is unknown. For that reason, narrow-band laser optical filters are regularly placed in front of the camera to filter out non-laser light.

Time-of-flight (ToF) cameras operate by measuring the time of flight of infrared light or laser light by pulse, frequency, or amplitude modulation. They are convenient in medium range because their accuracy is relatively constant over an extended operating range. For close-range applications like 3-D modeling, however, their accuracy is insufficient.

1.4.2 Optical, Reflective 3-D Sensors

The most prominent commercial 3-D modeling systems are indeed within the family of optical, reflective 3-D sensors:

- The SICK LMS 200 by SICK AG is a 2-D laser range scanner extensively used in combination with navigation tasks.

- Z+F IMAGER® by Zoller&Fröhlich GmbH extends data gathering to full 3-D around the sensor, at the cost of dropping real-time capability.

- The Velodyne Lidar™ HDL-64E by Velodyne Lidar, Inc. is similar to the Z+F sensor as it also delivers 3-D data; despite of its higher price and big size, its high data rate output (5-15 Hz) made it very popular for traffic challenges like the DARPA grand challenges 2005 and 2007.

- It is worth mentioning the stereo vision solutions of Point Grey Research, Inc., as they have been extensively used by roboticists and the computer vision community in general.

- Time-of-Flight (ToF) cameras are non-scanning LIDARs (imaging radars). They produce depth images at medium range by active, time-of-flight signal processing. Noteworthy products are by pmdtechnologies GmbH and the SwissRanger™ by CSEM SA.

- The Kinect 3-D sensor by Microsoft Corporation and PrimeSense has been ground-breaking due to its accurate, high data rate output and its low price. The sensor capabilities are comparable to the ones of ToF cameras at much lower cost and with better resolution and precision. The sensor, however, is limited by its active projection of an infrared pattern so that outdoor operation is compromised.

- The most widespread type of 3-D sensors for 3-D modeling are laser stripe profiler units attached to passive robotic manipulators by companies like FARO Technologies Inc., KREON Technologies, RSI GmbH, Metris NV, and ShapeGrabber Inc. Their products excel in reconstruction accuracy. The use or passive arms is, however, inconvenient to manual operation of the sensor (refer to Chapter 5).

- In industry, static sensors are usually more convenient than moving sensors. Isra Vision AG manufactures static 3-D sensors by multiline triangulation.

- There has been a trend towards more mobility in recent years. For example, laser triangulation heads tracked by attached electromagnetic devices by Polhemus Inc., as well as visual pose tracking by Creaform Inc. The latter option is, however, active w.r.t. the scene as it requires adhesive markers and projects infrared illumination. This type of sensor will be addressed in more detail in the next section and in Section 5.2.

Commercial and research scanners are largely dedicated to a single task. They are therefore not convenient for robust, multisensory deployment e.g. for applied research. Many applications like 3-D modeling, scene understanding, navigation, or exploration usually have conflicting requirements concerning range, accuracy, acquisition speed, or illumination, so that multisensory devices are actually required. In addition, the abovementioned commercial sensor systems neither combine strengths of individual sensors nor are able to evade their weaknesses by fusion of their data. This idea motivated the integration of different sensor components into the DLR 3D-Modeler (Suppa *et al.*, 2007).

1.4.3 3-D Modeling Systems that Rely on Registration by Pose Tracking

It is often impossible to acquire a complete 3-D model in a single measurement step owing to e.g. object self-occlusion, object size, or limited field of view; this is especially true at close range. Multiple views (or multiple sensors) are regularly deployed in order to fuse their 2.5-D images into a registered 3-D model.

A straightforward option to register range images (2.5-D) is based on their own geometry. Depending on the acquired scene, however, this option may be precluded if the surfaces do not feature salient 3-D regions, or in the case of 1-D range data e.g. when using laser stripe profilers.

A widespread alternative for permanent registration of depth images in realtime is to externally track the pose of the modeling device so that range data can be directly represented in a common reference frame, in realtime and irrespective of the range data quality (Hilton and Illingworth, 2000). A range of pose tracking systems, robotic manipulators, turntables, or electromagnetic devices are commonly used for this purpose. These options are inconvenient for three reasons: First, they limit the mobility of the user; second, they require accurate synchronization and extrinsic calibration w.r.t. the range sensor; and third, they are (by far) the largest and most expensive part of the 3-D modeling system.

In Section 5.2 the dominant commercial 3-D modeling systems in this concern are reviewed; these systems either use inconvenient external reference systems, or opt for visual pose tracking relying on active illumination and adhesive markers on the scene. In that section is a list past research work on *passive* visual pose tracking, which did not, however, run in realtime. In (Strobl *et al.*, 2009a) I appointed the self-referenced DLR 3D-Modeler to amend that vacancy.

1.5 Contribution of the Thesis

By following the design guidelines in Section 1.2 the approach explained in Section 1.3 was utilized. Since the state of the art in Section 1.4 has proven inadequate to my intentions, a number of innovations have contributed to the research community as well as to internal projects at DLR. This thesis reports these contributions in four different topics.

The fundamental contributions due to the physical instantiation of the DLR 3D-Modeler are as follows: The variability of the multipurpose sensor head integrating different sensors is unparalleled both in the research community and in the commercial market (Suppa *et al.*, 2007). The dual, crosshair laser stripe profiler premiered in the DLR 3D-Modeler, also standing out due to its lack of optical filtering of laser light as will be reported in Section 2.2.2. The compact size and the utmost integration of the mechanical parts and electronics to be controlled by informatics (*i.e.*, mechatronics) is paragon of the excellence of the Institute of Robotics and Mechatronics. Its modularity, allowing for flexibly attaching further sensors (e.g. the inertial measurement unit) and different pose reference systems has been crucial to the many findings reported in this work and others. Most of these contributions resulted from teamwork within the Institute of Robotics and Mechatronics, hence will not be addressed in more detail in this thesis. A potential commercialization of the DLR 3D-Modeler is believed to yield the most favorably priced system in the market.

Next I categorize the contributions into four topics:

1.5.1 The Importance of Exact Modeling of Sensors

It is central to successful geometric computer vision to rely on accurate sensor models. I would like to lay stress on these two contributions within Chapter 2:

- Section 2.2.1 presents in-depth investigations to understand the underlying operational principles of digital cameras in order to substantiate the choice of compact models of general validity; these have to be specific to the camera used and will support their successful parametrization in Section 3.2.
- Section 2.2.2 produces a novel model of the dual, crosshair laser stripe profiler.

1.5.2 The Importance of Accurate Parametrization

Chapter 3 perhaps contains most contributions within this thesis. The main objectives of my contributions have been as follows:

1. To choose sound calibration methods. If maximum likelihood estimation on systems with Gaussian errors is intended, the chosen method has to optimize parameters by minimizing *the actual* residual errors if we really want to achieve highest accuracy.

2. To choose a calibration method that is simple in order to minimize the risks of potential mistakes committed by the user.

3. To define how to gather data that are valid to the calibration method chosen.

These are the novel contributions in Chapter 3:

- In 2006 I introduced a novel method for optimal extrinsic calibration of cameras (also called hand-eye calibration) in (Strobl and Hirzinger, 2006). The method is currently being referenced as state-of-the-art. In detail, the method optimizes the hand-eye and object-base rigid body transformations by the joint minimization of translational and rotational residual errors in the kinematic chain of the reference system (e.g. the robotic manipulator). It is worth noting that translational and rotational errors are being weighted by a device-specific parameter that can be automatically estimated during the same optimization. Within the same work, we additionally extended the well-known approach in (Horaud and Dornaika, 1995) for better performance (refer to Section 3.3).

- In Section 3.4 I note that the calibration object is rarely being specified as accurately as really expected by the intrinsic or extrinsic camera calibration algorithms. I present two novel methods to amend that shortcoming: The first one was originally introduced in 2008 in (Strobl and Hirzinger, 2008); it models the calibration object by two parameters, viz. its aspect ratio and its absolute scale, which can be estimated during intrinsic and extrinsic camera calibration respectively. The second, a more complicated method, was introduced in 2011 (Strobl and Hirzinger, 2011); it optimizes the *whole* geometry of the calibration object during intrinsic and extrinsic camera calibrations. Validation experiments demonstrate that these methods should be preferred to traditional camera calibration unless the user can provide the dimensions of the calibration object with highest accuracy.

- In Section 3.5 I introduce another calibration method for intrinsic and extrinsic calibration of cameras featuring a narrow angular field of view. It is difficult to obtain the required evidence on perspectivity in the calibration images using these cameras. That evidence is regularly being used by traditional methods to differentiate between perspectivity effects (due to the location of the camera) and the scaling of the camera itself. In the absence of this information, traditional calibration methods are compromised. In (Strobl *et al.*, 2009b) we devised a method for joint intrinsic and extrinsic calibration of this type of camera by sensibly using the pose readings of the attached pose reference system (e.g. a robotic manipulator).

- The traditional calibration methods together with all novel methods addressed above have been implemented in a calibration toolbox called DLR

CalDe and DLR CalLab (Strobl *et al.*, 2005) that is freely available worldwide (for academic purposes only). I am main author of DLR CalLab (the calibration part of the toolbox), and my colleagues Wolfgang Sepp and Stefan Fuchs developed DLR CalDe (the corner detection software). This contribution has allegedly been, to date, the most widespread contribution in the context of this thesis. It is safe to say that it is ranked in the top three among the freely-available camera calibration toolboxes worldwide. Beyond learning my lessons on algorithmic and computer programming, I learned a lot about maintaining a software package for an active community of users.

- In Section 3.6 I shall present the calibration method of one of the main sensors within the DLR 3D-Modeler: the laser stripe profiler (LSP). This novel contribution was originally presented in (Strobl *et al.*, 2004). The method is based on a prior intrinsic and extrinsic calibration of its component camera(s). After that, the pose of the laser plane is estimated by the novel method. Instead of using precision calibration targets, the method merely consists in scanning a planar surface of unknown pose. The procedure is highly unlabored yet yields high accuracy. In Section 3.7 I present a variant of this method for the laser range scanner (LRS) of the DLR 3D-Modeler.

- In Section 3.9 I propose a concept for combined calibration of all component sensors of the DLR 3D-Modeler w.r.t. different external tracking systems. The approach has proven very useful for rapid deployment of the DLR 3D-Modeler as it is much faster than performing separate extrinsic calibration for all of its sensor components.

The diagram in Fig. 1.2 depicts the functional interaction between the calibration procedures of the main component sensors of the DLR 3D-Modeler. Starting out on the left-hand side, the intrinsic calibration of the (stereo) camera solely requires perspectively-distorted images of a checkerboard calibration pattern. If the novel methods in Section 3.4 are used, the dimensions of the corners on the calibration pattern need not be known with high precision. In the case of stereo vision, the absolute distance between two corners has to be provided, *unless* a subsequent hand-eye calibration is performed. Indeed, hand-eye calibration of the (stereo) camera usually takes place immediately after the intrinsic calibration stage. For that purpose we use the absolute extrinsics of the former intrinsic calibration (which are a by-product of that calibration), comparing them with the rigid body transformations delivered by the pose tracking system in question (e.g. a robotic manipulator like the FaroArm Gold, the Kuka KR 16 or the DLR Lightweight Robot III, or infrared tracking systems like the ARTtrack2) and minimize the resulting discrepancies. After that, the hand-eye transformation of the camera w.r.t. the **T**ool **C**enter **P**oint (TCP) frame S_T of the pose tracking device is used for the calibration of the laser coordinate frames of both, the laser stripe profiler (LSP), and the laser range scanner (LRS) in Sections 3.6 and 3.7.

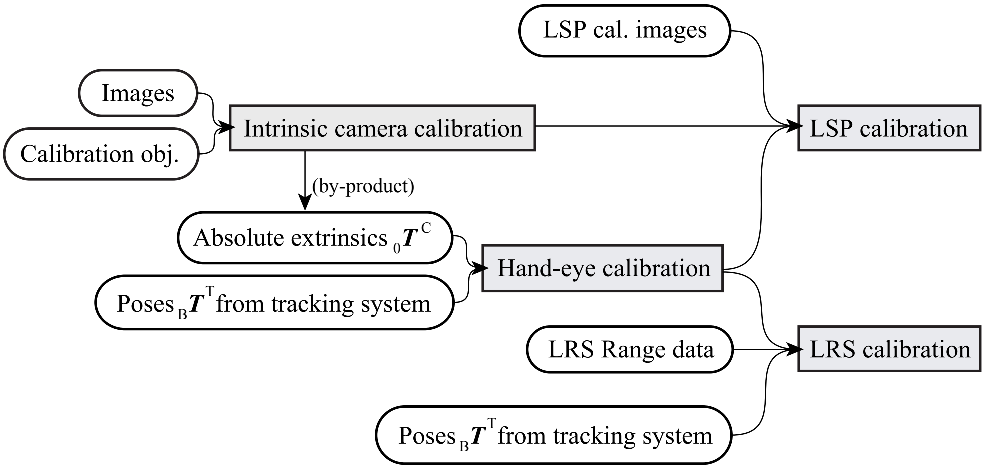

LSP cal. images
Images
Intrinsic camera calibration
LSP calibration
Calibration obj.
(by-product)
Absolute extrinsics ${}_0\boldsymbol{T}^{\mathrm{C}}$
Hand-eye calibration
Poses ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$ from tracking system
LRS Range data
LRS calibration
Poses ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$ from tracking system

Figure 1.2: Functional interaction between the calibration procedures.

1.5.3 The Importance of Robust Operation of Sensors

It would be desirable not to ruin the abovementioned precise computations by careless measurements, including image processing and stereo triangulation. In Chapter 4 I detail the methods required for robust data processing (mostly images) that, together with the pose readings of the pose reference system, result in 3-D pointclouds for further representation.

I mention two contributions:

- Theoretical investigations on the operation range of the component sensors as well as on their expected range estimation accuracies are detailed in Sections 4.2.5 and 4.2.6 in the case of the stereo camera, in Sections 4.3.4 and 4.3.5 in the case of the LSP, and in Sections 4.4.4 and 4.4.5 in the case of the LRS. The joint representation of their expected range estimation accuracies can be seen in Section 4.5.

- In Section 4.3.2 I present a novel, robust approach for the segmentation of laser stripe projections on images that were *not* filtered to laser light; the approach was originally published in (Strobl *et al.*, 2004). The use of unfiltered images of a laser stripe profiler is rare as it is far easier to work with filtered images. However, these would preclude us from performing stereo vision (Section 4.2), visual pose tracking (Chapter 5), visual texturing and image augmentation (Section 5.4.6) on the very same cameras. The approach presented in Section 4.3.2 features a cascade of detection and validation stages delivering the 2-D coordinates of laser stripe projections with sub-pixel accuracy. This procedure is in line with the "softwarization" paradigm addressed in Section 1.2, as more complex computations now allow one to get rid of extra hardware (the optical filter) that was impeding further use of the images in the first place.

1.5.4 Accurate, Passive Visual Pose Tracking

Accurate, passive visual pose tracking of the DLR 3D-Modeler in Chapter 5 constitutes a novel contribution in the realm of 3-D modeling devices; it presents the first hand-held 3-D modeling device for close-range applications that localizes itself passively from its own images in realtime, at a high data rate. Ever since its original publication in (Strobl *et al.*, 2009a), the approach acquired worldwide renown, as it was rated as a finalist to the best paper award at the well-known IROS conference in 2009. The approach is intended for the user to perform customary 3-D modeling as originally presented in (Suppa *et al.*, 2007), but doing without the external positioning systems like the FaroArm Gold, the Kuka KR 16, the DLR Lightweight Robot III, or the ARTtrack2 that constrain the system in size, mobility, and cost. Again, this is clearly in line with the "softwarization" paradigm introduced in Section 1.2.

The approach is based on high-rate tracking of natural, distinct features in the images of the main camera of the DLR 3D-Modeler. It is worth noting that the main difficulty in this context is that 3-D modeling is being performed at close range, where the displacement of features between frames is bigger than at long range because translational motions cannot be neglected anymore. In addition, the calculations involved have to perform in real-time, parallel to all other DLR 3D-Modeler-related computations like LSP triangulation, stereo vision, online meshing, and augmented image representation. In general, overall success can only be achieved by careful engineering of all key processes: relative motion is delivered at high-rate from feature tracking on a monocular image stream using a novel, robust V-GPS algorithm characterized by its efficiency and accuracy; in turn, feature tracking is based upon an accelerated KLT feature tracker (Mair *et al.*, 2010b), cast into the Active Matching paradigm for improved performance at close range, see (Strobl *et al.*, 2011). In order to detach feature set structure estimation from high-rate tracking at the frontend, feature-based stereo vision is being frugally triggered (at keyframe instants only) to compute accurate 3-D feature sets; in case of interrupted pose tracking, contingent appearance-based relocalization on SURF features is provided; finally, potential loop closures are utilized to increase accuracy in motion estimation performing sparse, hybrid bundle adjustment (BA), delivering refined motion history to the online meshing algorithm for timely display.

1.6 Outline of the Thesis

This thesis is structured having the principle of causality in mind. Its serial structure is as follows: In Chapter 1 I introduce the need for an innovative, multisensory 3-D modeling device that really complies with the "maximum effectiveness" paradigm. A short outline is delivered concerning the guidelines that resulted in the development of the DLR 3D-Modeler. In addition, the market of 3-D sensors has been introduced with regard to their principles of operation. I pointed at optical, reflective sensors as the more convenient sensor components to fulfill the task of close-range 3-D modeling. After that, I listed the main contributions of this thesis.

Chapter 2 presents the sensor components mounted on the DLR 3D-Modeler, describing their respective image formation processes. These system models are essential for valid 3-D reconstruction in Chapter 4 (and perhaps visual pose tracking in Chapter 5), provided the component sensors have been accurately calibrated as explained in Chapter 3. The latter chapter includes many novel contributions, as I noticed several wrongdoings when using traditional calibration methods, for example when disregarding precise information on the geometry of the calibration target, when using standard methods to calibrate cameras with narrow angular field of view, when extrinsically calibrating cameras w.r.t. noisy pose tracking systems, or when choosing inconvenient methods for calibrating a laser stripe profiler. In addition, I shall mention the calibration software DLR CalDe and DLR CalLab that we at DLR freely distribute on the internet (Strobl *et al.*, 2005).

In Chapter 4 I shall present the required calculations for 3-D reconstruction out of raw data (e.g. a live image stream). Some sensor components within the DLR 3D-Modeler are constrained owing to our original intention of gathering as much information as possible. For instance, the laser stripe profiler may not be filtered to laser light in order not to preclude concurrent texturing, image augmentation, stereo vision, and visual pose tracking. It is therefore important to devise robust processing algorithms that still deliver robust results even though the source material might be compromised, e.g. specked with spurious light reflections. These extra calculations are in line with the abovementioned guideline on "softwarization" for creating more effective sensors. In this chapter I also report on expected operation ranges as well as on the precision of the sensors subject to the measurement range.

Passive, visual pose tracking in realtime in Chapter 5 is a novel contribution in the context of 3-D modeling systems. It allows for accurate pose tracking in realtime without the need for external positioning systems, which, nearly without exception, represent the most expensive hardware part of the system. The method is based on high-rate tracking of natural, distinct features in the images of the main camera of the DLR 3D-Modeler. The diversity of supporting algorithms that are required for this application make for a dedicated section on state of the art, that effectively extends the last Section 1.4. This novel approach was well received by the computer vision community, being awarded a best paper finalist award at the IROS conference in 2009.

Even though every chapter features its own experiments that validate particular methods, in Appendix B I list robotic systems that include (at least) part of the algorithms developed in the course of this thesis, as well as closely related methods including one patented method. In addition, the calibration toolbox DLR CalDe and DLR CalLab is detailed in Section B.3.

Chapter 6 summarizes the contributions of this thesis and I shall mention open directions for research in the hope that sensors based on this technology that explicitly form 3-D models will eventually push forward the key technological area of service robotics in order to achieve critical mass and become widespread across society.

"Everything should be made as simple as possible, but not simpler."
—Albert Einstein, 1933

2 System Modeling

2.1 Introduction

Computer vision is largely about inverting the image formation process; consequently, accurate modeling of the perception processes taking place within the DLR 3D-Modeler plays a central role in this thesis.

Unlike qualitative tasks like 2-D image understanding and art appreciation by humans, **geometric** computer vision is a quantitative task that requires a thorough knowledge of the process of image formation with regard to its inner geometry. Since image formation is a complex process, it can be of course reflected by an accordingly complex model. In extreme cases, inscrutable devices can be even modeled by a comprehensive mapping of all possible inputs (e.g. 3-D instances and ambient illumination) onto their corresponding output projections, which is commonly referred to as a look-up table (LUT) model. Nonetheless it is only by choosing a **compact** (*i.e.*, non-redundant) model of operation that we shall eventually support efficient instantiation from these models in Chapter 4. In addition, minimal, non-redundant models that reflect the underlying physical principles of the device are more likely to feature **general validity** compared with extended models overfitted to data. Last but not least, compact model selection supports its own accurate parametrization (*i.e.*, calibration) in Chapter 3.

It is indeed judicious to draw on experts' work that incorporate all potential variations of the operating model to increase accuracy. Regrettably, this process entails risks, like the danger of overparametrization in the choice of the model (overfitting), or even the risk of following established "dogmas" that, in reality, only hold subject to the hardware at hand. By way of illustration, researchers regularly opt for advanced optical distortion models for cameras in the hope that they will lead to more accurate results as they, of course, do lead to somewhat smaller residual errors after calibration; more often than not, however, they lead to parameters that are overfitted to datasets, thus not valid in eventual, general operation. In (Strobl *et al.*, 2009b) we make the case for tighter models that ought not to harm more than help.

In this chapter I detail on the operating models required for accurate parametrization in Chapter 3 as well as efficient operation in Chapters 4 and 5 of all sensors mounted on the DLR 3D-Modeler.

2.2 The DLR 3D-Modeler Components

The DLR 3D-Modeler is a multipurpose, multisensory platform for geometric and visual perception (Suppa *et al.*, 2007). It combines complementary sensors in a compact, generic way, see Fig. 2.1. The main approaches for depth acquisition include stereo vision, structured light, and laser scanning. The sensor interfaces were unified in order to simplify the inclusion of further sensors. The sensor principles can be compared, and the best one chosen for a specific task. Evading and clearing sensor weaknesses can be also accomplished. We aim at robustness through data fusion. Current applications comprise 3-D modeling, visual tracking and servoing, exploration, path planning, and object recognition e.g. as the perception head of the humanoid robot “Justin” (Borst *et al.*, 2009), see Section B.2.1 within Appendix B.

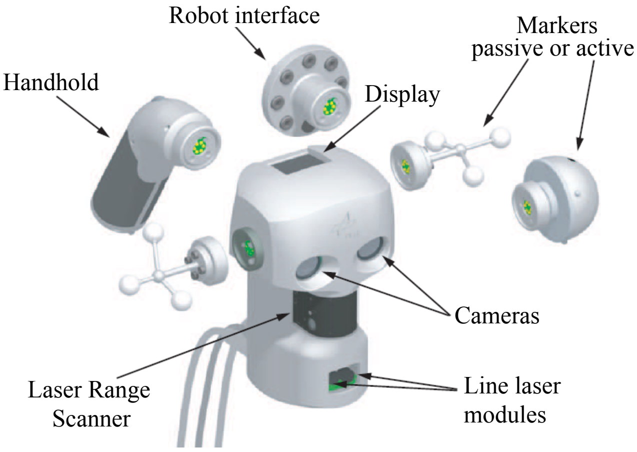


Figure 2.1: The DLR 3D-Modeler and its components (Fig. 1.1 reprint).

Further highlights are its low weight and power consumption, accurate and adaptable synchronization of internal and external sensors, on-board computing power on an embedded Linux operating system (with its own display, buttons, and a mouse wheel), generic mechanical interfaces, unified communication to computers via FireWire®, as well as an extensive, congruent inhouse software suite. Note the electronic boards inside the DLR 3D-Modeler in Fig. 2.2.

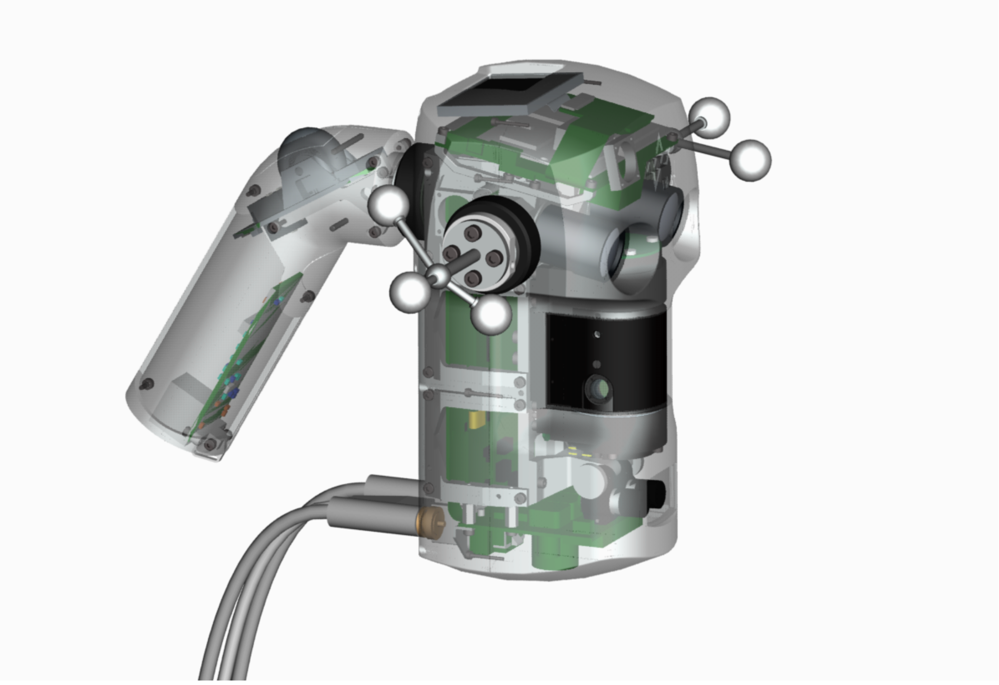


Figure 2.2: Inner view of the DLR 3D-Modeler and its electronics.

Its principal sensory components are (cf. Fig. 2.1):

- The stereo camera consists of two AVT Marlin F-046C progressive scan cameras, resolution 780×582, separated 50 mm from each other, featuring 6 mm Sony® VCL-06S12XM objectives. The base distance and the focal length were chosen to cover the sensing range from 30 cm up to 2 m, refer to Section 4.2.5. The implemented stereo algorithm is Semiglobal Matching (SGM) and it is detailed in Section 4.2.3 and in (Hirschmüller, 2008).

- The DLR Laser Stripe Profiler (LSP) (Strobl *et al.*, 2004; Suppa *et al.*, 2007) features two laser beams that sequentially project stripes on a surface. The stripes are recorded by the cameras and reconstructed by triangulation—this is the dual, crosshair operational mode; operation with only one laser is still possible. The LSP delivers close- to mid-range data, cf. Section 4.3.4. It is worth noting that the LSP works without optical filters on the cameras, as they would render stereo vision, texturing, and visual pose tracking in Chapter 5 impossible.

- The DLR Laser Range Scanner (LRS) (Hacker *et al.*, 1997; Kielhöfer, 2003) also operates by laser light triangulation. A visible, weak laser ray is continuously rotated and its reflection is dynamically recorded by a position sensitive detector. Because of its robustness and small size, it is used as a high-definition, short-range sensor. It is worth noting that its wide scan angle is especially convenient in robotic applications.

- The AscTec AutoPilot inertial measurement unit (IMU) by Ascending Technologies GmbH can be optionally attached to the device. Attitude estimation is on six degrees of freedom (6 DoF) at 1 kHz leveraging three gyros, three accelerometers, and three magnetometers. It also features onboard data fusion and a second, idle 60 MHz ARM processor still available for the user. It only weighs 19.6 g and is size 10 × 50 × 50 mm.

- A contact probe can be optionally mounted on the DLR 3D-Modeler, see Fig. 2.3. Note that the probe is rigid and should only be used in the case of passive pose tracking systems like the ARTtrack2 or visual pose tracking as in Chapter 5. In the case of active pose tracking systems like robotic manipulators, the use of this mechanical probe on rigid objects is only possible if the manipulator is compliant (probably controlled by force), e.g. the DLR Lightweight Robot III.

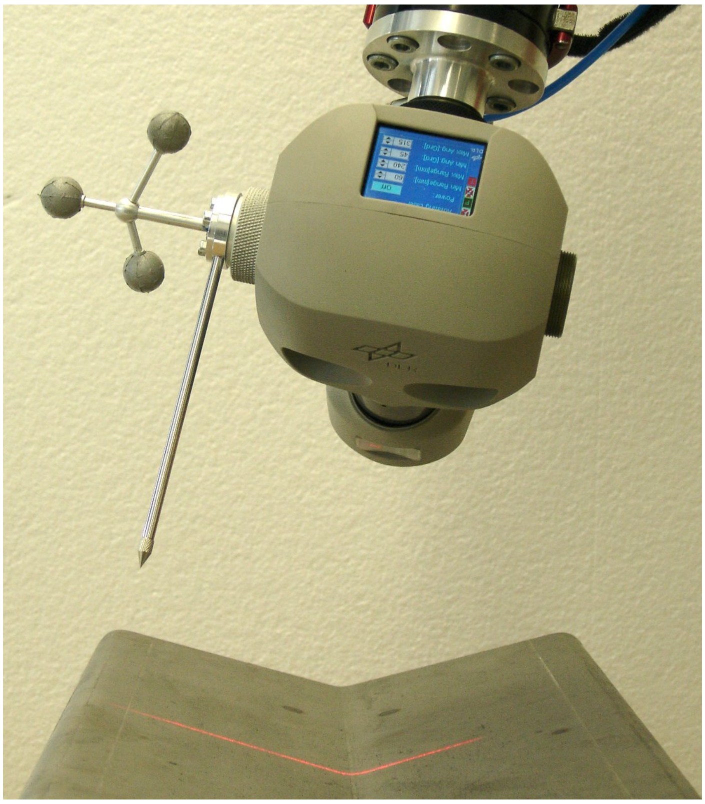


Figure 2.3: Mechanical contact probe mounted on the DLR 3D-Modeler.

In the following sections I shall select convenient operation models for the abovementioned sensors, along with their mechanical layout with respect to (w.r.t.) optional external tracking systems like the FaroArm Gold, the ARTtrack2, the Kuka KR 16 robotic manipulator, or the DLR Lightweight Robot III. In addition, I detail on the historical development of the sensors, as model selection goes hand in hand with their technological progress.

2.2.1 The Stereo Camera

Description

Cameras are devices for the visual representation of general, 3-D scenes *perspectively*, *i.e.*, in a similar way as perceived by the human eye. This representation is in the form of a central projection through the center of projection of the camera (e.g. a pinhole) onto the planar, 2-D image sensor. Note that by inverting this image formation process without any prior information of the scene, it is only possible to infer the direction of view in Euclidean space of the projected features, but not their depth. Overall, perspective projection can be summed up by these two principles: Close objects project bigger, and differently distant objects may project onto the same region, *i.e.*, range gets lost.

A stereo camera is a camera system composed of two or more cameras (mostly rigidly attached to each other) that is capable of immediate 3-D reconstruction of the scene (including depth) by triangulation using stereo vision algorithms. Stereo vision (or stereopsis) is a perception process leading to depth information out from different pespective projections of the scene, from different locations. In computer vision it usually concerns two video cameras, even though monocular cameras are also capable of 3-D reconstruction by stereo vision using images from different vantage points (of course, delayed in time and up to similarity, *i.e.*, unscaled). On the understanding of a pointwise treatment of the scene, stereo vision is capable of full 3-D reconstruction out of the inversion of the image formation processes for every single camera (*i.e.*, out of the inferred view directions of the projected features for every camera), along with depth estimation by optimal triangulation. By fusing view directions registered at different locations, feature triangulation in Euclidean space is possible *if* the relative transformation between cameras is known (cf. Section 4.2.2). The first approach to processing depth images was mechanical, on analog images by Eduard von Orel in 1907. Current analytic approaches focus on digital imaginery and are processed by computers.

The implemented stereo vision algorithm is the Semiglobal Matching method (SGM) by Heiko Hirschmüller (Hirschmüller, 2008). A global smoothness constraint that supports pixelwise matching is used as a cost function. Additionally, SGM uses a mutual information-based matching cost for compensating radiometric differences of input images. The algorithm also includes a number of post-processing steps for refinement. It is currently amongst the top-ranked in the Middlebury Stereo Vision Page (Scharstein and Szeliski, 2002) and has been already instantiated on different platforms like CPUs, GPUs and FPGAs.

Digital cameras are widely used in robotics as they are light, affordable, have a small footprint and consume limited amounts of energy. They furthermore allow for a very accurate parametrization of a simple, yet accurate operating model. In addition, cameras gather plenty of radiometric and geometric information within a single, rapid measuring cycle. Recent growth in computing power allows for elaborate image processing in customary rates like 25 to 50 Hz. Furthermore, they are non-contact sensors, thus free-floating, and operate passively without the need to influence the environment which they measure. They also allow for intrinsically synchronized measurements between different image-based sensing components, e.g. with simultaneous operation of the LSP, stereo vision, 3-D model texturing, and 6-D visual pose tracking as in Chapter 5.

Digital cameras have also taken over as the preferred sensors for simultaneous localization and mapping (SLAM). Laser range finders (LIDARs and LADARs) had been traditionally used to this end, as they directly measure surface distances with high precision. Traditional range finders, however, only scan 2-D stripes of the scene, which limits their use to flat floor scenarios—still they encounter difficulties with common objects such as tables or shelves. In addition, laser range finders are larger in size, heavier, and consume more power compared with passive sensors such as video cameras, and even the simplest range finders are two to three orders of magnitude more expensive than a camera.

Historical Remarks

Even though the ancient Greeks were aware of many concepts of projective geometry, they erroneously conceived the eye as an active device *emitting* radiation. It was the Persian scientist Ibn al-Haytham (Latinized: Alhacen) at the beginning of the 11th century that crucially identified the eye as a passive device *receiving* radiation. This observation allowed him to state the basics of perspectivity, to give a detailed description of the human eye, and to produce the reportedly first artificial camera: the *camera obscura*—out of scientific, not artistic interest. Artists like Leonardo da Vinci and Albrecht Dürer in the 15th century as well as the Dutch Masters in the 17th century used a *camera obscura* as a drawing aid for correct perspective representation of scenes (Hockney, 2006). The conceptual **pinhole camera** instantiated by the *camera obscura* consists of an opaque box pinholed in a side. Light from the scene passes through the pinhole projecting onto the opposite side of the box, both mirrored and inverted upside-down. Using mirrors, the image can be further projected onto tracing paper for manual transcriptions that accurately depict a perspective representation of the scene, see Fig. 2.4.

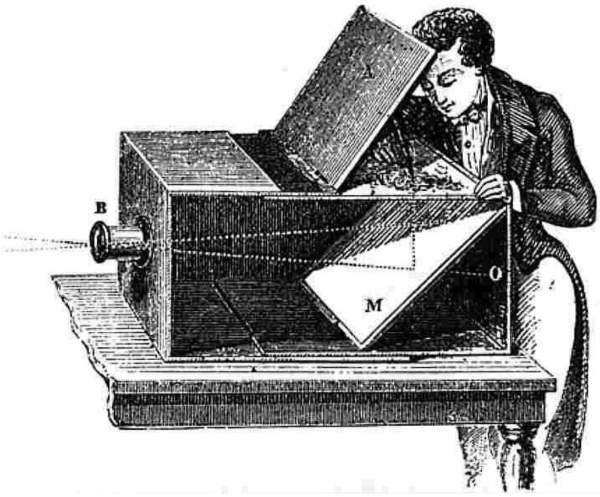


Figure 2.4: An 18th century artist drawing with a *camera obscura* featuring a lens (B) and a mirror (M). *Source: 18th century dictionary illustration.*

In general, the smaller the pinhole is made, the sharper the image results; on the other hand, however, a tiny pinhole yields to distortion by diffraction and, furthermore, the resulting images lack of brightness. For this reason, **optical lenses** were introduced in order to increase brightness while largely maintaining sharpness—similar to human eyes. This was proposed by Leonardo in the 15th century; it was Daniele Barbaro in the 16th century that pioneered the use of optical lenses in this type of cameras instead of actual pinholes. Optical lenses were also used in the realm of astronomical telescopes. Their design was perfected using the law of refraction of Willebrord Snell and René Descartes at the beginning of the 17th century, as well as the investigations on optical aberrations in the 19th century by James Clerk Maxwell and Ernst Abbe.

A major contribution for automatic, more veritable perspective representations is the work of Joseph Nicephore Niepce, Louis Daguerre, and William Henry Fox Talbot in the early 19th century by the invention of the daguerreotype and the negative/positive photographic process, see Fig. 2.5. It is no coincidence that lens design was subject to substantial improvements for the years to come, since, especially in this context, the use of optical lenses further allows for reasonable shutter times.

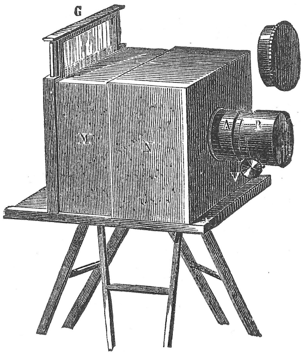


Figure 2.5: Daguerreotype camera built by Alphones Giroux in 1839. Note the silvered copper positive plate (G). *Source: WestLicht Photographica Auction* `www.westlicht-auction.com`.

The last milestone to current state of the art on off-the-shelf cameras has been the development of **electronic sensors** (charge-coupled devices (CCD) or complementary metal-oxide-semiconductor (CMOS) chips, see Fig. 2.6) for electronic acquisition of images in the second half of the 20th century. In conjunction with personal computers, they enable an overarching digital process with acquisition, storage, and image processing becoming fully digital.

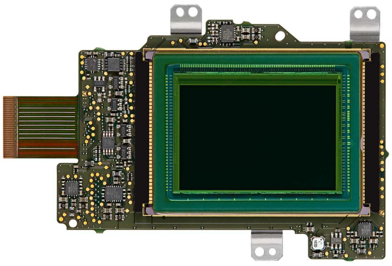


Figure 2.6: The CMOS image sensor first mounted on a Leica M camera. *Source:* `de.leica-camera.com`.

The Perspectivity Equation

The camera operating model presented here focuses on the geometry of the camera. I am not carrying out a photometric study, *i.e.*, I will not treat brightness and color of the perceived radiation, but only its location on the chip. Sharpness is also not considered. This type of camera modeling is common to geometric computer vision applications and will enable the user to infer in 3-D Euclidean space from the evidence of 2-D image projections.

It is remarkable that the geometric model of ancient *pinhole* cameras still holds for accurately describing the main functioning principle of modern cameras with complex lenses. The pinhole camera model represents perspective projection, *i.e.*, the mapping of the 3-D world scene onto a 2-D imaging plane, by rays of light passing through a (conceptual) point called *center of projection*, *focal point* or *focus* of the camera. The camera reference frame S_C is located at the focal point and its axis ${}_\mathrm{C}\boldsymbol{z}$ is in the direction defined by the point on the imaging plane of minimum Euclidean distance to the focal point and the focal point itself, *i.e.*, it is perpendicular to the imaging plane, see Fig. 2.7. That point on the projection plane is called *principal point*, and therefore the axis ${}_\mathrm{C}\boldsymbol{z}$ may be also called *principal axis* or *optical axis.* The distance between the principal point and the the focal point is called *focal length* f. Any imaging plane distant one distance unit to the focal point (*i.e.*, $f = 1$) is called a normalized image plane and is represented by the 2-D normalized image frame S_I. The origin of S_I is located at its principal point and its two main axes are both parallel and in the same direction as ${}_\mathrm{C}\boldsymbol{x}$ and ${}_\mathrm{C}\boldsymbol{y}$ in the camera frame S_C. The direction of the latter axes is not yet being specified as, for reasons of convenience, it will be related to the actual (digital) sensing plane.

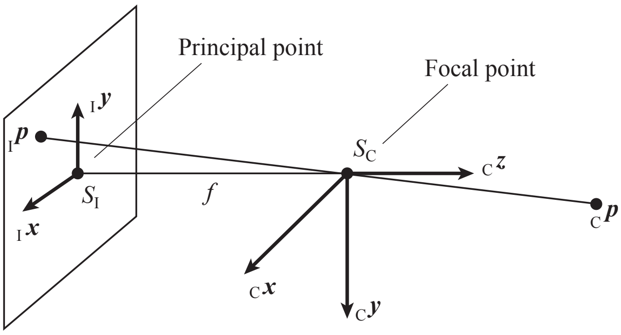

Principal point
Focal point
${}_\mathrm{I}\boldsymbol{y}$
${}_\mathrm{I}\boldsymbol{p}$
S_C
${}_\mathrm{C}\boldsymbol{z}$
S_I
f
${}_\mathrm{C}\boldsymbol{p}$
${}_\mathrm{I}\boldsymbol{x}$
${}_\mathrm{C}\boldsymbol{x}$
${}_\mathrm{C}\boldsymbol{y}$

Figure 2.7: Perspective projection of feature ${}_\mathrm{C}\boldsymbol{p}$ unto the image plane S_I.

Using the Thales' theorem of similar triangles and representing both main directions on the image frame S_I in matricial form, a particular 3-D world feature $\boldsymbol{p}$, represented in S_C as ${}_\mathrm{C}\boldsymbol{p} = [{}_\mathrm{C}x, {}_\mathrm{C}y, {}_\mathrm{C}z]^\mathsf{T}$, projects onto S_I as follows:

$$
{}_\mathrm{I}\boldsymbol{p} = \begin{bmatrix} {}_\mathrm{I}x \\ {}_\mathrm{I}y \end{bmatrix} = (-) \begin{bmatrix} {}_\mathrm{C}x / {}_\mathrm{C}z \\ {}_\mathrm{C}y / {}_\mathrm{C}z \end{bmatrix} , \tag{2.1}
$$

which clearly entails the loss of positioning information in one dimension, viz. the absolute distance to the image projection. From this it follows that the location of the 3-D original feature cannot be fully recovered from any finite projection ${}_{\mathrm{I}}\boldsymbol{p}$. It is worth noting the minus sign representing the inverted image formation; it is common practice to drop this sign in order to avoid using inverted images; this is readily done by relocating the (virtual) image frame S_{I} in front of the focal point, *i.e.*, $({}_{\mathrm{I}}x,\, {}_{\mathrm{I}}y) \mapsto (-{}_{\mathrm{I}}x,\, -{}_{\mathrm{I}}y)$.

More in detail, the pinhole camera model can be derived from the thin lens camera model in the case of smaller aperture sizes; the thin lens camera model is in turn a particularization of the more general thick lens camera model. The latter model includes the effects of thick lenses except for their aberrations, refer to Section 2.2.1. The reader can find a more detailed description of the nature of this type of projection in Refs. (Faugeras *et al.*, 2001; Hartley and Zisserman, 2004; McGlone *et al.*, 2004).

Digital Sensors

The actual imaging plane is currently being instantiated by an electronic, discrete imaging sensor like CCD or CMOS chips. This type of sensors decisively affect the formation of the digital image through the so-called *digitization process*. This process relates the eventual **p**icture **el**ements (*pel*s) with the original, normalized coordinates ${}_{\mathrm{I}}\boldsymbol{p}$ explained above, depending on the particular geometry of the **s**ensing **el**ements (*sel*s) of the chip, *i.e.*, their (perhaps non-rectangular) side sizes s_x and s_y, as well as its actual location w.r.t. the focal point. The chip's location is specified by both, the position of the principal point as well as its focal length f; the orientation of the imaging plane is already defined as it corresponds to ${}_{\mathrm{C}}\boldsymbol{z}$, and its in-plane orientation is set for ${}_{\mathrm{I}}\boldsymbol{x}$ to correspond with ${}_{\mathrm{C}}\boldsymbol{x}$. The 2-D reference frame ideally attached to the sensor plane is called memory frame S_{M}. I choose to locate it at the upper-left corner of the digital image (viz. at the center of the upper-left corner pixel), being its ${}_{\mathrm{M}}\boldsymbol{x}$ axis parallel and in the same direction as ${}_{\mathrm{I}}\boldsymbol{x}$ in S_{I}, see Fig. 2.8. The location of the principal point in S_{M} can then be especified by its coordinates u_0 and v_0.

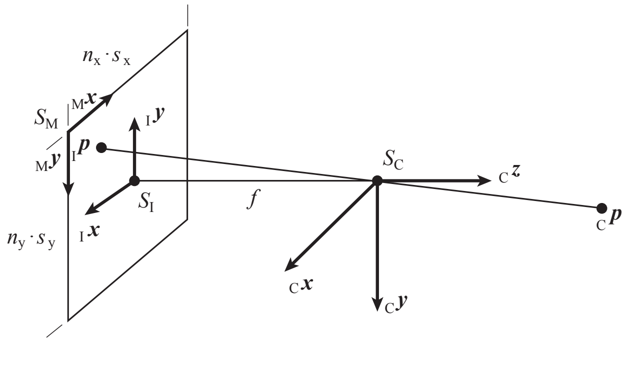

$n_{\mathrm{x}} \cdot s_{\mathrm{x}}$
S_{M}
${}_{\mathrm{M}}\boldsymbol{x}$
${}_{\mathrm{I}}\boldsymbol{y}$
${}_{\mathrm{I}}\boldsymbol{p}$
${}_{\mathrm{M}}\boldsymbol{y}$
S_{C}
${}_{\mathrm{C}}\boldsymbol{z}$
S_{I}
f
${}_{\mathrm{C}}\boldsymbol{p}$
$n_{\mathrm{y}} \cdot s_{\mathrm{y}}$
${}_{\mathrm{I}}\boldsymbol{x}$
${}_{\mathrm{C}}\boldsymbol{x}$
${}_{\mathrm{C}}\boldsymbol{y}$

Figure 2.8: Perspective projection of feature ${}_{\mathrm{C}}\boldsymbol{p}$ unto the image memory frame S_{M}.

These facts call for an extension of Eq. (2.1) as follows:

$$ {}_{\mathrm{M}}\boldsymbol{p} = \begin{bmatrix} {}_{\mathrm{M}}x \\ {}_{\mathrm{M}}y \end{bmatrix} = \begin{bmatrix} f/s_x \, {}_{\mathrm{I}}x + u_0 \\ f/s_y \, {}_{\mathrm{I}}y + v_0 \end{bmatrix} = \begin{bmatrix} f/s_x \, {}_{\mathrm{C}}x/{}_{\mathrm{C}}z + u_0 \\ f/s_y \, {}_{\mathrm{C}}y/{}_{\mathrm{C}}z + v_0 \end{bmatrix} . \tag{2.2} $$

More in general, skewed imaging sensors are allowed so that ${}_{\mathrm{M}}x$ and ${}_{\mathrm{I}}y$ are not independent anymore even if ${}_{\mathrm{M}}x$ and ${}_{\mathrm{I}}x$ still stick parallel to each other.

For a relative angle λ between ${}_{\mathrm{M}}x$ and ${}_{\mathrm{M}}y$ the former equation extends to:

$$
{}_{\mathrm{M}}\boldsymbol{p} = \begin{bmatrix} {}_{\mathrm{M}}x \\ {}_{\mathrm{M}}y \end{bmatrix} = \begin{bmatrix} f/s_x \, {}_{\mathrm{I}}x \, - \, f/s_x \, {}_{\mathrm{I}}y \, \cot\lambda \, + \, u_0 \\ f/s_y \, (\sin\lambda)^{-1} \, {}_{\mathrm{I}}y \, + \, v_0 \end{bmatrix} =
$$

$$
= \begin{bmatrix} f/s_x \, {}_{\mathrm{C}}x/{}_{\mathrm{C}}z \, - \, f/s_x \, {}_{\mathrm{C}}y/{}_{\mathrm{C}}z \, \cot\lambda \, + \, u_0 \\ f/s_y \, (\sin\lambda)^{-1} \, {}_{\mathrm{C}}y/{}_{\mathrm{C}}z \, + \, v_0 \end{bmatrix} . \qquad (2.3)
$$

These equations represent a *nonlinear relationship* between the 3-D point ${}_{\mathrm{C}}\boldsymbol{p}$ and its 2-D projection ${}_{\mathrm{M}}\boldsymbol{p}$. By using the homogeneous coordinates $(\bar{})$ for projected features ${}_{\mathrm{M}}\bar{\boldsymbol{p}}$, a simpler, *linear projective* formulation arises:

$$
{}_{\mathrm{M}}\bar{\boldsymbol{p}} = \begin{bmatrix} {}_{\mathrm{M}}x \\ {}_{\mathrm{M}}y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f/s_x & -f/s_x \, \cot\lambda & u_0 \\ 0 & f/s_y \, (\sin\lambda)^{-1} & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}_{(3\times3)}} \begin{bmatrix} {}_{\mathrm{I}}x \\ {}_{\mathrm{I}}y \\ 1 \end{bmatrix} \propto \boldsymbol{A}_{(3\times3)} \begin{bmatrix} {}_{\mathrm{C}}x \\ {}_{\mathrm{C}}y \\ {}_{\mathrm{C}}z \end{bmatrix} , \qquad (2.4)
$$

where $\boldsymbol{A}$ is the intrinsic matrix composed of the *intrinsic parameters* that represent the internal orientation of the camera. It can be also read as an affine transformation between the normalized image frame S_{I} in Eq. (2.1) and the memory frame S_{M}. In addition, the homogeneous formulation may factor in points at infinity, which may be of advantage in computer vision applications.

Since camera parameters f, s_x, s_y, and λ are entangled in the coefficients within $\boldsymbol{A}$, it is not possible to estimate them separately based on external, image-based measurements alone. However, in order to estimate in 3-D out from 2-D projections, it is not really necessary to have a through knowledge of these physical parameters anyway, but only of the composed coefficients. Consequently we set:

$$
\alpha \triangleq f/s_x \qquad (2.5)
$$

$$
\beta \triangleq f/s_y \, (\sin\lambda)^{-1} \qquad (2.6)
$$

$$
\gamma \triangleq -\, f/s_x \, \cot\lambda \qquad (2.7)
$$

and, as a result:

$$
{}_{\mathrm{M}}\bar{\boldsymbol{p}} = \begin{bmatrix} {}_{\mathrm{M}}x \\ {}_{\mathrm{M}}y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}_{(3\times3)}} \begin{bmatrix} {}_{\mathrm{I}}x \\ {}_{\mathrm{I}}y \\ 1 \end{bmatrix} \propto \boldsymbol{A}_{(3\times3)} \begin{bmatrix} {}_{\mathrm{C}}x \\ {}_{\mathrm{C}}y \\ {}_{\mathrm{C}}z \end{bmatrix} , \qquad (2.8)
$$

which is an adequate, general representation of the pinhole camera model—a simpler algebra at the expense of a straightforward geometric interpretation, see (Tsai, 1987; Faugeras and Toscani, 1987). It is worth noting that, according to the interpretation of the skew parameter γ as a skewing of the *sel*s, it is admittedly very unlikely for this type of distortion to still happen using modern sensors. Therefore, in general it is not recommended to release this degree of freedom except when using special cameras (e.g. bellows cameras or camera systems taking images of perspectively projected images).

Lens Distortion

The lens is generally the most expensive and least understood part of any camera. As stated above, projecting rays do not really pass through a single focal point but through complex lens units—keeping rays close and bundled but going separate ways, see Fig. 2.9.

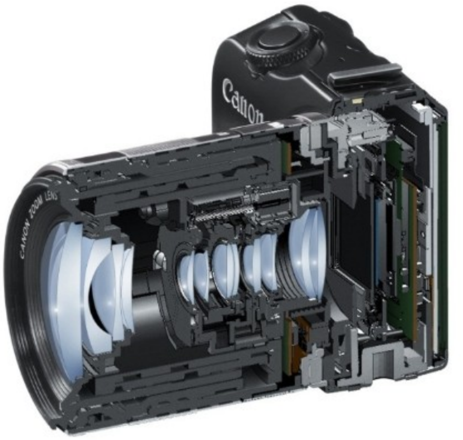


Figure 2.9: Cutaway view of an off-the-shelf digital camera. Its lens unit is composed of a number of individually moving lenses. *Source:* `www.canon.de`.

The complex path of light within a lens certainly accounts for deviations from the straight line projection assumption mentioned above, which is cause for a number of potential *optical aberrations*. Here we are concerned with the monochromatic aberrations that distort the geometrical shape of the whole image, for example with the Petzval field curvature or with distortion; we are not concerned with pointwise sharpness issues like spherical aberration, coma, or astigmatism. It is well known that the Petzval field curvature can be compensated for by a suitable combination of positive and negative lenses and stops[1] (Hecht, 1998; Born *et al.*, 1999). Unfortunately, lens designers rarely cope with distortion, which is most accentuated in cameras with bigger angular fields of view (AOV). The reason is that all aberrations are factors being concurrently optimized during lens design and sharpness issues are being considered more critical than geometric accuracy (Kingslake, 1992; Stroebel, 1999; Born *et al.*, 1999)—and rightly so. Unlike most of the other aberrations (including the chromatic ones), the effects of distortion can be readily compensated for by a posterior computation step called undistortion; consequently, more often than not lens distortion is being left unattended during the process of lens design.

In the following, the three main geometric lens distortion effects are being addressed. Even though lens distortion declaredly stems from the fact that the pinhole camera model is not exact, we still intend formulations of lens distortion that *extend* the former pinhole camera model without replacing it. The reason behind is the strength and efficiency of the linear projective formulation of the pinhole camera model that must be retained. In

[1] Aperture stops are circular, perforated discs that limit the incoming light to its central pinhole; they are regularly used between or within lenses in order to contain the effects of different types of aberrations.

detail, we aim at a pre-processing stage undist$(\cdot)$ for image undistortion, *i.e.*, undist$(\cdot)$: $({}_{\mathrm{I}}x_{\mathrm{d}},\, {}_{\mathrm{I}}y_{\mathrm{d}}) \mapsto ({}_{\mathrm{I}}x_{\mathrm{u}},\, {}_{\mathrm{I}}y_{\mathrm{u}})$, so that subsequent perspective reprojection of these virtual (nonobservable), **u**ndistorted projections using the standard pinhole camera model in Eq. (2.1) perfectly matches with the actual imaging directions in S_{C}. To recap:

$$\begin{bmatrix} {}_{\mathrm{C}}x/{}_{\mathrm{C}}z \\ {}_{\mathrm{C}}y/{}_{\mathrm{C}}z \end{bmatrix} = \begin{bmatrix} {}_{\mathrm{I}}x_{\mathrm{u}} \\ {}_{\mathrm{I}}y_{\mathrm{u}} \end{bmatrix} = \mathrm{undist}\left(\begin{bmatrix} {}_{\mathrm{I}}x_{\mathrm{d}} \\ {}_{\mathrm{I}}y_{\mathrm{d}} \end{bmatrix} \right) \Big/ \; {}_{\mathrm{M}}\bar{\boldsymbol{p}}_{\mathrm{u}} = \begin{bmatrix} {}_{\mathrm{M}}x_{\mathrm{u}} \\ {}_{\mathrm{M}}y_{\mathrm{u}} \\ 1 \end{bmatrix} = \boldsymbol{A}_{(3\times 3)} \begin{bmatrix} {}_{\mathrm{I}}x_{\mathrm{u}} \\ {}_{\mathrm{I}}y_{\mathrm{u}} \\ 1 \end{bmatrix} . \tag{2.9}$$

This pre-processing stage will often make use of mathematical models of the underlying physical principles of lens distortion, even though phenomenological approaches to lens undistortion could also suffice.

It is worth noting that the following, conventional models for undistortion only depend on view directions and not on projection distances. It is conventionally accepted that the pinhole camera model is only valid beyond very close distances of approximately 30 times the focal length (Luhmann *et al.*, 2006; Magill, 1955; Brown, 1966; Fryer; Duane C. Brown, 1986); undistortion models should at least match this operating range. Furthermore, I consider lens distortion as a constant effect because focal length is a constant parameter in most computer vision applications. In the case of camera systems with varying magnification (*i.e.*, zoom), I refer the reader to (Magill, 1955).

The following lens distortion models address all potential lens distortions of most camera systems with regular angular field of view (*i.e.*, at most 120°). The first of them is related with the imperfect design of the lens unit, whereas the two last distortion models concern improper lens and camera assembly.

- ***Radial distortion***, *Seidel distortion* or *barrel or pincushion distortion* is the last of the Seidel aberrations and causes a radial displacement δ_r of projections w.r.t. the center of distortion. In the absence of other kinds of deviations like improper assembly of the lens unit, the center of distortion ought to coincide with the principal point of projection. Radial distortion is due to varying camera magnification (*i.e.*, different focal length) in relation to the angular distance of the projecting ray to the optical axis—irrespective of its angle around the optical axis (therefore, it is also called *symmetric distortion*). Since the effect is responsible for actually straight lines being rendered curved (except for lines that pass through the center of radial distortion), it is sometimes also called *curvilinear distortion.*

  In detail, the divergent magnifications in the case of *oblique* light pencils traversing the lens are direct consequence either of the use of thick lenses or of the use of aperture stops in systems of thin lenses, see Fig. 2.10. Depending on the position of these stops (and not on their aperture sizes[2]),

[2] Indeed, at very small apertures of the diaphragm, distortion is the only noticeable aberration (Conrady, 1958). This observation is currently being used to treat presbyophia in elderly patients, implanting rings on their eyes to increase sharpness based on the principle of sharper projection by smaller apertures, e.g. using the KAMRA Inlays$^{\mathrm{TM}}$ system.

the magnification varies from radially displacing projections *towards* the center of radial distortion (*barrel distortion*, $\delta_r < 0$) to displacing the projections *away* from it (*pincushion distortion*, $\delta_r > 0$), cf. Fig. 2.10. Complex lenses may very well present both barrel and pincushion distortions. In order to minimize this effect, lens designers have to plant the stops between or within lenses (Jenkins and White, 1976; Hecht, 1998).

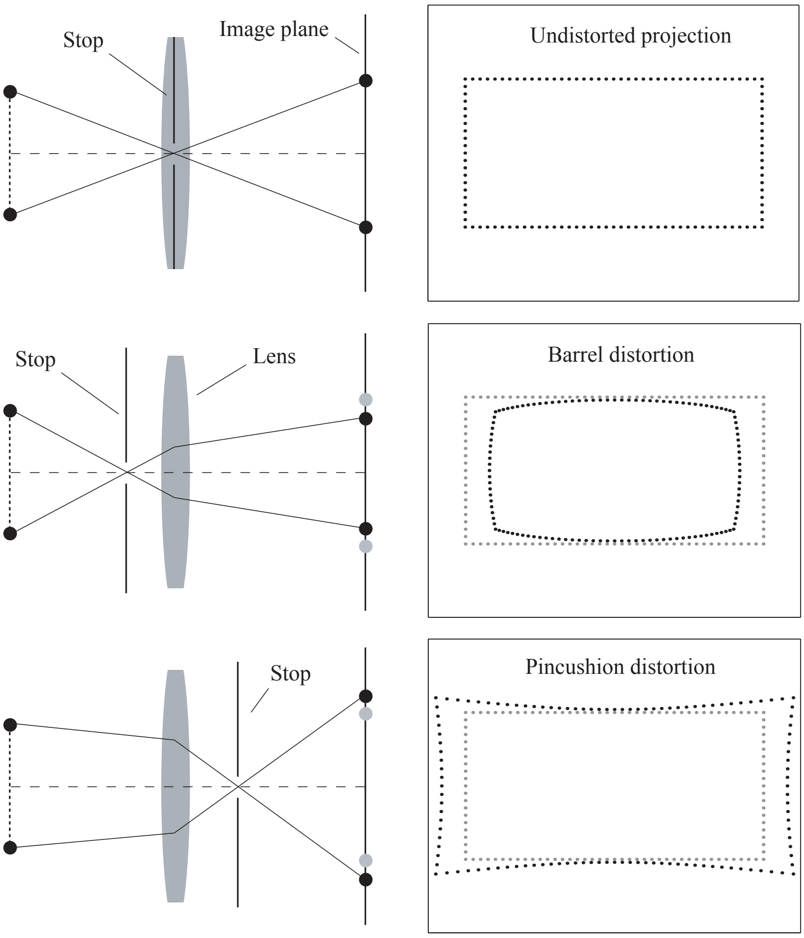


Figure 2.10: Placing stops limits the effects of different types of aberrations. Stops within the lens reduce lens distortion effects improving the depth of field. Stop discs outside the lens produce either barrel or pincushion lens distortion.

Taking into account Snell's law of refraction as well as the geometry of the camera (including the potential use of stops), radial distortion is expected to grow *as the cube* of the distance of the normalized projection ${}_{\mathrm{I}}\boldsymbol{p}_{\mathrm{u}}$ to the principal point if we use a third-order Taylor expansion of the sine function in the law of refraction of Snell. This is the so-called cubic law of the distortion—refer to Conrady's study on extra-axial projections in (Conrady, 1958, 1960). The normalized projections ${}_{\mathrm{I}}\boldsymbol{p}_{\mathrm{u}}$ are determined using the pinhole camera model in Eq. (2.9), or rather the theorem of Lagrange.[3] The addition of higher-order decomposition terms to Snell's

[3] The theorem of Lagrange determines the linear magnification of the image of a small

refraction law yields the following analytic expression for radial distortion:

$$\delta_{\mathrm{r}}(\rho) = k_1\rho^3 + k_2\rho^5 + k_3\rho^7 + O(\rho^9) \tag{2.10}$$

where ρ is the radial distance from the center of radial distortion to the expected, normalized projections ${}_{\mathrm{I}}\boldsymbol{p}_{\mathrm{u}}$, *i.e.*, $\rho = \sqrt{{}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2}$, and k_1, k_2, k_3 are the coefficients of radial distortion. The Cartesian representation of Eq. (2.10) depends on the in-plane angular position $\vartheta = \arctan({}^{\mathrm{I}}y_{\mathrm{u}}/{}_{\mathrm{I}}x_{\mathrm{u}})$ of the feature in the reference frame S_{I} as follows:

$$\delta_{\mathrm{r_x}} = \delta_{\mathrm{r}}\cos\vartheta = \delta_{\mathrm{r}}\ {}^{\mathrm{I}}x_{\mathrm{u}}/\rho = {}_{\mathrm{I}}x_{\mathrm{u}}\,(k_1\rho^2 + k_2\rho^4 + k_3\rho^6 + O(\rho^8)) \tag{2.11}$$

$$\delta_{\mathrm{r_y}} = \delta_{\mathrm{r}}\sin\vartheta = \delta_{\mathrm{r}}\ {}^{\mathrm{I}}y_{\mathrm{u}}/\rho = {}_{\mathrm{I}}y_{\mathrm{u}}\,(k_1\rho^2 + k_2\rho^4 + k_3\rho^6 + O(\rho^8)) \tag{2.12}$$

that can be directly added to the normalized projections ${}_{\mathrm{I}}\boldsymbol{p}_{\mathrm{u}}$ in order to get the distorted, normalized projections ${}_{\mathrm{I}}\boldsymbol{p}_{\mathrm{d_{radial}}}$:

$${}_{\mathrm{I}}\boldsymbol{p}_{\mathrm{d_{radial}}} = \begin{bmatrix} {}_{\mathrm{I}}x_{\mathrm{u}} + \delta_{\mathrm{r_x}} \\ {}_{\mathrm{I}}y_{\mathrm{u}} + \delta_{\mathrm{r_y}} \end{bmatrix} = \begin{bmatrix} {}_{\mathrm{I}}x_{\mathrm{u}}\,(1 + k_1\rho^2 + k_2\rho^4 + k_3\rho^6 + O(\rho^8)) \\ {}_{\mathrm{I}}y_{\mathrm{u}}\,(1 + k_1\rho^2 + k_2\rho^4 + k_3\rho^6 + O(\rho^8)) \end{bmatrix} . \tag{2.13}$$

This is the so-called u → d (undistorted-to-distorted) formulation that is conform to physical refraction laws. This formulation enables direct calculation of distorted projections out from the ideal, undistorted ones. Unfortunately, in computer vision applications it is the opposite calculations that we are regularly interested in (d→u), and a direct analytical inversion of this formulation is not easily possible. However, an iterative solution for the estimation of undistorted reprojections out of actually tracked (˜), distorted ones ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d_{radial}}}$ can be implemented by repeated distortion of some initial, undistorted projections, updating them until their distorted counterparts match the actually tracked distorted projections ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d_{radial}}}$ in the first place, see Alg. 1. The fastest option is to use the actually tracked, distorted projections themselves as initial (undistorted) values for the first iteration. Alternatively, an extensive look-up table (LUT) can be calculated in advance. As an exception, the simplifying case of third-degree distortion (*i.e.*, $k_1 \in \mathbb{R}$, $k_i = 0\ \forall i \in \mathbb{N}, i > 1$) allows direct undistortion using the Cardan method, see (Devernay and Faugeras, 1995). The formulation can be also approximated for rapid undistortion, see Refs. (Melen, 1994; Heikkilä and Silvén, 1997; Heikkilä, 2000; Wei and De Ma, 1994; Mallon and Whelan, 2004).

It is worth mentioning that, in the literature, many authors use similar formulae to directly *undistort* actual, distorted image projections ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}$, instead of *distorting* normalized projections ${}_{\mathrm{I}}\boldsymbol{p}_{\mathrm{u}}$ (*i.e.*, u→d). I call this inverted approach distorted-to-undistorted formulation (d→u), see Refs. (Tsai, 1987; Willson and Shafer, 1994; Stein, 1993, 1997; Wei and De Ma, 1994; Devernay and Faugeras, 1995; Zhang, 1996; Prescott and McLean, 1997; Heikkilä, 2000; Devernay and Faugeras, 2001; El Melegy and Farag, 2003; Sagawa *et al.*, 2005; Tamaki, 2005). The authors directly compute

object produced by the refraction of *paraxial* rays at a spherical surface (Conrady, 1958).

Algorithm 1 Iterative algorithm for undistortion of feature projections. Note that a faster implementation is possible using analytical Jacobians of the reprojection equations, refer to the documentation in (Strobl *et al.*, 2005).

```
**Require:** Actual (distorted) feature projections ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}_{\mathrm{radial}}}$, intrinsic matrix $\boldsymbol{A}$
    and distortion parameters $\{k_1, k_2, \ldots\}$.
  ${}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{u}} \triangleq {}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}_{\mathrm{radial}}}$        {*initialization using distorted projections*}
  Set desired precision $P$ (e.g. to 0.01 pixels).
  **repeat**
    $\begin{bmatrix} u_{\mathrm{temp}} \\ v_{\mathrm{temp}} \end{bmatrix} = {}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{u}} - \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$
    ${}_{\mathrm{I}}\hat{v}_{\mathrm{u}} = {}^{v_{\mathrm{temp}}}\!/_{\beta}$
    ${}_{\mathrm{I}}\hat{u}_{\mathrm{u}} = {}^{(u_{\mathrm{temp}} - \gamma\, {}_{\mathrm{I}}\hat{v}_{\mathrm{u}})}\!/_{\alpha}$
    ${}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{d}_{\mathrm{radial}}} = {}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{u}} + \begin{bmatrix} k_1\, u_{\mathrm{temp}}\, \left({}_{\mathrm{I}}\hat{u}_{\mathrm{u}}{}^2 + {}_{\mathrm{I}}\hat{v}_{\mathrm{u}}{}^2\right) + k_2\, u_{\mathrm{temp}}\, \left({}_{\mathrm{I}}\hat{u}_{\mathrm{u}}{}^2 + {}_{\mathrm{I}}\hat{v}_{\mathrm{u}}{}^2\right)^2 + \ldots \\ k_1\, v_{\mathrm{temp}}\, \left({}_{\mathrm{I}}\hat{u}_{\mathrm{u}}{}^2 + {}_{\mathrm{I}}\hat{v}_{\mathrm{u}}{}^2\right) + k_2\, v_{\mathrm{temp}}\, \left({}_{\mathrm{I}}\hat{u}_{\mathrm{u}}{}^2 + {}_{\mathrm{I}}\hat{v}_{\mathrm{u}}{}^2\right)^2 + \ldots \end{bmatrix}$
    $\boldsymbol{E} = {}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{d}_{\mathrm{radial}}} - {}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}_{\mathrm{radial}}}$        {*distortion error*}
    ${}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{u}} = {}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{u}} - \boldsymbol{E}$        {*update undistorted projections*}
  **until** norm($\boldsymbol{E}$) $< P$        {*four iterations should suffice*}
  **return** ${}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{d}_{\mathrm{radial}}}$.
```

the radial distance ρ out from distorted image projections ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}$, which enables direct undistortion of projections without the need for an iterative process like Alg. 1 or an LUT. The approach is incorrect on a strict, physical ground, but my own simulations show that it may also deliver accurate results—provided that the camera calibration process also follows the d→u formulation. It is worth noting that, on all tested cameras, the physically-conform u→d formulation did actually deliver more accurate results, cf. (Strobl and Hirzinger, 2008).

The presented formulation based on Snell's refraction law is the customary model representation. However, other phenomenological models exist: polynomial terms were used by Ebner and others in (Ebner, 1976; Konecny and Lehmann, 1985), and a similar approach using a bicubic model was introduced by Kilpelä in (Kilpelä, 1980). In (Faugeras and Toscani, 1987) the authors correct distortion using bilinear transformations in image patches. Fitzgibbon and Brauer-Burchardt *et al.* introduced the division model for catadioptric systems in (Fitzgibbon, 2001; Brauer-Burchardt and Voss, 2001), also used by Barreto in (Barreto, 2006). After that, Perwass and Sommer extended it into the inversion model, see (Perwass and Sommer, 2006). In (Claus and Fitzgibbon, 2005) the authors present a model based on a rational function. Devernay and Faugeras propose the field of view (FOV) model (mainly for fish-eye lenses) in (Devernay and Faugeras, 2001). Qiu and De Ma employ a non-parametric model, see (Qiu and Ma, 1995), and Ma *et al.* present a further polynomial distortion model in (Ma *et al.*, 2003; Ma *et al.*, 2003). Last, it is worth mentioning the non-phenomenological approach in (Wang *et al.*, 2008) where the authors develop the abovementioned formulation u→d taking geometrical inaccuracies into account.

- The ***decentering distortion*** model aims at compensating for potentially erroneous assembly of lenses within a lens unit concerning their centering accuracy to each other as well as w.r.t. the imaging frame, *i.e.*, whether all components are strictly collinear to each other. Different from radial distortion where deviations only show radial components, this type of distortion features both, radial and tangential components.

  The original formulation seems to have been originally delivered by Conrady in (Conrady, 1919, 1958, 1960) and further explained by his son-in-law Kingslake in (Kingslake, 1992); it was also used in (Brown, 1966, 1971; Weng *et al.*, 1992) and beyond. Decentering distortion contains radial and tangential components as follows:

$$\delta_{\mathrm{d_r}} = 3(j_1\rho^2 + j_2\rho^4 + \cdots)\sin(\vartheta - \vartheta_0) \tag{2.14}$$

$$\delta_{\mathrm{d_t}} = (j_1\rho^2 + j_2\rho^4 + \cdots)\cos(\vartheta - \vartheta_0) \tag{2.15}$$

  where $\vartheta = \arctan({}_{\mathrm{I}}y_{\mathrm{u}}/{}_{\mathrm{I}}x_{\mathrm{u}})$ is, again, the in-plane angular position of the feature in the reference frame S_{I}, and ϑ_0 sets the angular direction of maximum tangential decentering distortion, see Fig. 2.11.

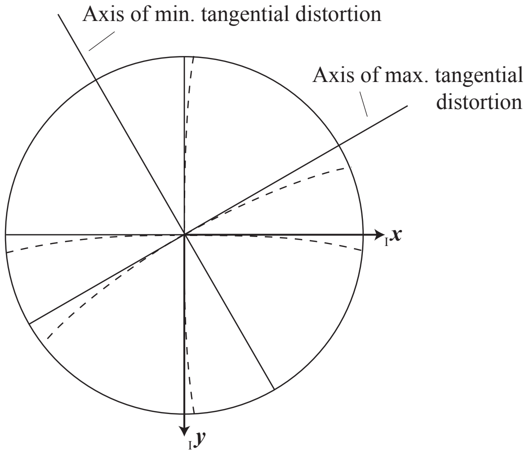

Axis of min. tangential distortion
Axis of max. tangential distortion
Ix
Iy

Figure 2.11: Angular directions of maximum and minimum tangential distortions and their effects (dashed curves).

In Cartesian representation we have:

$$\begin{bmatrix}\delta_{\mathrm{d_x}}\\ \delta_{\mathrm{d_y}}\end{bmatrix} = \begin{bmatrix}\cos\vartheta & -\sin\vartheta\\ \sin\vartheta & \cos\vartheta\end{bmatrix}\begin{bmatrix}\delta_{\mathrm{d_r}}\\ \delta_{\mathrm{d_t}}\end{bmatrix} \quad , \tag{2.16}$$

where, letting $p_1 = -j_1 \sin\vartheta_0$ and $p_2 = j_1 \cos\vartheta_0$, we substitute Eqs. (2.14) and (2.15) in Eq. (2.16) and, releasing j_1, we have:

$$\begin{aligned} \delta_{\mathrm{d_x}} &= p_1\,(3\,{}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2) \;+\; 2\,p_2\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}} \;+ O\left(({}_{\mathrm{I}}x_{\mathrm{u}},\,{}_{\mathrm{I}}y_{\mathrm{u}})^4\right) \quad , &(2.17)\\ \delta_{\mathrm{d_y}} &= 2\,p_1\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}} \;+\; p_2({}_{\mathrm{I}}x_{\mathrm{u}}^2 + 3\,{}_{\mathrm{I}}y_{\mathrm{u}}^2) \;+ O\left(({}_{\mathrm{I}}x_{\mathrm{u}},\,{}_{\mathrm{I}}y_{\mathrm{u}})^4\right) \quad . &(2.18)\end{aligned}$$

If the further DoF j_2 is required, letting $p_3 = -j_2 \sin\vartheta_0$ and $p_4 = j_2 \cos\vartheta_0$, the former equations extend to:

$$\begin{aligned} \delta_{\mathrm{d_x}} &= p_1\,(3\,{}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2) \;+\; 2\,p_2\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}}\\ &\quad + \rho^2\left(p_3\,(3\,{}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2\,) \;+\; 2\,p_4\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}}\right) + O\left(({}_{\mathrm{I}}x_{\mathrm{u}},\,{}_{\mathrm{I}}y_{\mathrm{u}})^6\right) \quad , &(2.19)\\ \delta_{\mathrm{d_y}} &= 2\,p_1\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}} \;+\; p_2({}_{\mathrm{I}}x_{\mathrm{u}}^2 + 3\,{}_{\mathrm{I}}y_{\mathrm{u}}^2)\\ &\quad + \rho^2\left(\,2\,p_3\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}} \;+\; p_4\,({}_{\mathrm{I}}x_{\mathrm{u}}^2 + 3\,{}_{\mathrm{I}}y_{\mathrm{u}}^2\,)\right) + O\left(({}_{\mathrm{I}}x_{\mathrm{u}},\,{}_{\mathrm{I}}y_{\mathrm{u}})^6\right) \quad . &(2.20)\end{aligned}$$

It is worth noting that in Eqs. (2.17) and (2.18) both parameters p_1 and p_2 stem from j_1 and ϑ_0 and have to be estimated independently of each other; the situation is different in Eqs. (2.19) and (2.20) where p_3 and p_4 *are* correlated since $p_4 = {}^{(p_3\,p_2)}\!/_{p_1}$.

The decentering distortion model has a close connection to the former radial distortion model: I already mentioned that radial distortion is defined w.r.t. the principal point of the camera in S_{I}. However, the attentive reader may have noticed that radial distortion of the lens unit and the principal point of the camera are essentially isolated entities that, subject to the accuracy of the assembly process, may actually differ to each other. As a consequence, in the presence of radial distortion, the estimated location of the principal point primarily locates the origin of this distortion (Weng *et al.*, 1992; Willson and Shafer, 1994), so that the actual principal point is shifted. Any deviation from the actual projective principal point implies a different orientation and translation of the imaging sensor (Tsai, 1987). These effects are usually small and can be neglected. However, some cameras really require a decoupling of both, the actual origin of radial distortion and the projective principal point; in (Stein, 1993) the author describes how decentering distortion in its first DoF (Eqs. (2.17) and (2.18)) is equivalent to releasing the center of radial distortion. Indeed, in many experiments I did came upon this effect: Extremely cheap cameras like webcams, which components have not been assembled with the required accuracy, may indeed benefit from the release of the center of radial distortion by expecting decentering distortion. As far as my experience goes, this only concerns *extremely* cheap, low-end cameras; the user should then consider this distortion model during calibration in Section 3.2. As a matter of fact, this seems the only context in which explicit modeling of decentering distortion seems justified.

- ***Thin prism distortion*** also arises from imperfect lens assembly of the component lenses of lens units or of its sensing frame. An example is tilting of lens components or the sensing image frame. This distortion has been traditionally modeled in the literature by a thin prism effect both, with radial and tangential components, see (Brown, 1966):

  $$\delta_{\mathrm{t_r}} = (i_1\,\rho^2 + i_2\,\rho^4 + \cdots)\sin(\vartheta - \vartheta_1) \tag{2.21}$$
  $$\delta_{\mathrm{t_t}} = (i_1\,\rho^2 + i_2\,\rho^4 + \cdots)\cos(\vartheta - \vartheta_1) \tag{2.22}$$

  where $\vartheta = \arctan({}_{\mathrm{I}}y_{\mathrm{u}}/{}_{\mathrm{I}}x_{\mathrm{u}})$ is, again, the in-plane angular position of the feature in the reference frame S_{I}, and ϑ_1 sets the angular direction of maximum tangential thin prism distortion.

  In Cartesian representation and letting $s_1 = -i_1 \sin\vartheta_1$ and $s_2 = i_1\cos\vartheta_1$ we have:

  $$\begin{bmatrix}\delta_{\mathrm{t_x}}\\ \delta_{\mathrm{t_y}}\end{bmatrix} = \begin{bmatrix}\cos\vartheta & -\sin\vartheta\\ \sin\vartheta & \cos\vartheta\end{bmatrix}\begin{bmatrix}\delta_{\mathrm{t_r}}\\ \delta_{\mathrm{t_t}}\end{bmatrix} = \begin{bmatrix}s_1\,({}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2) + O\left(({}_{\mathrm{I}}x_{\mathrm{u}},\,{}_{\mathrm{I}}y_{\mathrm{u}})^4\right)\\ s_2\,({}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2) + O\left(({}_{\mathrm{I}}x_{\mathrm{u}},\,{}_{\mathrm{I}}y_{\mathrm{u}})^4\right)\end{bmatrix}. \tag{2.23}$$

  This distortion model is not being required in modern cameras.

Radial, decentering, and thin prism distortion models may be added together. For instance, in the case of one DoF for each of the models, we have:

$$\begin{aligned}\delta_{\mathrm{x}} &= \delta_{\mathrm{r_x}} + \delta_{\mathrm{d_x}} + \delta_{\mathrm{t_x}} = k_1\,{}_{\mathrm{I}}x_{\mathrm{u}}({}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2)\\ &\qquad + p_1\,(3\,{}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2) + 2\,p_2\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}}\\ &\qquad + s_1\,({}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2) \end{aligned}\tag{2.24}$$
$$\begin{aligned}\delta_{\mathrm{y}} &= \delta_{\mathrm{r_y}} + \delta_{\mathrm{d_y}} + \delta_{\mathrm{t_y}} = k_1\,{}_{\mathrm{I}}y_{\mathrm{u}}({}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2)\\ &\qquad + 2\,p_1\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}} + p_2({}_{\mathrm{I}}x_{\mathrm{u}}^2 + 3\,{}_{\mathrm{I}}y_{\mathrm{u}}^2)\\ &\qquad + s_2\,({}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2) \end{aligned}\tag{2.25}$$

or, in a more compact way, letting $g_1 = s_1 + p_1$, $g_2 = s_2 + p_2$, $g_3 = 2\,p_1$ and $g_4 = 2\,p_2$:

$$\delta_{\mathrm{x}} = (g_1 + g_3)\,{}_{\mathrm{I}}x_{\mathrm{u}}{}^2 + g_4\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}} + g_1\,{}_{\mathrm{I}}y_{\mathrm{u}}{}^2 + k_1\,{}_{\mathrm{I}}x_{\mathrm{u}}\,({}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2) \tag{2.26}$$
$$\delta_{\mathrm{y}} = g_2\,{}_{\mathrm{I}}x_{\mathrm{u}}{}^2 + g_3\,{}_{\mathrm{I}}x_{\mathrm{u}}\,{}_{\mathrm{I}}y_{\mathrm{u}} + (g_2 + g_4)\,{}_{\mathrm{I}}y_{\mathrm{u}}{}^2 + k_1\,{}_{\mathrm{I}}y_{\mathrm{u}}\,({}_{\mathrm{I}}x_{\mathrm{u}}^2 + {}_{\mathrm{I}}y_{\mathrm{u}}^2) \tag{2.27}$$

with the result that

$$\begin{bmatrix}{}_{\mathrm{I}}x_{\mathrm{d}}\\ {}_{\mathrm{I}}y_{\mathrm{d}}\end{bmatrix} = \begin{bmatrix}{}_{\mathrm{I}}x_{\mathrm{u}} + \delta_{\mathrm{x}}\,({}_{\mathrm{I}}x_{\mathrm{u}},\,{}_{\mathrm{I}}y_{\mathrm{u}})\\ {}_{\mathrm{I}}y_{\mathrm{u}} + \delta_{\mathrm{y}}\,({}_{\mathrm{I}}x_{\mathrm{u}},\,{}_{\mathrm{I}}y_{\mathrm{u}})\end{bmatrix}\quad, \tag{2.28}$$

which can be also used in the context of the undistortion function undist$(\cdot)$ in Eq. (2.9), together with the actual projections ${}_{\mathrm{I}}\tilde{\boldsymbol{p}}_{\mathrm{d}}$, *i.e.*,

$${}_{\mathrm{M}}\bar{\boldsymbol{p}}_{\mathrm{u}} = \begin{bmatrix}{}_{\mathrm{M}}\tilde{x}_{\mathrm{u}}\\ {}_{\mathrm{M}}\tilde{y}_{\mathrm{u}}\\ 1\end{bmatrix} = \begin{bmatrix}\alpha & \gamma & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}{}_{\mathrm{I}}x_{\mathrm{u}}\\ {}_{\mathrm{I}}y_{\mathrm{u}}\\ 1\end{bmatrix} = \begin{bmatrix}\alpha & \gamma & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\mathrm{undist}\,({}_{\mathrm{I}}\tilde{\boldsymbol{p}}_{\mathrm{d}})\\ 1\end{bmatrix} \tag{2.29}$$

to obtain implicit, normalized projections $\{_{\mathrm{I}}\hat{x}_{\mathrm{u}},\ _{\mathrm{I}}\hat{y}_{\mathrm{u}}\}$ that, in turn, allow us to infer 3-D directions in Euclidean space in S_{C}.

Note that a general statement on the number of DoF required for accurate modeling of lens distortion is not possible, as camera systems are highly diverse. It is mostly during camera calibration that the user can assess the significance of the released parameters for any particular camera, see Section 3.2. The only valid general statement is that the smaller the AOV (*i.e.*, the bigger the ratio focal length to size of the imaging sensor), the less effect radial distortion has in the image.

These distortion parameters are then considered part of the interior orientation of the camera, *i.e.*, included in the intrinsic parameters.

It is worth noting that, even though the actual image is distorted, this does not mean that the user needs to undistort the whole image in advance of image processing e.g. in Chapter 4. Much on the contrary, undistortion (and image warping in general) would falsify noise models (due to averaging) and introduce further errors like aliasing; therefore, image processing on the original, distorted footage is preferable (except in special cases e.g. when straight lines have to be found in an efficient way).

Extrinsic Geometry

Until now the camera operation has been modeled w.r.t. the Euclidean camera frame of the camera S_{C}, see Eq. (2.8). However, that reference frame lies within the camera (more specifically, at the frontal area of the lens unit); results on that reference frame are not always useful. In general, world coordinates in the world frame S_0 are preferred, which have to be transformed into camera coordinates in S_{C}. We can generalize Eq. (2.8) in world coordinates as follows:

$$_{\mathrm{M}}\bar{\boldsymbol{p}}_{\mathrm{u}} = \begin{bmatrix} _{\mathrm{M}}x_{\mathrm{u}} \\ _{\mathrm{M}}y_{\mathrm{u}} \\ 1 \end{bmatrix} \propto \underbrace{\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}_{(3\times3)}} \begin{bmatrix} _{\mathrm{C}}x \\ _{\mathrm{C}}y \\ _{\mathrm{C}}z \end{bmatrix} = \underbrace{\boldsymbol{A}_{(3\times3)}\ _{\mathrm{C}}\boldsymbol{T}^{0}_{(3\times4)}}_{\boldsymbol{P}_{(3\times4)}} \begin{bmatrix} _{0}x \\ _{0}y \\ _{0}z \\ 1 \end{bmatrix} = \boldsymbol{P}_{(3\times4)}\ _{0}\bar{\boldsymbol{p}}\ , \tag{2.30}$$

where $_{\mathrm{C}}\boldsymbol{T}^{0}_{(3\times4)} = [\boldsymbol{R}\,\boldsymbol{t}\,]$ is the rigid body transformation between the camera frame S_{C} and the world frame S_0. The resulting matrix $\boldsymbol{P}_{(3\times4)}$ is called perspective projection matrix; it consists of the camera intrinsic matrix $\boldsymbol{A}$ and the transformation $_{\mathrm{C}}\boldsymbol{T}^{0}_{(3\times4)}$.

On occasions, the transformation $_{\mathrm{C}}\boldsymbol{T}^{0}_{(3\times4)}$ is neither useful nor easy to compute. For instance, cameras are usually mounted on robots to be able to actively control the camera's field of view (FOV). In this case, it is possible to determine the transformation $_{\mathrm{C}}\boldsymbol{T}^{0}_{(3\times4)}$ with the help of intermediate transformations related to the manipulator readings as follows:

$$_{\mathrm{C}}\boldsymbol{T}^{0}_{(3\times4)} = {}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}_{(3\times4)}\ {}_{\mathrm{T}}\widetilde{\boldsymbol{T}}^{\mathrm{B}}\ {}_{\mathrm{B}}\boldsymbol{T}^{0}\ , \tag{2.31}$$

where $_{\mathrm{T}}\widetilde{\boldsymbol{T}}^{\mathrm{B}}$ is the *homogeneous* (size 4×4) transformation matrix between the

tool **c**enter **p**oint (TCP) reference frame S_{T} of the robot and its base reference frame S_{B}. Its value is measured $(\widetilde{\;})$ in the form of the manipulator's readings. Furthermore, ${}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}{}_{(3\times4)}$ and ${}_{\mathrm{B}}\boldsymbol{T}^{0}$ are *static* transformations, so-called eye-hand and base-to-world transformations, which can be estimated from images, see Section 3.3. Altogether:

$$ {}_{\mathrm{M}}\bar{\boldsymbol{p}}_{\mathrm{u}} = \begin{bmatrix} {}_{\mathrm{M}}x_{\mathrm{u}} \\ {}_{\mathrm{M}}y_{\mathrm{u}} \\ 1 \end{bmatrix} \propto \underbrace{\boldsymbol{A}_{(3\times3)}\; {}_{\mathrm{C}}\boldsymbol{T}^{T}_{(3\times4)}\; {}_{\mathrm{T}}\widetilde{\boldsymbol{T}}^{\mathrm{B}}\; {}_{\mathrm{B}}\boldsymbol{T}^{0}}_{\boldsymbol{P}_{(3\times4)}} \begin{bmatrix} {}_{0}x \\ {}_{0}y \\ {}_{0}z \\ 1 \end{bmatrix} = \boldsymbol{P}_{(3\times4)}\; {}_{0}\bar{\boldsymbol{p}}\,. \qquad (2.32) $$

In contrast to the intrinsic parameters (composed of the the intrinsic matrix $\boldsymbol{A}$ and the lens distortion parameters), these unknown transformations ${}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}$ and ${}_{\mathrm{B}}\boldsymbol{T}^{0}$ are called *extrinsic parameters* as they represent the external orientation of the camera, *i.e.*, the pose of S_{C} w.r.t. external frames.

2.2.2 The DLR Laser Stripe Profiler

Description

The DLR Laser Stripe Profiler (LSP) consists of one or two laser beams generating planes of laser light together with one or two video cameras (Strobl *et al.*, 2004; Suppa *et al.*, 2007). Its fundamental principle of range sensing is optical laser light triangulation as illustrated in Fig. 2.12: The laser beam spreads through a cylindrical lens to a laser plane, illuminating a stripe on a surface, and the video camera records its reflection. From this projection, the LSP effectively delivers a contour of depths over the emitted laser plane by triangulation, *i.e.*, intersecting the laser plane with the rays of sight corresponding to the laser stripe projections in S_{I}. Of course, the pose of the plane w.r.t. the camera as well as the intrinsic parameters of the camera are required for valid triangulation, refer to Section 3.6.

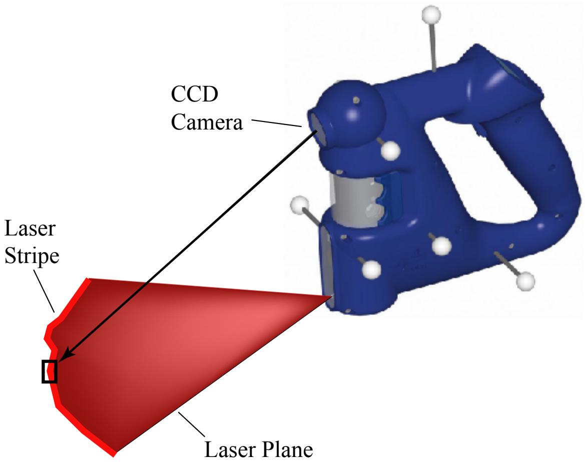


Figure 2.12: The Laser Stripe Profiler at an older version of the DLR 3D-Modeler.

Perhaps the best description of its operational principle is that of the LSP acting as a modified stereo camera as explained in the last section, with the exception that one of the cameras has become active, being substituted by a laser plane generator. By doing this, the correspondence problem between features in both cameras is alleviated as it is easier and more robust to search for laser projections rather than to look for feature similarity—this is especially so if the camera is filtered to laser light wavelength. It is worth noting that the LSP mounted on the DLR 3D-Modeler works without optical filters on the cameras, as they would make concurrent stereo vision, texturing, or visual pose tracking as in Chapter 5 impossible.

On the other hand, the LSP is fundamentally limited to 2-D geometric information about the scene, *i.e.*, range data spread in 1-D only, whereas stereo cameras produce dense, 3-D geometric information. In the ideal case, the LSP delivers a contour of depths corresponding to the projection of the laser plane onto the scene. It is therefore necessary to sweep the sensor across the scene to gather 3-D information, registering the gathered stripes in 3-D e.g. by means of pose tracking of the camera.

It is worth noting that the laser power can be regulated, with a maximum allowed power of 5 mW. The LSP is registered laser safety class 2M that cannot harm the user's eyes except in cases of prolonged stare directly at the laser beam ($t > 0.25$ s). In addition, laser beams are triggered in a pulsed mode so that they are only active during shutter time of their respective cameras—thereby minimizing their effective irradiation.

Because of its precision, robustness, the extended operating range (cf. Section 4.3.4), and the speed of acquisition, the LSP is widely used within the DLR 3D-Modeler as a high definition, short- to middle-range sensor.

Description—Dual, Crosshair LSP

It is usually said that scanning with this type of sensor is virtually like spray can painting; this is especially so if the sensor is being hand-held by the user. However, this is not completely true, as the user does not have the freedom to horizontally move in the direction of the laser stripe while scanning, but only to sweep up and down the sensor. What is more, while automatically scanning with a robotic manipulator, this fact constraints the robot motion and entails the waste of one of its valuable DoF.

In order to get rid of this constraint, a second laser beam has been integrated that illuminates perpendicularly to the former stripe (from here *crosshair*), see Fig. 2.13 and the conference paper to this novel contribution (Suppa *et al.*, 2007). Due to construction-related constraints, both laser beams have to be closely placed, which entails an inconvenient reduction of the basis distances between each laser plane and the main camera—compromising accuracy at that. It comes handy that the DLR 3D-Modeler features a second camera; we decide then to use both cameras *sequentially*, each one performing single LSP with its farthest laser plane, so that basis distances remain long. We call this configuration the *dual, crosshair LSP*.

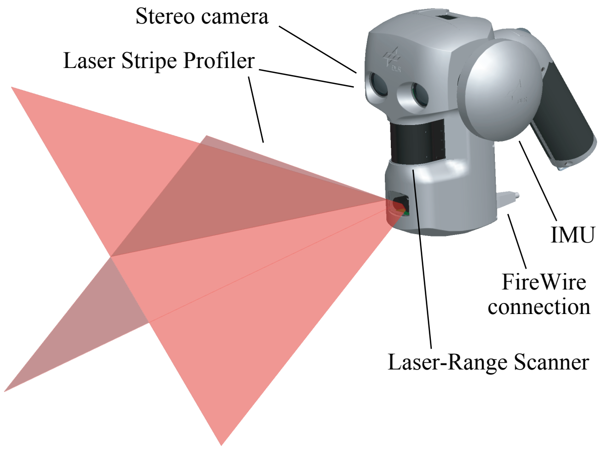


Figure 2.13: The crosshair LSP within the DLR 3D-Modeler.

This novel crosshair configuration yields the following advantages:

1. One DoF in motion planning during scanning is released w.r.t. the traditional LSP.

2. The amount of surface-related information gained in any motion direction increases, cf. Fig. 2.14.

3. Since both single LSPs may be triggered in a complementary way, it is possible to duplicate sensing rate provided each camera can still work at highest speed (limiting their shutter time).

This novel development in 2004 was the first laser stripe profiler of its sort; paralelly, Creaform Inc. developed the HandyScan 3D—commercially available in 2005 following the same paradigm.

Triangulation between Laser and Camera

The simple triangulation process of the LSP is depicted in Fig. 2.12. The hybrid triangulation process is analogous but with pulsed laser planes of the crosshair LSP, each operating in simple triangulation process with its more distant camera, see Fig. 2.13.

In this section I detail the laser plane model and the simple triangulation process using the camera model already presented in Section 2.2.1, the camera calibration method in Section 3.2, as well as the laser calibration process in Section 3.6. I shall provide the mathematical formulation required to compute 3-D coordinates of projected lased stripes in the camera reference frame S_{C} or in the TCP reference frame S_{T}.

Since LSP measurements essentially originate from images, I make use of the whole formulae in the last Section 2.2.1. As I shall mention in Section 3.6, it is convenient to implicitly exploit the latter model in order to keep its model simple so that its calibration process is subject to as less degrees of freedom as possible. I adopt Eq. (2.32) aiming at 3-D results ${}_{\mathrm{T}}\boldsymbol{p}$ in the TCP frame S_{T} of the DLR 3D-Modeler as follows:

$$
{}_{\mathrm{M}}\bar{\boldsymbol{p}}_{\mathrm{u}} = \begin{bmatrix} {}_{\mathrm{M}}x_{\mathrm{u}} \\ {}_{\mathrm{M}}y_{\mathrm{u}} \\ 1 \end{bmatrix} \propto \underbrace{\boldsymbol{A}_{(3\times 3)}\ {}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}_{(3\times 4)}}_{\boldsymbol{P}_{(3\times 4)}} \begin{bmatrix} {}_{\mathrm{T}}x \\ {}_{\mathrm{T}}y \\ {}_{\mathrm{T}}z \\ 1 \end{bmatrix} = \boldsymbol{P}_{(3\times 4)}\ {}_{\mathrm{T}}\bar{\boldsymbol{p}} \quad . \tag{2.33}
$$

Note that the undistorted, virtual projections ${}_{\mathrm{M}}\boldsymbol{p}_{\mathrm{u}}$ have to be calculated out from actual, distorted projections ${}_{\mathrm{M}}\boldsymbol{p}_{\mathrm{d}}$ in a previons stage of undistortion as explained in Eq. (2.9), Section 2.2.1.

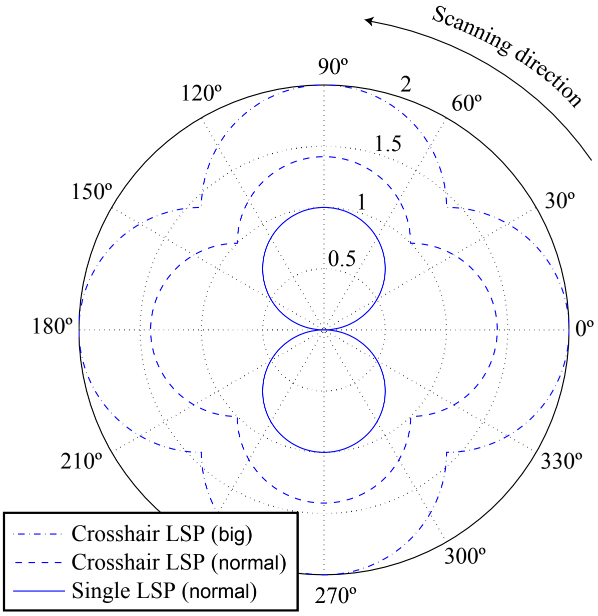


Figure 2.14: This graph shows, in polar coordinates, the surface area scanned at constant speed and range w.r.t. the object, in every scanning direction. The single LSP does not yield any information when sweeping it horizontally (see 0° and 180° in the graph), whereas the dual crosshair LSP gains a rather constant amount of information irrespective of the scanning direction (flower-like contours). Furthermore, as the stripe is being obliquely projected in the images, the FOV increases in size: (normal) refers to a crosshair LSP with laser planes featuring the same opening angle as in the case of a single LSP, whereas (big) refers to a crosshair LSP with wider opening angles corresponding to the diagonal projection of laser stripes on the images.

More in detail, the perspective projection matrix $\boldsymbol{P}_{(3\times4)}$ is composed of intrinsic and extrinsic camera parameters as follows:

$$\begin{aligned}\boldsymbol{P}_{(3\times4)} &= \boldsymbol{A}_{(3\times3)}\ {}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}_{(3\times4)} = \begin{bmatrix}\alpha & \gamma & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_{\mathrm{x}}\\ r_{21} & r_{22} & r_{23} & t_{\mathrm{y}}\\ r_{31} & r_{32} & r_{33} & t_{\mathrm{z}}\end{bmatrix} = \\ &= \left[\begin{array}{ccc|c}\alpha r_{11}+\gamma r_{21}+u_0 r_{31} & \alpha r_{12}+\gamma r_{22}+u_0 r_{32} & \alpha r_{13}+\gamma r_{23}+u_0 r_{33} & \alpha t_{\mathrm{x}}+\gamma t_{\mathrm{y}}+u_0 t_{\mathrm{z}}\\ \beta r_{21}+v_0 r_{31} & \beta r_{22}+v_0 r_{32} & \beta r_{23}+v_0 r_{33} & \beta t_{\mathrm{y}}+v_0 t_{\mathrm{z}}\\ r_{31} & r_{32} & r_{33} & t_{\mathrm{z}}\end{array}\right] \\ &= \left[\begin{array}{c|c}\boldsymbol{q}_1^{\mathrm{T}} & q_{14}\\ \boldsymbol{q}_2^{\mathrm{T}} & q_{24}\\ \boldsymbol{q}_3^{\mathrm{T}} & q_{34}\end{array}\right] . \end{aligned} \tag{2.34}$$

This system of equations is underdetermined for finding ${}_{\mathrm{T}}\boldsymbol{p}$ as it is only possible to obtain the direction of projection $\|{}_{\mathrm{C}}\boldsymbol{p}\|$. It is by the intersection of that direction with the laser plane that the matrix of the system of equations really gets full rank.

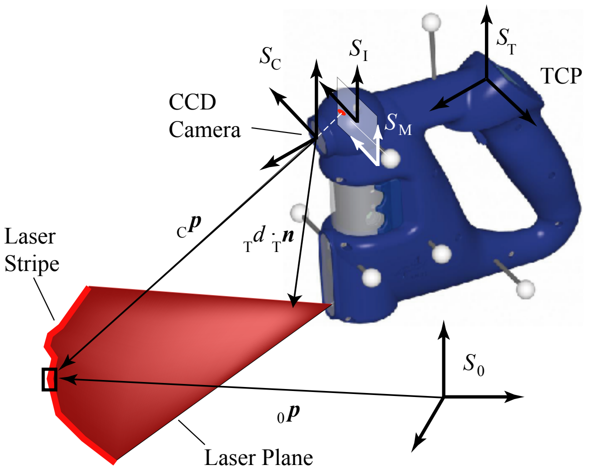


Figure 2.15: 3-D reconstruction at the LSP of the DLR 3D-Modeler.

In Fig. 2.15 the single triangulation process of the LSP is depicted. From that figure it is clear that the only remaining geometry to be described is the pose of the laser plane w.r.t. the camera reference frame S_{C}. The laser plane originates in a single laser beam that is spread out by passing through a cylindrical lens. Both, the laser illuminant and the camera, are rigidly fixed w.r.t. the TCP frame S_{T}, hence we have the fixed geometrical relationship:

$$\left[{}_{\mathrm{T}}\boldsymbol{n}^{\mathsf{T}}\ {}_{\mathrm{T}}d\right]{}_{\mathrm{T}}\bar{\boldsymbol{p}} = 0 \quad , \quad {}_{\mathrm{T}}d > 0 \quad , \tag{2.35}$$

called Hessian normal form of the plane in the TCP reference frame S_{T}, that defines the laser plane points in relation to the pose of the laser plane w.r.t. the camera frame S_{C}: ${}_{\mathrm{T}}\boldsymbol{n}$ is the normal vector to the laser plane in S_{T} and ${}_{\mathrm{T}}d$ is the distance on that vector between the origin of S_{T} and the laser plane.

These parameters feature 3 DoF (two for the orientation of ${}_{\mathrm{T}}\boldsymbol{n}$ and one for the distance ${}_{\mathrm{T}}d$) and are determined during the calibration stage in Section 3.6. The laser plane must not meet the optical center of its respective camera.

Only now we are in a position to solve for ${}_{\mathrm{T}}\boldsymbol{p}$ by joining Eqs. (2.33) and (2.35) as follows:

$$\left.\begin{array}{rcl} {}_{\mathrm{M}}x_{\mathrm{u}} \overset{(2.33)}{=} \dfrac{\boldsymbol{q}_1^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}} + q_{14}}{\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}} + q_{34}} & \equiv & \left(\boldsymbol{q}_1^{\mathsf{T}} - \boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{M}}x_{\mathrm{u}}\right){}_{\mathrm{T}}\hat{\boldsymbol{p}} = q_{34}\,{}_{\mathrm{M}}x_{\mathrm{u}} - q_{14} \\ {}_{\mathrm{M}}y_{\mathrm{u}} \overset{(2.33)}{=} \dfrac{\boldsymbol{q}_2^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}} + q_{24}}{\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}} + q_{34}} & \equiv & \left(\boldsymbol{q}_2^{\mathsf{T}} - \boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{M}}y_{\mathrm{u}}\right){}_{\mathrm{T}}\hat{\boldsymbol{p}} = q_{34}\,{}_{\mathrm{M}}y_{\mathrm{u}} - q_{24} \\ {}_{\mathrm{T}}\boldsymbol{n}^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}} + {}_{\mathrm{T}}d \overset{(2.35)}{=} 0 & \equiv & {}_{\mathrm{T}}\boldsymbol{n}^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}} = -{}_{\mathrm{T}}d \end{array}\right\} , \quad (2.36)$$

which represent a linear system of equations in the form $\boldsymbol{F}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}} = \boldsymbol{b}$. In all practical cases the system's solution reads

$${}_{\mathrm{T}}\hat{\boldsymbol{p}} = \boldsymbol{F}^{-1}\boldsymbol{b} \quad . \quad (2.37)$$

Extrinsic Geometry

The triangulation equations in the last section are conceived in general terms relating a laser plane with its TCP reference frame S_{T} or its camera frame S_{C}. However, as mentioned above, there exist two laser planes and two cameras and their results have to be correctly registered to each other. Further, the TCP of the DLR 3D-Modeler may correspond to different external tracking systems like the infrared tracking system ARTtrack2, robotic manipulators—either passive like the FaroArm Gold or active like the Kuka KR 16, or even by purely vision-based pose tracking as in Chapter 5. To that effect, Eq. (2.33) has to be adjusted accordingly for both laser units of the crosshair LSP at the DLR 3D-Modeler as follows:

$${}_{\mathrm{M_{left}}}\bar{\boldsymbol{p}}_{\mathrm{u}} = \begin{bmatrix} {}_{\mathrm{M_{left}}}x_{\mathrm{u}} \\ {}_{\mathrm{M_{left}}}y_{\mathrm{u}} \\ 1 \end{bmatrix} \propto \underbrace{\boldsymbol{A}_{\mathrm{left}}\;{}_{\mathrm{C_{left}}}\boldsymbol{T}^{\mathrm{T}}_{(3\times4)}}_{\boldsymbol{P}_{\mathrm{left}}} \begin{bmatrix} {}_{\mathrm{T}}x_{\mathrm{left}} \\ {}_{\mathrm{T}}y_{\mathrm{left}} \\ {}_{\mathrm{T}}z_{\mathrm{left}} \\ 1 \end{bmatrix} = \boldsymbol{P}_{\mathrm{left}}\;{}_{\mathrm{T}}\bar{\boldsymbol{p}}_{\mathrm{left}} \quad , \quad (2.38)$$

$${}_{\mathrm{M_{right}}}\bar{\boldsymbol{p}}_{\mathrm{u}} = \begin{bmatrix} {}_{\mathrm{M_{right}}}x_{\mathrm{u}} \\ {}_{\mathrm{M_{right}}}y_{\mathrm{u}} \\ 1 \end{bmatrix} \propto \underbrace{\boldsymbol{A}_{\mathrm{left}}\;{}_{\mathrm{C_{right}}}\boldsymbol{T}^{\mathrm{T}}_{(3\times4)}}_{\boldsymbol{P}_{\mathrm{right}}} \begin{bmatrix} {}_{\mathrm{T}}x_{\mathrm{right}} \\ {}_{\mathrm{T}}y_{\mathrm{right}} \\ {}_{\mathrm{T}}z_{\mathrm{right}} \\ 1 \end{bmatrix} = \boldsymbol{P}_{\mathrm{right}}\;{}_{\mathrm{T}}\bar{\boldsymbol{p}}_{\mathrm{right}} \; , \quad (2.39)$$

where ${}_{\mathrm{C_{right}}}\boldsymbol{T}^{\mathrm{T}} = {}_{\mathrm{C_{right}}}\boldsymbol{T}^{\mathrm{C_{left}}}\,{}_{\mathrm{C_{left}}}\boldsymbol{T}^{\mathrm{T}}$; ${}_{\mathrm{C_{right}}}\boldsymbol{T}^{\mathrm{C_{left}}}$ results from the stereo camera calibration, see Section 3.2. It holds:

$$\boldsymbol{P}_{\mathrm{left}} = \boldsymbol{A}_{\mathrm{left}}\;{}_{\mathrm{C_{left}}}\boldsymbol{T}^{\mathrm{T}}_{(3\times4)} = \left[\begin{array}{c|c} {}_{\mathrm{l}}\boldsymbol{q}_1^{\mathsf{T}} & {}_{\mathrm{l}}q_{14} \\ {}_{\mathrm{l}}\boldsymbol{q}_2^{\mathsf{T}} & {}_{\mathrm{l}}q_{24} \\ {}_{\mathrm{l}}\boldsymbol{q}_3^{\mathsf{T}} & {}_{\mathrm{l}}q_{34} \end{array}\right] \quad , \quad (2.40)$$

$$\boldsymbol{P}_{\mathrm{right}} = \boldsymbol{A}_{\mathrm{right}}\;{}_{\mathrm{C_{right}}}\boldsymbol{T}^{\mathrm{T}}_{(3\times4)} = \left[\begin{array}{c|c} {}_{\mathrm{r}}\boldsymbol{q}_1^{\mathsf{T}} & {}_{\mathrm{r}}q_{14} \\ {}_{\mathrm{r}}\boldsymbol{q}_2^{\mathsf{T}} & {}_{\mathrm{r}}q_{24} \\ {}_{\mathrm{r}}\boldsymbol{q}_3^{\mathsf{T}} & {}_{\mathrm{r}}q_{34} \end{array}\right] \quad . \quad (2.41)$$

Instead of using Eq. (2.35), Eqs. (2.38) and (2.39) now meet the following laser planes:

$$\begin{bmatrix} {}_{\mathrm{T}}\boldsymbol{n}^{\mathsf{T}}_{\mathrm{right}} & {}_{\mathrm{T}}d_{\mathrm{right}} \end{bmatrix} {}_{\mathrm{T}}\bar{\boldsymbol{p}}_{\mathrm{left}} = 0 \quad \text{and} \tag{2.42}$$

$$\begin{bmatrix} {}_{\mathrm{T}}\boldsymbol{n}^{\mathsf{T}}_{\mathrm{left}} & {}_{\mathrm{T}}d_{\mathrm{left}} \end{bmatrix} {}_{\mathrm{T}}\bar{\boldsymbol{p}}_{\mathrm{right}} = 0 \quad , \tag{2.43}$$

respectively, where $\boldsymbol{p}_{\{\mathrm{left,right}\}}$ are points seen with the {left,right} camera, and $\boldsymbol{n}_{\{\mathrm{right,left}\}}$ and $d_{\{\mathrm{right,left}\}}$ are laser plane parameters of the {right,left} plane. Remember that projections obtained using the left camera relate to the right laser plane and vice versa. In the end, these lead to the following systems of equations:

$$\left.\begin{aligned}
{}_{\mathrm{M_{left}}}x_{\mathrm{u}} &\overset{(2.38)}{=} \frac{{}_{\mathrm{l}}\boldsymbol{q}_1^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{left}} + {}_{\mathrm{l}}q_{14}}{{}_{\mathrm{l}}\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{left}} + {}_{\mathrm{l}}q_{34}} &&\equiv \left({}_{\mathrm{l}}\boldsymbol{q}_1^{\mathsf{T}} - {}_{\mathrm{l}}\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{M_{left}}}x_{\mathrm{u}}\right){}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{left}} = {}_{\mathrm{l}}q_{34}\,{}_{\mathrm{M_{left}}}x_{\mathrm{u}} - {}_{\mathrm{l}}q_{14}\\
{}_{\mathrm{M_{left}}}y_{\mathrm{u}} &\overset{(2.38)}{=} \frac{{}_{\mathrm{l}}\boldsymbol{q}_2^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{left}} + {}_{\mathrm{l}}q_{24}}{{}_{\mathrm{l}}\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{left}} + {}_{\mathrm{l}}q_{34}} &&\equiv \left({}_{\mathrm{l}}\boldsymbol{q}_2^{\mathsf{T}} - {}_{\mathrm{l}}\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{M_{left}}}y_{\mathrm{u}}\right){}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{left}} = {}_{\mathrm{l}}q_{34}\,{}_{\mathrm{M_{left}}}y_{\mathrm{u}} - {}_{\mathrm{l}}q_{24}\\
{}_{\mathrm{T}}\boldsymbol{n}^{\mathsf{T}}_{\mathrm{right}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{left}} + {}_{\mathrm{T}}d_{\mathrm{right}} &\overset{(2.42)}{=} 0 &&\equiv {}_{\mathrm{T}}\boldsymbol{n}^{\mathsf{T}}_{\mathrm{right}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{left}} = -{}_{\mathrm{T}}d_{\mathrm{right}}
\end{aligned}\right\} \tag{2.44}$$

$$\left.\begin{aligned}
{}_{\mathrm{M_{right}}}x_{\mathrm{u}} &\overset{(2.39)}{=} \frac{{}_{\mathrm{r}}\boldsymbol{q}_1^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{right}} + {}_{\mathrm{r}}q_{14}}{{}_{\mathrm{r}}\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{right}} + {}_{\mathrm{r}}q_{34}} &&\equiv \left({}_{\mathrm{r}}\boldsymbol{q}_1^{\mathsf{T}} - {}_{\mathrm{r}}\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{M_{right}}}x_{\mathrm{u}}\right){}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{right}} = {}_{\mathrm{r}}q_{34}\,{}_{\mathrm{M_{right}}}x_{\mathrm{u}} - {}_{\mathrm{r}}q_{14}\\
{}_{\mathrm{M_{right}}}y_{\mathrm{u}} &\overset{(2.39)}{=} \frac{{}_{\mathrm{r}}\boldsymbol{q}_2^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{right}} + {}_{\mathrm{r}}q_{24}}{{}_{\mathrm{r}}\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{right}} + {}_{\mathrm{r}}q_{34}} &&\equiv \left({}_{\mathrm{r}}\boldsymbol{q}_2^{\mathsf{T}} - {}_{\mathrm{r}}\boldsymbol{q}_3^{\mathsf{T}}\,{}_{\mathrm{M_{right}}}y_{\mathrm{u}}\right){}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{right}} = {}_{\mathrm{r}}q_{34}\,{}_{\mathrm{M_{right}}}y_{\mathrm{u}} - {}_{\mathrm{r}}q_{24}\\
{}_{\mathrm{T}}\boldsymbol{n}^{\mathsf{T}}_{\mathrm{left}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{right}} + {}_{\mathrm{T}}d_{\mathrm{left}} &\overset{(2.43)}{=} 0 &&\equiv {}_{\mathrm{T}}\boldsymbol{n}^{\mathsf{T}}_{\mathrm{left}}\,{}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{right}} = -{}_{\mathrm{T}}d_{\mathrm{left}}
\end{aligned}\right\} \tag{2.45}$$

which yield both solutions

$${}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{left}} = \boldsymbol{F}^{-1}_{\mathrm{left}}\boldsymbol{b}_{\mathrm{left}} \quad \text{and} \quad {}_{\mathrm{T}}\hat{\boldsymbol{p}}_{\mathrm{right}} = \boldsymbol{F}^{-1}_{\mathrm{right}}\boldsymbol{b}_{\mathrm{right}} \tag{2.46}$$

on the same reference frame S_{T}.

It is precisely in the TCP reference frame S_{T} where absolute pose information is available. Subject to the reference system used, the results in Eqs. (2.44) and (2.45) can be transformed to the world frame S_0 directly, *i.e.*, ${}_0\bar{\hat{\boldsymbol{p}}} = {}_0\boldsymbol{T}^{\mathrm{T}}\,{}_{\mathrm{T}}\bar{\hat{\boldsymbol{p}}}$ or through the base reference frame S_{B} of a robot manipulator, *i.e.*, ${}_0\bar{\hat{\boldsymbol{p}}} = {}_0\boldsymbol{T}^{\mathrm{B}}\,{}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}\,{}_{\mathrm{T}}\bar{\hat{\boldsymbol{p}}}$.

Using visual pose tracking as in Chapter 5 represents a special case: Both, the perspective projection matrices and the plane equations, have to be simplified as the common TCP reference frame S_{T} is rendered superfluous. In the end, it comes down to identifying S_{T} with e.g. $S_{\mathrm{C_{left}}}$, *i.e.*, ${}_{\mathrm{C_{left}}}\boldsymbol{T}^{\mathrm{T}} = \boldsymbol{I}_{(4\times4)}$, in Eqs. (2.38), (2.39), (2.42) and (2.43), with the result that the systems of equations (2.44) and (2.45) now yield data in the camera reference frame $S_{\mathrm{C_{left}}}$.

2.2.3 The DLR Laser Range Scanner

Description

The DLR Laser Range Scanner (LRS) is a sensor similar to the LSP as it also operates by laser light triangulation (Hacker *et al.*, 1997; Kielhöfer, 2003), see Fig. 2.16.

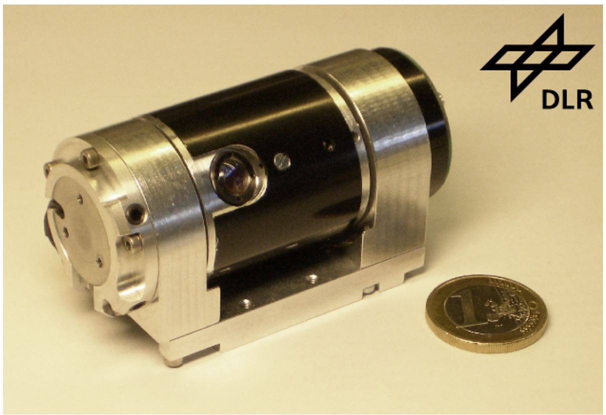


Figure 2.16: The DLR Laser Range Scanner.

Instead of generating a laser plane, however, the LRS emits a weak, pulsed laser ray originated from a laser diode focused by a highly refracting microlens. The single ray is continuously rotated to generate a planar area of singular laser projections, *i.e.*, the axis of rotation of the laser ray is perpendicular to it, see Fig. 2.17. During its rotation, a number of measurements of pulsed laser projections are being taken, obtaining in the end a 1-D countour of ranges to the surface of the scene as in the case of the LSP.

Still, some differences between the LRS and the LSP are to be mentioned:

1. In the case of the LRS, any range contour can be subdivided into single measurements (pulsed laser spots) that are being taken sequentially. This was not the case for the LSP where the whole contour projection was imaged by the camera instantly. Consequently, in the case of the LRS it is possible to directly infer on the *laser ray direction* for every single measured reflection, whereas in the case of the LSP it is only possible to infer on the view direction from the camera, as the laser beam origin remains unknown. This difference is nonrelevant in most applications, except for autonomous environment exploration: Note that positive measurements on a known ray direction are not only useful to infer on the scene's surface, but also to infer on void area between the laser beam origin and the detected surface. In the case of the LRS, that direction can be precisely controlled if the sensor has been previously calibrated, see Section 3.7. In the case of the LSP, however, range directions in S_C depend on the (unknown) scene, hence cannot be specified in advance and directly used for purposive, autonomous exploration.

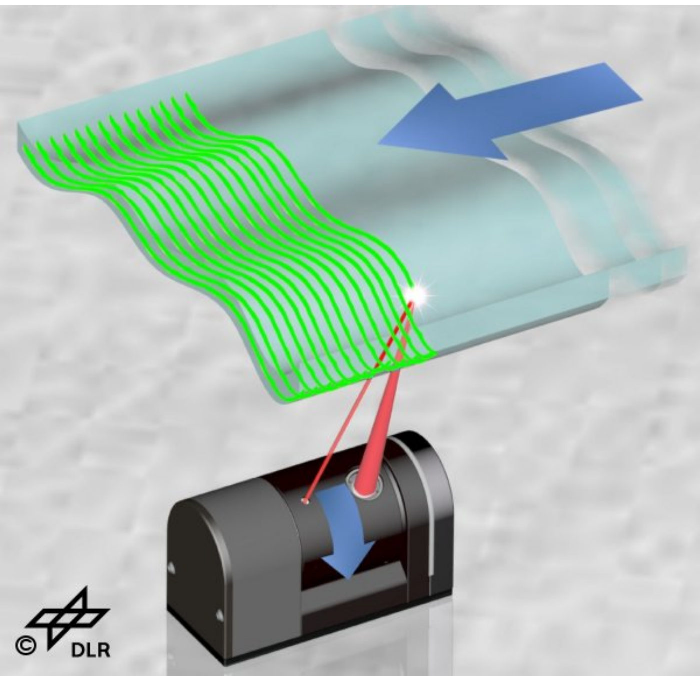


Figure 2.17: Principle of operation of the DLR Laser Range Scanner.

2. As an undesirable side effect of laser beam rotation at contant speed, all sequential, single measurements are being taken at different times. Precise synchronization and time interpolation is required in order to be able to accurately fuse data with e.g. movable external positioning systems.

3. A further undesirable side effect of its rotatory motion is, of course, the presence of (rapidly) moving parts in the sensor head.

4. In order to record laser reflections from a rotating laser ray, it is convenient for the laser projection detector to move along with the rotating laser beam. Since it is difficult to rotate a camera at this speed and, at the same time, keep the sensor size small, it was decided to use an electronic position sensitive device (PSD) instead of a camera (Hacker *et al.*, 1997). This device can be filtered to only detect laser light at 670 nm frequency and allows for the production of a self-contained device. In addition, the PSD is controlled jointly with the laser beam power to overcome varying reflection characteristics of the surface: The emitted laser power is dynamically adapted to the intensity of the reflected signal following a logarithmic scale, for every single measurement spot. In this way, the LRS achieves high robustness against varying surface materials. However, the big drawback of the inclusion of a PSD instead of using an external camera is the extended calibration requirements: First, an internal method for PSD calibration has to be devised similar to regular camera calibration in Section 3.2. Second, it is now necessary to obtain the precise coordinates of the local reference frame S_{LRS} to be able to

register LRS data. Remember that, in the case of the LSP, the camera center can be used to locate the sensor instead, so that only the laser plane location w.r.t. the camera is required, *i.e.*, 3 DoF; now, the external calibration of the LRS also requires its position, *i.e.*, 6 DoF are to be estimated altogether, see Section 3.7.

5. In order to keep the size of the sensor small and lower its power requirements, the sensor's basis distance between the virtual laser plane and the PSD is kept small and the laser light power is also low. Hence, the sensor is limited to close-range applications up to 20 cm. The LRS is size 75 × 32 × 44 mm.

6. Its wide scan angle (cf. Fig. 2.17) is especially convenient in robotic applications like autonomous exploration (Suppa, 2008; Kriegel *et al.*, 2012).

Geometry

In this work I leave the intrinsic calibration and operation of the LRS aside and refer the reader to the original work in (Hacker *et al.*, 1997; Kielhöfer, 2003). The valid output of the intrinsic operation of the LRS is a series of measured (˜) ranges $\tilde{d}$ along with their corresponding acquisition angle $\tilde{\phi}$ and time $\tilde{t}$. I avoid the use of subscripts for the sake of clarity. We can easily reconstruct the measured Euclidean coordinates using the cylindrical range sensor description in (Bodenmüller, 2009) as follows:

$$
{}_{\mathrm{LRS}}\boldsymbol{p} = \begin{bmatrix} {}_{\mathrm{LRS}}x \\ {}_{\mathrm{LRS}}y \\ {}_{\mathrm{LRS}}z \end{bmatrix} = {}_{\mathrm{LRS}}\boldsymbol{R}_{\mathrm{y}}^{\mathrm{Rotor}}(\tilde{\phi}) \begin{bmatrix} 0 \\ 0 \\ \tilde{d} \end{bmatrix} = \begin{bmatrix} \cos\tilde{\phi} & 0 & \sin\tilde{\phi} \\ 0 & 1 & 0 \\ -\sin\tilde{\phi} & 0 & \cos\tilde{\phi} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \tilde{d} \end{bmatrix} . \tag{2.47}
$$

Range data $\tilde{d}$ is in the ${}_{\mathrm{Rotor}}\boldsymbol{z}$ axis of the rotor; the rotor rotates around its own ${}_{\mathrm{Rotor}}\boldsymbol{y}$ axis that is coincident with the axis ${}_{\mathrm{LRS}}\boldsymbol{y}$ of the stator reference frame S_{LRS} as in the rotation matrix ${}_{\mathrm{LRS}}\boldsymbol{R}_{\mathrm{y}}^{\mathrm{Rotor}}$. The reference frames are depicted in the following Fig. 2.18.

Because of its precision, the robust data acquisition capabilities and its small size, the LRS is used within the DLR 3D-Modeler as a high definition, short-range sensor.

Extrinsic Geometry

As in the case of the LSP, the LRS is fundamentally limited to 2-D geometric information about the scene, *i.e.*, range data spread in 1-D only. In the ideal case, the LRS delivers a contour of depths according to the laser projections onto the scene. In order to gather 3-D information of the scene, it is still necessary to sweep the sensor across the scene, registering the gathered laser projections in 3-D by means of external pose tracking systems like the ARTtrack2, the FaroArm Gold or the Kuka KR 16. Therefore, the global representation of LRS data in the world reference frame S_0 usually requires intermediate frames

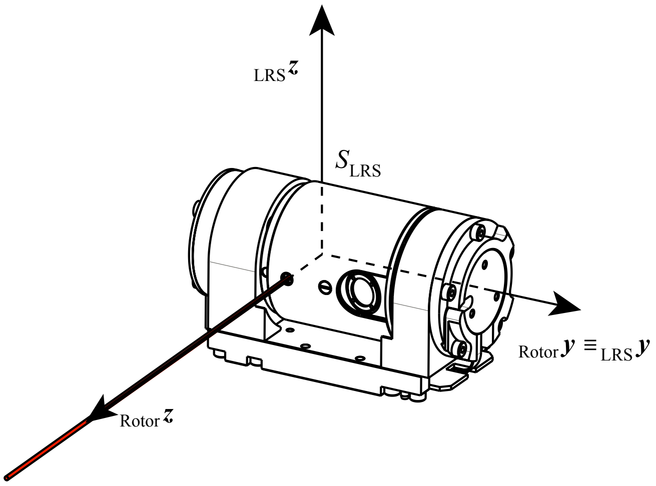


Figure 2.18: The reference frame of the LRS S_{LRS} and the reference frame of the rotor S_{Rotor} share the position of their origin as well as their axis $\boldsymbol{y}$.

like the TCP frame of the robotic manipulator S_{T} and the base reference frame of the same manipulator S_{B} as follows:

$$ {}_0\bar{\boldsymbol{p}} = \begin{bmatrix} {}_0x \\ {}_0y \\ {}_0z \\ 1 \end{bmatrix} = {}_0\boldsymbol{T}^{\text{B}}\ {}_{\text{B}}\widetilde{\boldsymbol{T}}^{\text{T}}\ {}_{\text{T}}\boldsymbol{T}^{\text{LRS}}_{(3\times4)}\ {}_{\text{LRS}}\boldsymbol{R}^{\text{Rotor}}_{\text{y}}(\tilde{\phi}) \begin{bmatrix} 0 \\ 0 \\ \tilde{d}_{\text{LRS}} \end{bmatrix} \quad , \tag{2.48} $$

where ${}_0\boldsymbol{T}^{\text{B}}$ and ${}_{\text{B}}\boldsymbol{T}^{\text{T}}$ are the homogeneous transformation matrices between S_0 and S_{B}, and S_{B} and S_{T}, respectively. These matrices are usually being estimated in the context of the extrinsic calibration of attached cameras, tracked by the external reference system in position and orientation, see Section 3.3. ${}_{\text{T}}\boldsymbol{T}^{\text{LRS}}_{(3\times4)}$ is the transformation matrix between S_{T} and S_{LRS}; it is size 3×4 and will be estimated using the novel extrinsic calibration process detailed in Section 3.7.

2.2.4 The Inertial Measurement Unit

Description

Inertial measurement units (IMUs) are enclosed electronic devices that yield 6-D motion estimation in a passive, self-contained way. To that end, IMUs combine gyroscopes that measure rotational speed and accelerometers that measure linear accelerations, and sometimes magnetometers as well. The usual configuration includes three gyroscopes and three accelerometers in orthonormal arrangement in order to independently measure their three respective DoF in orientation and translation, respectively. Since in this context motion estimation regularly builds upon last motion estimates, the IMU can be considered a dead reckoning sensor. However, in most applications the gravity vector is being used as an absolute measurement in order to partially constrain motion estimation.

Perhaps the most inconvenient drawback of using an IMU is the fact that gyroscopes and accelerometers deliver higher order pose tracking information, *i.e.*, their derivatives angular speed and linear acceleration. These values have to be integrated in time so that pose estimation inevitably drifts. What is more, variable biases (drifts) and static offsets in gyroscope readings have to be accounted for. In the case of the accelerometers, zero-mean noise is dominant, which also accounts for strong drifts in translation estimation after repeated integration. In addition, orientation errors directly impact translational estimations. Further errors like misalignment and nonorthogonality of the sensor components, asymmetric scale factors and nonlinearity errors can also occur. External difficulties are its synchronization and geometric calibration w.r.t. external devices.

In recent years inertial sensors are being deployed as a complementary sensor to cameras. Indeed, IMUs perfectly complement off-the-shelf cameras both in measuring rate and in temporal precision: On the one hand, regular cameras take, say 25 images per second and estimations from their images are, in principle, equally accurate all the time, *i.e.*, they are *absolute measurements.* On the other hand, IMUs yield data at kHz rates and their pose tracking readings are far more accurate when they lie close together in time rather than when they lie far apart since, as mentioned above, they provide *relative measurements.* These inconsistencies can be used to compensate each other sensor, aiming at more robust estimations.

This can be done either by fusing final pose outcomes from both sensors (either stochastically or just in time), or directly for one sensor to support pose estimation within the other sensor's estimation process. The latter approach is employed in this work: I opt for the unidirectional support of visual processing by the IMU, refer to Section 5.4.2. IMU data that are close in time are used to support image processing (in the images where they coincide in time). I are not using visual measurements to support IMU-based pose tracking.

The DLR 3D-Modeler can be extended by a rigidly attached IMU, the AscTec AutoPilot by Ascending Technologies GmbH. Attitude estimation is on six DoF at 1 kHz leveraging three gyros, three accelerometers, and three magnetometers. The IMU performs on-board data fusion and features a second, idle 60 MHz ARM processor available for the user. It only weighs 19.6 g and is size $10 \times 50 \times 50$ mm.

Geometry

In this section I do not cope with internal geometry of the IMU nor with their drifts in rotation rate and their integration. The reason is that I shall not use inertial data in the long run, but intertwined between image frames, *i.e.*, within 40 ms, where drifts can be neglected. The geometrical drifts in the reference frame of the IMU S_{IMU} are already being compensated for internally within the AscTec AutoPilot, on a regular basis.

Indeed, in Section 5.4.2 I shall only make use of differential rotations in pitch angle ($\widetilde{\Delta P}$), in yaw angle ($\widetilde{\Delta Y}$) and in roll angle ($\widetilde{\Delta R}$), integrated over 40 ms. These values can be easily put together to produce the differential rotation matrix $\boldsymbol{R}^{i–1,i}$ between imaging instants i–1 and i in advance of image processing as follows:

$$\boldsymbol{R}^{i–1,i}_{\mathrm{IMU}} = \boldsymbol{R}^{i–1,i}_{\mathrm{roll}}\,\boldsymbol{R}^{i–1,i}_{\mathrm{yaw}}\,\boldsymbol{R}^{i–1,i}_{\mathrm{pitch}} = \begin{bmatrix} \mathrm{cr\,cy} & -\mathrm{sr\,cp} + \mathrm{cr\,sy\,sp} & \mathrm{sr\,sp} + \mathrm{cr\,sy\,cp} \\ \mathrm{sr\,cy} & \mathrm{cr\,cp} + \mathrm{sr\,sy\,sp} & -\mathrm{cr\,sp} + \mathrm{sr\,sy\,cp} \\ -\mathrm{sy} & \mathrm{cy\,sp} & \mathrm{cy\,cp} \end{bmatrix} \tag{2.49}$$

where

$$\begin{aligned} \mathrm{sp} &= \sin\widetilde{\Delta P} & \mathrm{cp} &= \cos\widetilde{\Delta P} \\ \mathrm{sy} &= \sin\widetilde{\Delta Y} & \mathrm{cy} &= \cos\widetilde{\Delta Y} \\ \mathrm{sr} &= \sin\widetilde{\Delta R} & \mathrm{cr} &= \cos\widetilde{\Delta R} \quad . \end{aligned}$$

Extrinsic Geometry

Accurate motion information in S_{IMU} is, however, not useful for direct insertion in e.g. visual pose tracking algorithms as in Chapter 5. In order to be able to represent the readings of the IMU in the camera reference frame S_{C}, the rigid body transformation between both frames S_{IMU} and S_{C} has to be estimated (see Section 3.8). Since I only make use of the rotational rates of the IMU, this relationship boils down to:

$${}_{\mathrm{C}}\boldsymbol{R}^{i–1,i}_{\mathrm{IMU}} = {}_{\mathrm{C}}\widehat{\boldsymbol{R}}^{\mathrm{T}}\,{}_{\mathrm{T}}\widehat{\boldsymbol{R}}^{\mathrm{IMU}}\,\boldsymbol{R}^{i–1,i}_{\mathrm{IMU}} = {}_{\mathrm{C}}\widehat{\boldsymbol{R}}^{\mathrm{T}}\,{}_{\mathrm{T}}\widehat{\boldsymbol{R}}^{\mathrm{IMU}}\,\boldsymbol{R}^{i–1,i}_{\mathrm{roll}}\,\boldsymbol{R}^{i–1,i}_{\mathrm{yaw}}\,\boldsymbol{R}^{i–1,i}_{\mathrm{pitch}} \quad , \tag{2.50}$$

where ${}_{\mathrm{C}}\widehat{\boldsymbol{R}}^{\mathrm{T}}$ stems from the extrinsic calibration of the stereo camera in Section 3.3 and ${}_{\mathrm{T}}\widehat{\boldsymbol{R}}^{\mathrm{IMU}}$ is the required transformation between S_{IMU} and S_{T}, refer to Section 3.8.

2.3 Absolute Pose Tracking Systems

2.3.1 Description

In general, it is not possible for a 3-D modeling device to acquire a complete model at one single measurement step due to its limited field of operation, object self-occlusion, or object size—this particularly holds for close-range 3-D modeling systems. The 3-D geometrical information gathered from a single vantage point is limited, so that multiple views (or multiple sensors) are required in order to subsequently merge data to a single 3-D model (Chen *et al.*, 2000).

Merging range data (also referred to as 2.5-D data) can be performed following either of these oppositional approaches: On the one hand, 2.5-D data can be acquired freely from arbitrary unknown positions, and after that merged to a single 3-D model by software registration. On the other hand, parallel position and orientation (pose) tracking of the 3-D sensor makes it possible to directly represent results in the world frame S_0, in realtime. The DLR 3D-Modeler realizes the second approach. For this purpose, tracking systems like the infrared optical tracking system ARTtrack2, redundant robotic manipulators like the Kuka KR 16 or the DLR Lightweight Robot III, or passive arms like the FaroArm Gold, turntables, CMMs or electromagnetic devices are commonly used. The original DLR 3D-Modeler in (Suppa *et al.*, 2007) realized the second approach. I tracked the pose of the DLR 3D-Modeler by fixing it to the end-effector of a FaroArm Gold featuring 7 passive joints or, alternatively, utilized the ARTtrack2 infrared optical tracking system or the Kuka KR 16 robotic manipulator. In reality, pose tracking refers here to the TCP of the tracking system—every other sensor system has to be calibrated w.r.t. it, refer to Section 3.3.

The abovementioned options are, however, extremely limiting for the following reasons:

1. They limit the user's mobility, thus the usability of the sensor. Even infrared tracking systems restrict motion, especially in two rotational DoF. In the case of robotic manipulators, some of them feature a 7th DoF for improved autonomy; however, their usability as part of a 3-D modeling system remains very low.

2. The final performance of the whole system will strongly depend on accurate synchronization and extrinsic hand-eye calibration, which are cumbersome, error-prone processes (Strobl and Hirzinger, 2006; Bodenmüller *et al.*, 2007). What is more, the hand-eye attachment cannot be rearranged without an additional extrinsic recalibration of the sensor.

3. It turns out that every external positioning system mentioned above represents the largest and most expensive part of any 3-D modeling system.

In Chapter 5 I present novel algorithms for image-based motion estimation from the images of the stereo camera presented in Section 2.2.1. This approach overcomes all of the abovementioned limitations.

2.3.2 Geometry

In this section I present a general model for any of the tracking system mentioned above, without going into detail on their own principles of operation.

With the exception of visual pose tracking in Chapter 5, all tracking systems feature their own reference frames between which pose readings are delivered; these frames are completely independent of the 3-D modeling device. In order for the latter to take advantage of pose tracking information, it has to be rigidly attached to one of that reference frames. Since robotics is a dominant application area for the DLR 3D-Modeler, we refer to the reference frame where our device is rigidly attached to (where pose tracking information is available) as the **t**ool **c**enter **p**oint (TCP) reference frame S_{T}. The particularity of the visual tracking system in Chapter 5 is that $S_{\mathrm{T}} \triangleq S_{\mathrm{C}}$, *i.e.*, the camera reference frame S_{C} directly acts as a reference frame for pose tracking. Furthermore, every tracking system delivers motion estimation w.r.t. its own fixed reference frame, called base reference frame S_B (in the case of visual pose tracking, $S_\mathrm{B} \triangleq S_\mathrm{C}$ *at the initial tracking instant*). In general it holds:

$$
{}_\mathrm{C}\boldsymbol{T}^0 = {}_\mathrm{C}\boldsymbol{T}^\mathrm{T}\,{}_\mathrm{T}\widetilde{\boldsymbol{T}}^\mathrm{B}\,{}_\mathrm{B}\boldsymbol{T}^0 \quad , \tag{2.51}
$$

where S_0 represents the world reference frame, and the homogeneous transformation matrix ${}_\mathrm{T}\widetilde{\boldsymbol{T}}^\mathrm{B}$ directly stems from the output of the pose tracking system at a particular instant. The transformations ${}_\mathrm{C}\boldsymbol{T}^\mathrm{T}$ and ${}_\mathrm{B}\boldsymbol{T}^0$ are called extrinsic transformations; they are required to transform motion readings into the camera frame S_C. Note that these transformations can be estimated out of visual data as explained in Section 3.3.

2.4 Summary

In this chapter I described the underlying operational principles of all sensors potentially involved in the task of 3-D modeling using the DLR 3D-Modeler. I also laid out their mathematical models for accurate parametrization in the next chapter 3 as well as utilization in chapters 4 and 5 and Appendix B.

I started out motivating the search for accurate, compact sensor models of general validity. In addition, these models have to observe the requirement for eventual, accurate parametrization in Chapter 3. After that, I listed the relevant components of the DLR 3D-Modeler. Since video cameras are the central sensor of this thesis, I elaborate in great detail on the perspective model of cameras, their historic development, and finally state the projective formulation that is going to be extensively used in the next chapters. It follows the modeling of the operational principle of the DLR Laser Stripe Profiler (LSP) as it is closely related to the operation of cameras; the LSP is perhaps the dominant range sensor of the DLR 3D-Modeler. In particular, I introduce the novel development of the dual, crosshair LSP. Following the LSP, I introduce the DLR Laser Range Scanner (LRS) as it shares the operational principle of the LSP, but it is independent of the stereo camera. An additional, modular inertial measurement unit (IMU) can be also used to support image processing in Chapter 5. Alternatively, the user must revert to absolute pose tracking systems, which conclude this chapter.

"Be precise. A lack of precision is dangerous when the margin of error is small."
—Donald Rumsfeld, WSJ, 2001

3 Accurate Parametrization of System Models

3.1 Introduction

In line with last chapter's motivation, it is clear that geometric computer vision is a quantitative task that heavily relies on accurate measurements. Accurate measurements, in turn, rely on accurate parametrization of the operating models listed in Chapter 2, *i.e.*, the accurate calibration of the sensor components.

Sensor calibration is a complex task as many aspects have to be considered at the same time. Consequently, it is a dangerous process where, more often than not, researchers commit smaller errors that mislead them to wrong parametrizations. It goes without saying that erroneously calibrated sensors severely compromise both, the development of novel algorithms based on their data as well as their eventual performance. In a nutshell, a **correct method** has to be chosen and **valid data** has to be collected for it. Additionally, a helpful calibration software should enable the user to choose the correct sensor **model** (avoiding unnecessary DoF, see Chapter 2) during calibration.

With regard to choosing a valid calibration **method**, two aspects have to be considered: *First*, whether the method is sound (e.g. how to fit the sensor model operation to actual calibration data); at best, the chosen method should minimize the actual errors in the system model for the sake of statistical optimality. Many calibration methods end up computing a solution to the parametrization problem in the most straightforward—perhaps efficient—way, e.g. linear least squares solutions instead of nonlinear optimizations; the adoption of this type of methods invariably involves a loss of accuracy. *Second*, whether the method is convenient or entails severe costs like expensive calibration objects, inappropriate models, or the requirement for very precise external measurements. The best bet is to choose a calibration method that both, delivers highest accuracy and does so in a flexible way, *i.e.*, imposing as less restrictions as possible in order to avoid human mistakes otherwise bound to occur. These conception rules will lead the novel developments in the present chapter.

With regard to acquiring **valid data**, the limitation is rarely physical (e.g. cameras with smaller AOV or limited focal depth, inaccurate external pose readings, etc.), but rather consequence of the lack of understanding by the user of the underlying principles of the optimization problem (for instance, the conditioning of the optimization problem). Admittedly, on occasions these errors could be avoided by more readable documentation of the proposed algorithms. These limitations do not only concern inadequate sensor measurements but also inaccurate metadata required by the optimization method. To cite an example of inadequate measurements: In the context of camera calibration, note the predominant use of central projections of the calibration object that do not fill the whole image frame, or rather note the widespread use of images orthogonal to a planar calibration object. A prevalent example of inaccurate metadata concerns standard camera calibration without accurate previous measurement of the actual dimensions—or even the planarity—of the calibration object. I shall address these limitations in Section 3.2.

A last topic addressed in this chapter is the design of a sensible concept for combined calibration or recalibration of all component sensors of the system w.r.t. external pose tracking systems, *i.e.*, of a “system calibration” concept in Section 3.9. In the case of the DLR 3D-Modeler, most sensors depend on each other and a frugal, synergistic approach to calibrate the whole system is required.

I would like to further stress the significance of an accurate parametrization of system models, *i.e.*, of sensor calibration. Note that this may be considered a further instance of the present technological paradigm of the “softwarization” of hardware, as meticulous calibration (*i.e.*, intensive optimization) of simple models really make for a better sensor.

3.2 Intrinsic (Stereo) Camera Calibration

3.2.1 Introduction

Camera calibration is the process of estimating the parameters of a camera model that is capable of adequately reflecting the underlying operational principle of the actual camera. This is usually accomplished by *comparing its expected, model-based operation with the actually collected data*, followed by a sensible minimization of the resulting discrepancies. The parameterized model will enable the user to infer in 3-D Euclidean space from the evidence of the 2-D information in the image projections.

Accurately calibrated cameras are prerequisite to most vision-based algorithms. However, researchers still find it challenging to achieve the required accuracy in particular areas like stereo vision (Section 4.2) or SLAM (see Chapter 5). Ever since the advent of high-resolution, stereo vision algorithms are demanding higher accuracy in calibration (*i.e.*, more accurate epipolar geometry, cf. Section 4.2.2) to keep computational costs in practical terms; SLAM puts similar requirements on calibration accuracy, mainly to reduce dead reckoning drift, improving overall performance.

In this work I proceed on the assumption that the camera cannot intrinsically change during operation, but only extrinsically in its pose w.r.t. the scene. It is therefore possible for the user to estimate its parameters in advance of regular operation. Moreover, since cameras record their environment in a passive way, it makes sense to set conditions on the scene structure during calibration in order to support both robustness and accuracy. These conditions may concern inserting a priori knowledge of the metric structure of the scene (e.g. a calibration object) in order to both, maximize the amount of data for calibration as well as to improve the estimations of expected projections. As mentioned above, it is by comparing these estimations with the actual projections that the optimal parameters will be eventually delivered.

The following guidelines apply when designing a scene with a priori knowledge used for calibration:

- The more *diverse* the scene is (e.g. a general 3-D scene), the more independent evidence for the calibration will be.
- The more *accurate* knowledge of the scene exists, the more accurate predictions of the camera operation will be.

Unfortunately, these points imply a trade-off since optimal conditions (e.g. general, precisely known scenes) suggest elaborated and expensive calibration setups, which are cumbersome for general computer vision applications. On the other hand, less advantageous but convenient conditions, such as mere point correspondences, are not sufficient for accurate camera calibration; that approach pertains to the family of self-calibration, which are usually less reliable than standard camera calibration, see (Triggs, 1998; Liebowitz and Zisserman, 1999; Remondino and Fraser, 2006; Civera *et al.*, 2009).

But for all that, research on camera calibration for computer vision has arrived at a point where most of its components have become standard practice: The perspective projection model (pinhole camera model), see (Faugeras and Luong, 2004) and Section 2.2.1; the radial and tangential lens distortion models, see (Weng *et al.*, 1992) and Section 2.2.1; the imaging noise assumption (Matthies and Shafer, 1987; Sun and Cooperstock, 2006); feature detection algorithms (Mallon and Whelan, 2007); the planar calibration object (Tsai, 1987); and even the estimation algorithms (Zhang, 2000; Sturm and Maybank, 1999) meet the demands of the computer vision community. In the next section I summarize the most significant contributions that led us to this point.

3.2.2 State of the Art

Until the mid eigthies there only was photogrammetric work. It mainly relied on full-scale nonlinear optimizations for elaborate projection models and calibration objects, see (Brown, 1971; Faig, 1975). Since photogrammetry is about *accurately* measuring objects by photographies, they went to considerable length into elaborate calibration procedures. This was not suitable for computer vision applications since first, their hardware requirements (including computational) were too high and, second, the complexity of their camera models exceeded the required for solid-state imaging devices, see Section 2.2.1. The work by Abdel-Aziz and Karara paved the way for computer vision applications; their direct linear transformation (DLT) basically finds solutions to linear equations using the basic camera model of collinearity (Abdel Aziz and Karara, 1971). However, since ignoring lens distortion is mostly unacceptable (refer to Section 2.2.1), Tsai in the mid eigthies introduced a calibration method with a more complete camera model instead (Tsai, 1987). He was able to simplify the formulation by using the radial alignment constraint, which reduces the dimensionality of the problem and allows its decomposition in two independent stages—at the risk of a loss of radial geometric information when the lens distortion is small. Severe scene restrictions still apply: The procedure requires either 3-D calibration objects or accurately shifting a planar calibration plate, *i.e.*, a 2.5-D scene. Similar methods were proposed in (Weng *et al.*, 1992) and (Faugeras and Toscani, 1987). The former method employed an extensive lens distortion model; therefore, it relies on an iterative coupling of local, nonlinear optimizations.

A major contribution towards simplicity in camera calibration was simultaneously made in the late nineties by Zhang (Zhang, 2000) and Sturm and Maybank (Sturm and Maybank, 1999). The suitability of their algorithm in computer vision applications made both, their algorithmic and the used models, current standard practice. They presented a closed-form solution by linear least squares techniques for the initialization of a nonlinear optimization. Most importantly, they relax conditions on the scene allowing for *freely* moving a *precisely known* planar calibration pattern for collecting data—compared to 3-D or 2.5-D objects before. In short, it recovers the intrinsic camera parameters from readily obtained object-to-camera homographies using both, the pinhole camera model and rigid body motion constraints. Indeed, the perspective projection of a planar, known pattern suffices to differentiate between the pose of the calibration pattern and the scaling characteristics of the camera: Whereas the latter merely scales the image, the pose of the pattern dictates the perspective distortion of the projected pattern. The approach represented a step towards self-calibration since no implicit 3-D information was required anymore, but only in 2-D. Malm and Heyden extended this formulation in the case of stereo camera systems by the addition of further rigid body motion constraints (Malm and Heyden, 2001). In turn, Strobl and Hirzinger extended the same formulation in the case of inaccurate (or unknown) planar calibration patterns, see (Strobl and Hirzinger, 2008, 2011).

3.2.3 The Standard Method by Zhang, Sturm, and Maybank

The standard method by Zhang, Sturm, and Maybank essentially differentiates from earlier methods like the method of Tsai regarding the set of parameters to be estimated: Whereas the Tsai method is merely about estimating the parameters of the camera itself, Zhang, Sturm, and Maybank extend the set of unknown parameters to the poses of the camera w.r.t. the known planar pattern. At first sight, a gratuitous extension of the set of unknown parameters to parameters that are not really required in the first place seems inadequate, as the estimation of the camera parameters will necessarily become less precise. At second sight, however, the method relieves the user of the accurate measurement of the 3-D scene structure, or rather of the pose of the 2-D structure w.r.t. the camera; more often than not, these inconvenient measurements are accompanied with human errors and mistakes, which decrease accuracy to a higher extent than the new approach.

The new method merely requires taking images of a known, planar calibration object from N different vantage points.[1] The discrepancies between expected and actually detected projections are minimized to refine some initial values for both, the intrinsic parameters and the camera motion. If the camera is constituent part of an eye-in-hand system, the images for extrinsic hand-eye transformation can be collected at the same time, along with N pose readings of the end-effector at the imaging moments, see Section 3.3 and Fig. 3.1.

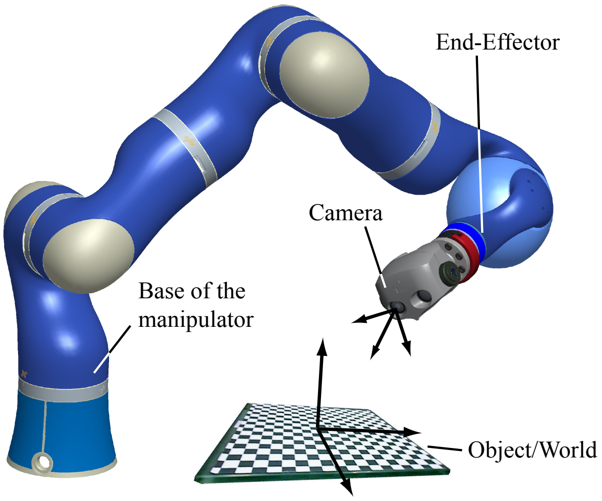


Figure 3.1: Stereo camera mounted at the top of the DLR Light-Weight Robot 3.

[1] Multiple vantage points are not always required for camera calibration, but they facilitate intrinsic initialization and allow for hand-eye calibration (Section 3.3). In addition, the central limit theorem states that, when the amount of independent and identically distributed (i.i.d.) data grows, error distributions tend to Gaussianity, which, in turn, facilitates optimal estimation by maximum likelihood estimation (especially in the case of the hand-eye calibration, see (Strobl and Hirzinger, 2006)). In fact, at least 8 vantage points are repeatedly suggested in the literature, cf. (Tsai, 1987; Triggs, 1998; Zhang, 2000; Sturm and Maybank, 1999; Strobl and Hirzinger, 2006).

As mentioned above, the dimensions of the planar calibration object have to be measured—this condition will be lifted in part in Section 3.4. In order to simplify this process, a repetitive pattern is commonly used, so that few measurement of the pattern are required if an accurate pattern model is provided. The most widespread planar pattern is a checherboard pattern.[2] The square corners are then perspectively projected by the camera onto 2-D images in S_{I} that stack up into the discrete memory array S_{M}, consistently with the discretized pinhole camera model in Section 2.2.1. Subsequently, the corners are detected and located in the images with sub-pixel accuracy (e.g. using DLR CalDe within the calibraton toolbox DLR CalDe and DLR CalLab (Strobl *et al.*, 2005)); feature location errors usually spread according to 2-D i.i.d. zero-mean Gaussian distributions; it is a tacit assumption that these errors actually encompass both, the pinhole camera model simplification error as well as the discretization error. Control points projections should spread all over the images in order to both, best condition the system equations, and to deliver a valid estimate of the distortion model parameters in the first place.

Initial Solution by Linear Least Squares

The main contribution of the standard calibration method in (Zhang, 2000; Sturm and Maybank, 1999) concerns the rapid calculation of reasonable initial values for both, the intrinsic camera parameters and the camera poses (also known as absolute extrinsics).

First, the visible control points or corners ${}_{0}\bar{\boldsymbol{p}}_i = [{}_{0}x_i\ {}_{0}y_i\ {}_{0}z_i\ 1]^{\mathsf{T}}$, $\forall i$ on the calibration plate are detected and localized in the image memory frame S_{M} at instant $n \in \{1, ..., N\}$. These distorted, measured $(\tilde{\ })$ projections ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}^{\{n,i\}}$ are to be compared with the expected, undistorted projections ${}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{u}}^{\{n,i\}}$, which are estimated $(\hat{\ })$ through Euclidean decomposition of the perspective projection matrix $\boldsymbol{P}$ as follows:

$$
{}_{\mathrm{M}}\bar{\boldsymbol{p}}_{\mathrm{u}} = \begin{bmatrix} {}_{\mathrm{M}}x_{\mathrm{u}} \\ {}_{\mathrm{M}}y_{\mathrm{u}} \\ 1 \end{bmatrix} \propto \underbrace{\boldsymbol{A}_{(3\times3)}\ {}_{\mathrm{C}}\boldsymbol{T}^{0}_{(3\times4)}}_{\boldsymbol{P}_{(3\times4)}}\ {}_{0}\bar{\boldsymbol{p}} = \underbrace{\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t} \end{bmatrix}}_{\boldsymbol{H}_{(3\times3)}} \begin{bmatrix} {}_{0}x \\ {}_{0}y \\ 1 \end{bmatrix} \quad . \tag{3.1}
$$

Here I drop subscripts n and i for the sake of clarity. Note that this equation differs from the general one in Eq. (3.5.3) in the absence of the third DoF in the position of the control points, *i.e.*, ${}_{0}z \triangleq 0$, so that $\boldsymbol{r}_3$ disappears as it becomes irrelevant. This is, of course, because the calibration plate is flat. It is also this fact that allows us to simplify to a linear projective transformation $\boldsymbol{H} = [\boldsymbol{h}_1\ \boldsymbol{h}_2\ \boldsymbol{h}_3]_{(3\times3)}$, which equals the homography between the calibration plane and the virtual, undistorted image. In projective geometry, homogeneous *plane* coordinates are transformed following a *linear* projective transformation called homography. Since the planar sensor of the camera and the planar target are approximately related by a projective transformation, N homographies $\widehat{\boldsymbol{H}}_n$

[2] Checkerboard patterns have been repeatedly appointed as most convenient in terms of potential accuracy in corners detection (Mallon and Whelan, 2007; Strobl *et al.*, 2005).

between image projections ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}}$ and pattern features ${}_{0}\boldsymbol{p}_i$ can be estimated ($\hat{\ }$) from at least four (three out of four non-collinear) correspondences i, for every image n, so that ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}} \propto \widehat{\boldsymbol{H}}_{n0}\boldsymbol{p}_i$, $\forall\, n \in \{1, \ldots, N\}$, $i \in \{1, \ldots, M\}$, see Appendix A. Of course, an exact homography is only possible w.r.t. *undistorted* projections ${}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{u}}^{\{n,\,i\}}$, but then undistortion parameters are not yet known until the end of the next optimization stage; therefore, distorted projections are used as a valid approximation.

We aim at the pinhole camera model represented by its intrinsic matrix $\boldsymbol{A}$, which together with one of the N rigid body transformations between the camera frame and the object frame ${}_{\mathrm{C}}\boldsymbol{T}_n^0 = [\boldsymbol{r}_1\, \boldsymbol{r}_2\, \boldsymbol{r}_3\, \boldsymbol{t}\,]$, project 3-D coordinates as in Eq. (3.1). For planar targets we have $\widehat{\boldsymbol{H}} \propto \boldsymbol{A}\ [\boldsymbol{r}_1\, \boldsymbol{r}_2\, \boldsymbol{t}]$. Since $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are orthonormal, we use the orthonormality restrictions[3] $\boldsymbol{r}_1 \cdot \boldsymbol{r}_2 = 0$, $\boldsymbol{r}_1 \cdot \boldsymbol{r}_1 = 1$, and $\boldsymbol{r}_2 \cdot \boldsymbol{r}_2 = 1$, *i.e.*, ${}_{\mathrm{C}}\boldsymbol{R}^0 \in SO(3)$, and sorting the scale out:

$$\left.\begin{array}{r}(A^{-1}\boldsymbol{h}_1)^{\mathsf{T}}\cdot(A^{-1}\boldsymbol{h}_2) = 0\\ (A^{-1}\boldsymbol{h}_1)^{\mathsf{T}}\cdot(A^{-1}\boldsymbol{h}_1)\\ -(A^{-1}\boldsymbol{h}_2)^{\mathsf{T}}\cdot(A^{-1}\boldsymbol{h}_2) = 0\end{array}\right\} \Leftrightarrow \left.\begin{array}{l}\boldsymbol{h}_1^{\mathsf{T}}\,\boldsymbol{\omega}_\infty\,\boldsymbol{h}_2 = 0\\ \boldsymbol{h}_1^{\mathsf{T}}\,\boldsymbol{\omega}_\infty\,\boldsymbol{h}_1 = \boldsymbol{h}_2^{\mathsf{T}}\,\boldsymbol{\omega}_\infty\,\boldsymbol{h}_2\end{array}\right\} \tag{3.2}$$

with the so-called absolute conic $\boldsymbol{\omega}_\infty = \boldsymbol{A}^{-\mathsf{T}}\boldsymbol{A}^{-1}$. These two equations hold for every N images, leading to $2N$ constraints for e.g. 5 intrinsic unknowns. In this particular case of 5 intrinsic unknowns, the system of equations can be solved for using a linear least squares criterion if at least **three different views** are available, *i.e.*, $N \geq 3$. Still, it is readily possible to do with less images if the number of unknowns is sensibly reduced, e.g. $\alpha \triangleq \beta$, $\gamma = 0$, and $\{u_0, v_0\}$ located at the middle of the image memory frame imply that a sole image is required for intrinsic calibration. It is worth noting that the solution only depends on the orientation of the plane and *not* on its distance or scale, *i.e.*, the formulation works both on Euclidean and similarity geometries, see (Faugeras and Luong, 2004) and Section 3.4.

If a stereo configuration exists (*i.e.*, N_{C} additional cameras rigidly attached to the main one), it is convenient to unify their absolute extrinsics to the absolute extrinsics of the main camera, see (Malm and Heyden, 2001). The rigid body constraint[4] ${}_{\mathrm{C}_c}\boldsymbol{T}_n^0 = {}_{\mathrm{C}_c}\boldsymbol{T}^{\mathrm{C}}\,{}_{\mathrm{C}}\boldsymbol{T}_n^0$ holds for every additional camera $\mathcal{C}_c$, $c \in \{1,\ 2,\ ...,\ N_{\mathrm{C}}\}$ so that $\boldsymbol{h}_1 = {}^{s_c}\!/_{s}\ {}_c\boldsymbol{H}_\infty\ {}_c\boldsymbol{h}_1$ and $\boldsymbol{h}_2 = {}^{s_c}\!/_{s}\ {}_c\boldsymbol{H}_\infty\ {}_c\boldsymbol{h}_2$ hold for every N images and N_{C} additional cameras; from these equations, the infinite homographies ${}_c\boldsymbol{H}_\infty = \boldsymbol{A}\,{}_{\mathrm{C}}\boldsymbol{R}^{\mathrm{C}_c}\,{}_c\boldsymbol{A}^{-1}$ can be estimated. These result in the following constraints that hold for their intrinsic matrices: $\boldsymbol{\omega}_\infty = {}_c\boldsymbol{H}_\infty^{\mathsf{T}}\ {}_c\boldsymbol{\omega}_\infty\ {}_c\boldsymbol{H}_\infty$. These six linear constraints may additionally stack up in the system of $2N$ linear equations (3.2) for N_{C} cameras.

[3] The geometrically inclined reader may prefer the interpretation concerning constraints on the critical points of each calibration plane by the intersection of the lines at infinity of the respective planes with the absolute conic (at the plane in infinity). This projective interpretation belongs together with a complexification of the homogeneous Euclidean vector space, refer to (Faugeras and Luong, 2004; Zhang, 2000; Sturm and Maybank, 1999).

[4] The camera-to-camera transformation ${}_{\mathrm{C}_c}\boldsymbol{T}^{\mathrm{C}}$ can be considered as an intrinsic parameter of a more abstract camera system concerning stereo cameras.

After determining the intrinsic matrix, the absolute extrinsics for every image n are readily computed from homographies as follows: $\boldsymbol{r}_1 = {}^1\!/_s\,\boldsymbol{A}^{-1}\boldsymbol{h}_1$, $\boldsymbol{r}_2 = {}^1\!/_s\,\boldsymbol{A}^{-1}\boldsymbol{h}_2$, $\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2$, $\boldsymbol{t} = {}^1\!/_s\,\boldsymbol{A}^{-1}\boldsymbol{h}_3$, and $s = ||\boldsymbol{A}^{-1}\boldsymbol{h}_1|| = ||\boldsymbol{A}^{-1}\boldsymbol{h}_2||$.

Unfortunately, lens distortion spoils the neat linear projective formulation of the perspective camera. Although it is possible to separately estimate its distortion parameters with a linear least squares criterion, this only works in alternation with the former intrinsic estimation (Zhang, 2000; Sánchez *et al.*, 2006). For accurate estimation, however, it is still necessary to perform a subsequent nonlinear optimization process, which can be as well initialized with the above described estimations. If the above initialization values lie within a broad convergence region in the parameter space, this nonlinear optimization will eventually be exclusive responsible for the final precision of the calibration process.

Final Solution by Nonlinear Optimization

On the basis of the maximum likelihood criterion, optimal calibration parameters will be obtained by sensibly minimizing the discrepancies of erroneous measurements w.r.t. the parameterized camera model. In our case, it is the detected projections ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}}$ of the control points in the images that are erroneous, viz. following an i.i.d. zero-mean Gaussian error distribution. It follows the optimal parameters estimation that minimizes the sum of squared prediction errors in the control points projections:

$$\widehat{\boldsymbol{\Omega}}_{\star} = \arg\min_{\widehat{\boldsymbol{\Omega}}} \sum_{n=1}^{N} \sum_{i} \left\| {}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}} - {}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}} \left(\widehat{\boldsymbol{\Omega}},\, \boldsymbol{\Upsilon}({}_{0}\boldsymbol{p}_i) \right) \right\|^2, \tag{3.3}$$

where ${}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}}$ are expected, distorted projections in S_{M} of the control points ${}_{0}\boldsymbol{p}_i$. These projections depend on both, the calibration parameters $\boldsymbol{\Omega}$ to be estimated (*i.e.*, the intrinsic matrix $\boldsymbol{A}$, the distortion parameters $\boldsymbol{k} = \{k_1,\, k_2,\, \ldots\}$, and absolute extrinsic parameters as by-products) and on the system model $\boldsymbol{\Upsilon}$ including the camera and lens distortion models as well as the calibration object model, e.g. ${}_{0}\boldsymbol{p}_i$.

In the case of a stereo camera system, the optimization can be readily extended as follows:

$$\widehat{\boldsymbol{\Omega}}_{\star}^{\mathrm{stereo}} = \arg\min_{\widehat{\boldsymbol{\Omega}}^{\mathrm{stereo}}} \sum_{c=1}^{C} \sum_{n=1}^{N} \sum_{i=1}^{M} \left\| {}^{c}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}} - {}^{c}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}} \left(\widehat{\boldsymbol{\Omega}}^{\mathrm{stereo}},\, \boldsymbol{\Upsilon}({}_{0}\boldsymbol{p}_i) \right) \right\|^2, \tag{3.4}$$

where the stereo calibration parameters $\widehat{\boldsymbol{\Omega}}^{\mathrm{stereo}}$ include the intrinsic parameters of further cameras ($\boldsymbol{A}_c$, $\boldsymbol{k}_c$) and their rigid, relative transformations ${}_{\mathrm{C}_c}\boldsymbol{T}^{\mathrm{C}}$ w.r.t. the main reference camera in S_{C}.

It is worth noting a variation of this method where lens distortion is being calibrated in advance, irrespective of the other intrinsic parameters of the camera. Indeed, lens distortion solely relies on the lens unit and not on the camera scaling parameters. Therefore, lens distortion can and should be estimated separately from regular intrinsic camera calibration. The predominant method is called plumb line method (or "straight lines have to be straight"); it consists in adjusting the distortion parameters so that they bend straight the distorted projections of actual lines in the scene, since it then corresponds to a linear projective transformation in homogeneous coordinates; refer to (Brown, 1971; Fryer; Duane C. Brown, 1986; Stein, 1993; Fryer *et al.*, 1994; Prescott and McLean, 1997; Devernay and Faugeras, 2001; Kang, 2001; El Melegy and Farag, 2003). An alternative method is self-calibration as in (Civera *et al.*, 2009). Note that partial imaging of the calibration object can lead to inaccuracy in the estimation of the origin of lens distortion (Malm and Heyden, 2001).

3.2.4 Summary

In this section 3.2 I introduced the standard camera calibration method currently used in computer vision vision applications (Zhang, 2000; Sturm and Maybank, 1999).

Starting out, I revisit the historic formation of the standard calibration method out of more complex calibration methods that required precision laboratory equipment and time consuming measurements (inadequate in our context). The standard method by Zhang, Sturm, and Maybank excels in its ease of use that, in turn, averts human or measurement mistakes that were otherwise bound to occur when using any of the former methods.

In a nutshell, the method requires images from a known, planar calibration object; unlike when using traditional approaches, it is not necessary to measure the pose of the calibration object w.r.t. the camera when taking calibration images. A first, ballpark solution is estimated by linear least squares methods using both, the 2-D homography between projected features in the image frame S_I and known features in the object frame S_0, as well as the rigid body motion constraints taking place during acquisition, see Eqs. (3.2). The initial solution hereby obtained is part of the bootstrap routine of a nonlinear optimization in Eq. (3.3) that considers the appropriate camera model as presented in Section 2.2.1. In the case of stereo cameras, Eq. (3.4) extends the former optimization taking the relative transformation between the cameras into account.

3.3 Extrinsic Camera Calibration

3.3.1 Introduction

Extrinsic camera calibration is the process of estimating the parameters that define the position and orientation (pose) of the (stereo) camera frame S_{C} w.r.t. some other reference frame external to the camera. The most widespread example are eye-in-hand systems where the camera (eye) is rigidly attached to the end-effector of an active robotic manipulator (hand) like the Kuka KR 16 or the DLR Lightweight Robot III (the pose of the TCP frame S_{T} w.r.t. their base frame S_{B} is operated), or the passive arm FaroArm Gold (the pose of the arm is reached by hand), see Fig. 3.2; in this case, the rigid body transformation ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ between the TCP frame S_{T} and the camera frame S_{C} is to be estimated (also called hand-eye transformation), refer to Section 2.3.

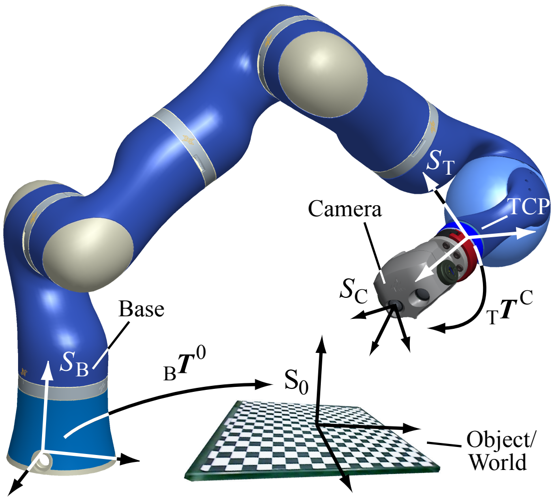


Figure 3.2: Rigid body transformation involved in the hand-eye calibration of a stereo camera mounted at the top of the DLR Light-Weight Robot 3.

In most applications there is a strong need for an accurate hand-eye calibration of the pose of S_{C} w.r.t. S_{T}. The reasons are mostly twofold:

a) to map sensor-centered measurements into the robot frame S_{B} or the world frame S_0, or

b) to allow for an accurate prediction of the pose of the sensor S_{C} on the basis of the robot arm motion.

In fact, these are often complementary aspects of the same problem.

3.3.2 State of the Art

There are two main approaches to estimate ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ when performing hand-eye calibration on the basis of both, the pose of the TCP frame S_{T} w.r.t. the robot base frame S_{B} ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$ and the pose of S_{C} w.r.t. the world/object frame S_0 ${}_0\boldsymbol{T}^{\mathrm{C}}$:

A. Move the hand and observe/perceive the motion of the eye or $\boldsymbol{AX} = \boldsymbol{XB}$, where $\boldsymbol{A}$ is the robot TCP motion ${}_{\mathrm{T}_1}\boldsymbol{T}^{\mathrm{T}_2}$, $\boldsymbol{B}$ the induced camera motion ${}_{\mathrm{C}_1}\boldsymbol{T}^{\mathrm{C}_2}$, and $\boldsymbol{X}$ is the hand-eye transformation ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ to be determined. This is the classical approach. Early solutions regard the rotational part of this equation decoupled from the translational one, yielding uncomplex and fast formulations. These are, however, error-prone as rotation estimation errors propagate to the translational part. Seminal articles propose least squares fitting of rotation, then translation, using the angle-axis representation (Shiu and Ahmad, 1989), and similar closed-form solutions in (Tsai and Lenz, 1989). The work in (Zhuang and Roth, 1991) simplified the formulation introducing quaternions for the estimation of the rotational part, in the same way as (Chou and Kamel, 1991), who make use of the singular value decomposition (SVD). Chen, for the first time, does not decouple rotational and translational terms by using the screw theory (Chen, 1991).

The work in (Wang, 1992) compares (Shiu and Ahmad, 1989) and (Tsai and Lenz, 1989), detecting a slight advantage for the latter. In (Zhuang and Shiu, 1993) the authors apply nonlinear optimization for both parts, minimizing a similar expression to Frobenius norms of homogeneous matrices of transformation errors. They additionally offer the possibility to disregard camera orientation for the estimation of the hand-eye transformation. A similar approach was presented in (Fassi and Legnani, 2005) and a nonlinear optimization in (Park and Martin, 1994), but again in the detached formulation. The work in (Lu and Chou, 1995) introduces the eight-space formulation based on quaternions, linearly optimizing both parts at the same time using the SVD. The work in (Horaud and Dornaika, 1995) nonlinearly optimizes both, the rotational (formulated with quaternions) and the translational parts, at the same time and with equal weights. Wei, Arbter, and Hirzinger in (Wei *et al.*, 1998) nonlinearly minimize algebraic distances performing simultaneous hand-eye and camera calibration. Daniilidis in (Daniilidis, 1999) introduces the dual quaternions—an algebraic representation of the screw theory to describe motions; these enable the author to find a fast joint solution for rotation and translation within linear formulation, again based on the SVD. The work in (Bayro–Corrochano *et al.*, 2000) also produces a linear solution of the coupled problem using the SVD, by the use of motors within the geometric algebra framework. Finally, Andreff *et al.* in (Andreff *et al.*, 2001) do the job properly, employing this particular formulation for X-from-motion applications—cf. with the last consideration in Section 3.3.3. They get rid of the nonlinear orthogonality constraint in $SO(3)$[5] by increasing the dimensionality of the rotational part, finally formulating the problem as a single homogeneous linear system.

[5] $SO(3)$ is the **S**pecial **O**rthogonal group of 3×3 matrices or rotation group for 3-D space.

B. Simultaneous estimation of the hand-eye transformation and the pose of the robot in the world or $\boldsymbol{AX} = \boldsymbol{ZB}$, where $\boldsymbol{A}$ is the pose of the camera frame S_{C} w.r.t. S_0 ${}_0\boldsymbol{T}^{\mathrm{C}}$, $\boldsymbol{B}$ the pose of the robot TCP S_{T} w.r.t. S_{B} ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$, and $\boldsymbol{X}$ and $\boldsymbol{Z}$ are the eye-hand and world-base transformations to be determined, *i.e.*, ${}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}$ and ${}_0\boldsymbol{T}^{\mathrm{B}}$, respectively. To the best of my knowledge it was Wang in (Wang, 1992) who first submitted this formulation explicitly for hand-eye calibration. Surprisingly, none of the other approaches refer to him in this respect.[6] Zhuang *et al.* in (Zhuang *et al.*, 1994) apply quaternions in order to get a simple linear solution of the rotational part by the use of the SVD. Rémy *et al.* in (Rémy *et al.*, 1997) nonlinearly optimize both parts by minimizing reprojected 3-D Euclidean error distances in S_0. Dornaika and Horaud in (Dornaika and Horaud, 1998) solve the rotational problem linearly using quaternions, and they also nonlinearly optimize both parts by minimization of Frobenius norms (with equal weights for all components) and two penalty functions—see Section 3.3.4. Other approaches integrate the hand-eye calibration with the intrinsic camera calibration and minimize the RMS of the reprojection errors in the image frame S_I (Zhuang *et al.*, 1995; Malm, 2003).

The optimization criteria for both approaches are often sub-optimal and, regrettably, no attention has been paid to proper parametrization of the components of the optimization formulae. Since the purpose of model-based[7] calibration is the accurate parametrization of the system model, *maximum-accuracy, optimal calibration is only achieved when minimizing model fitting errors with regard to the actually erroneous elements.* Here I propose a metric on the group of rigid transformations $SE(3)$ for this purpose.

Moreover, with the exception of (Rémy *et al.*, 1997), a thorough comparison of these very different approaches is missing. Here I show the most accurate algorithms along with a novel one—for both common approaches—and justify their use in relation to the nature of the problem.

3.3.3 Problem Description

Let ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$ be the homogeneous transformation relating the pose of the base frame S_{B} to the pose of the TCP frame S_{T}. ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$ results from the calibrated forward kinematic model of the robot, the encoder readings of every joint, and possibly its control parameters. Let again ${}_0\boldsymbol{T}^{\mathrm{C}}$ be the homogeneous transformation relating the pose of the object/world frame S_0 to the pose of the camera frame S_{C} (regardless of whether we use monocular or stereo vision). ${}_0\boldsymbol{T}^{\mathrm{C}}$ stems from the absolute extrinsic parameters of the camera calibration process in Section 3.2, refer to (Zhang, 2000).

[6] Admittedly, the main contribution in (Wang, 1992) was the comparison of the algorithms in (Shiu and Ahmad, 1989) and (Tsai and Lenz, 1989). At the same time, however, he produced this second family of solutions. Wang himself criticizes his *class A* calibration procedure as it yields biased results unless an error-free ${}_0\boldsymbol{T}^{\mathrm{B}}$ is specified. He actually fails to realize the necessity of estimating ${}_0\boldsymbol{T}^{\mathrm{B}}$ *at the same time* in order to avoid measurement innacuracies or mistakes; he uses ad hoc external measurements instead.

[7] Approaches that do not rely on a physical model of the system may actually perform on ocassions better if they are purposefully calibrated. I therefore add the adjective *model-based* to most procedures in this work.

Two unknown transformations ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ and ${}_{0}\boldsymbol{T}^{\mathrm{B}}$ remain to be estimated. The latter does not require frequent recalibration, since manipulators are not usually shifted. On the contrary, the rigid pose of the camera frame S_{C} w.r.t. the TCP frame S_{T} has to be calibrated more often, since camera(s) may be removed or rotated. These transformations should not be measured by hand since the different frames are located inside the manipulator or the sensor.

In order to uniquely determine ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ (and perhaps ${}_{0}\boldsymbol{T}^{\mathrm{B}}$), at least $n = 3$ stations (*i.e.*, robot configurations) are required (Tsai and Lenz, 1989; Chen, 1991)—specifically two motions with nonparallel rotation axes.

Solution #1: $AX = ZB$

Next the direct formulation of the *predictive parametric model* described in the last section will be mathematically detailed. It enables us to *predict* values (e.g. ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$) on the basis of a *parametric* representation of the world (e.g. ${}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}$). These predictions, jointly with actual measurements, make it possible to refine optimally on this parametric representation of the world in Section 3.3.4.

This first formulation directly states the rigid transformations involved in the loop camera-TCP-base-world-camera:

$$
{}_{0}\boldsymbol{T}^{\mathrm{C}}{}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}} = {}_{0}\boldsymbol{T}^{\mathrm{B}}{}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}} \quad \rightleftharpoons \quad \begin{tikzcd} S_{\mathrm{C}} \arrow[r, "{}_{\mathrm{C}}T^{\mathrm{T}}"] & S_{\mathrm{T}} \\ S_{0} \arrow[u, "{}_{0}T^{\mathrm{C}}"] \arrow[r, "{}_{0}T^{\mathrm{B}}"'] \arrow[ur] & S_{\mathrm{B}} \arrow[u, "{}_{\mathrm{B}}T^{\mathrm{T}}"'] \end{tikzcd} \quad , \tag{3.5}
$$

which avoids further modeling (e.g. perspective projection as in Section 2.2.1 or detailed kinematic joint/link information). It introduces a significant constraint in order to ensure that both, ${}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}$ and ${}_{0}\boldsymbol{T}^{\mathrm{B}}$, are consistent with the actual system. The equation is usually decomposed into its rotational and translational parts:

$$
\left.\begin{aligned} {}_{0}\boldsymbol{R}^{\mathrm{C}}{}_{\mathrm{C}}\boldsymbol{R}^{\mathrm{T}} &= {}_{0}\boldsymbol{R}^{\mathrm{B}}{}_{\mathrm{B}}\boldsymbol{R}^{\mathrm{T}} \\ {}_{0}\boldsymbol{R}^{\mathrm{C}}{}_{\mathrm{C}}\boldsymbol{t}^{\mathrm{T}} + {}_{0}\boldsymbol{t}^{\mathrm{C}} &= {}_{0}\boldsymbol{R}^{\mathrm{B}}{}_{\mathrm{B}}\boldsymbol{t}^{\mathrm{T}} + {}_{0}\boldsymbol{t}^{\mathrm{B}} \end{aligned}\right\} \quad , \tag{3.6}
$$

where $\boldsymbol{R}$ and $\boldsymbol{t}$ are the rotational and translational components of the homogeneous transformation matrices $\boldsymbol{T}$, respectively.

The solution to this problem has been historically calculated in different ways (Wang, 1992; Zhuang *et al.*, 1994; Rémy *et al.*, 1997; Dornaika and Horaud, 1998).

Solution #2: $AX = XB$

Due to the fact that ${}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}$ is more often required than ${}_{0}\boldsymbol{T}^{\mathrm{B}}$, most approaches eliminate the latter by writing Eq. (3.5) at two different instants i and j yielding the well-known hand-eye formulation:

$$
{}_{\mathrm{C}_i}\boldsymbol{T}^{\mathrm{C}_j}{}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}} = {}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}{}_{\mathrm{T}_i}\boldsymbol{T}^{\mathrm{T}_j} \quad \rightleftharpoons \quad \begin{tikzcd} {}_{0}\boldsymbol{T}^{\mathrm{C}_j} \arrow[r, "{}_{\mathrm{C}}T^{\mathrm{T}},\ ({}_{0}T^{\mathrm{B}})"] & {}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}_j} \\ {}_{0}\boldsymbol{T}^{\mathrm{C}_i} \arrow[u, "{}_{\mathrm{C}_i}T^{\mathrm{C}_j}"] \arrow[r, "{}_{\mathrm{C}}T^{\mathrm{T}},\ ({}_{0}T^{\mathrm{B}})"] \arrow[ur] & {}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}_i} \arrow[u, "{}_{\mathrm{T}_i}T^{\mathrm{T}_j}"'] \end{tikzcd} \tag{3.7}
$$

or

$$
\left.\begin{aligned} {}_{\mathrm{C}_i}\boldsymbol{R}^{\mathrm{C}_j}{}_{\mathrm{C}}\boldsymbol{R}^{\mathrm{T}} &= {}_{\mathrm{C}}\boldsymbol{R}^{\mathrm{T}}{}_{\mathrm{T}_i}\boldsymbol{R}^{\mathrm{T}_j} \\ {}_{\mathrm{C}_i}\boldsymbol{R}^{\mathrm{C}_j}{}_{\mathrm{C}}\boldsymbol{t}^{\mathrm{T}} + {}_{\mathrm{C}_i}\boldsymbol{t}^{\mathrm{C}_j} &= {}_{\mathrm{C}}\boldsymbol{R}^{\mathrm{T}}{}_{\mathrm{T}_i}\boldsymbol{t}^{\mathrm{T}_j} + {}_{\mathrm{C}}\boldsymbol{t}^{\mathrm{T}} \end{aligned}\right\} \quad , \tag{3.8}
$$

first formulated in (Shiu and Ahmad, 1989; Tsai and Lenz, 1989). This problem was geometrically analyzed in (Fassi and Legnani, 2005). When writing ${}_{\mathrm{C}_i}\boldsymbol{T}^{\mathrm{C}_j} = {}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}\,{}_{\mathrm{T}_i}\boldsymbol{T}^{\mathrm{T}_j}\,{}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$, it becomes clear that ${}_{\mathrm{C}_i}\boldsymbol{T}^{\mathrm{C}_j}$ and ${}_{\mathrm{T}_i}\boldsymbol{T}^{\mathrm{T}_j}$ are the same rigid body transformation assessed in different reference frames.

The solution has been also calculated in different ways: (Shiu and Ahmad, 1989; Tsai and Lenz, 1989; Zhuang and Roth, 1991; Chou and Kamel, 1991; Chen, 1991; Wang, 1992; Zhuang and Shiu, 1993; Fassi and Legnani, 2005; Park and Martin, 1994; Lu and Chou, 1995; Horaud and Dornaika, 1995; Wei *et al.*, 1998; Daniilidis, 1999; Bayro–Corrochano *et al.*, 2000; Andreff *et al.*, 2001). If necessary, the dual equation ${}_{0_i}\boldsymbol{T}^{0_j}\,{}_{0}\boldsymbol{T}^{\mathrm{B}} = {}_{0}\boldsymbol{T}^{\mathrm{B}}\,{}_{\mathrm{B}_i}\boldsymbol{T}^{\mathrm{B}_j}$ (where S_0 moves w.r.t. S_{C} and S_{B} w.r.t. S_{T}) enables the estimation of ${}_{0}\boldsymbol{T}^{\mathrm{B}}$.

Choice of formulation

The above equations do not hold exactly in the case of noise and $N > 3$ stations due to the erroneous rigid body transformations ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}_i}$ and ${}_{0}\boldsymbol{T}^{\mathrm{C}_i}$ for Solution #1 or ${}_{\mathrm{T}_i}\boldsymbol{T}^{\mathrm{T}_j}$ and ${}_{\mathrm{C}_i}\boldsymbol{T}^{\mathrm{C}_j}$ for Solution #2. The proper *optimal* way of accurately estimating ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ and ${}_{0}\boldsymbol{T}^{\mathrm{B}}$ is thus *optimally* correcting these erroneous measurements at every station $i \,/\, i \in \mathbb{N}_1,\, i \leq N$, depending on both, their geometric error models and the predictions/estimations from the formulae.

The **M**aximum **L**ikelihood method (ML) selects the model (e.g. ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$) where the probability of the observed data (e.g. ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}_i}$) is highest or, in other words, where its incompatibility with the model is minimized. A typical cost function for Gaussian error distributions is the sum of covariance-weighted squared predictions of these errors. Then, the resulting model parameters have zero bias, lowest variance, and maximum probability if flat prior (MacKay, 2003).

In the next section a metric for rigid body transformation errors is presented. Experiments using the real systems FaroArm Gold and ARTtrack2 suggest zero-mean Gaussian distributions[8] *if* this error metric is applied to the transformation ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$, see Section 3.3.5. Naturally, these errors are much larger than the ones in ${}_{0}\boldsymbol{T}^{\mathrm{C}}$; hence the errors in ${}_{0}\boldsymbol{T}^{\mathrm{C}}$ should not be considered (Tsai and Lenz, 1989). These experiments make it clear that, for this particular eye-in-hand framework, it is possible to *optimally* correct ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}_i}$ by minimizing a sum of covariance-weighted squared prediction errors in the context of the $\boldsymbol{AX} = \boldsymbol{ZB}$ formulation in order to *optimally* estimate ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ and ${}_{\mathrm{B}}\boldsymbol{T}^{0}$.

In the case of the $\boldsymbol{AX} = \boldsymbol{XB}$ formulation, the abovementioned considerations do *not* hold in the context of eye-in-hand systems using e.g. robotic manipulators, thus the proposed method is not optimal anymore; the reason for this is that ${}_{\mathrm{T}_i}\boldsymbol{T}^{\mathrm{T}_j}$ and ${}_{\mathrm{C}_i}\boldsymbol{T}^{\mathrm{C}_j}$ do not necessarily show Gaussian errors in this metric, but rather nonlinear functions of them. In the case of pose-from-motion problems where the camera motion is being estimated from its own images as in Chapter 5 or (Andreff *et al.*, 2001), however, the motion ${}_{\mathrm{C}_i}\boldsymbol{T}^{\mathrm{C}_j}$ *may* actually produce Gaussian errors in this metric (viz. much larger than the ones in ${}_{\mathrm{T}_i}\boldsymbol{T}^{\mathrm{T}_j}$ so that the latter can be ignored). More research in this direction is required.

[8] Of course, the metric in translation error (in squared form) shows a χ^2 distribution.

3.3.4 Minimizing Residual Errors on $\boldsymbol{SE}(3)$

In this section a novel distance metric on the Euclidean group of rigid body motions $SE(3)$ is presented. Elements in this group are represented as a couple $\{\boldsymbol{R}, \boldsymbol{t}\}$ where $\boldsymbol{R} \in SO(3) \,/\, \boldsymbol{R} \in \mathbb{R}^{3\times3} : \boldsymbol{R}^{\mathsf{T}}\boldsymbol{R} = \boldsymbol{I}$, $\det(\boldsymbol{R}) = 1$ and $\boldsymbol{t} \in \mathbb{R}^3$. The problem of finding a metric for $SE(3)$ can be presented in terms of a real valued objective function $\mathcal{O} : (SO(3) \times \mathbb{R}^3) \rightarrow \mathbb{R}$ which measures the suitability of the unknowns to fit the data. This metric will make it possible to optimally solve the hand-eye calibration problem.

Many choices exist in the literature for a metric in the context of the hand-eye calibration problem, but none directly refers to the actually erroneous transformations ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}_i}$ and ${}_{0}\boldsymbol{T}^{\mathrm{C}_i}$, hence they do not allow for optimal estimation following the ML criteria. In the following I present some significant approaches that will be also addressed within the simulations and experiments in Section 3.3.5.

a) The residuals in *linear optimization methods* are highly diverse: angles, rotation matrix elements, quaternion distances, etc., as well as nonlinear algebraic transformations of them.

b) A relevant metric is presented in (Horaud and Dornaika, 1995). The authors optimize nonlinearly for ${}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}$ in the context of the $\boldsymbol{AX} = \boldsymbol{XB}$ formulation as follows:

$$\{\boldsymbol{q}, \boldsymbol{t}\}^{\star} = \arg\min_{\boldsymbol{q},\boldsymbol{t}} (\mathcal{O}_H) = \arg\min_{\boldsymbol{q},\boldsymbol{t}} \left(f(\boldsymbol{q}, \boldsymbol{t}) + \lambda(1 - \boldsymbol{q} * \bar{\boldsymbol{q}})^2 \right) \qquad (3.9)$$

with

$$\begin{aligned} f(\boldsymbol{q}, \boldsymbol{t}) \; = \; & \lambda_1 \sum_{i=1}^{N-1} \left\| \boldsymbol{v}_i' - \boldsymbol{q} * \boldsymbol{v}_i * \bar{\boldsymbol{q}} \right\|^2 + \\ & \lambda_2 \sum_{i=1}^{N-1} \left\| \boldsymbol{q} * {}_{\mathrm{T}_i}\boldsymbol{t}^{\mathrm{T}_{i+1}} * \bar{\boldsymbol{q}} - \left({}_{\mathrm{C}_i}\boldsymbol{R}^{\mathrm{C}_{i+1}} - \boldsymbol{I} \right) \boldsymbol{t} - {}_{\mathrm{C}_i}\boldsymbol{t}^{\mathrm{C}_{i+1}} \right\|^2 \quad , \end{aligned} \qquad (3.10)$$

using weights $\lambda_1 = \lambda_2 = 1$ and $\lambda = 2 \cdot 10^6$. The latter factor λ guarantees the consistency of the quaternion vector $\boldsymbol{q}$ representing ${}_{\mathrm{C}}\boldsymbol{R}^{\mathrm{T}}$. $\boldsymbol{v}_i$ and $\boldsymbol{v}_i'$ are the eigenvectors associated with the unitary eigenvalues of ${}_{\mathrm{T}_i}\boldsymbol{R}^{\mathrm{T}_{i+1}}$ and ${}_{\mathrm{C}_i}\boldsymbol{R}^{\mathrm{C}_{i+1}}$, respectively. The objective function $\mathcal{O}_H$ has the form of a sum of squares of nonlinear functions and can be minimized e.g. using the Levenberg-Marquardt algorithm. This method considerably improved accuracy w.r.t. to earlier work.

c) The same authors, now in (Dornaika and Horaud, 1998), apply a different metric for the $\boldsymbol{AX} = \boldsymbol{ZB}$ formulation:

$$\begin{aligned} \left\{ {}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}, {}_{0}\boldsymbol{T}^{\mathrm{B}} \right\}^{\star} = \; & \arg\min_{{}_{\mathrm{T}}T^{\mathrm{C}},\, {}_{0}T^{\mathrm{B}}} (\mathcal{O}_D) = \arg\min_{{}_{\mathrm{T}}T^{\mathrm{C}},\, {}_{0}T^{\mathrm{B}}} \Big(f\left({}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}, {}_{0}\boldsymbol{T}^{\mathrm{B}} \right) + \\ & \lambda_3 \left\| {}_{\mathrm{T}}\boldsymbol{R}^{\mathrm{C}} {}_{\mathrm{C}}\boldsymbol{R}^{\mathrm{T}} - \boldsymbol{I} \right\|^2 + \lambda_4 \left\| {}_{0}\boldsymbol{R}^{\mathrm{B}} {}_{\mathrm{B}}\boldsymbol{R}^{0} - \boldsymbol{I} \right\|^2 \Big) \end{aligned} \qquad (3.11)$$

with

$$
\begin{aligned}
f\left({}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}},\, {}_{0}\boldsymbol{T}^{\mathrm{B}}\right) = \lambda_1 \sum_{i=1}^{N} \left\| {}_{0}\boldsymbol{R}^{\mathrm{C}}\,{}_{\mathrm{C}}\boldsymbol{R}^{\mathrm{T}} - {}_{0}\boldsymbol{R}^{\mathrm{B}}\,{}_{\mathrm{B}}\boldsymbol{R}^{\mathrm{T}} \right\|^2 + \\
\lambda_2 \sum_{i=1}^{N} \left\| {}_{0}\boldsymbol{R}^{\mathrm{C}}\,{}_{\mathrm{C}}\boldsymbol{t}^{\mathrm{T}} + {}_{0}\boldsymbol{t}^{\mathrm{C}} - {}_{0}\boldsymbol{R}^{\mathrm{B}}\,{}_{\mathrm{B}}\boldsymbol{t}^{\mathrm{T}} - {}_{0}\boldsymbol{t}^{\mathrm{B}} \right\|^2 \quad ,
\end{aligned}
\tag{3.12}
$$

again with weights $\lambda_1 = \lambda_2 = 1$ and $\lambda_3 = \lambda_4 = 10^6$. This time, the objective function $\mathcal{O}_D$ minimizes the Frobenius norm *of a residual rotation matrix.*

d) Daniilidis in (Daniilidis, 1999) presents a unified and fast way of formulating the $\boldsymbol{AX} = \boldsymbol{XB}$ problem using dual-quaternions, which are the algebraic counterparts of screws and are valid for both, rotational and translational components. His method aims at avoiding the abovementioned weighting problem by using the compact formulation $\breve{\boldsymbol{a}} = \breve{\boldsymbol{q}}\,\breve{\boldsymbol{b}}\,\bar{\breve{\boldsymbol{q}}}$, which can be solved linearly using the SVD.

Note the absence of any weighting criteria between positional and orientational residuals—sometimes even the weighting factor is missing; in my view, the sensible choice of these factors is essential to purposefully aim at optimal, total estimators. In addition, parametrization criteria have a strong influence in the abovementioned methods: the scaling of variables (*i.e.*, the choice of units relating meaningfully to the problem structure) and, more generally, matrix preconditioning (the choice of the linear combinations of parameters to use) are crucial for convergence and, unfortunately, play an unexpected role in the weighting issue (Hartley, 1997).

I next describe a sound objective function $\mathcal{O}$ based on an error metric on $SE(3)$. For the first time it addresses the following objectives:

1. To *optimally* reduce actual system errors,
2. to allow for a natural weighting of the rotational and translational components, and
3. for the algorithm to be able to autonomously adjust the latter weighting factors.

In addition, the proposed metric sorts well with both, the $\boldsymbol{AX}=\boldsymbol{XB}$ and the $\boldsymbol{AX}=\boldsymbol{ZB}$ formulations.

Metric for rotation error

Any rigid body transformation can be modeled as a rotation in $SO(3)$ by an angle θ about an axis $\boldsymbol{p}$ through the origin, followed by a translation $\boldsymbol{t}$ in $\mathbb{R}^3$. Rotation can thus be represented by three independent parameters: the rotation angle θ and two angles $\{\alpha,\, \beta\}$ defining the axis of rotation $\boldsymbol{p}$.

Actual residual error rotations such as the rotational residual at S_{T}

$$
{}_{\Delta\mathrm{T}}\widetilde{\boldsymbol{R}} = {}_{\mathrm{T}}\widetilde{\boldsymbol{R}}^{\mathrm{B}}\,{}_{\mathrm{B}}\widehat{\boldsymbol{R}}^{\mathrm{T}} \quad ,
\tag{3.13}
$$

where ${}_{\mathrm{B}}\widetilde{\boldsymbol{R}}^{\mathrm{T}}$ is measured (e.g. from joint encoders) and ${}_{\mathrm{B}}\widehat{\boldsymbol{R}}^{\mathrm{T}}$ is estimated by e.g. Eq. (3.5), usually present small angles θ. Experiments[9] show arbitrary (randomly distributed) axes of rotation for these residual rotations in most tracking systems. Hence *the axes of rotation of residual rotations are disregarded.* The following rotational error metric is proposed:

$$\mathcal{O}_i^{\mathrm{rot}} = {}_{\Delta\mathrm{T}}\theta_i = \pm \arccos\left(\frac{\mathrm{trace}({}_{\Delta\mathrm{T}}\widetilde{\boldsymbol{R}}_i) - 1}{2}\right) = {}_{\Delta\mathrm{B}}\theta_i \quad . \tag{3.14}$$

This scalar, geometrically-defined metric[10] in $SO(3)$ gauges *the actual residual rotation error* and is frame-invariant, *i.e.*, ${}_{\Delta\mathrm{T}}\theta_i = {}_{\Delta\mathrm{B}}\theta_i$. In (Wang, 1992) the similar metric $\mathcal{O}_i^{\mathrm{rot}\prime} = \mathcal{N}({}_{\mathrm{B}}\widetilde{\boldsymbol{R}}^{\mathrm{T}_i}, {}_{\mathrm{B}}\widehat{\boldsymbol{R}}^{\mathrm{T}_i}) = 2\,\sin(|{}_{\Delta\mathrm{T}}\theta_i|/2)$ is introduced for assessment of the accuracy of the calibration; its geometrical interpretation was the Euclidean distance between head ends of rotated unit vectors.

Metric for translation error

The natural metric of translation in Euclidean space is the Euclidean distance. In the context of rigid body motions, the Euclidean distance is *not*, however, frame-invariant, as

$$\begin{aligned}
{}_{\mathrm{T}_i}\tilde{\boldsymbol{t}} = {}_{\mathrm{B}}\hat{\boldsymbol{t}}^{\mathrm{T}_i} - {}_{\mathrm{B}}\tilde{\boldsymbol{t}}^{\mathrm{T}_i} \neq {}_{\mathrm{B}_i}\tilde{\boldsymbol{t}} = {}_{\mathrm{T}_i}\hat{\boldsymbol{t}}^{\mathrm{B}} - {}_{\mathrm{T}_i}\tilde{\boldsymbol{t}}^{\mathrm{B}} \quad \Leftrightarrow \quad & \{{}_{\Delta\mathrm{T}}\theta_i = {}_{\Delta\mathrm{B}}\theta_i > 0\}\,, \\
& \{\|{}_{\mathrm{T}_i}\tilde{\boldsymbol{t}}\| > 0 \ \vee \ \|{}_{\mathrm{B}_i}\tilde{\boldsymbol{t}}\| > 0\}\ .
\end{aligned} \tag{3.15}$$

In addition, it is difficult to choose the reference frame that actually shows the most significant translation errors, as this depends on the pose tracking system used. In absence of further model information, I therefore choose not to use a single Euclidean residual distance as a metric for translation error; the equitable balance between these two symmetrical Euclidean distances is chosen instead:

$$\mathcal{O}_i^{\mathrm{tra}} = (\,\|{}_{\mathrm{T}_i}\tilde{\boldsymbol{t}}\| + \|{}_{\mathrm{B}_i}\tilde{\boldsymbol{t}}\|\,)\,/\,2 \quad . \tag{3.16}$$

Combination of both metrics

As stated above, the ML method estimates the *optimal* model (e.g. ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$) by means of the minimization of the sum of covariance-weighted squared prediction errors with zero-mean Gaussian error distributions. Hence the total transformation error cost function results in:

$$\mathcal{O}_i = \frac{(\mathcal{O}_i^{\mathrm{rot}})^2}{{}^{\star}\sigma_{\mathrm{rot}}^2} + \frac{(\mathcal{O}_i^{\mathrm{tra}})^2}{{}^{\star}\sigma_{\mathrm{tra}}^2} \quad , \tag{3.17}$$

[9] Experiments on pose accuracy for both, the FaroArm Gold and the ARTtrack2 system, were performed by comparing their readings with the absolute extrinsics obtained from stereo camera calibration.

[10] The trace of a rotation matrix $\boldsymbol{R}$ is independent of the coordinate system used (as long as it is orthonormal). Hence it matches the sum of the eigenvalues of $\boldsymbol{R}$, that is $1 + (\cos\theta + i\sin\theta) + (\cos\theta - i\sin\theta) = 1 + 2\cos\theta \Rightarrow \theta = \pm\arccos((r_{11} + r_{22} + r_{33} - 1)/2)$.

where ${}^{\star}\sigma_{\mathrm{rot}}^2$ and ${}^{\star}\sigma_{\mathrm{tra}}^2$ are the 2nd central moments of the independent Gaussian **p**robability **d**ensity **f**unctions (pdfs) in rotation and translation error, respectively. Eventually e.g. in the case of the $\boldsymbol{AX} = \boldsymbol{ZB}$ formulation:

$$
\begin{aligned}
\left\{{}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}},\, {}_{\mathrm{B}}\boldsymbol{T}^{0}\right\}^{\star} &= \arg \min_{{}_{\mathrm{T}}T^{\mathrm{C}},\, {}_{\mathrm{B}}T^{0}} \left(\sum_{i=1}^{N} \frac{(\mathcal{O}_i^{\mathrm{rot}})^2}{{}^{\star}\sigma_{\mathrm{rot}}^2} + \frac{(\mathcal{O}_i^{\mathrm{tra}})^2}{{}^{\star}\sigma_{\mathrm{tra}}^2} \right) \\
&= \arg \min_{{}_{\mathrm{T}}T^{\mathrm{C}},\, {}_{\mathrm{B}}T^{0}} \left(\sum_{i=1}^{N} (\mathcal{O}_i^{\mathrm{rot}})^2 + \frac{(\mathcal{O}_i^{\mathrm{tra}})^2}{({}^{\star}\sigma_{\mathrm{tra}}/{}^{\star}\sigma_{\mathrm{rot}})^2} \right) \quad , \qquad (3.18)
\end{aligned}
$$

where ${}^{\star}\sigma_{\mathrm{tra}}/{}^{\star}\sigma_{\mathrm{rot}}$ is the *position/orientation precision ratio*, which is now the only required weighting parameter for optimal estimation. Numerical optimizations are to be used to find the solutions.

Automatic, optimal weighting

The sigma values above refer to the actual precision characteristics of the particular pose tracking system being used. Even though these parameters could be determined by experiments, in this section I bring forward a more convenient approach that estimates the abovementioned ratio based exclusively on the same data used for calibration. This is a further appealing virtue of the metrics presented above.

It is well known that the mean square rotational and translational residuals tend to their actual 2nd central moments when $N \to \infty$, *i.e.*, for optimal values of $\{{}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}},\, {}_{\mathrm{B}}\boldsymbol{T}^{0}\}^{\star}$ it holds

$$\sum_{i=1}^{N} ({}^{\star}\mathcal{O}_i^{\mathrm{rot}})^2/N \to {}^{\star}\sigma_{\mathrm{rot}}^2 \quad \text{and} \qquad (3.19)$$

$$\sum_{i=1}^{N} ({}^{\star}\mathcal{O}_i^{\mathrm{tra}})^2/N \to {}^{\star}\sigma_{\mathrm{tra}}^2 \quad . \qquad (3.20)$$

Fortunately, non-optimal but approximated model parameters $\{{}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}},\, {}_{\mathrm{B}}\boldsymbol{T}^{0}\}$ resulting from an optimization with arbitrary prior weightings (e.g. a unitary position/orientation precision ratio) yield *improved* weighting parameters, so that it also holds that

$$\sigma_{\mathrm{rot}}^2 = \sum_{i=1}^{N} (\mathcal{O}_i^{\mathrm{rot}})^2/N \rightsquigarrow {}^{\star}\sigma_{\mathrm{rot}}^2 \quad \text{and} \qquad (3.21)$$

$$\sigma_{\mathrm{tra}}^2 = \sum_{i=1}^{N} (\mathcal{O}_i^{\mathrm{tra}})^2/N \rightsquigarrow {}^{\star}\sigma_{\mathrm{tra}}^2 \quad . \qquad (3.22)$$

Both, simulations and experiments, suggest that this process always converges to the optimal position/orientation precision ratio if the approximated sigma values σ_{rot} and σ_{tra} are updated with the optimized rotational and translational residuals after every optimization; after about 3 iterations the results become optimal, refer to Fig. 3.3. The camera calibration toolbox DLR CalDe and DLR CalLab in (Strobl *et al.*, 2005) implements this algorithm.

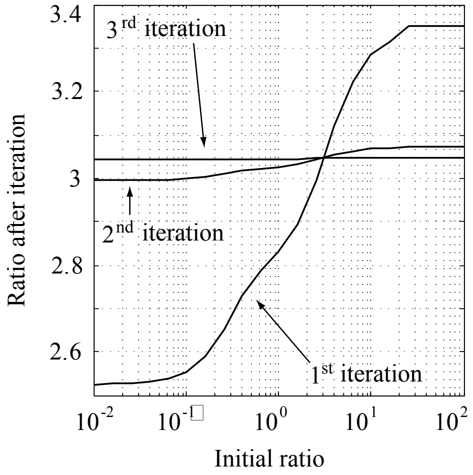


Figure 3.3: Position/orientation precision ratio adaption: For any initial ratio chosen (axis of abscissas), an optimization following Eq. (3.18) produces residuals that can be used to, in turn, update the precision ratio (axis of ordinates). The optimal precision ratio will be reached after three iterations.

3.3.5 Simulations and Experiments

Simulations

Simulations were conducted to compare both solutions $\boldsymbol{AX}=\boldsymbol{ZB}$ and $\boldsymbol{AX}=\boldsymbol{XB}$ applying the proposed metric, as well as five other representative methods. Simulations facilitate the comparison of *estimated* and *actual* values, as—unlike in experiments—actual values are known. In particular, these simulations compare nonlinear minimization on $SE(3)$ as proposed in the last section (from now on "*SE*" for $\boldsymbol{AX}=\boldsymbol{ZB}$ and "*se*" for $\boldsymbol{AX}=\boldsymbol{XB}$) with the five well-known methods briefly presented in the last section: linear least squares solution in rotation ("*li*"), nonlinear minimization (Horaud and Dornaika, 1995) ("*nl*"), and dual-quaternion (Daniilidis, 1999) ("*dq*") approaches for $\boldsymbol{AX}=\boldsymbol{XB}$; linear least squares solution in rotation ("*LI*") and nonlinear minimization of Frobenius norms (Dornaika and Horaud, 1998) ("*FN*") approaches for $\boldsymbol{AX}=\boldsymbol{ZB}$.

On the one hand, most of these methods lack of a convenient rotation/translation weighting policy. On the other hand, the novel metric proposed here is able to perform weighting automatically. I take advantage of this situation and study some of these methods in relation to their weighting parameters and, in the case of the method in (Horaud and Dornaika, 1995), I introduce a novel weighting policy in order to boost performance: It simply aims at bringing both, translational (Euclidean distances) and rotational (quaternion distances) errors, to the the same order of magnitude.

The simulation is conducted as follows: 9 camera stations were generated at each altitude ${}_0z$ of 25 and 45 cm w.r.t. S_0 with both ${}_0x$ and ${}_0y \in \{-30,\ 0,\ 30\}$ cm, *i.e.*, $n=18$ stations altogether are used for calibration. All stations focus on the origin of S_0. Since the simulation results are presented statistically, this series was randomly repeated 100 times in the form of a Monte Carlo simulation. Nominal, arbitrary—but realistic—values for the unknown transformations ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ and ${}_{\mathrm{B}}\boldsymbol{T}^{0}$ are generated for every calibration attempt, being the only restriction as follows: $\left\| {}_{\mathrm{T}}\boldsymbol{t}^{\mathrm{C}} \right\| = \left\| {}_{\mathrm{B}}\boldsymbol{t}^{0} \right\| = 30$ cm. These enable the calculation of the *actual* ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}_i}$ transformations. In turn, random errors provide ${}_{\mathrm{B}}\widetilde{\boldsymbol{T}}^{\mathrm{T}}$. The Monte Carlo simulations for the methods *nl*, *FN*, *SE*, and *se* were repeated 21 times with different weighting factors. The average results assess accuracy and error minimization performance.

As previously stated in Section 3.3.3, the robot arm pose measurement ${}_{\mathrm{B}}\widetilde{\boldsymbol{T}}^{\mathrm{T}}$ is expected to be the main source of perturbation. Here two different noisy models are simulated: *noise model* #1 applies for general, time-independent inaccurate pose data, and *noise model* #2 to general time-independent inaccurate *motion* data—which is time-*dependent* inaccurate pose data since it implies growing pose innacuracy over time. It will be shown that the latter are best dealt using the $\boldsymbol{AX} = \boldsymbol{XB}$ formulation, whereas the former are best dealt using the $\boldsymbol{AX} = \boldsymbol{ZB}$ formulation. In particular, measurement noise was included for the *noise model* #1 in orientation[11] with ${}_{\Delta\mathrm{T}}\widetilde{\boldsymbol{R}}$ and ${}_{\Delta\mathrm{B}}\widetilde{\boldsymbol{R}}$, having rotation angles θ granted to be unbiased, with Gaussian pdfs with $\sigma_\theta = 0.15\,°$. The axes of rotation $\boldsymbol{p}$ of these rotation matrices are uniformly distributed, *i.e.*, $\alpha \in [-90°, 90°)$ with $\mathrm{pdf}(\alpha) = 180^{-1}\,[°]^{-1}$ and $\beta \in [-90°, 90°)$ with $\mathrm{pdf}(\beta) \propto \arcsin(\beta/90)\,[°]^{-1}$. In position, the Euclidean residual distances ${}_{\mathrm{T}}\tilde{\boldsymbol{t}}$ and ${}_{\mathrm{B}}\tilde{\boldsymbol{t}}$ also present real Gaussian pdfs with $\sigma_t = 0.35$ mm, and their directions are again uniformly distributed. For *noise model* #2 I use ${}_{\mathrm{T}_i}\widetilde{\boldsymbol{R}}^{\Delta\mathrm{T}_j} = {}_{\mathrm{T}_j}\widetilde{\boldsymbol{R}}^{\mathrm{T}_i}\,{}_{\mathrm{T}_i}\widehat{\boldsymbol{R}}^{\mathrm{T}_j}$ and ${}_{\mathrm{T}_i}\tilde{\boldsymbol{t}}^{\Delta\mathrm{T}_j} = {}_{\mathrm{T}_i}\hat{\boldsymbol{t}}^{\mathrm{T}_j} - {}_{\mathrm{T}_i}\tilde{\boldsymbol{t}}^{\mathrm{T}_j}$.

Robustness analyses in the presence of varying noise levels were also performed. The conclusions are in conformity with prior works: for common applications the superiority of an algorithm does not critically depend on the level of noise. Consequently, these studies are not reported here.

Accuracy analysis with synthetic data

Next both accuracy and precision[12] in the estimation of ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ and ${}_{\mathrm{B}}\boldsymbol{T}^{0}$ are examined in relation to the chosen method, the error model, and the weighting parameters.

Primarily the simulations reflect the operation of the Maximum Likelihood (ML) method as stated in Section 3.3.3: Error standard deviations are much larger than biases—at least 10 times. In addition, under mild regularity conditions on the measurement distributions, the posterior distributions of the ML estimates converge asymptotically in probability to Gaussians. Therefore the accuracy analyses performed here focus on the 2nd central moments of the estimation errors.

Figs. 3.4 and 3.5 show the standard deviations in position and orientation estimation for ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ and ${}_{0}\boldsymbol{T}^{\mathrm{B}}$. The figures totally differ in the fact that in Fig. 3.4 the $\boldsymbol{AX} = \boldsymbol{ZB}$ approaches (upper case) show better performance, whereas in

[11] Note that this error model does not completely correspond to the metric proposed in Section 3.3.4. Here rotational errors appear both in S_{T} and S_{B}.

[12] Accuracy refers to the agreement of estimations and actual values, whereas precision refers to the repeatability of estimations.

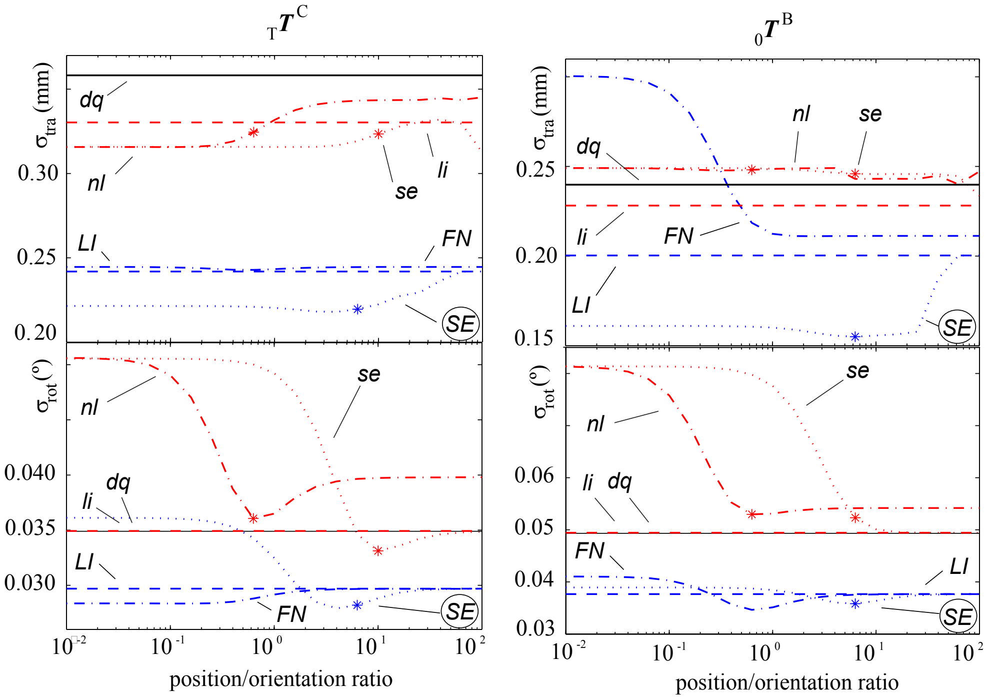


Figure 3.4: Standard deviations of the parameter errors with *noise model* #1. Optimal values regarding weighting are marked with '*'.

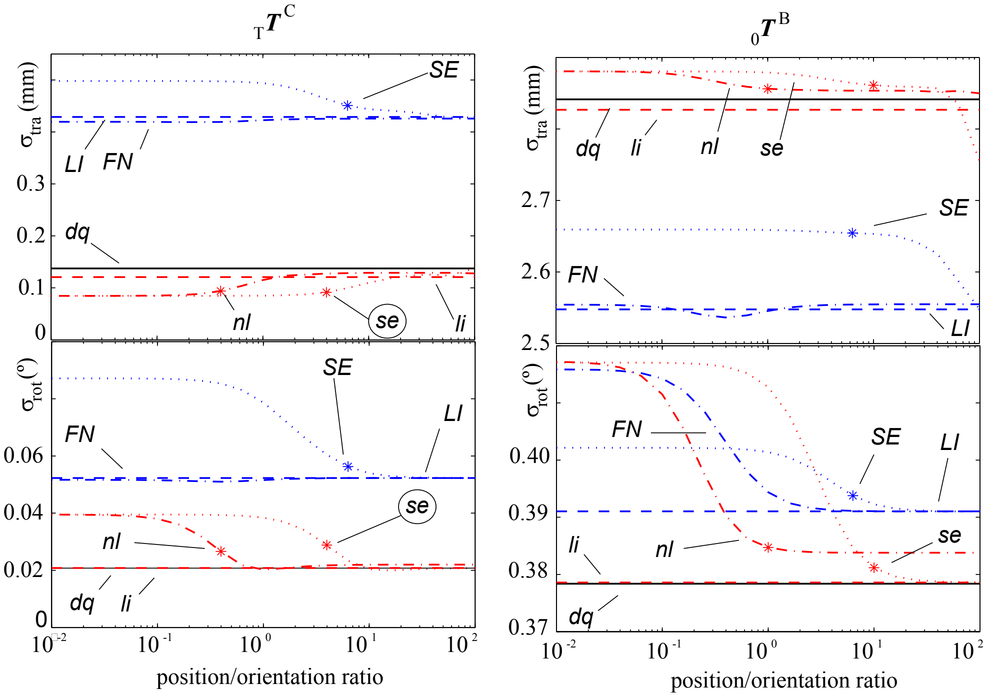


Figure 3.5: Standard deviations of the parameter errors with *noise model* #2. Optimal values regarding weighting are marked with '*'.

Fig. 3.5 (${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$) the $\boldsymbol{AX}=\boldsymbol{XB}$ approaches (lower case) do. In particular, for both approaches the methods developed here (*SE* and *se*) show highest precision. In addition, proper weighting proves crucial to optimal estimation—this applies with novelty to the method *nl* in (Horaud and Dornaika, 1995) as well as to both methods presented here (*SE* and *se*). Note that without this weighting policy but with $\lambda_1 = \lambda_2 = 1$ (*i.e.*, ratio $= 10^0$) the *nl* method would perform even worse. Besides, the results with *noise model* #2 in the right-hand side of Fig. 3.5 do not show relevant findings, since for this model neither of the solutions is optimal for the estimation of ${}_0\boldsymbol{T}^{\mathrm{B}}$.

Although parametrization and conditioning have been taken into account, in the simulations the dual-quaternion approach does not work any better than the linear estimation methods do. Perhaps, this fast method can be useful in the context of *X-from-motion* problems, see (Schmidt *et al.*, 2005).

Model matching analysis with synthetic calibration data

Figs. 3.6 and 3.7 show the 2nd central moments of the error metrics presented in Section 3.3.4 for the very same simulations. They help to understand that:

- linear solutions minimizing rotational errors (*li*) represent only one potential extreme solution for the hand-eye calibration problem,
- it is *critical* for optimal calibration to be able to find a proper weighting,
- approaches for the solution $\boldsymbol{AX}=\boldsymbol{XB}$ perform poorly with *noise model* #1 because the error reduction that they get in ${}_{\mathrm{T}_i}\boldsymbol{T}^{\mathrm{T}_j}$ is lower than the error reduction the approaches for the solution $\boldsymbol{AX}=\boldsymbol{ZB}$ can get in ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}_i}$ (the opposite applies when using *noise model* #2).

Experiments

In this section the performance of the different algorithms in real systems is demonstrated. The pose of S_{T} w.r.t. S_{B} stems from the output of the ART-track2 system. ${}_0\boldsymbol{T}^{\mathrm{C}}$ is in turn provided by a stereo camera calibraton algorithm derived from the monocular one in (Zhang, 2000), using DLR CalDe and DLR CalLab (Strobl *et al.*, 2005). Experiments aim at verifying correct operation of the methods in real systems. As in the simulations above, estimation accuracy would be evidence of correct operation. Unfortunately, calibration accuracy can not be *directly* assessed as ground-truth information is missing in experiments. It is possible, however, to *indirectly* evaluate calibration performance by verifying the cabability of the system model to match the calibration output, as the results that best fit the model stem from an optimal calibration process (MacKay, 2003). For instance, testing the ability to predict ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$ on the basis of ${}_0\boldsymbol{T}^{\mathrm{C}}$ and the optimal solutions ${}^{\star}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$ and ${}^{\star}_{\mathrm{B}}\widehat{\boldsymbol{T}}^{0}$ in several verification stations; these experiments are presented last in this section. Alternatively, it is still possible to *indirectly* evaluate calibration performance by using task-dependent metrics.[13]

[13] For instance, you may evaluate the ability to predict camera poses, or image projections, using only measurement data. These practices are, however, undesirable since the assessments

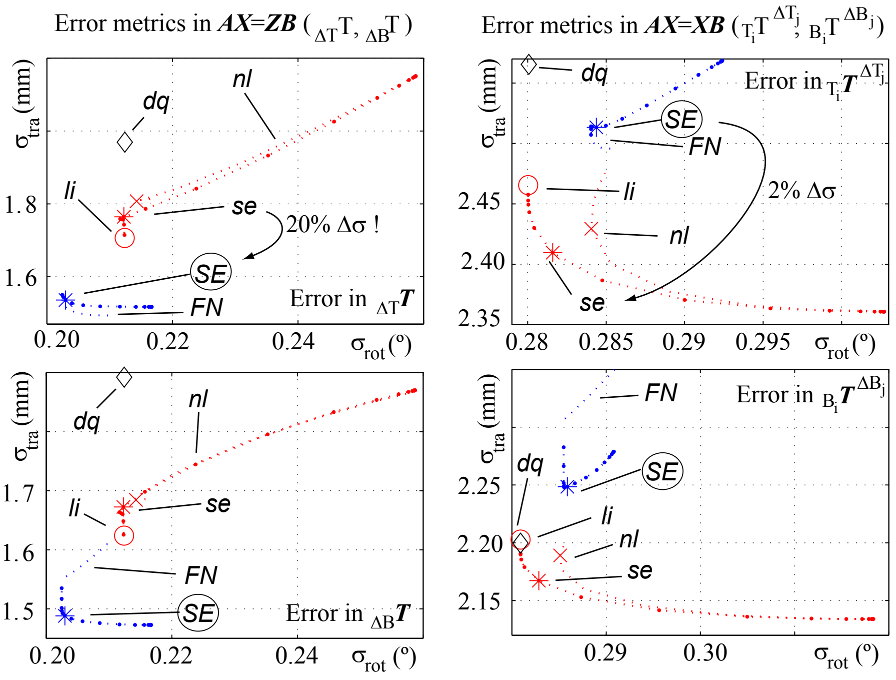


Figure 3.6: Standard deviations of different error metrics with *noise model* #1. Optimal values regarding weighting are marked with '*'.

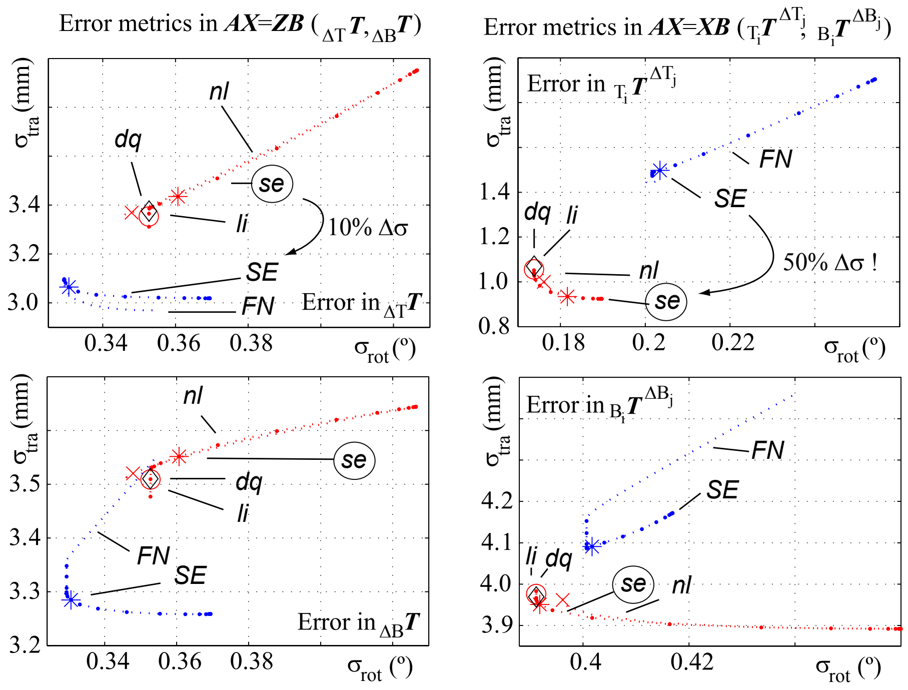
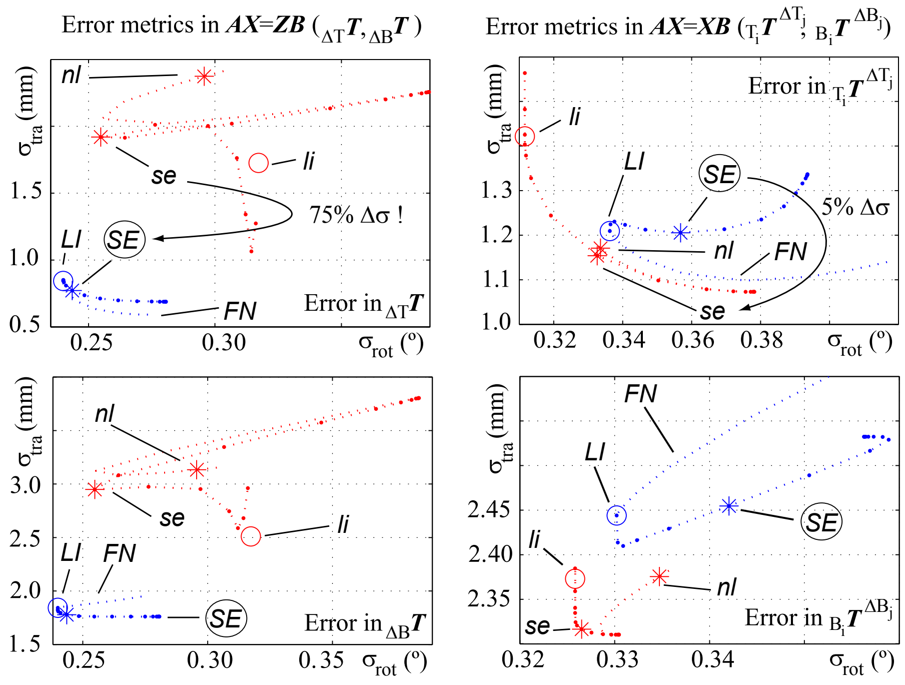


Figure 3.7: Standard deviations of different error metrics with *noise model* #2. Optimal values regarding weighting are marked with '*'.

Model matching analysis with real calibration data

Following on from the last section, I next observe the model matching capability of different calibration algorithms on real experiments in order to relate them to their virtual counterparts in the simulations section. Again, the 2nd central moments of the metrics presented in Section 3.3.4 are shown in Fig. 3.8 for an experiment consisting of 10 stations. Results resemble to a great extent Fig. 3.6, in a slightly different order of magnitude.[14] This suggests that the system presents *noise model* #1. In this case, the simulations section points to the method *SE* (minimization on $SE(3)$ within $\boldsymbol{AX} = \boldsymbol{ZB}$) in order to optimally estimate $_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ and $_{\mathrm{B}}\boldsymbol{T}^{0}$.


Error metrics in $\boldsymbol{AX}$=$\boldsymbol{ZB}$ ($_{\Delta\mathrm{T}}\boldsymbol{T}$, $_{\Delta\mathrm{B}}\boldsymbol{T}$)

Error metrics in $\boldsymbol{AX}$=$\boldsymbol{XB}$ ($_{\mathrm{T_i}}\boldsymbol{T}^{\Delta\mathrm{T_j}}$; $_{\mathrm{B_i}}\boldsymbol{T}^{\Delta\mathrm{B_j}}$)

Figure 3.8: Standard deviations of different error metrics in a real hand-eye calibration. Optimal values regarding weighting are marked with '*'.

will incorporate task-dependent requirements. For instance, for a very long link $_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$, the accuracy of camera pose estimation would be strongly influenced by the TCP orientation error. In this case orientation error minimization algorithms would be preferred—which certainly *do not* parameterize the system model properly as explained in Section 3.3.4.

[14] An important attribute of the automatic weighting method exposed in Section 3.3.4 is its independency of the order of magnitude of the system's noise—the only required parameter being a *ratio* of the precisions in the system.

Model matching analysis with real verification data

The latter results do not suffice to indirectly evaluate model-based calibration performance, since the calibration process purposefully forced these particular data to comply with the parameterized model. To correctly verify the predictive capability of Eq. (3.5) in a general case, I acquired data external to the calibration process in the form of 27 additional TCP stations and their corresponding camera pose estimations. Table 3.1 presents the 2nd central moments of the metrics in Section 3.3.4 for 27 verification stations with noisy ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$. In the upper table the calibration results ${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$ and ${}_{\mathrm{B}}\widehat{\boldsymbol{T}}^{0}$ from the last section have been used. It is good news that the *verification* stations still present low discrepancies between measurements and predictions. In the lower table the calibration results of an ex**t**ensive (-*t*) hand-eye calibration with 37 stations were used. The latter show better results, as expected. This is due to the few stations used for the former calibration[15] as well as to remaining modeling errors (e.g. in camera pose estimation). The proposed *SE-t* approach is the least affected in this concern. Apart from that, all results confirm superior performance for the *SE* estimation approach.

Table 3.1: 2nd central moments of the metrics presented in Section 3.3.4 during 27 verification stations with noisy base-to-TCP transformations ${}_{\mathrm{B}}\boldsymbol{T}^{\mathrm{T}}$. In the upper table the calibration results ${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$ and ${}_{\mathrm{B}}\widehat{\boldsymbol{T}}^{0}$ obtained in the last section (*i.e.*, using 10 stations for calibration) have been used. In the lower table the calibration results ${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$ and ${}_{\mathrm{B}}\widehat{\boldsymbol{T}}^{0}$ of a more comprehensive calibration including the latter 10 stations as well as the 27 verification stations (*i.e.*, 37 stations in total) have been used.

	${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$ and ${}_{\mathrm{B}}\widehat{\boldsymbol{T}}^{0}$ from previous calibration (10 stations)							
	$\boldsymbol{AX}=\boldsymbol{ZB}$			$\boldsymbol{AX}=\boldsymbol{XB}$				
	SE	*FN*	*LI*	*se*	*nl*	*dq*	*li*	
σ_θ	0.210	0.213	0.213	0.227	0.239	0.516	0.270	[°]
σ_t	1.343	1.388	1.386	2.401	2.597	3.735	1.969	[mm]

	${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$ and ${}_{\mathrm{B}}\widehat{\boldsymbol{T}}^{0}$ from comprehensive calibration (37 stations)							
	$\boldsymbol{AX}=\boldsymbol{ZB}$			$\boldsymbol{AX}=\boldsymbol{XB}$				
	SE-t	*FN-t*	*LI-t*	*se-t*	*nl-t*	*dq-t*	*li-t*	
σ_θ	0.199	0.200	0.200	0.215	0.216	0.341	0.241	[°]
σ_t	1.171	1.217	1.216	1.463	1.390	2.424	1.549	[mm]

[15] Optimal estimation provides unbiased results only for numerous erroneous data. Here the common number of 10 stations is used.

3.3.6 Summary

In this section 3.3 I presented a calibration method for eye-in-hand systems that estimates both, the hand-eye transformation ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ as well as the robot-to-world transformation ${}_{\mathrm{B}}\boldsymbol{T}^{0}$. Eye-in-hand systems attaching cameras at the end-effectors of robotic manipulators are the most common approach currently used to promote their autonomy.

Starting out, I distinguish between the two common solutions of the hand-eye calibration problem: the $\boldsymbol{AX} = \boldsymbol{ZB}$ and the $\boldsymbol{AX} = \boldsymbol{XB}$ formulations. I furthermore categorize relevant literature in this respect.

Different from traditional approaches that minimize residuals irrespective of their physical meaning, the proposed optimization takes place in terms of a parametrization of a realistic stochastic model. In order to perform optimally in the context of the Maximum Likelihood method, a metric on the group of the rigid transformations $SE(3)$ together with an experimentally validated error model are proposed for nonlinear optimization. The translational and rotational components of the residuals are weighted with the particular precision characteristics of the manipulator. This metric allows for automatically adapting its weights to the precision characteristics of the system. The novel metric works well with both common formulations $\boldsymbol{AX}{=}\boldsymbol{XB}$ and $\boldsymbol{AX}{=}\boldsymbol{ZB}$, and makes use of them in accordance with the nature of the problem.

A performance comparison w.r.t. the most representative, traditional approaches has proven favorable to the presented method. Additionally, a novel weighting policy for the well-known approach in (Horaud and Dornaika, 1995) has been produced.

This section was adapted from the original publication in (Strobl and Hirzinger, 2006).

3.4 Caveat #1: The Accuracy of the Calibration Object

3.4.1 Introduction

As mentioned in Section 3.2.1, most models related with camera calibration are widely accepted in the computer vision community and it does not seem necessary to question their suitability anymore. However, it is apparent that there still exists one potential error source that has not yet been addressed, namely the allegedly known positions of the control points in the calibration object; on the one hand, the accurate knowledge of these corners supports calibration accuracy, but on the other hand it can very easily feed incorrect data into the estimation. In fact, it is often the case that *the pattern on the calibration plate is inaccurately imprinted.* Off-the-shelf printers especially fail in scaling the pattern, which independently (and regularly) occurs in its two main perpendicular directions—skew patterns rarely occur. It is standard practice to carry out subsequent 2-D measurements of the positions of the control points in order to cope with this problem, which is difficult to perform by hand with high accuracy.[16]

What is more, on a number of occasions the user will not even be measuring the pattern printout; even worse, they may crumple it up or just wrinkle it and fold it to warehouse and use it again in the future. It goes without saying that, if the pattern does not lie flattened on the table, the whole calibration object model is lifted.

In order to cope with the former error concerning badly scaled calibration patterns, I present a novel method that rescales the pattern back to ground truth with only two parameters. For this purpose, the *scaling factor κ* and the *aspect ratio ν* are introduced for the parameterization of the calibration pattern.

In order to cope with the latter error concerning patterns imperfections beyond homogeneous scaling, I present an additional novel approach that concurrently optimizes *the whole scene structure* in a compact, accurate way.

3.4.2 State of the Art in Calibration by Scene Structure Estimation

In the early years of computer vision, camera calibration was a cumbersome process. 3-D knowledge of the scene structure was a hard requirement (Faugeras and Toscani, 1987; Faugeras, 1993; Tsai, 1986) and high quality targets were difficult to achieve. In contrast to this, the possibility to *estimate* the scene structure also exists, as a by-product, along with regular camera calibration. In fact, this was the most important trend in camera calibration since the inclusion of distortion models. Tsai strikes this new path by two calibration methods, where the pose of the target (either 3-D or a planar, accurately shifted target) w.r.t. the camera is being estimated (Tsai, 1986, 1987). The most significant contribution, however, was simultaneously presented in the late nineties

[16] Conventional rulers are accurate to say 1 mm markers (Sun and Cooperstock, 2006). The reader is invited to check different rulers against each other.

by Zhang, Sturm and Maybank (Zhang, 2000; Sturm and Maybank, 1999). Their approach allows free motion of a *precisely known* planar calibration target. Their formulation obtains an approximate solution for both, the target pose and the camera parameters, from the readily obtained object-to-camera homographies (Appendix A), by means of rigid body motion constraints. The approach is flexible and accurate enough to become standard practice to this day. This is *not* because extensive 3-D knowledge of the scene directly compromises calibration accuracy—the contrary is true, but because its flexibility and simplicity prevent damage to calibration owing to human inaccuracies and mistakes (Sun and Cooperstock, 2006; Strobl and Hirzinger, 2008).

It is pertinent to address this trend towards camera-to-target pose estimation in the context of scene structure estimation, even though they do not yet include the geometry of the calibration target into the optimization. From the camera's point of view, the scene structure is equally determined by both, the target's geometry and its relative pose w.r.t. the camera. In other words, the 6 DoF of the target's pose combine with local target geometry to form the actual scene structure that eventually projects unto the camera. This trend therefore provides a clear indication to *additionally estimate the target's geometry.*

In fact, a few authors already made an attempt at this. Since manufacturing accurate 3-D calibration targets is more laborious than manufacturing planar ones, the approach was first taken in 3-D by Lavest *et al.* (Lavest *et al.*, 1998). Even though their results seem convincing, the method did not become popular, probably because it is formulated in 3-D and, from 1999 on, researchers largely opted for planar calibration targets.

Planar calibration objects do indeed provide convenient ground truth for camera calibration for different reasons: First, they are easy to manufacture, use, and store; second, they are naturally well adapted to the calibration of lens distortion since they can easily fill whole images; third and most importantly, high geometrical accuracy can be (cheaply) achieved. However, that is unfortunately not the case for the 2-D pattern imprinted on it. Regular printers dramatically lack of accuracy and it is therefore standard practice to gauge the pattern by hand, which is in turn prone to errors because of the use of inaccurate or inappropriate rulers, or even the indolent commitment of the user.

We were the first to deal with structure estimation in 2-D (Strobl and Hirzinger, 2008). We noticed that off-the-shelf printers systematically cause errors both, in global scale and in aspect ratio of the printed pattern. The pattern is then minimally modeled by these two parameters, which can be simultaneously estimated during intrinsic and hand-eye calibrations, respectively. I am presenting that method in the next section.

Albarelli *et al.* go one step further (Albarelli *et al.*, 2010); they observe anisotropic error distribution in reprojected object coordinates after calibration, which leads them to believe that significant, systematic pattern errors are actually pervasive—however small. In addition, the considerable reduction *in residual reprojection errors* reached after full camera and scene structure optimization further strengthens their position, that slightest pattern corrections really imply a significant accuracy improvement in camera calibration.

Even though their initial rationale is wrong (anisotropic error distributions are actually expected after nonlinear reprojection of isotropic image noise), their approach is undoubtedly convenient, at least when major inaccuracies in the calibration target occur.

Although the algorithms in (Lavest *et al.*, 1998) and (Albarelli *et al.*, 2010) are indeed very similar (incidentally, the latter fail to cite the former), their conveyed messages differ. While Lavest *et al.* claim that, by using their method, small inaccuracies will not get to harm camera calibration, Albarelli *et al.* on the other hand affirm that, by target geometry optimization, the user will even be able to come by accuracy levels that are otherwise virtually impossible to achieve at moderate cost. Whatever message they convey, the two main differences in their methods are: *First*, Albarelli *et al.* assume planar patterns and have the opportunity to make use of the following convenient algorithms (Zhang, 2000; Sturm and Maybank, 1999; Strobl and Hirzinger, 2008), whereas Lavest *et al.* require 3-D calibration targets and a laborious initialization step. *Second*, Lavest *et al.* seem to directly include *all* 3-D geometry of the target into the optimization, for several images, without further ado; Albarelli *et al.* on the contrary are forced to construct an iterative algorithm that decouples geometric estimation of the target from intrinsic parameters estimation. In (Strobl and Hirzinger, 2011) the authors rework this last detail, delivering a tight parametrization of the full scene structure that will be presented below.

3.4.3 Estimating Aspect Ratio and Absolute Scale of the Planar Calibration Object

In the above lines the dangers arising from the requirement of accurate knowledge of the scene structure are discussed. If the latter requirements are lifted, optimal estimation by sensible minimization of reprojection discrepancies must be re-engineered. In this section, the standard formulation for camera calibration in Section 3.2 will be adapted to this new paradigm—while still taking advantage of a priori knowledge of both, its planarity and the regularity of its pattern. I claim that highest accuracy camera calibration is still possible by this means. This would be a significant contribution in order to avoid commonplace mistakes and therefore increase calibration accuracy.

I propose a parameterization for the grid pattern of the planar calibration object by two parameters only: the scaling factor κ and the aspect ratio ν, see Fig. 3.9. This is a convenient parameterization not only because it very closely corresponds to the actual limitations of conventional printing equipment, but also because the effects of these parameters on the calibration process can be clearly differentiated: whereas an erroneous aspect ratio ν does affect the estimation of the intrinsic parameters, an erroneous scaling factor κ still allows optimal intrinsic calibration; it only affects (in range) the absolute extrinsics of each camera. Furthermore, they are a tight object model representation that makes it still possible for the calibration process to take advantage of accurate knowledge of the scene in the form of the high planarity and regularity of the imprinted pattern. The new method concerns the simultaneous estimation of these parameters during camera calibration.

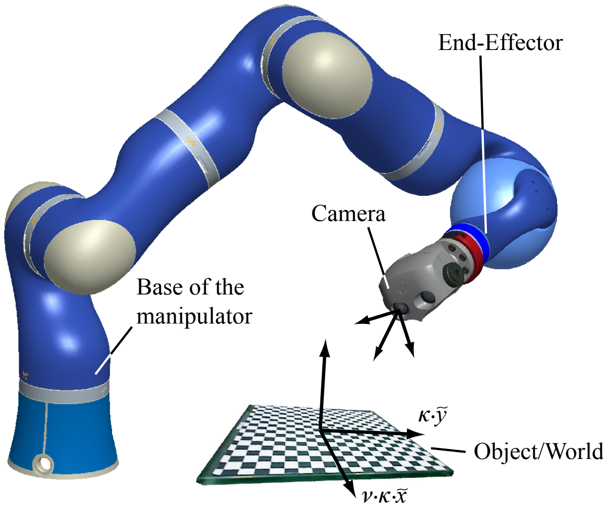


Figure 3.9: Stereo camera mounted at the top of the DLR Light-Weight Robot 3. Note the scaling of the calibration plate by the aspect ratio ν and the absolute scale κ.

I build upon the standard planar approach in Section 3.2 as well as the hand-eye calibration approach presented in Section 3.3. First, the intrinsic parameters are roughly estimated with linear least squares techniques. Second, the complete set of parameters of the camera model is refined by nonlinear optimization. Third, the extrinsic parameters are also roughly estimated. Last, the extrinsics are refined by nonlinear optimization.

The modified initial, intrinsic closed-form solution

Like most optimization processes that are formulated as residual minimization problems, camera calibration is vulnerable to local solutions. The current standard for its initialization stems from Refs. (Zhang, 2000; Sturm and Maybank, 1999) and has been explained in Section 3.2.3.

In the presence of badly scaled patterns, however, the actual Euclidean coordinates of the control points ${}_0\boldsymbol{p}_i$ are no longer known, but only the erroneously scaled ones ${}_0\breve{\boldsymbol{p}}_i$. Therefore, the solution of the system of Eqs. (3.2) may now lead strongly biased results. That is because the decomposition of the calculated homographies $\widehat{\boldsymbol{H}}$ (so that $\hat{\boldsymbol{m}} \propto \widehat{\boldsymbol{H}}\ [{}_0\breve{x}\ {}_0\breve{y}\ 1]^{\mathsf{T}}$) has changed. Now:

$$\widehat{\boldsymbol{H}} \propto \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t} \end{bmatrix} \begin{bmatrix} \kappa\nu & 0 & 0 \\ 0 & \kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{such that} \quad \left.\begin{aligned} \boldsymbol{r}_1 \cdot \boldsymbol{r}_2 &= 0 \\ \boldsymbol{r}_1 \cdot \boldsymbol{r}_1 &= 1 \\ \boldsymbol{r}_2 \cdot \boldsymbol{r}_2 &= 1 \end{aligned}\right\} \qquad (3.23)$$

$\forall\ \nu, \kappa \in \mathbb{R}\,/\,\nu, \kappa \neq 0$. It follows:

$$\left.\begin{aligned} (A^{-1}\boldsymbol{h}_1)^{\mathsf{T}} \cdot (A^{-1}\boldsymbol{h}_2) &= 0 \\ 1/\nu^2 \cdot (A^{-1}\boldsymbol{h}_1)^{\mathsf{T}} \cdot (A^{-1}\boldsymbol{h}_1) & \\ -(A^{-1}\boldsymbol{h}_2)^{\mathsf{T}} \cdot (A^{-1}\boldsymbol{h}_2) &= 0 \end{aligned}\right\} \Leftrightarrow \left.\begin{aligned} \boldsymbol{h}_1^{\mathsf{T}}\, \boldsymbol{\omega}_\infty\, \boldsymbol{h}_2 &= 0 \\ \boldsymbol{h}_1^{\mathsf{T}}\, \boldsymbol{\omega}_\infty\, \boldsymbol{h}_1 &= \nu^2 \cdot \boldsymbol{h}_2^{\mathsf{T}}\, \boldsymbol{\omega}_\infty\, \boldsymbol{h}_2 \end{aligned}\right\}, \qquad (3.24)$$

cf. Eq. (3.2). The introduction of one further unknown parameter (the aspect ratio ν) does away with the former linear formulation in part. Although ν could be otherwise calculated as it is unique for N images, we do not need to bother about it since intrinsic calibration is not about estimating the value of ν, but about estimating the value of the intrinsic parameters only. It is possible to orthogonalize both $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ (*i.e.*, $\boldsymbol{h}_1^\top \boldsymbol{\omega}_\infty \boldsymbol{h}_2 = 0$) without normalizing them (*i.e.*, without forcing $\boldsymbol{h}_1^\top \boldsymbol{\omega}_\infty \boldsymbol{h}_1 = \nu^2 \cdot \boldsymbol{h}_2^\top \boldsymbol{\omega}_\infty \boldsymbol{h}_2$) thereby resulting in N constraints for 5 intrinsic unknowns, and there still are more equations than unknowns after all. In addition, the stereo constraints introduced in (Malm and Heyden, 2001) still hold in this case—unmodified. This formulation is now closer to the actual intrinsic and absolute extrinsic values than the traditional one in the ubiquitous case of erroneous knowledge of the aspect ratio ν.

However, I recommend to decrease the number of unknowns in this first estimation. It is a pointless effort to aim at success in accurately estimating very sensitive parameters, such as the skew parameter γ or the principal point $[u_0, v_0]$, prior to the estimation of the lens distortion. It is advisable to include some prior knowledge of the parameters in the following form: $\gamma = 0$ and $[u_0, v_0]$ be located at the image center. The remaining parameters to be estimated are the scale factors α and β—along with the absolute extrinsics of the camera. It is more likely that the estimations resulting from this method fall in the convergence region required for successful nonlinear optimization, rather than the numerous former parameters afflicted with biases. Alternatively, the iterative method in (Sánchez *et al.*, 2006) can be also used with the omission of the normalization constraint. Nonetheless it is fair to say that the traditional approach does also mostly fall in the convergence region of the eventual nonlinear optimization stage.

After determining the intrinsic matrix $\boldsymbol{A}$, the absolute extrinsics for every image n can be computed from Eq. (3.23). If we introduce the parameter s as the proportionality parameter between both terms of the equation, it follows: $\boldsymbol{r}_1 = \nu/_{s\kappa}\, \boldsymbol{A}^{-1}\boldsymbol{h}_1$, $\boldsymbol{r}_2 = 1/_{s\kappa}\, \boldsymbol{A}^{-1}\boldsymbol{h}_2$, $\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2$, $\boldsymbol{t} = 1/_{s\kappa}\, \boldsymbol{A}^{-1}\boldsymbol{h}_3$, $s\kappa = ||\boldsymbol{A}^{-1}\boldsymbol{h}_1||$, and we can even estimate the aspect ratio $\nu = ||\boldsymbol{A}^{-1}\boldsymbol{h}_1|| / ||\boldsymbol{A}^{-1}\boldsymbol{h}_2||$. Again, these object-to-camera absolute extrinsics (as well as the camera-to-camera translations in the case of stereo) may be incorrectly scaled (in range) *if* the assumed scaling factor κ is far from reality; this also holds after nonlinear optimization of the intrinsic parameters. It is only the potential extrinsic calibration that will be able to correct them. However, the intrinsic parameters remain unaffected as they can be estimated irrespective of the scaling factor κ—recall Eqs. (3.24).

These initial values pave the way for a nonlinear optimization of the intrinsic parameters in the spirit of Eq. (3.3) and subsequent extrinsic calibration—the only difference being the additional unknown parameters aspect ratio ν and scaling factor κ. In this context, I next present two different methods that solve for the intrinsic and extrinsic unknowns in the form of a nonlinear optimization.

Method #1: estimating the aspect ratio $\boldsymbol{\nu}$ from the minimization of reprojection errors; estimating the scaling factor $\boldsymbol{\kappa}$ from the minimization of extrinsic transformation errors

Since an erroneous aspect ratio ν implies systematic errors between estimated $\hat{\boldsymbol{p}}$ and actually projected $\tilde{\boldsymbol{p}}$ control points in the image, the assumption on the Gaussian distribution of the error metric is violated which prevents (unbiased) optimal estimation of the intrinsic parameters. In addition to that, simulations show that released intrinsic parameters cannot completely compensate for these systematic errors if multiple images from different vantage points are taken. From these observations it follows that *first*, only the correct aspect ratio value ν truly minimizes reprojection errors after nonlinear optimization (see Fig. 3.13), and *second*, the aspect ratio ν can be estimated *at the same time* along with the other intrinsic and absolute extrinsic parameters. In this way, the erroneous data to minimize during the optimization process is, again, the projections ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\, i\}}$ of the control points in the images. The following minimization provides now the optimal intrinsic parameters:

$$\widehat{\boldsymbol{\Omega}}_{\star}^{\mathrm{s}} = \arg\min_{\widehat{\boldsymbol{\Omega}}^{\mathrm{s}}} \sum_{n=1}^{N} \sum_{i} \left\| {}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\, i\}} - {}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\, i\}} \left(\widehat{\boldsymbol{\Omega}}^{\mathrm{s}}(\hat{\nu}),\ \boldsymbol{\Upsilon}({}_{0}\check{\boldsymbol{p}}_{i}) \right) \right\|^{2} . \tag{3.25}$$

In contrast to Eq. (3.3), the optimization vector of calibration parameters $\boldsymbol{\Omega}^{\mathrm{s}}$ includes now the aspect ratio ν that, in turn, together with the erroneous object model in the vector of system models $\boldsymbol{\Upsilon}$, eventually generates ${}_{0}\hat{\boldsymbol{p}}_{i} = [\kappa{\cdot}\nu{\cdot}{}_{0}\check{x}_{i}\ \ \kappa{\cdot}{}_{0}\check{y}_{i}\ \ 0\ \ 1]^{\mathsf{T}}$.

If a subsequent hand-eye calibration is required, it has to be considered the above mentioned fact that the optimally ($\star$) estimated absolute extrinsics ${}^{\star}_{\mathrm{C}}\widehat{\boldsymbol{T}}_{n}^{0}$ may strongly differ from the actual ones. The transformation errors in the rotational and translational metrics within $SE(3)$ do not present the required unbiased Gaussian distributions anymore, and the optimal estimation process becomes strongly corrupted, cf. (Strobl and Hirzinger, 2006). Therefore, the hand-eye calibration algorithm has to be modified in order to estimate the scaling factor κ in which the intrinsic calibration was actually performed. In doing so, the absolute extrinsics (and, in the case of stereo cameras, also the camera-to-camera transformations) have to be scaled accordingly. Since the released extrinsic parameters cannot compensate for erroneous scales/ranges in all the absolute extrinsics at the same time, the simultaneous estimation of the hand-eye transformation and the scaling factor κ for multiple images tends to restore the error distribution to its reputed unbiased Gaussian nature, and, as a consequence, the extrinsic calibration along with it to optimal (unbiased) operation:

$$\left\{ {}^{\star}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}},\, {}^{\star}_{\mathrm{B}}\widehat{\boldsymbol{T}}^{0}, \hat{\kappa}_{\star} \right\} = \arg \min_{{}_{\mathrm{T}}\widehat{T}^{\mathrm{C}},\, {}_{\mathrm{B}}\widehat{T}^{0},\, \hat{\kappa}} \sum_{n=1}^{N} \mathcal{O}_{n} \left(\Phi({}^{\star}_{\mathrm{C}}\widehat{\boldsymbol{T}}_{n}^{0},\ \hat{\kappa}),\ {}_{\mathrm{B}}\widetilde{\boldsymbol{T}}^{\mathrm{T}},\ \ldots \right) \tag{3.26}$$

where the function Φ scales the estimated absolute extrinsics ${}_{\mathrm{C}}\widehat{\boldsymbol{T}}_{\star}^{0}$ in range according to the estimated scaling factor $\hat{\kappa}$. The reader is invited to compare Eq. (3.26) with the original Eq. (3.18) in Section 3.3.

Method #2: estimating both, aspect ratio ν and scaling factor κ, by minimizing extrinsic transformation errors

Alternatively, and only if a subsequent hand-eye calibration has to be performed, it is also possible to estimate the aspect ratio ν, and again the scaling factor κ, by minimizing the extrinsic transformation errors, see Section 3.3. As mentioned above, any intrinsic calibration with incorrect aspect ratio ν will yield erroneous parameters, thus erroneous absolute extrinsics—not only in their range w.r.t. the calibration object. This will necessarily compromise hand-eye calibration even if it also estimates κ; therefore, once again, only the correct value for the aspect ratio ν will make it to truly minimize the extrinsic residuals:

$$\left\{{}^{\star}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}_{n},\ {}^{\star}_{\mathrm{B}}\widehat{\boldsymbol{T}}^{0},\ \hat{\kappa}_{\star},\ \hat{\nu}_{\star}\right\} = \arg \min_{{}_{\mathrm{B}}\widehat{T}^{\mathrm{C}}_{;}\,{}_{\mathrm{B}}\widehat{T}^{0},\,\hat{\kappa},\,\hat{\nu}} \sum_{n=1}^{N} \mathcal{O}_n \left(\Phi({}^{\star}_{\mathrm{C}}\widehat{\boldsymbol{T}}^{0}_{n},\ \hat{\kappa}),\ {}_{\mathrm{B}}\widetilde{\boldsymbol{T}}^{\mathrm{T}}_{n},\ \ldots \right) \tag{3.27}$$

with

$${}^{\star}_{\mathrm{C}}\widehat{\boldsymbol{T}}^{0} \in \arg\min_{\widehat{\boldsymbol{\Omega}}} \sum_{n=1}^{N} \sum_{i} \left\| {}_{\mathrm{M}}\tilde{\boldsymbol{p}}^{\{n,\,i\}}_{\mathrm{d}} - {}_{\mathrm{M}}\hat{\boldsymbol{p}}^{\{n,\,i\}}_{\mathrm{d}} \left(\widehat{\boldsymbol{\Omega}},\ \boldsymbol{\Upsilon}^{\mathrm{s}}(\hat{\nu},\, {}_{0}\check{\boldsymbol{p}}_{i}) \right) \right\|^{2} \quad , \tag{3.28}$$

where ν is not included in the optimization vector of the intrinsic estimation $\boldsymbol{\Omega}$ anymore, but in the new vector of system models $\boldsymbol{\Upsilon}^{\mathrm{s}}$. This method is computationally more expensive since a complete optimization for ${}^{\star}_{\mathrm{C}}\widehat{\boldsymbol{T}}^{0}_{n}$ is taking place for every single extrinsic iteration. Its main motivation are systems where the positioning accuracy is very high, the errors in the chosen metrics in $SE(3)$ are close to Gaussian, and the imaging errors are neither small nor Gaussian (e.g. with very low resolution cameras or oddly distorted images). In this case, a feasible solution can be obtained as follows: A first solution by *Method #1* serves as a good initialization for *Method #2*. In this way, *Method #2* performs a very restricted local search on ν over both, the traditional intrinsic optimization in Eq. (3.3) and the subsequent extrinsic optimization in Eq. (3.26).

Simulations and Experiments

Simulations

Simulations were conducted in order to illustrate the fundamental weaknesses of the traditional calibration methods and to put the novel methods presented in the last section to the proof. Ground truth data was adopted from the intrinsic and extrinsic results of the real, monocular camera calibration in the below experiments section (left-hand side data in Fig. 3.10) as well as assumed pattern dimensions. Note that I am using *monocular* data since the perspective projection equations are in this case worse conditioned than in the case of stereo. In short, the ideal image projections and robot motions were calculated, and noisy image and positioning data was generated on them. Next, the effects of errors in the assumed pattern dimensions and noise levels are studied.

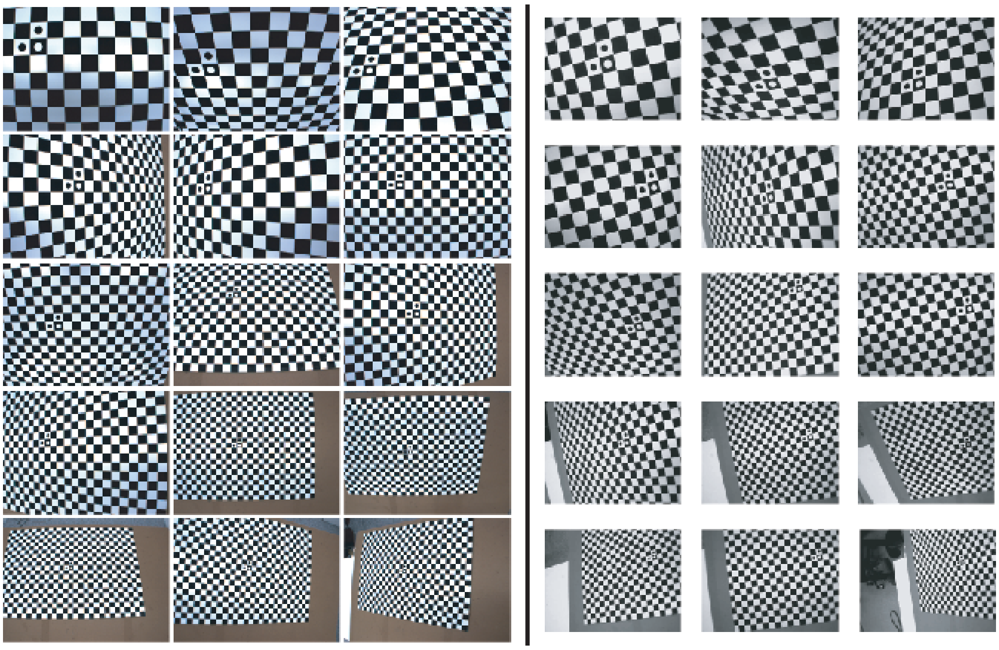


Figure 3.10: Fifteen images used for calibration with the AVT Marlin camera (left, 780× 580 pixels) and the Typhoon$^{\mathrm{TM}}$ EasyCam camera (right, 640×480 pixels). The extrinsic poses were taken by a Kuka KR 16. The pattern is size A2.

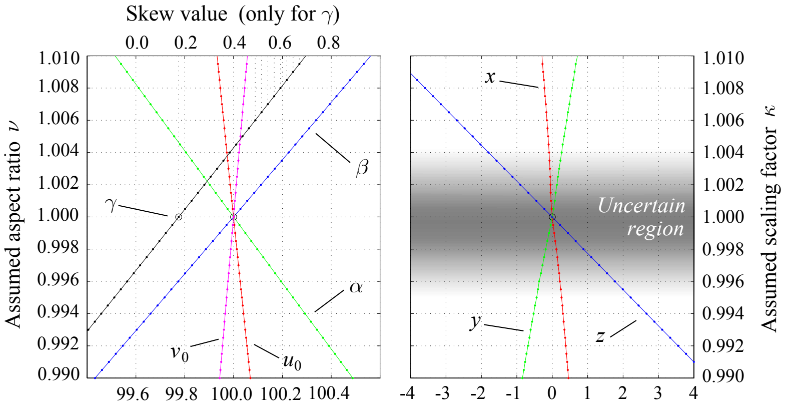
Simulation #1: Effects of erroneous pattern dimensions

In this section the errors in the estimation of the parameters of the camera model after traditional calibrations with inaccurate knowledge of both pattern parameters ν and κ of 1% (*i.e.*, from 0.99 to 1.01) is shown—ground truth reads $\nu = \kappa = 1$. Afterwards, the performance of the calibrated camera is assessed.

Noisy visual data was generated over the ideal image projections with $\sigma_{\mathrm{x}} = \sigma_{\mathrm{y}} = 0.15$ pixels. For the extrinsic calibration, random noisy transformations ${}_{0}\widetilde{\boldsymbol{T}}^{\mathrm{T}}$ were generated from N ideal absolute extrinsics along with the ground truth hand-eye transformation. The noise was added to the ideal pose of the end-effector of the robot as follows: The angles θ of the angle-axis representation $\{\theta, \boldsymbol{p}\}$ of the added noise follow a zero-mean Gaussian distribution with $\sigma_\theta = 0.05°$ and their axes $\boldsymbol{p}$ are uniformly distributed, *i.e.*, their azimut and elevation angles ϕ and ψ are $\phi \in [-90°, 90°)$ according to the **p**robability **d**ensity **f**unction $\text{pdf}(\phi) = 180^{-1}\,[°]^{-1}$ and $\psi \in [-90°, 90°)$ with $\text{pdf}(\psi) \propto \arcsin(\psi/90)\,[°]^{-1}$. The translation errors $\boldsymbol{t}$ also follow a zero-mean Gaussian distribution in range with $\sigma_t = 0.25$ mm and their directions are, again, uniformly distributed.

Simulation #1: 1) Erroneous estimation of parameters

Figs. 3.11 and 3.12 show erroneously estimated camera parameters in relation to the assumed values for the parameters ν and κ. Fig. 3.11 (a) shows significant drifts in the intrinsic parameters of the camera model in relation to the assumed aspect ratio (irrespective of κ). Fig. 3.12 does the same for the resulting absolute extrinsics. The results with $\nu \neq 1$ will of course also imply erroneous extrinsic calibration. Even if $\nu = 1$ there still exists the possibility that an erroneous scaling factor $\kappa \neq 1$ yielded badly scaled absolute extrinsics (in range), even though the intrinsic parameters were optimally estimated. Fig. 3.11 (b) shows the error in the estimation of the hand-eye transformation in this last case.



(a) Estimated vs actual intrinsics [percent] (b) Errors in the hand-eye transformation [*mm*]

Figure 3.11: Percent of error in the intrinsic parameters (a) and translation error in the hand-eye transformation (b) in relation to the pattern scaling parameters assumed for traditional calibration. The actual parameters are $\nu = 1$ and $\kappa = 1$.

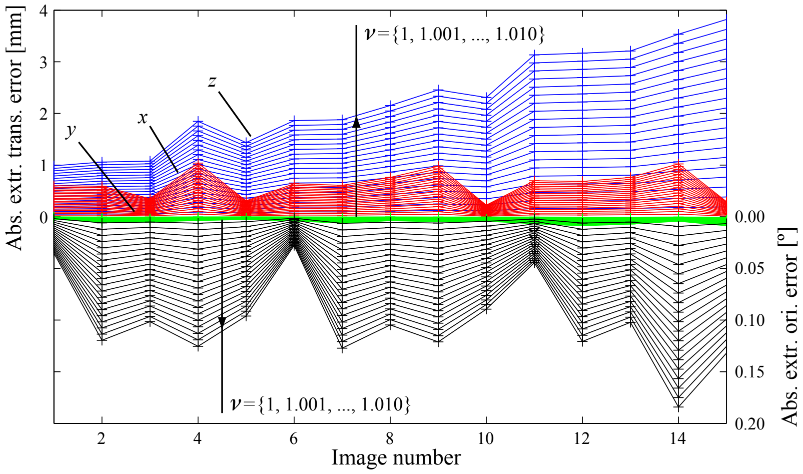


Figure 3.12: Error in translation and orientation of the absolute extrinsics in relation to the aspect ratio assumed for traditional calibration. In reality $\nu = 1$.

Simulation #1: 2) Performance after erroneous calibration

Of course, the abovementioned erroneous camera calibration sharply deteriorates performance. In Fig. 3.13 the **r**oot **m**ean **s**quare error (RMS) of the intrinsic reprojection errors between expected and actually imaged projections of the *real* calibration object for all images are shown. The expected projections are obtained using the intrinsic parameters from traditional calibrations with $\tilde{\nu} \in [0.99, 1.01]$, with ground truth absolute extrinsics. For optimal calibration ($\nu = 1$) the projection error is expectedly minimal and identical to the "*virtual*" residual after calibration (0.21 pixels RMS). On the contrary, for traditional calibrations with $\nu \neq 1$ the error scales up (approximately linearly) to e.g. 0.4 pixels RMS for $\nu = 0.9975$ (0.25 % aspect ratio error, *i.e.*, only 0.75 mm discrepancy between the x and y lengths when measuring a 30×30 cm section of the pattern as a whole). In addition, the "*virtual*" residuals after calibration with $\tilde{\nu} \in [0.99, 1.01]$ are depicted; these reflect the operation of Eq. (3.25) relative to the assumed aspect ratio ν, where erroneous absolute extrinsics try their hardest to compensate for erroneous intrinsic parameters.

If extrinsic calibration follows the intrinsic one, Section 3.4.3 showed that the extrinsic estimation may also become inaccurate, and naturally the eventual performance will get worse as well. The set of curves on the right-hand side in Fig. 3.13 show the projection errors where the actual noisy readings of the manipulator ${}_0\widetilde{\boldsymbol{T}}^{\mathrm{T}}_n$, along with the traditionally estimated hand-eye transformations ${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$ with $\tilde{\nu} \in [1.00, 1.01]$ and $\tilde{\kappa} \in [0.99, 1.01]$, take the place of the former absolute extrinsics. For the ground truth parameters ($\nu = \kappa = 1$) the error scales up to 0.65 pixels RMS. The (small) noise in the manipulator readings accounts for this increment. Slightly erroneous pattern parameters skyrocket this error.

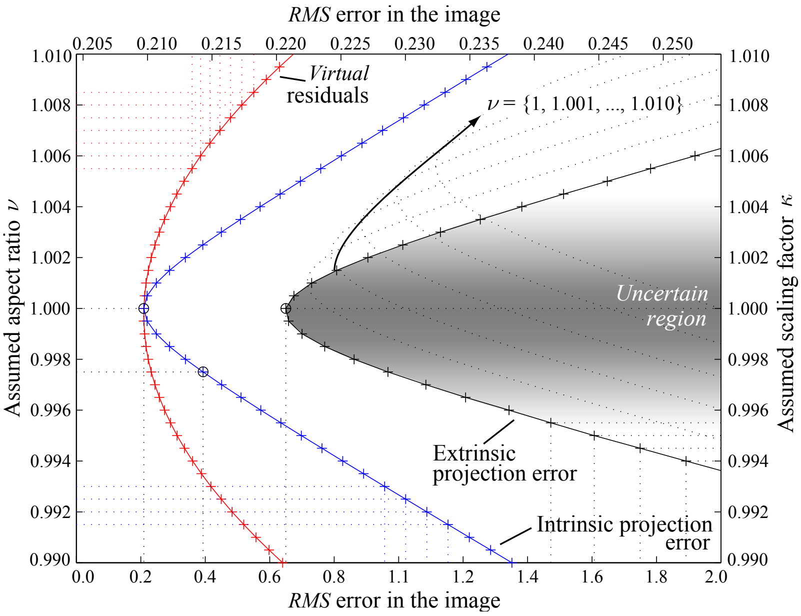


Figure 3.13: Image projection errors in relation to the scaling parameters for traditional calibration. The actual parameters: $\nu = 1$, $\kappa = 1$. Note the different axes.

It is worth noting that, in this last case and in the case of traditional hand-eye calibration, it clearly exists a *fundamental uncertainty region* where it is not possible for the user to assess calibration accuracy, since it is subject to the absolute accuracy of the ruler at hand. For incorrect aspect ratio $\tilde{\nu}$ this is not clearly defined, since relative dimensions can be determined with high precision even when using inaccurate rulers.

Simulation #2: Convergence of the novel estimation methods under noise

Simulations were conducted with variable noise levels in the control points projections detected in the images, as well as in positioning accuracy of the robotic manipulator. Fig. 3.14 shows "*virtual*" residuals after traditional intrinsic calibrations with different image noise levels $\sigma_{\{u,v\}} \in [0.1,\ 1.0]$ pixels and assumed aspect ratios $\tilde{\nu} \in [0.99,\ 1.01]$, as percentage w.r.t. the optimal results when $\nu = 1$. The residuals reflect the operation of Eq. (3.25). The minimum residual is unequivocal for the optimal solution $\nu = 1$ and shows that, in this context, *the erroneous intrinsic and absolute extrinsic parameters cannot completely compensate for erroneous knowledge of the aspect ratio of the calibration pattern* (refer to Section 3.4.3). This result is basis for the intrinsic optimization in Eq. (3.25) of *Method #1*, since it clearly shows the existence of an unique, unbiased minimum for the optimization. Similar results are achieved for the extrinsic calibration with codetermination of the scaling factor in Eq. (3.26), as well as for the optimizations of *Method #2*. In general, the methods do not only converge for the initial parameters shown in these simulations, but for significantly worse ones; aspect ratio and scaling factor errors of up to only $\pm 1\,\%$ were used in this section in order to visualize the absence of biases in the final estimations.

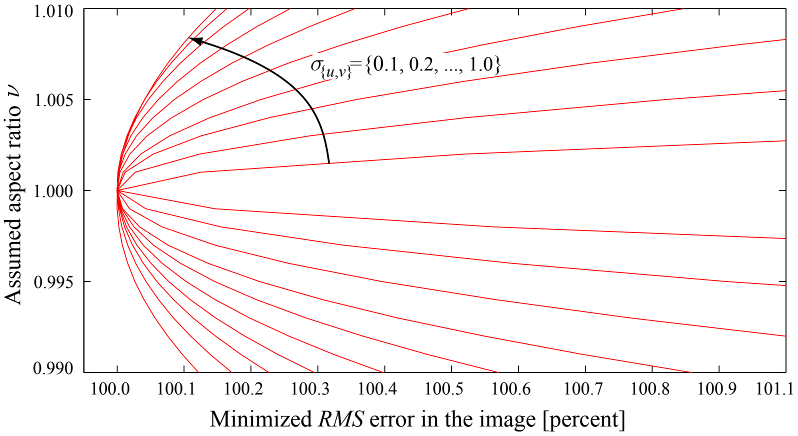


Figure 3.14: Minimized *virtual* image projection errors in relation to the aspect ratio and different image noises $\sigma_{\{u,v\}} \in \{0.1,\ 0.2,\ ...,\ 1.0\}$ pixels.

Experiments

In this section the performance of the algorithms in real systems is studied in order to validate both, models and algorithms. Since the validity of the traditional camera calibration methods is out of the question, here only the novelty concerning the codetermination of the pattern scaling parameters ν and κ is evaluated. Fortunately, and in contrast to the case of intrinsic parameter estimation, it is possible to directly assess the accuracy in the determination of these pattern parameters, since they can also be directly measured. If the determination is accurate, the estimation of the further parameters is necessarily the equivalent of the well-established traditional calibration methods, which consequently validates the novel methods in this work for the systems in test.

With the idea of validating in a wide range of systems, two different (monocular) cameras were used: On the one hand, an accurate progressive scan AVT Marlin camera with a SVGA 1/2" Sony CCD chip and a Sony VCL-06S12XM 6 mm objective worth $1,500 altogether; on the other an off-the-shelf VGA 1/4" CMOS 4-6 mm Typhoon$^{\mathrm{TM}}$ EasyCam webcam worth $15. Both cameras are rigidly attached to the end-effector of a precise robot manipulator Kuka KR 16 and take pictures of a precisely imprinted A2-sized checkerboard calibration plate, see Fig. 3.10.

Initially, ground truth data was obtained by visually measuring an extended patch of the checkerboard pattern with a metallic precision ruler—it was assumed that the checkerboard pattern regularly spreads in x and y directions. Specifically, the lengths d_{x} and d_{y} of the segments defined by 28 and 19 squares the size of $u_{\mathrm{x}} \times u_{\mathrm{y}}$ ($u_{\mathrm{x}} \approx u_{\mathrm{y}} \approx 2\,\mathrm{cm}$) were measured, and the optimal parameters $\nu_\star = \kappa_\star = 1$ were assigned to them. After that, *Methods #1* and *#2* were used to estimate the correcting parameters $\hat{\nu}$ and $\hat{\kappa}$ against potentially erroneous pattern data, which in turn lead to the estimated lengths $\hat{d}_{\mathrm{x}}$ and $\hat{d}_{\mathrm{y}}$. The results in Table 3.2 show a formidable consistency of the estimated and measured dimensions, even though the algorithms were initialized with dramatically wrong dimensions like $u_{\mathrm{x}} = 3\,\mathrm{cm}$ and $u_{\mathrm{y}} = 1\,\mathrm{cm}$.

Table 3.2: Calibration results using *Methods #1* and *#2* w.r.t. traditional calibration.

		κ	ν	d_{x}	d_{y}	$\mathrm{RMS}_{\mathrm{int}}$
Precision ruler		1.00000	1.00000	559.6	379.0	—
AVT Marlin	*Method #1*	0.99906	1.00096	559.59	378.63	0.1735
	Method #2	0.99936	0.99910	559.34	379.55	0.1737
Typhoon$^{\mathrm{TM}}$ EasyCam	*Method #1*	0.99967	1.00056	559.47	378.78	0.6452
	Method #2	1.00081	0.99780	558.82	379.31	0.6453
				[mm]	[mm]	[pixels]

These values are more accurate than the ones any user is able to obtain with the sole aid of a regular ruler over a smaller patch of the pattern. To be precise, the upper and lower lengths d_{x} of the measured patch on the actual pattern differ as much as 0.3 mm; here the mean value of 559.6 mm has been used. In addition to that, all measurements on the actual pattern were the result of an interpolation between results using two different metallic precision rulers, which differ in length about 0.3 mm over their whole length of 1 m. Therfore, the above results come by the highest measurable accuracy that the authors were able to achieve.

Concerning the computational cost of the approaches, *Method #1* hardly affects the costs, especially if the parameters are reasonably close to the true values. *Method #1* increases the lengths of the optimization vectors in only one parameter each, being them usually the size of $5 + 2 + (6 \cdot N)$ parameters intrinsically, and $6 \cdot 2$ extrinsically. In contrast, *Method #2* does significantly increase costs, since it implies an iterative process of numerical optimizations. However, it is clear that computational cost is a quite immaterial issue for traditional calibration.

In this work the Levenberg-Marquardt optimization method was used both, for the intrinsic and the extrinsic calibrations.

Discussion

In general systems, where the Gaussian image noise assumption largely holds, optimal intrinsic camera calibration is only attained if the aspect ratio of the pattern is perfectly known. Since this never holds outside of simulation scenarios, the user should opt for one of the two methods mentioned above if he or she is not able to determine the aspect ratio of the imprinted pattern with an accuracy of say one part in a thousand (*i.e.*, 0.3 mm in a 30×30 cm patch). As to which method to use, the fact that the projection residuals in the images are mostly numerous and small, and conversely the camera vantage points fewer (typically 10 to 15) and their positioning errors of arbitrary size (depending on the system), suggests that the former errors distributions show much more a Gaussian nature than the latter ones. Therefore, *Method #1* should usually perform more accurately than *Method #2* for the codetermination of the aspect ratio ν. As regards the extrinsic calibration, the user should also opt for one of these methods if he or she is not able to determine the absolute size of the plate with an accuracy of say one part in a thousand (*i.e.*, 0.3 mm accuracy in 30 cm), which actually is *very* often the case.

3.4.4 Estimating the Full Structure of the Scene

In this section I present a novel approach that performs optimal, maximum likelihood camera calibration as in the last section, but now in the presence of a more drastic deformation of the pattern on the calibration object. As mentioned above, it is often the case that the user will not measure the pattern printout, or he or she will even be crumpling it up or wrinkling it, folding it to warehouse and use it again in the future. It goes without saying that, if the pattern does not lie flattened on the table, the whole calibration object model is lifted.

In order to cope with these errors, an approach that concurrently optimizes the whole scene structure in a compact, accurate way is constructed. In detail, the formulation extends the camera extrinsic parameters into a tight parametrization of the whole scene structure. When considering together the target's geometry and its relative pose w.r.t. the camera, they form together the scene structure. A parametrization of the scene structure using both, a rigid body transformation with 6 DoF and $3 \cdot M$ Euclidean coordinates for all M feature points in 3-D is clearly overparameterized, cannot be estimated unambiguously, and will not converge during nonlinear optimization. A tight parametrization is achieved e.g. by merely releasing $3 \cdot M$ Euclidean coordinates. However, it is sensible to take advantage of the relative transformation ${}_{\mathrm{C}}\boldsymbol{T}_n^0$ between the reference frame of the calibration object S_0 and the camera frame frame S_{C}, because the local geometry model will then still hold, unmodified, from a different vantage point, which is convenient for multi-view optimization. However, in this case the local geometry model of the calibration target ought to be restricted to $3 \cdot M - 6$ parameters. The authors in (Lavest *et al.*, 1998) do not mention this issue, which may have been another reason for the limited popularity of their approach. In (Albarelli *et al.*, 2010) the authors encounter this problem; they deal with it by strictly decoupling target geometry and camera parameters estimation in an iterative way. While the latter approach should work, it is not necessary to detach scene structure estimation from intrinsic optimization if a tight parametrization is used. The perspective distortion captured by images ought to be sufficient to distinguish between camera magnification (*i.e.*, focal length) and the structure of the scene (*i.e.*, the geometry of the calibration target and N poses ${}_{\mathrm{C}}\boldsymbol{T}^0$, up to scale) during optimization by multi-view calibration. Furthermore, their rescaling step back to original absolute scale is superfluous, as correct monocular intrinsic calibration is possible irrespective of absolute scale (Strobl and Hirzinger, 2008).

In this section I present a calibration method that completely releases target geometry and performs jointly with intrinsic parameters estimation. The approach is similar to the standard calibration methods in (Zhang, 2000; Sturm and Maybank, 1999; Strobl and Hirzinger, 2008). Expected, model-based operation is compared with actual projections; after that, the resulting discrepancies are minimized by tuning parameters in the projection model. In this work the main modification w.r.t. standard methods will be the *tight* release of the target's geometry during the final nonlinear optimization. Critically, requirements on the calibration target are now drastically lifted so that unmeasured patterns (e.g., a checkerboard printed on paper using off-the-shelf printers) can be used, even on an uneven surface. The only requirement now is that the pattern remains static during calibration—unless it is rigid material. If stereo camera calibration is intended, a sole scale parameter (e.g., absolute distance between two arbitrary corner features) is required. A potential hand-eye calibration in turn waives this last requirement.

Initial Solution

It is of paramount importance for accurate camera calibration to precisely and robustly detect calibration target features on the images. In fact, Lavest *et al.* argue that, by following this paradigm of concurrent target geometry estimation, the calibration results will no longer depend on the (lack of) accuracy of the pattern, but mainly on the accuracy of feature detection (Lavest *et al.*, 1998). Planar checkerboard patterns are certainly convenient in terms of (subpixel) localization accuracy of their corners (Mallon and Whelan, 2007; Strobl *et al.*, 2005), thus my method is conceived for (not restricted to) this type of data.

Like most optimization processes that are formulated as residual minimization problems, camera calibration is vulnerable to local solutions. The current standard for its initialization stems from (Zhang, 2000; Sturm and Maybank, 1999) and has been explained in Section 3.2.3. In Section 3.4.3 I modified the traditional formulation in case of imprinted patterns with unknown aspect ratio and absolute scale. The solution produced hereby is irrespective both, of the absolute scale and of the aspect ratio of the planar pattern, and it suffices to bootstrap nonlinear optimization.

Since optical distortion has not yet been compensated for during the initialization method in Section 3.4.3, the user may insert a nonlinear optimization in order to support eventual convergence. At this point, the user may choose between the traditional approach in Section 3.2.3 and the novel approach in Section 3.4.3, where the pattern aspect ratio is being estimated.

However, if the expected (prior to printing) pattern dimensions are provided and off-the-shelf printers are used, experiments show that this whole step can be readily skipped.

Simultaneous intrinsic camera calibration and full scene structure estimation

As stated above, it is often sensible to fully release scene structure, extending optimization parameters to the target's geometry. Three recent approaches were reviewed that are either erroneous, incomplete, or needlessly complex. Here I bring forward a novel target parametrization that is perfect complement to the N relative transformations ${}_{\mathrm{C}}\boldsymbol{T}_n^0$, to jointly model full scene structure.

Target geometry is a parameter to reprojection in Eq. (3.3), but it is not part of the optimization parameters $\boldsymbol{\Omega}$. The blunt inclusion of M 3-D target points is suggested in (Lavest *et al.*, 1998). Unfortunately, this leads to overparametrization when coupled with the N unknown transformations ${}_{\mathrm{C}}\widehat{\boldsymbol{T}}_n^0$

($3 \cdot M + 6$ DoF at every station n) and estimations change uncontrollably during optimization, which precludes absolute convergence. To obtain a tight representation, 6 DoF have to be subtracted from the geometric model of the target (now $3 \cdot M - 6$ DoF) to overall $3 \cdot M$ DoF at every station n—and scene structure is uniquely defined. However, since intrinsic camera calibration is possible irrespective of the absolute scale of the scene, a further DoF has to be subtracted. In Fig. 3.15 the 7 DoF that are excluded from optimization are depicted; they involve three corner features—their choice is arbitrary as long as they are non-collinear. Feature $\boldsymbol{p}_1 = [0\,0\,0]^\mathsf{T}$ is fixed to be pattern origin since else it couples with the translational part of ${}_{\mathrm{C}}\boldsymbol{T}^0_n$. Two other fixed points are $\boldsymbol{p}_2 = [d\,0\,0]^\mathsf{T}$ and $\boldsymbol{p}_3 = [x_3\, y_3\, 0]^\mathsf{T}$. $y_2 \triangleq 0$, $z_2 \triangleq 0$, and $z_3 \triangleq 0$ fix the target orientation so that it will not get coupled with the orientation in ${}_{\mathrm{C}}\widehat{\boldsymbol{T}}^0_n$ during estimation. $x_2 \triangleq d$ fixes the absolute pattern scale to an *arbitrary* value. In spite of these constraints, the target geometry is still released up to its absolute scale—which cannot be estimated during intrinsic calibration after all.

The new optimization parameters $\boldsymbol{\Omega}^+$ include x_3, y_3, and $\boldsymbol{p}_i\ \forall i \in \{4, \dots, M\}$, *i.e.*, $3 \cdot (M - 3) + 2$ variables:

$$\widehat{\boldsymbol{\Omega}}^+_\star = \arg\min_{\widehat{\boldsymbol{\Omega}}^+} \sum_{n=1}^{N} \sum_{i=1}^{M} \left\| {}_{\mathrm{M}}\tilde{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}} - {}_{\mathrm{M}}\hat{\boldsymbol{p}}_{\mathrm{d}}^{\{n,\,i\}} \left(\widehat{\boldsymbol{\Omega}}^+,\, d \right) \right\|^2 . \qquad (3.29)$$

In doing so, we cast the former Eq. (3.3) into a much harder optimization task as the parameters vector length skyrockets from e.g. $5 + 2 + 6 \cdot N$ to $5 + 2 + 6 \cdot N + 3 \cdot (M - 3) + 2$, where $M \gg N$. Being the residuals vector already long (up to $2 \cdot M \cdot N$), the required Jacobian matrix increases exponentially in size. Even though computing efficiency is uncritical in camera calibration, I recommend providing Jacobian sparsity *patterns* to this optimization. The use of *analytical* Jacobians is here, however, discouraged as residuals are in distorted image space, Jacobians are hard to get, and it is too costly to perform variable substitution on them in the first place.

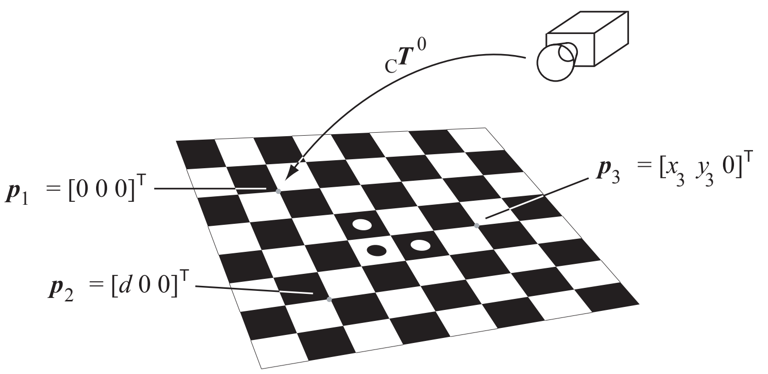

${}_{\mathrm{C}}\boldsymbol{T}^0$
$\boldsymbol{p}_1 = [0\ 0\ 0]^\mathsf{T}$
$\boldsymbol{p}_3 = [x_3\ y_3\ 0]^\mathsf{T}$
$\boldsymbol{p}_2 = [d\ 0\ 0]^\mathsf{T}$

Figure 3.15: Pattern features $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ that will be (in part) fixed during joint intrinsic and full scene structure optimization.

Extension #1: Stereo camera calibration

A natural extension of this work is in the case of stereo cameras:

$$\widehat{\boldsymbol{\Omega}}^{\oplus}_{\star} = \arg\min_{\widehat{\boldsymbol{\Omega}}^{\oplus}} \sum_{c=1}^{C}\sum_{n=1}^{N}\sum_{i=1}^{M} \left\| {}^{c}_{\mathrm{M}}\tilde{\boldsymbol{p}}^{\{n,\,i\}}_{\mathrm{d}} - {}^{c}_{\mathrm{M}}\hat{\boldsymbol{p}}^{\{n,\,i\}}_{\mathrm{d}}\left(\widehat{\boldsymbol{\Omega}}^{\oplus},\, d\right)\right\|^2 . \tag{3.30}$$

Compared to $\boldsymbol{\Omega}^{+}$, the optimization parameters $\boldsymbol{\Omega}^{\oplus}$ additionally include the intrinsic parameters of further cameras ($\boldsymbol{A}_c$, $\boldsymbol{k}_c$) and their rigid, relative transformations ${}_{\mathrm{C}_c}\boldsymbol{T}^{\mathrm{C}}$ w.r.t. the main reference camera in S_{C}, cf. Eq. (3.4). If two cameras are used, the length of the residuals vector amounts to up to $2{\cdot}2{\cdot}M{\cdot}N$, and the length of the parameters vector to, e.g., $(5{+}2){\cdot}2{+}6{+}6{\cdot}N{+}3{\cdot}(M{-}3){+}2$. It is worth noting that, if the stereo camera is to be calibrated to correct metric scale, the value of d is *not* arbitrary anymore; the user has to provide a valid distance d between two (arbitrary) features on the pattern of the calibration object.

Extension #2: Hand-eye calibration

Hand-eye calibration is the process of estimating the rigid body transformation ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ relating the end-effector frame of e.g. a robot manipulator (hand, S_{T}) to the reference camera frame (eye, S_{C}) mounted on it, refer to Section 3.3. Similar to stereo calibration, the standard hand-eye calibration *requires* correct metric scale. Since more often than not hand-eye calibration is decoupled from intrinsic camera calibration, the hand-eye calibration method presented in Section 3.4.3, *Method #1*, still holds. In a nutshell: The discrepancies ($\mathcal{O}_n$) between *expected* and *measured* transformations are minimized, see Eq. (3.26) together with Eq. (3.18) in Section 3.3. Expected eye locations ${}^{\star}_{\mathrm{C}}\widehat{\boldsymbol{T}}^{0}_{n}$ stem from intrinsic calibration (they are called absolute extrinsics and are a by-product of intrinsic camera calibration); measured transformations ${}_{\mathrm{B}}\widetilde{\boldsymbol{T}}^{\mathrm{T}}_{n}$ stem from the noisy motion readings of the manipulator. Note that here the absolute scale d can be simultaneously estimated *during* optimization. Following the notation in Sections 3.3 and 3.4.3:

$$\left\{{}^{\star}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}},\, {}^{\star}_{\mathrm{B}}\widehat{\boldsymbol{T}}^{0},\, \hat{d}_{\star}\right\} = \arg\min_{{}_{\mathrm{T}}\widehat{T}^{\mathrm{C}},\, {}_{\mathrm{B}}\widehat{T}^{0},\, \hat{d}} \sum_{n=1}^{N}\mathcal{O}_n\left(\Phi({}^{\star}_{\mathrm{C}}\widehat{\boldsymbol{T}}^{0}_{n},\, \hat{d}),\, {}_{\mathrm{B}}\widetilde{\boldsymbol{T}}^{\mathrm{T}},\, \ldots\right) \tag{3.31}$$

where the function Φ scales the transformations ${}^{\star}_{\mathrm{C}}\widehat{\boldsymbol{T}}^{0}_{n}$ in range—according to the scaling factor $\hat{d}$ being estimated. If this method is used, the user does not even need to provide a valid distance d for stereo calibration; in the case of stereo cameras, he or she only needs to rescale potential transformations ${}^{\star}_{\mathrm{C}_c}\widehat{\boldsymbol{T}}^{\mathrm{C}}$ back to correct metric scale using $\hat{d}_{\star}$.

Experiments

In this section the results of the above method are analyzed, both on calibration data and in independent validation experiments. After that I shall discuss on the utility of the presented approach.

A stereo camera was used consisting of two progressive scan AVT Marlin cameras with SVGA 1/2” Sony CCD chips and Sony VCL-06S12XM 6 mm objectives; experiments show that a radial distortion model using only two parameters (3rd and 5th degree) suffice to model the optics’ geometric distortion. Stereo base distance is approximately 5 cm.

Two calibration targets are used: On the one hand a precision pattern size A3 printed on a metallic plate; on the other hand a printed A3 sheet of paper with the same checkerboard pattern of $14 \times 20 = 280$ corner features. The distance between features is approx. 2 cm. The paper pattern was folded previous to calibration to affect its planarity, thus represents a worst-case scenario, see Fig. 3.16. In both cases, the calibration consists of 12 *tilted* images at three different heights w.r.t. the calibration pattern (20, 40 and 80 cm). Of course, not all corner points are seen in every image.

At this point it is worth mentioning the reason for taking additional images at different heights, since usually 4 or 8 images suffice for camera calibration: It is critical to optical distortion estimation to fill in images with features, so that distortion can be correctly estimated in the image corners (Strobl *et al.*, 2005). Naturally, some features in the image corners might be imaged only once. Using my novel method, those lone features are now totally released in 3-D to match their actual image projections, thus will not enforce correct distortion model parametrization. *To avoid lone features, I additionally take distant images.*

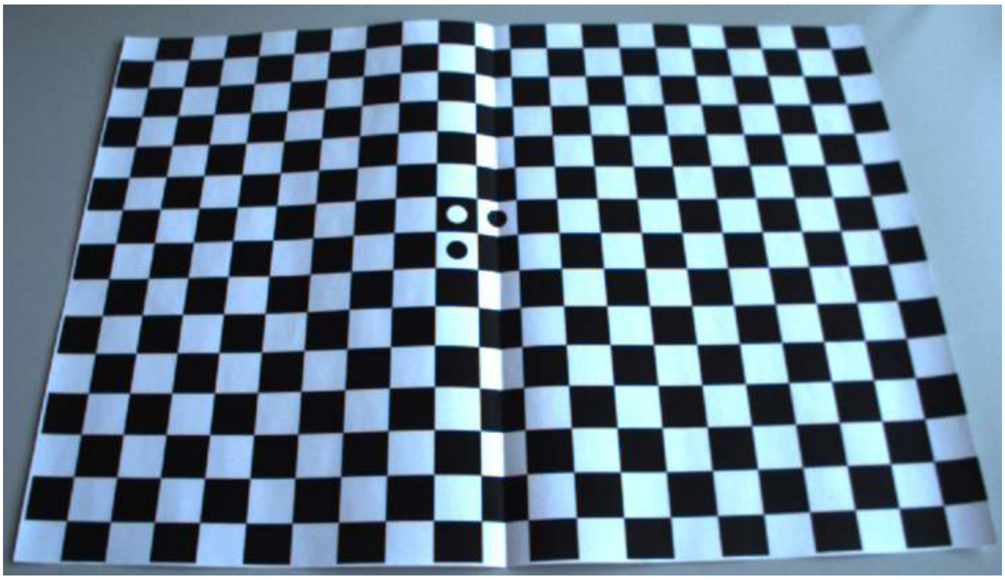


Figure 3.16: Wrinkled paper calibration target size A3.

Calibration starts out by accurately detecting and locating corners in the images using DLR CalDe (Strobl *et al.*, 2005), with sub-pixel accuracy. Since calibration will also estimate the target's geometry, it is not necessary to provide accurate pattern dimensions—experiments in (Strobl and Hirzinger, 2008) showed strong convergence in a similar scenario. However, the metallic plate was initially meant to deliver ground-truth geometry, or rather to show the potential precision in target geometry estimation, thus I do adopt accurate pattern dimensions in that case (actual square size is 19.985 × 19.950 mm).

The optimization method in Eqs. (3.29) and (3.30) was implemented in MATLAB®; the `lsqnonlin()` Levenberg-Marquardt optimization function is used, viz. in its large-scale variant. I choose to provide Jacobian patterns for its sparse numerical implementation to keep computational costs low.

Joint optimization of camera and scene

Next I show the resulting camera parameters and scene structure as well as the residuals after calibration both in image and in 3-D target coordinates.

Using an accurate, planar metallic target: Planar calibration targets imprinted on metallic plates provide both structural stability and high planarity. This is a best-case scenario to camera calibration, thus less profit is expected from concurrent scene structure estimation.

In fact, both monocular and stereo joint intrinsic and full scene structure estimations deliver almost identical camera parameters w.r.t. the standard approaches, cf. Tables 3.4 and 3.3. The reason is the very slight optimization of the pattern structure achieved, see Fig. 3.19, in the region of a tenth of a millimeter. The target optimization is mainly in its 2-D imprinted pattern because, apparently, it is still subject to inaccurate printing errors similar to off-the-shelf paper printers, see Fig. 3.18 (a). Fig. 3.19 (a) shows a planarity correction in the order of a tenth over 200 mm—a very slight bending of the plate.

A remarkable result is, however, the significant reduction both in image and object reprojection residuals, see Figs. 3.17 and 3.18. Image reprojection residual errors are measured by their **R**oot **M**ean **S**quare error (RMS). Nevertheless, these reductions result from calibration-related minimizations and their potential effects in final accuracy still have to be experimentally verified, see the results below.

Table 3.3: Estimated intrinsic parameters after standard (Std.) and simultaneous scene structure and **monocular** calibration (Full), using a precision target.

	${}_{\mathrm{L}}\alpha$	${}_{\mathrm{L}}\beta$	${}_{\mathrm{L}}u_0$	${}_{\mathrm{L}}v_0$	${}_{\mathrm{L}}k_1$	${}_{\mathrm{L}}k_2$	RMS
Std.	724.58	723.93	372.44	272.17	−0.1960	0.0994	0.151
Full	724.50	723.69	371.92	271.08	−0.1955	0.0975	0.063

Table 3.4: Estimated intrinsic parameters after standard (Std.) and simultaneous scene structure and stereo calibration (Full) for both cameras of the **stereo** camera, using a precision target. Bottom: Resulting RMS reprojection error for both cameras after standard (Std.) and novel (Full) calibration.

	Left camera						
	${}_{\mathrm{L}}\alpha$	${}_{\mathrm{L}}\beta$	${}_{\mathrm{L}}u_0$	${}_{\mathrm{L}}v_0$	${}_{\mathrm{L}}k_1$	${}_{\mathrm{L}}k_2$	RMS
Std.	724.79	724.12	372.42	272.34	−0.1963	0.0995	0.155
Full	724.32	724.35	372.20	271.22	−0.1973	0.0993	0.078

	Right camera						
	${}_{\mathrm{R}}\alpha$	${}_{\mathrm{R}}\beta$	${}_{\mathrm{R}}u_0$	${}_{\mathrm{R}}v_0$	${}_{\mathrm{R}}k_1$	${}_{\mathrm{R}}k_2$	RMS
Std.	728.31	728.01	391.73	270.35	−0.1962	0.1008	0.173
Full	727.85	728.40	391.55	269.23	−0.1982	0.1033	0.077

	Both cameras
	RMS
Std.	0.165
Full	0.077

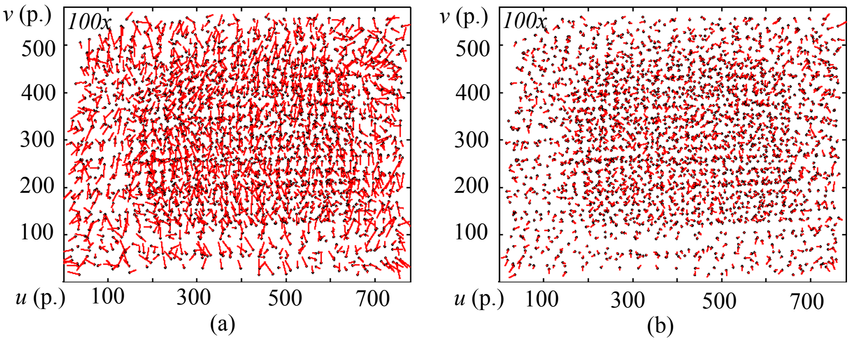


Figure 3.17: Magnified (100×) image reprojection errors for all 12 left calibration images after std. camera calibration (a) and after full estimation (b), using a precision pattern. RMS error reduces from 0.151 to 0.063 pixels.

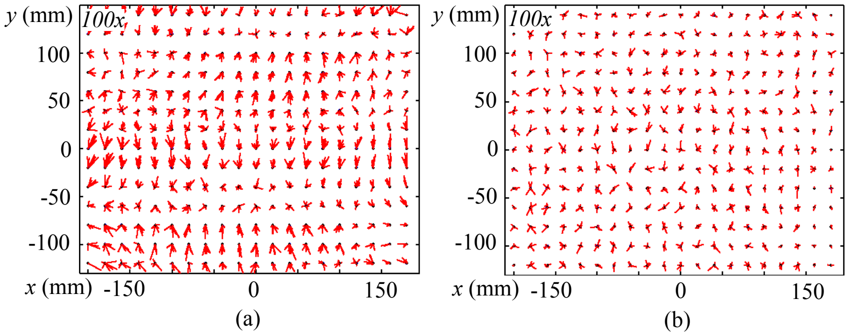
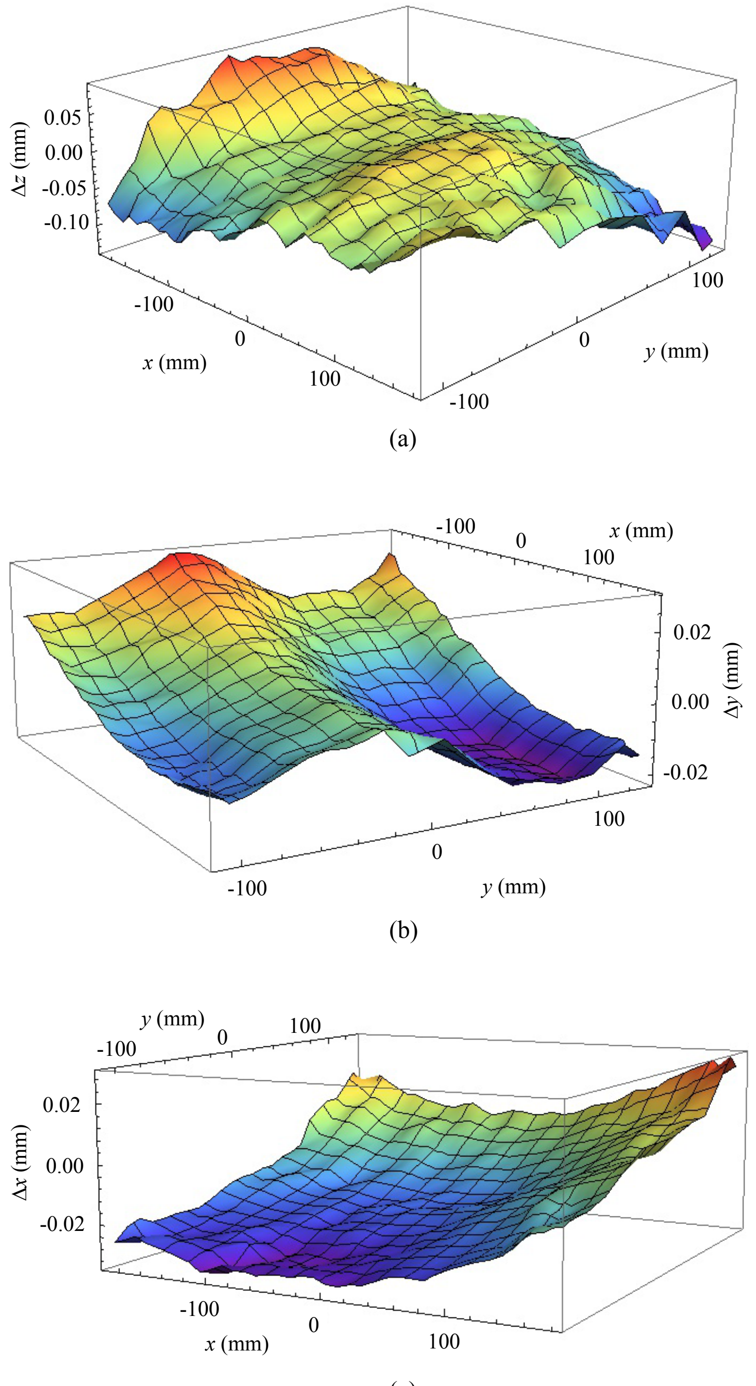


Figure 3.18: Perpendicular projection of magnified (100×) object reprojection errors for all 12 left calibration images after standard camera calibration (a) and after full estimation (b), using a metallic precision pattern.



(a)

(b)

(c)

Figure 3.19: Corrected feature positions Δz (height), Δy and Δx (in 2-D) after joint intrinsic and full scene structure estimation on the precision target. Corrections are consistent after monocular and stereo approaches.

Using an unknown, wrinkled paper target: Checkerboard patterns on paper using off-the-shelf printers are the most convenient calibration targets that still guarantee accuracy and repeatability in detection and localization of their corner projections, through several images.[17] Indeed, printed patterns are the most used calibration targets worldwide (Mallon and Whelan, 2007; Strobl *et al.*, 2005). Researchers struggle to stick them on planar surfaces and, more often than not, to measure up their dimensions. Eventually they get humid and bumpy and need to be replaced.

For reasons of space I am addressing a worst-case scenario where the pattern is *not* being measured after printing. I assume corner distances of 2 cm as in the original PostScript® file. On top of that, the paper target has a folding mark in the middle so that it is clearly non-planar, see Fig. 3.16.

Standard camera calibration cannot deliver accurate results over this pattern, see Tables 3.6 and 3.5. The image reprojection residuals after calibration in Fig. 3.20 (a) are very high, owing to strong systematic errors in the object model, see Fig. 3.21 (a). The proposed methods *do* compensate for these object model errors, see Figs. 3.20 (b) and 3.21 (b), so that the intrinsic camera parameters virtually match former results in Tables 3.4 and 3.3.[18] The object model optimization performed during calibration is depicted in Fig. 3.22. The results correspond with the expected deformation showing unevenness of approximately 6 mm.

A further drawback of using the standard method with this type of patterns is that measuring its dimensions is difficult, as the pattern is delicate and easy to deform. By using the novel method this step is rendered superfluous. In the case of stereo calibration, the input of a single absolute distance between two arbitrary pattern corners suffices, cf. Eq. (3.30). If hand-eye calibration is additionally performed, the user can spare this last measurement.

Table 3.5: Intrinsics after standard (Std.) and simultaneous scene structure and **monocular** calibration (Full), using an unknown, wrinkled paper.

	${}_{\mathrm{L}}\alpha$	${}_{\mathrm{L}}\beta$	${}_{\mathrm{L}}u_0$	${}_{\mathrm{L}}v_0$	${}_{\mathrm{L}}k_1$	${}_{\mathrm{L}}k_2$	RMS
Std.	718.65	723.05	370.78	268.53	−0.2518	0.1721	2.105
Full	724.35	723.58	372.18	270.90	−0.1943	0.0946	0.069

[17] The only more convenient calibration target is unstructured scenery (self-calibration), which does not, however, guarantee accurate and robust feature detection and localization.

[18] In the case of stereo calibration, residuals do not quite reach the levels of the metallic pattern, cf. Tables 3.4 and 3.6; if the paper was not folded but directly put on a table after printing, results do match exactly, irrespective of natural paper bending. The difference can be explained either by noisy detection of pattern features due to local shadows, or by stagnant convergence of the nonlinear optimization. Either way, the validity of the parametrization is not stated by the calibration RMS but by independent validation experiments as in the next section.

Table 3.6: Estimated intrinsic parameters after standard (Std.) and simultaneous scene structure and stereo calibration (Full) for both cameras of the **stereo** camera, using an unknown, wrinkled paper target. Bottom: Resulting RMS reprojection error for both cameras after standard (Std.) and novel (Full) calibration.

	Left camera						
	${}_{\mathrm{L}}\alpha$	${}_{\mathrm{L}}\beta$	${}_{\mathrm{L}}u_0$	${}_{\mathrm{L}}v_0$	${}_{\mathrm{L}}k_1$	${}_{\mathrm{L}}k_2$	RMS
Std.	718.99	724.10	362.60	268.97	−0.2534	0.1706	2.180
Full	724.71	724.07	372.53	270.87	−0.1968	0.0971	0.111

	Right camera						
	${}_{\mathrm{R}}\alpha$	${}_{\mathrm{R}}\beta$	${}_{\mathrm{R}}u_0$	${}_{\mathrm{R}}v_0$	${}_{\mathrm{R}}k_1$	${}_{\mathrm{R}}k_2$	RMS
Std.	719.64	724.71	393.26	271.03	−0.2050	0.0840	2.084
Full	728.18	728.08	391.74	268.89	−0.1981	0.1013	0.115

	Both cameras
	RMS
Std.	2.133
Full	0.113

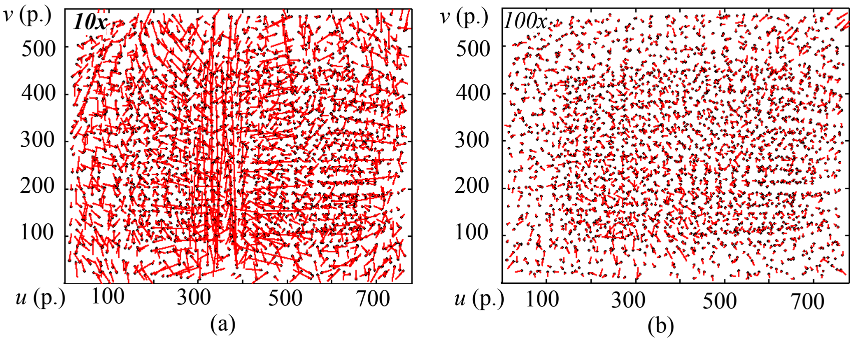


Figure 3.20: Magnified (10×) image reprojection errors for all 12 left calibration images after std. camera calibration (a) and after full estimation (b), using a wrinkled paper pattern. RMS error reduces from 2.105 to 0.069 pixels.

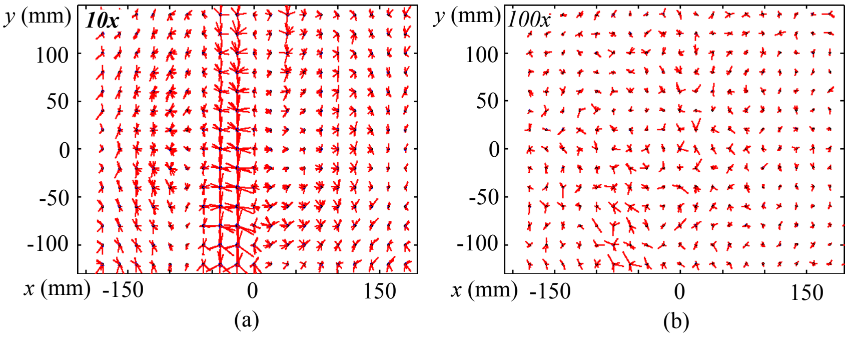
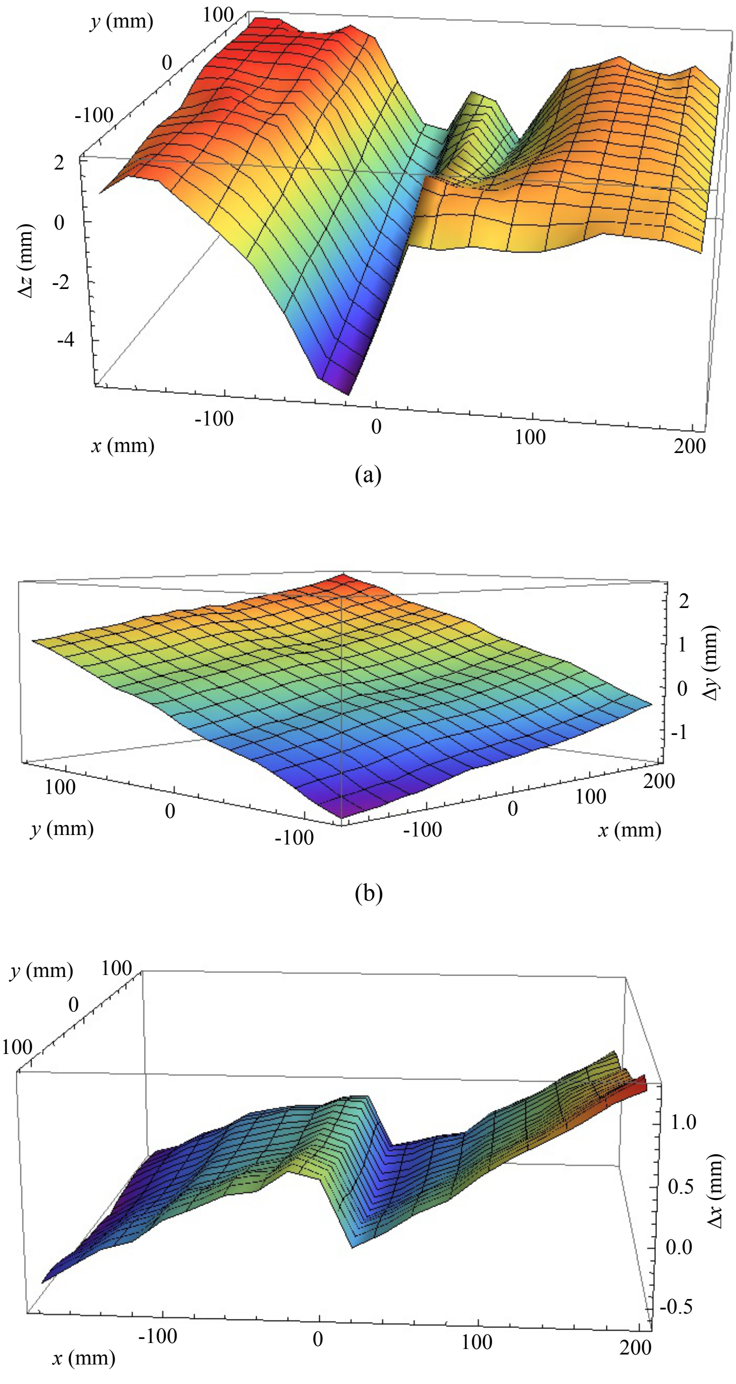


Figure 3.21: Perpendicular projection of magnified (10×) object reprojection errors for all 12 left calibration images after standard camera calibration (a) and after full estimation (b), using a wrinkled paper pattern.



(a)

(b)

(c)

Figure 3.22: Corrected feature positions Δz (height), Δy and Δx (in 2-D) after joint intrinsic and full scene structure estimation on the paper target. Corrections are consistent after monocular and stereo approaches.

Accuracy evaluation

Next I show stereo triangulation results on image data independent from calibration data; these results will be used to check calibration methods against each other. I replicate the validation experiment in (Albarelli *et al.*, 2010), which measures the distance d between two rigid points in 3-D space, refer to Fig. 3.23. The camera continuously moves in the direction of its optical axis. In order to reach optimal feature localization accuracy, I take two particular corner features in a checkerboard pattern that is standing perpendicular to the camera motion.

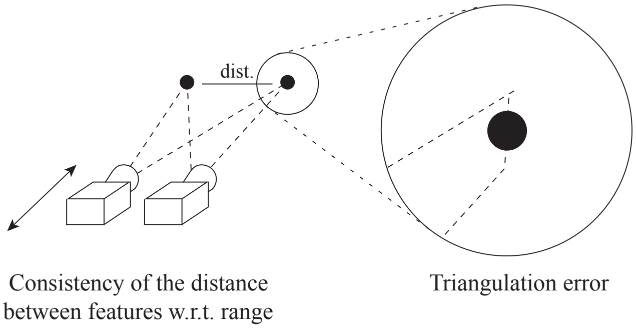


Figure 3.23: Validation by stereo vision: I measure the Euclidean distance d between two features for different camera-to-feature ranges.

The features are located approximately $d = 22$ cm apart from each other. The measured distance d is, however, irrelevant to my analysis as it ultimately depends on the accuracy when measuring the pattern scale *by hand* during calibration, which is naturally limited. A valid hint for calibration accuracy is, however, the *consistency* of the distance estimation at different triangulation ranges (Albarelli *et al.*, 2010). Fig. 3.24 (a) shows that, both with and without full scene structure estimation, the metallic plate-based stereo camera calibration delivers near-constant estimations that drift half a millimeter (out of 220 mm) from 0.3 to 1 m range. Paper target-based calibration causes a major drift of 2 mm unless full structure estimation is performed—then results again match the former.

Flawless stereo triangulation is of course impeded by inaccurate feature detection and imperfect camera calibration—*i.e.*, estimated ray directions will not intersect. I choose the 3-D point $\boldsymbol{i}$ in the middle of the segment of minimum distance between the left (camera L) and the right (camera R) stereo rays ${}_{\mathrm{L}}\boldsymbol{l}$ and ${}_{\mathrm{R}}\boldsymbol{r}$ as the triangulation result for a particular feature, see the detail at the right-hand side of Fig. 3.23. Mathematically, it can be represented as follows:

$$\begin{aligned} {}_{\mathrm{L}}\boldsymbol{i} &= L\,{}_{\mathrm{L}}\boldsymbol{l} + \frac{N}{2}\,{}_{\mathrm{L}}\boldsymbol{n} = R\left({}_{\mathrm{L}}\widehat{\boldsymbol{R}}_{\star}^{\mathrm{R}}\,{}_{\mathrm{R}}\boldsymbol{r}\right) + {}_{\mathrm{L}}\hat{\boldsymbol{t}}^{\mathrm{R}} - \frac{N}{2}\,{}_{\mathrm{L}}\boldsymbol{n} \quad / \\ {}_{\mathrm{L}}\boldsymbol{n} &= {}_{\mathrm{L}}\boldsymbol{l} \times {}_{\mathrm{L}}\widehat{\boldsymbol{R}}_{\star}^{\mathrm{R}}\,{}_{\mathrm{R}}\boldsymbol{r}, \quad L \in \mathbb{R}, \quad N \in \mathbb{R}, \quad R \in \mathbb{R} \quad . \end{aligned} \tag{3.32}$$

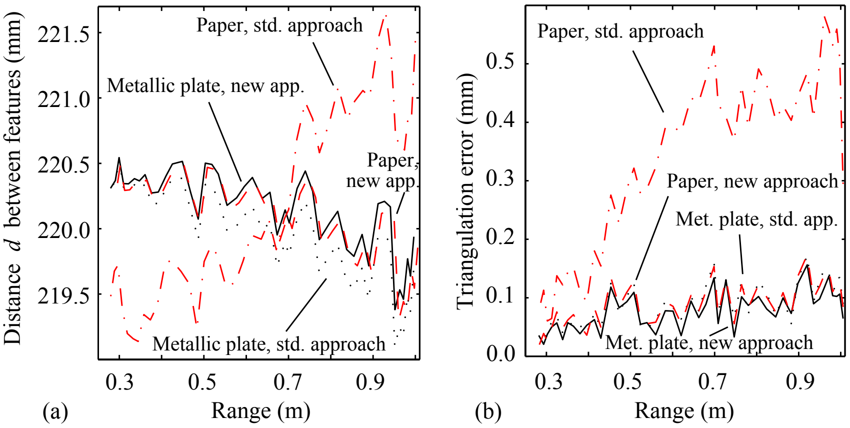


Figure 3.24: Validation by stereo: (a) Distance d between rigid points and (b) mean value of the triangulation error, w.r.t. camera range.

Eq. (3.32) forms a linear system of 3 equations and 3 unknowns L, N and R that is solved by LU factorization. Similar to the consistency in distance estimation in Fig. 3.24 (a), the minimum distance N between stereo reprojection rays here also indicates calibration accuracy. Fig. 3.24 (b) shows its mean value for both corner points w.r.t. camera range. For the metallic plate-based calibration, stereo triangulation is performed with half a tenth of a millimeter triangulation error at any distance tested. Scene structure estimation does slightly improve consistency (9.9% error decrease). Results are clearer for the paper target-based calibration, where triangulation errors increase to four tenths of a millimeter at far range if the standard calibration method is used. If scene structure estimation was performed, error levels shrink again to half a tenth of a millimeter (**72**% error decrease), exactly as when using the metallic plate.[19]

It is worth noting that it is the estimated $(\widehat{\ })$ extrinsic rigid transformation between cameras ${}_{\mathrm{L}}\widehat{\boldsymbol{T}}^{\mathrm{R}}$ that is mainly responsible for the results presented here. Unlike in the experiment presented in Ref. (Albarelli *et al.*, 2010), in this work the stereo transformation *fully* results from the full structure estimation paradigm introduced above. Furthermore the examined range extends to 1 m.

I have also compared both, the standard and the proposed method, using *dense* stereo vision methods. Fig. 3.25 shows a carafe (viz. its disparity reconstruction) as seen by the humanoid robot “Justin” (Borst *et al.*, 2009) using its stereo camera head (the DLR 3D-Modeler) and the SGM stereo vision algorithm (Hirschmüller, 2008). The novel method presented in this section allows for more complete results, particularly in untextured edges parallel to the epipolar line.

[19] More specifically: 7.8% worse than after full scene structure estimation using the metallic plate, but then 3.4% *better* than standard calibration using the precision metallic pattern.

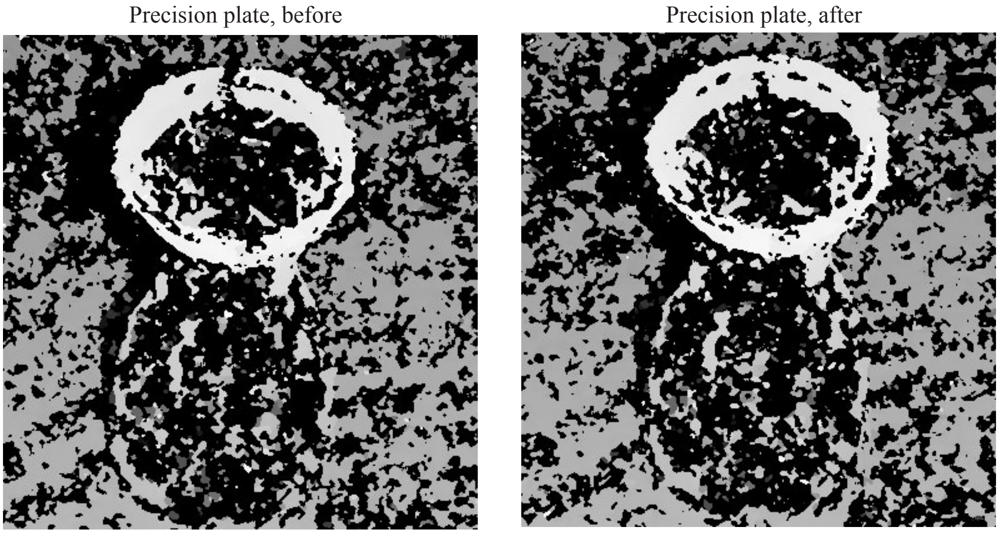


Figure 3.25: Validation by *dense* stereo vision (Hirschmüller, 2008): Standard stereo camera calibration using a metallic precision plate delivers worse results (left) than when releasing its pattern structure following the novel method presented in this section (right)—even if the calibration pattern of the metallic precision plate had been very accurately measured before.

Discussion

At first the above results make for somewhat of a disappointment. If camera calibration is dutifully performed, less extra accuracy is attained by simultaneous estimation of full scene structure.[20] All things considered, however, it is very difficult for most users to produce an exact calibration target and, on top of that, it comes at no cost to calibration accuracy to perform simultaneous intrinsic and full scene structure estimation as long as two slight limitations are observed: First, to avoid gathering features in image corners with exclusive support; additional images are encouraged where the pattern is fully captured.[21] Second, the calibration target has to remain static unless it is rigid material; it is the camera and not the calibration target that should be shifted for grabbing images.

The experiments above show that simultaneous intrinsic and full scene structure estimation should be performed in any situation where the calibration target is expected to be nearly planar. Apart from delivering results at least as accurate as from a flawless standard implementation, the method deskills

[20] The authors in (Albarelli *et al.*, 2010) observe that, using their method on an accurate planar target, scene structure is optimized prior to camera parameters since this minimizes residuals faster—they cannot provide an explanation for that. My read of this phenomenon is that, since it is only scene structure optimization that minimizes residual errors, camera parameters do not *significantly* change. Standard, least squares optimization with abundant, redundant data *already compensated* for the former structure inaccuracies, thereby delivering optimal, accurate intrinsic parameters in the first place.

[21] For that matter, it is widespread to *only* take this type of images during camera calibration anyway; incidentally, this is a harmful habit to accurate calibration.

the calibration procedure, thus prevents damage from pattern inaccuracies and human mistakes. This is especially true in the case of printed paper patterns or bigger targets (e.g. patterns projected by an overhead projector), which are difficult to measure accurately. In view of the blatant similarity to bundle adjustment—gold standard for structure from motion approaches, the current methods have the potential to be considered *gold standard for pinhole camera calibration using planar targets.*

3.4.5 Summary

In this section 3.4 I identified and addressed the problem of widespread inaccurate knowledge of the 3-D geometry of the pattern imprinted on calibration targets; this type of patterns are being predominantly used in the context of camera calibration. I note that highly accurate knowledge of the dimensions of the calibration pattern rarely exists, and furthermore that this violation has negative effects on the proper estimation of the camera parameters.

Chessboard calibration patters are employed because the alleged homogeneity of their geometric structure allows for the user to extrapolate measured coordinates still with high accuracy. The fact that off-the-shelf printers lack of accuracy when scaling the pattern in both its main directions goes, however, more often than not unnoticed. In addition, calibration patterns on paper may provide a bumpy structure that is not in accordance with the standard camera calibration method in Section 3.2, as users may fail to flatten the pattern on a flat surface. What is more, a frightening large number of users skip over the necessary step of accurately measuring the homogeneous chessboard pattern. I show that these facts are the cause of significant calibration errors that will not allow for useful computer vision algorithms as intended.

I also noticed the non-availability of appropriate methods in the literature that address this topic. Still, an overview of the literature on camera calibration presents a tendency to decrease the complexity of the calibration object, motivated by the fact that this deskills the calibration procedure. I elaborate on this motivation and suggest that, in fact, this trend is appropriate, as less complex objects prevent damage to the calibration due to metric inaccuracies. Yet I take the matter further, easing requirements of knowledge of the metric dimensions of the calibration pattern.

In the following, I presented two methods that intend to increase camera calibration accuracy irrespective of the cooperation of the user, *i.e.*, even in the case of indolent users that do not pay attention to the validity of the calibration object model. Incidentally, during final experiments I find out that the proposed methods deliver higher accuracy even for meticulous users that do take the prior step to accurately measure the imprinted pattern. As illustrated in Fig. 3.26, I intend to bring *all* users to accuracy levels typical of meticulous users.

First, I bring forward a preliminary approach that accounts for printing errors of off-the-shelf printers. It turns out that there exists a simple parameterization of the checkerboard pattern composed of its aspect ratio and its absolute scale that, on the one hand, corresponds to the actual inaccuracies resulting from regular printing equipment, and on the other allows for optimal intrinsic and extrinsic calibration irrespective of their actual values. It becomes clear that accurate intrinsic calibration is possible irrespective of the absolute scale of the scene; more importantly, the aspect ratio of the calibration pattern can be optimized *at the same time* during intrinsic camera calibration. Consequently, in Eq. (3.25) I extend the formulation of the standard approach in Section 3.2 with the aspect ratio of the pattern, and in Eq. (3.26) I do so with the hand-eye calibration in Section 3.3. The algorithm was originalally presented in (Strobl and Hirzinger, 2008), outperforming conventional methods that require accurate knowledge of the pattern dimensions.

We then received feedback from the scientific community in (Albarelli *et al.*, 2010). Albarelli *et al.* note that significant, systematic pattern errors are pervasive in calibration object patterns; regrettably, they fail to deliver an optimal formulation to cope with that errors. In this section I revisited the alternative that we proposed in (Strobl and Hirzinger, 2011); it optimizes the full structure of the calibration object in a minimal way and does indeed lead to better results than using either the standard or the preliminary approach above. Simulations as well as experiments confirm this last claim.

It is worth noting that, as a by-product and in line with one of the main topics of this thesis, I provide a more convenient, flexible algorithm that deskills the camera calibration process leading to more effective perception systems, refer to Section 1.1.

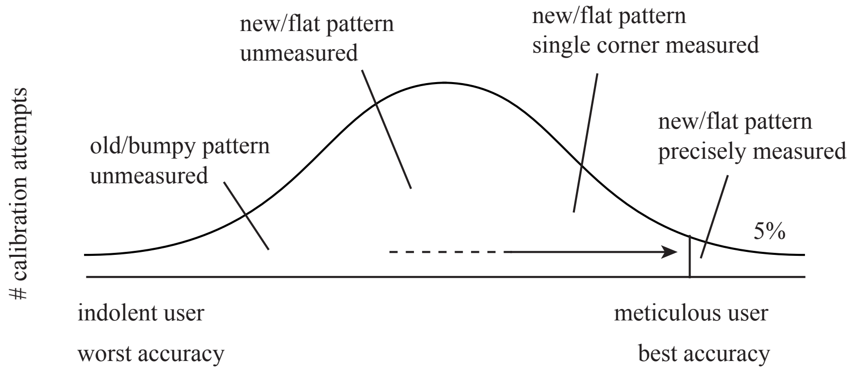


Figure 3.26: By virtue of my experience with users of the calibration toolbox DLR CalDe and DLR CalLab (Strobl *et al.*, 2005), I picture the frequency of calibration attempts regarding the rigor of the user when providing the calibration object model to the calibration algorithm.

3.5 Caveat #2: Cameras with Narrow Angular Field of View

3.5.1 Introduction

I furthermore consider the issue of calibrating a camera with narrow angular field of view (AOV) using the standard camera calibration method in Section 3.2. Many applications in robotics and beyond make use of cameras with narrow AOV; these cameras either show a long focal length (e.g. laparoscopic cameras or satellite cameras for docking maneuvers from distant rendezvous, refer to Section B.2.3 within Appendix B) or cameras that are limited in their field of view by application-specific obstacles (e.g. a car wheel in the context of vision-based car wheel mounting in assembly lines, refer to Section B.2.6 within Appendix B).

Perspective distortion in images is direct consequence of the use of pinhole model-like cameras, see Section 2.2.1. Camera calibration as in Section 3.2 regularly uses the perspectivity distortion captured in the calibration images to discern camera range and focal length. Camera range is sole responsible for the perspective distortion shown in the images, whereas focal length merely scales the whole image homogeneously. Regrettably, the narrower the AOV, the more difficult it is to show perspective distortion in the calibration images, see Fig. 3.27, hence standard camera calibration gets badly conditioned.

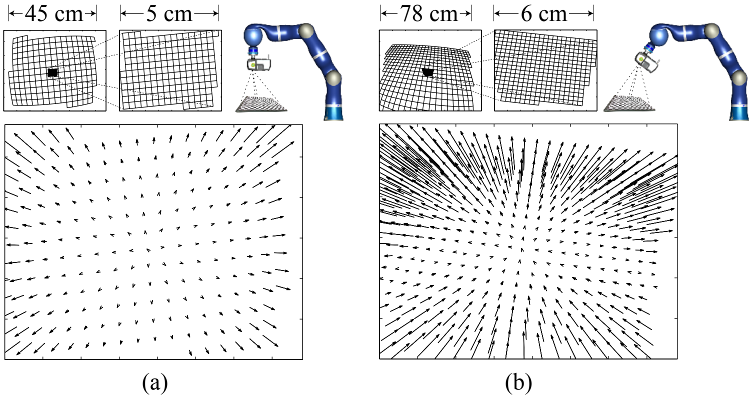


Figure 3.27: Camera projection of the corners of chessboard calibration patterns distant 32 cm, (a) perpendicular to the principal axis of the camera, and (b) distant 41 cm and tilted 37° w.r.t. the principal axis of the camera. Projections are shown for two different scaling parameters $\alpha = \beta = 482$ ($86° \times 65°$) and $\alpha = \beta = 4820$ ($9.5° \times 7°$). Radial lens distortion is fixed to a realistic value of $k_1 = 0.155$, and image size is 780×580. Of course, the sizes of the object pattern differ for each projection. Corresponding points within each image are linked together in order to show the very significant evidence on perspective distortion **from oblique views** (b). The residuals in (a) are sole consequence of the static, radial lens distortion. At the top both projections are separately depicted; in addition, two illustrations show the mentioned vantage points with cameras mounted at the DLR Light-Weight Robot 3.

From this, I propose an alternative method that compensates for the loss in perspectivity by utilizing the pose readings of a robotic manipulator like the Kuka KR 16 or the tracking system ARTtrack2. The proposed method facilitates accurate pose estimation by nonlinear optimization, minimizing reprojection errors *and* errors in the manipulator transformations at the same time, similar to the hand-eye calibration method in Section 3.3. Accurate pose estimation in turn enables accurate parametrization of a perspective camera.

3.5.2 State of the Art

The foundation pillars for standard, perspective camera calibration are:

a) appropriate definition of the camera and calibration object model,

b) successful initial estimation of its parameters,

c) availability of enough evidence on perspective distortion,

d) and finally the proper estimation of the scene structure.

Whenever one of these pillars is shaking, the accuracy of standard camera calibration is compromised.

It is remarkable that the geometric model of ancient *pinhole* cameras still holds for accurately describing the main functioning principle of a number of modern cameras (a). An approach for accurate, simple parameter initialization within this model (b) was proposed in (Zhang, 2000; Sturm and Maybank, 1999); this approach proved extremely useful, thus most successful. Strobl and Hirzinger noted in (Strobl and Hirzinger, 2008, 2011) a predominant error source for correct scene structure estimation (d) and brought forward an alternative formulation. Now I focus on a critical aspect concerning the remaining pillar: the requirement for satisfactory evidence on perspective distortion, in particular in relation to the limited angular field of view (AOV) of some cameras.

Perspective distortion is direct consequence of the use of pinhole model-like cameras. It describes the mapping of a 3-D scene onto its 2-D image and can be roughly summed up by these two circumstances: close objects project bigger, and differently distant objects may project onto the same region—*i.e.*, range gets lost. These circumstances are regularly used for camera calibration since they help to discriminate between the Euclidean structure of both the scene and the camera, and the camera magnifying characteristics themselves. The images in Fig. 3.27 show different perspective distortion effects on images of a planar pattern in relation to both the external orientation of the camera—(a) against (b)—and different magnifying characteristics of its perspective camera model—within each figure.

In this respect, I address the issue of calibrating a camera with very limited AOV. It is difficult to gather enough evidence on perspective distortion with that type of cameras, thus calibration accuracy gets compromised. Even though the particularities of wide AOV have been often addressed (Brandt and Kannala, 2006), to the best of my knowledge the issue in this paper has been left largely untreated in computer vision—apart from changing into affine camera approximations.[22] According to my experience on maintaining the camera calibration toolbox DLR CalDe and DLR CalLab (Strobl *et al.*, 2005), there is less concern among the users about applying regular camera calibration methods for camera systems with extremely *narrow* AOV.[23] Of course, there exist photogrammetric approaches to deal with this problem since it occurs very frequently in that field; these methods and their required equipments (e.g. optical collimators) are, however, rarely available outside photogrammetric labs.

What makes it all the worse are the *numerous applications* of narrow AOV cameras, and what is more that these applications are mostly justified precisely by the high accuracy that they are supposed to provide. A collection of examples: long focal length cameras for feedback control of robotic manipulators in industry (e.g. laser beam welding), high-accuracy positioning by gazing at landmarks in structured environments (Davison, 1999), foveal vision e.g. for anthropomorphic research (F. Seara *et al.*, 2003), manufacturing inspection in intricate cavities, etc. In practice, it is largely only possible with this type of cameras to further increase the already high accuracy of current robotic manipulators.

3.5.3 The Role of the Focal Length in the Pinhole Camera Model

The pinhole camera model is the main part of the projection model of most cameras in computer vision applications. It represents the perspective projection taking place when mapping the 3-D world scene onto the 2-D imaging plane by rays of light passing through a (conceptual) point, the camera focal point or focus. In reality, the imaging plane is usually instantiated by an electronic imaging sensor like charge-coupled devices (CCD) or CMOS chips, and the focus of the camera is located at the aperture center of the frontal lens. Further potential parts of the camera model are the digitization process, the lens distortion model, or the extrinsic rigid body transformation from the camera to an external point, see Section 2.2.1.

The geometrical mapping of 3-D points ${}_0\boldsymbol{p}$ in the world/object reference frame S_0 onto their projections ${}_\mathrm{M}\bar{\boldsymbol{p}}_\mathrm{u}$ in the memory plane S_M has been already formulated mathematically in Eq. (3.5.3) as follows:

$$
{}_\mathrm{M}\bar{\boldsymbol{p}}_\mathrm{u} = \begin{bmatrix} {}_\mathrm{M}x_\mathrm{u} \\ {}_\mathrm{M}y_\mathrm{u} \\ 1 \end{bmatrix} \propto \underbrace{\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}_{(3\times3)}} \begin{bmatrix} {}_\mathrm{C}x \\ {}_\mathrm{C}y \\ {}_\mathrm{C}z \end{bmatrix} = \underbrace{\boldsymbol{A}_{(3\times3)}\ {}_\mathrm{C}\boldsymbol{T}^0_{(3\times4)}}_{\boldsymbol{P}_{(3\times4)}} \begin{bmatrix} {}_0x \\ {}_0y \\ {}_0z \\ 1 \end{bmatrix} = \boldsymbol{P}_{(3\times4)}\ {}_0\bar{\boldsymbol{p}}\ ,
$$

[22] Affine camera models are tolerable approximations of perspective projection cameras when the AOV and the relative variation of depth are small. Their models are linear (instead of merely linear projective in the case of perspective models), thus allow for linear algebra solutions (instead of nonlinear solutions). In addition, affine camera calibration is better conditioned for narrow AOV. Still, affine camera models are very limiting approximations (Hartley and Zisserman, 2004; Faugeras *et al.*, 2001; Christy and Horaud, 1996).

[23] Bad relative positioning choices between camera and calibration plate is reportedly the other most common reason for erroneous camera calibration (Strobl *et al.*, 2005).

where $\boldsymbol{P}$ is the perspective projection matrix, which consists of the camera intrinsic matrix $\boldsymbol{A}$ and the rigid body transformation ${}_{\mathrm{C}}\boldsymbol{T}^0$ from the **c**amera frame S_{C} to the object/world frame S_0 at a particular imaging instant (time and point-related indexes have been omitted for the sake of clarity). The matrix $\boldsymbol{A}$ is in turn composed of the scaling parameters α and β, which are directly proportional to the focal length, the skew parameter γ, which represents slight skewness in the image plane coordinates, as well as the 2-D coordinates u_0 and v_0, which locate the principal point in the image frame. The principal point is supposed to be its closest point to the focal point—usually it is *not* (Willson and Shafer, 1994).

This is the generally established formulation but other formulations exist as well. In the past, the model was much more related to actual camera parameters like the sizes of the picture elements in different directions, or to the focal length (Tsai, 1987; Faugeras and Toscani, 1987). However, this does not pay off for both calibration and utilization of regular cameras, and all-encompassing parameters like α or β are currently preferred (Bouguet, 2002; Strobl *et al.*, 2005). Nonetheless, the user should bear in mind both their origin and nature.

The focal length is one of the main camera parameters that have to be taken into account *either in reconstruction*, in order to extract information from the image projection, *or in acquisition*, to determine both scene and camera relative poses so that the user eventually obtains the desired image projection. In reality, focal length defines the perpendicular distance between the focal point and the image frame; for instance, the angular area of the projected scene *reduces* when the focal length increases (paradoxically this is what we get to call image *amplification*), which is due to the limited size of the sensor chip.

But strictly speaking, in the perspective distortion issue it is all about *the pose* of the camera w.r.t. the scene, since it primarily defines the potential perspective distortion that we can expect from the whole scene—whereas the focal length relates to the AOV by narrowing or broadening it (thus determining the absolute scale of the projection) but without modifying its appearance. When one speaks of decreasing the perspective distortion by increasing the focal length what actually occurs is *either* that the projected scene reduced to a small section of the original one, without moving the camera nor the potential perspective distortion, *or* that the camera departed from the scene and the focal length had to be increased in order the same part of the original scene to remain on camera—losing some perspective distortion all this way. In this second case, increasing the focal length is just a by-product of the action of moving the camera since else the imaging chip would get a huge viewing area, wasting most of its valuable pixels for void space.[24]

[24] Since A. Hitchcock's *Vertigo*, filmmakers make an extensive use of this effect to provoke a disquieting sensation, or the character's reassessment of a situation.

In the next sections both, the performance of camera-based estimations like feature-based pose estimation, as well as the performance of standard camera calibration, are to be discussed in relation to the AOV of the camera. The simulation results mentioned in the next sections are strongly based on real data—therefore the seemingly arbitrary choice of coordinates. In all simulations, the poses of the camera w.r.t. the scene remain constant for every set of images; this allows for fair comparisons regarding precision in pose estimation. The focal length (and the size of the pattern) of course do vary. It is useful to first clarify the relationship between AOV and the focal length, which is a nonlinear one, see Fig. 3.28 (a).

In general, of course, the longer or shorter the focal length, the smaller or bigger the AOV, respectively. The reduction of AOV in a couple of degrees when it is already small, however, does take much bigger an increase in focal length than it would take if the AOV were bigger. This is inconvenient e.g. if it is required to represent simulation data w.r.t. the AOV (as we do here for more natural and general reading), since uniformly distributed sampling in focal length implies highly non-uniform distributions in AOV. This issue is easily handled by uniformly distributing *on the inverse* of the focal length, which almost linearly corresponds to the AOV, see Fig. 3.28 (b). In this work all simulations are going to be performed on this distribution—yet represented in AOV.

For the rest, customary camera parameters are used. It is worth mentioning that the resolution is invariably set to moderate 780 × 580 pixels—this value is relevant only in direct conjunction with the scaling parameters and the image processing noise, which follows an homogeneous, 2-D zero-mean Gaussian distribution with standard deviation $\sigma_{\{x,y\}} = 0.4$ pixels. This potentially maps to many actual camera systems.

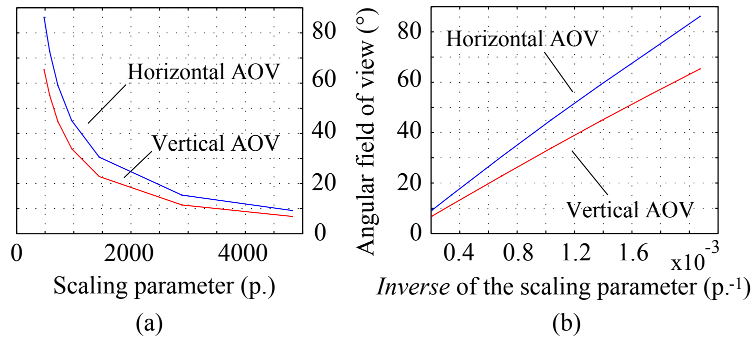
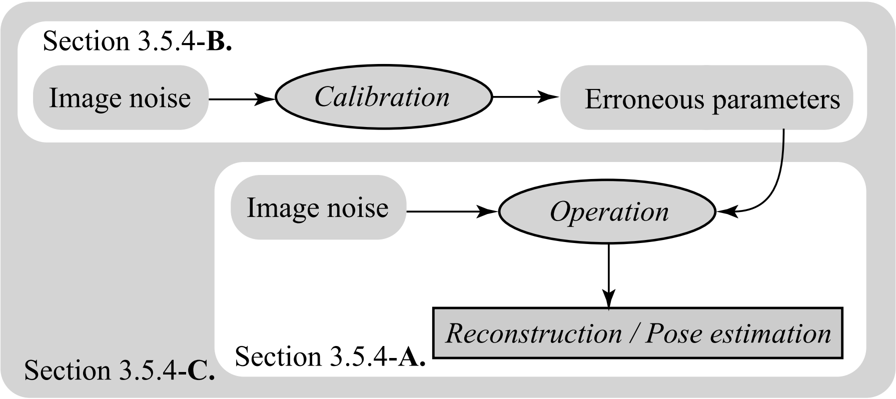


Figure 3.28: (a) Relation of the AOV with the scaling parameter $\alpha = \beta$, and (b) with its inverse. Typical values for radial distortion have been considered.

3.5.4 Erroneous Camera Model Utilization and Parametrization

One of the main points of this work is to assess the accuracy that we can really, *finally* expect from vision-based algorithms in general. On the way to the final application, the accuracy becomes compromised in several steps. It is indicated to separately study these error sources in this section.

The following sequence of events is responsible for inaccurate vision-based estimation in most computer vision applications that require calibrated cameras. Starting out from the calibration process: In the beginning was only image processing noise when detecting features in images for calibration. Through calibration we may get a parameterized camera model, but its values are erroneous to some extent, and what is more, even the model is only approximated. Eventually, in final camera operation, both the erroneously estimated camera model and *additional* image processing noise jointly affect the accuracy of the final estimation adversely.



Next the operation from noisy image processing (**A.**) is studied. Then I also consider erroneous calibration from noisy image processing (**B.**). On the basis of these latter results, I will extend the former initial results on noisy operation taking also camera model parametrization errors into account (**C.**).

A. Image-based pose estimation from noisy image processing

In this section I present ordinary results on camera *pose estimation* from known scenery on the pretentious assumption that both, the camera parametrization and its model, are totally accurate. The scenery corresponds to a perfectly known planar chessboard pattern as used for camera calibration. The projections of the pattern are affected by homogeneous Gaussian noise as above mentioned. The pose estimation algorithm is an optimal nonlinear optimization process that minimizes the sum of squared reprojection errors of the calibration pattern—the process is optimal provided that the estimation is initialized on the convex area of the absolute minimum. This frugal example is in preparation for more complex ones in the following.

In Fig. 3.29 the pose estimation precision for different AOVs is shown; the camera is at a fixed distance and perpendicular view to the plate. The curves result from 7 AOV points, uniformly sampled on the inverse scaling parameter space (or inverse focal length space). The data stem from a Monte Carlo simulation consisting of 1000 pose estimation optimizations repeated with independent image noise, for each AOV. The images in Fig. 3.27 (a) correspond to the horizontal extremes in Fig. 3.29, *i.e.*, with the widest and the narrowest AOVs.

The figure shows a considerable worsening of both, positioning and orientation estimation precision, for small AOVs—even though the camera model still holds perfectly. It was mentioned in Section 3.5.3 that it is the camera pose that is responsible for perspective distortion in the images. Since planar structure points from perpendicular images present similar distances, their images provide less *variation* in perspective distortion w.r.t. the camera pose (cf. Fig. 3.27 (a)), which comes near by affine projection and ambiguities like the Necker reversal. Therefore, pose estimation becomes bad conditioned. It is only due to both, the known structure and the known camera scaling (focal length), that at least the estimation of the range (absolute distance) is good conditioned (see z in Fig. 3.29).

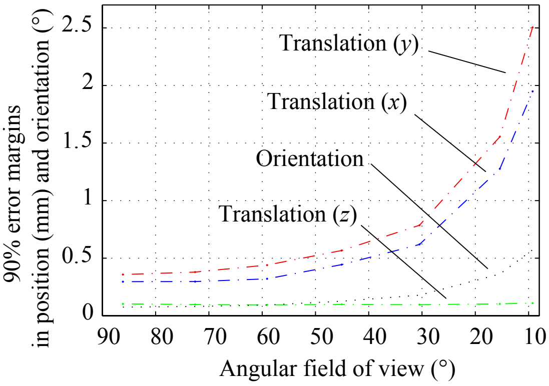


Figure 3.29: Accuracy (90% error margin) in position and orientation estimations w.r.t. the AOV, with camera range 32 cm and perpendicular view to the planar object. The estimation biases are insignificant.

It is interesting to compare this simulation with the results when the camera *is* tilted w.r.t. the calibration plate (Figs. 3.30 and 3.27 (b)). The perspective distortion due to an inclination of 37° is more pronounced because different ranges appear and differently distant parts project in different sizes, which makes the *relative* pose estimation better conditioned. This is so heavily pronounced that the accuracy becomes virtually independent of the actual focal length. Furthermore, the known scaling parameter of the camera along with the known structure of the pattern allow for accurate range estimation, thus *absolute* pose estimation.

Hence it is alarming news that perpendicular views to planar objects are most common both in final applications as well as during camera calibration.

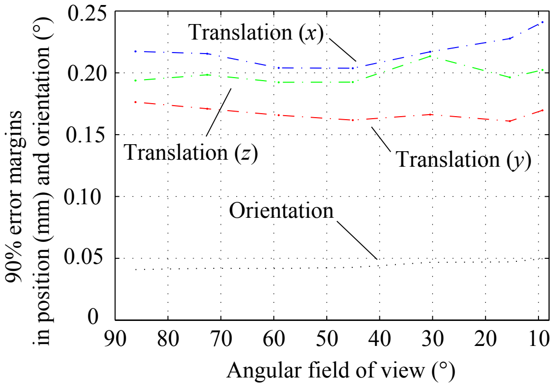


Figure 3.30: Positioning and orientation accuracy (90% error margin) w.r.t. the AOV, with camera range 41 cm and tilted 37° w.r.t. the perpendicular of the planar object. The estimation biases are insignificant.

These results still represent the minimum errors that the user should expect. Image processing errors rarely spread homogeneously in the image nor are clear of outliers, and neither the pinhole camera model nor its parametrization completely hold but in simulations. It is also worth mentioning that, on occasions, camera parametrization inaccuracies are implicitly assumed *within* the imaging noise in normal operation (Lorch *et al.*, 2002).

B. Erroneous camera model parametrization

Camera calibration is the process of estimating the parameters of a camera model that is capable of adequately reflecting the operation of the actual camera at hand, refer to Section 3.2. This section applies the most common algorithms in computer vision for camera calibration for different AOVs (Bouguet, 2002; Strobl *et al.*, 2005). Noteworthy details are the following: The used camera parametrization follows Ref. (Zhang, 2000), and the parameters initialization is also performed by the algorithms detailed in Refs. (Zhang, 2000; Sturm and Maybank, 1999). The algorithm requires a perfectly known calibration plate (Strobl and Hirzinger, 2008, 2011) which confines the user to close- to mid-range imaging. In a nutshell: The camera calibration process boils down to optimally estimating the pinhole camera parameters (mainly the focal length) by numerically minimizing image reprojection errors for several object views. In intrinsic camera calibration, several views are required mainly for the parameters initialization stage, see (Zhang, 2000; Sturm and Maybank, 1999). In extrinsic camera calibration, at least three views (specifically two rotation motions with nonparallel rotation axes) are required, see Section 3.3 and (Strobl and Hirzinger, 2006). In addition, the central limit theorem requests a sufficiently large amount of data—so does statistical optimality.

Both, the principal point location and the distortion parameters, are set to fixed, realistic values and are not being estimated. This is because potential variation of these parameters directly implies a motion of the camera frame;[25]

[25] The translation of the principal point in a pinhole camera model primarily implies shifting the origin of lens distortion (Weng *et al.*, 1992; Willson and Shafer, 1994), secondarily a

in turn, a motion of the camera frame S_{C} implies drifts in the remaining intrinsic parameters—including the focal length. This interplay would cover up the intrinsic weakness that I want to show in this section concerning the interdependence of the focal length estimation and the estimation of the camera poses in the presence of noisy image data and limited AOV. This adoption fixing distortion parameters is realistic since they can—and on occasions even should—be estimated *in advance* of pinhole camera model calibration (Devernay and Faugeras, 2001). Furthermore, lens distortion is scarcely noticeable in narrow AOV camera systems, cf. Fig. 3.27 (a). In addition to this, the ground-truth camera model used in this study also lacks of skewness, and the relative projection scaling α/β is enforced to unity (*i.e.*, $\alpha \triangleq \beta$). In this way, the only remaining camera parameter is the focal length, which is the central parameter of the pinhole camera model after all. These measures support the results on calibration accuracy presented here since they make this study a best case scenario for camera calibration, where fundamental weaknesses for general models are to be clearly identified.

Next the statistical results from 150 intrinsic calibrations for each of the 7 different focal lengths/AOVs are presented. For each calibration, image noise is independently generated for 12 convenient, different calibration images. In Fig. 3.31 the focal length estimation results (in the form of the scaling parameter) are compared to ground-truth. It can be seen that the estimation accuracy of the focal length strongly depends on the AOV of the camera; it worsens for narrow AOVs.

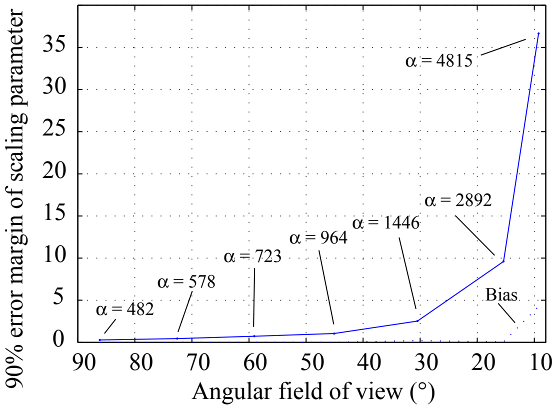


Figure 3.31: Scaling parameter estimation error (90% error margin) w.r.t. the actual scaling parameter α after 150 standard camera calibrations for each AOV.

Similar to the simulation above, the camera poses are also unknown and have to be estimated. In Figs. 3.32 and 3.33 the accuracies of these extrinsic estimations are depicted w.r.t. ground-truth for the two same images treated above, perpendicular and tilted (which are included in the 12 images used for calibration). The positioning accuracy (in this case its range, *i.e.*, $\sqrt{{}_{\mathrm{C}}x^2+{}_{\mathrm{C}}y^2+{}_{\mathrm{C}}z^2}$) worsens for narrower AOVs, similar to the above results in Fig. 3.29. However, in the case of tilted views, the results are very different since now they also suf-

rotation of the camera frame S_{C}, and third a slight displacement of S_{C} (Tsai, 1987).

fer from positioning inaccuracy, cf. Figs. 3.30 and 3.32. This was expected since the (erroneous) focal length is responsible for the absolute scaling of images, refer to Section 3.5.3.

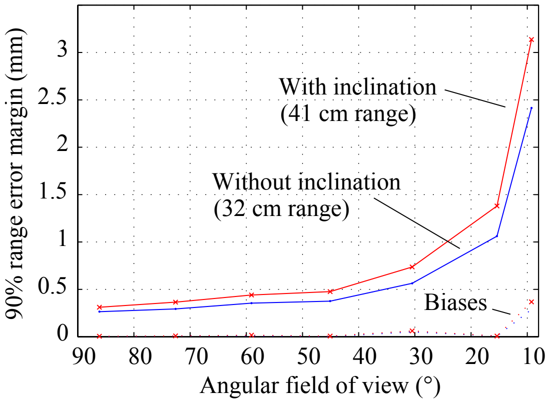


Figure 3.32: Range estimation error (90% error margin) after 150 standard camera calibrations for each AOV (for the images with and without inclination).

In general, two reasons account for the inaccurate estimation of the scaling parameter: On the one hand, a reduction of the AOV (without relocating the camera) implies that a smaller area of the scene will be seen, and therefore that there will be less evidence for accurate estimation—notwithstanding some more precision in the 2-D measurements. As I mentioned in Section 3.5.3, this is not because of any change in the potential perspectivity of the scene, but because of the limited size of the imaging chip. However, the comparison above made it clear that individual tilted images still contain perspectivity evidence for very accurate camera pose estimation. Exactly the same in camera calibration, it is the perspective distortion that differentiates camera range from focal length, and therefore one would expect that camera calibration does a better job in the estimation of the camera pose of tilted images, cf. Fig. 3.32. On the other hand, during the camera calibration process the intrinsic camera parameters are continously being shared between all calibration images. Erroneous pose estimation by certain images (e.g. the perpendicular ones, see Fig. 3.29) will spread to images with sufficient perspectivity information simply because they share the focal length parameter. This point intensifies my conflict with perpendicular images mentioned above, even though perpendicular images may be useful for reliable lens distortion estimation.

Fig. 3.33 shows that the accuracy of estimation of the camera orientation is not affected by concurrent estimation of the focal length (cf. with Figs. 3.29 and 3.30). Fig. 3.34 shows extreme correlation between range and focal length estimations and no correlation between orientation and focal length estimations. This is because the projective effects of camera rotations and focal length adaption are clearly differentiated.

The results in Figs. 3.32 and 3.33 could also help to define a threshold for the proper definition of a potential, subsequent hand-eye calibration as in Section 3.3.

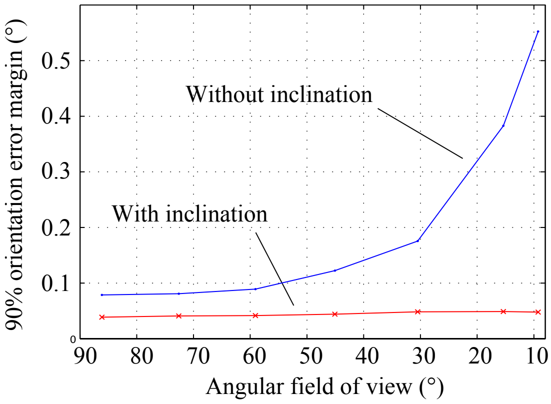


Figure 3.33: Orientation estimation error (90% error margin) after 150 standard camera calibrations for each AOV (for the images with and without inclin.).

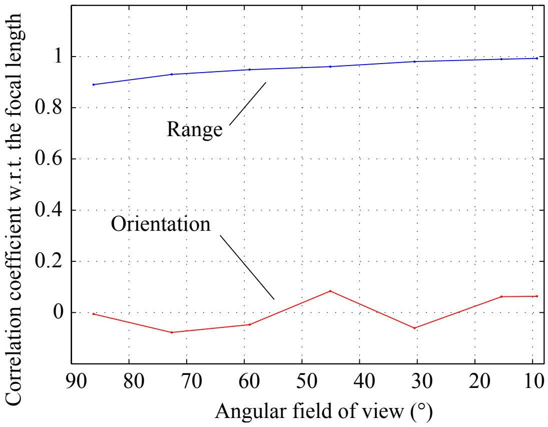


Figure 3.34: Correlation coefficient relating the focal length estimation error with the range and orientation estimation errors for a typical calibration image.

But for all that, it is often not sensible to validate results in relation to the estimation accuracy of particular parameters (for example, one may expect less accuracy in long-range pose estimation than in short-range, which is perfectly normal). In the next subsection the consequences of this issue in *final* camera operation will be shown.

C. Image-based estimation from noisy image processing *and* erroneous models

After each calibration process it is convenient to be able to properly assess the calibration results. The most common practice is to mention the RMS error in reprojection after intrinsic calibration. Whereas this is acceptable for regular cameras with reasonable AOV, proper camera and object models, and valid image processing and optimization processes, this practice is intrinsically wrong. This is because during optimization it is explicitly rewarded to minimize precisely that RMS error at expenses of the model parametrization. Two evils come on scene: erroneous camera parameters and wrongful reprojection residuals, thus wrongful assessment.

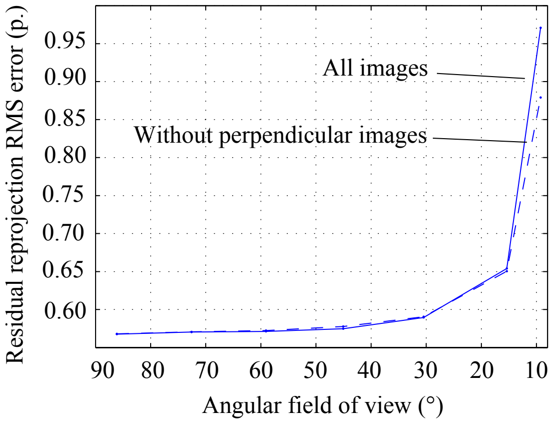


Figure 3.35: Residual reprojection RMS error after standard calibration *and* further erroneous reprojection, for each AOV.

In order to assess the model parametrization of the last section *in final operation*, the following simulation was performed: For *each* of the calibration results from the last section (*i.e.*, 150 intrinsic calibrations for each of the 7 different AOVs), *again* 250 sets of simulated noisy points were generated for all images, on the ground-truth projections *at the ground-truth camera poses*. Only in this way the real parametrization errors emerge—as opposed to the residuals after calibration—since the estimated camera poses are not a valid outcome of the calibration process. For perfect model parametrization, this RMS reprojection error should average the Gaussian image noise $\sqrt{\sigma_x^2+\sigma_y^2} =$ 0.56 pixels, but Fig. 3.35 shows that this only happens for wide AOVs; for narrow AOV the residuals fairly surpass that level.

Fig. 3.35 also shows the accuracy results when the 5 images with vantage angles between the principal axis of the camera and the perpendicular to the calibration plate lower than 15° are omitted from all calibrations. The remaining 7 images do yield slightly more accurate results. However, it is inconvenient to further omit tilted images since image processing performs worse with strong perspective distortion. In general, it is very difficult to further improve in this way.

A comparison between these results and the ones in Section 3.5.4-**A.** could bring light into the question of whether and when is it really appropriate to consider the calibration errors as an extra level of imaging noise during final camera operation.

3.5.5 Proposed Calibration Method: Joint Intrinsic and Hand-Eye Estimations

Inspired by photogrammetric procedures but recasting them in the typical computer vision scenario of robotics, I propose a calibration method that amends the deficiencies shown in the last section; I propose the use of pose readings of e.g. a robotic manipulator in order to support the intrinsic calibration of a camera, being the camera mounted on its end-effector. The method should be used in the case of images showing lack of perspectivity, *i.e.*, narrow AOV cameras, constrained camera placements (e.g. a deficient number of tilted images), or with visually impaired cameras like endoscopes. In addition, by doing so the processes of intrinsic and extrinsic camera calibration merge and former intrinsic inaccuracies do not harm the latter, potential extrinsic (hand-eye) calibration anymore.

The pose readings of the kinematic chain of the manipulator are represented by the rigid body motions ${}_{\mathrm{B}}\widetilde{\boldsymbol{T}}_i^{\mathrm{T}}$ between the **b**ase of the manipulator S_{B} and its end-effector (**T**CP) S_{T} in different instants i / $i\in\mathbb{N}_1$, $i\leq N$. Along with the fixed (yet unknown) object-to-base ${}_0\boldsymbol{T}^{\mathrm{B}}$ and hand-eye ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ transformations, they define the pose of the camera S_{C} in S_0: ${}_0\widehat{\boldsymbol{T}}_i^{\mathrm{C}} = {}_0\widehat{\boldsymbol{T}}^{\mathrm{B}}\,{}_{\mathrm{B}}\widetilde{\boldsymbol{T}}_i^{\mathrm{T}}\,{}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$. The idea suggests itself to directly include these extrinsic transformations in place of the unknown poses of the camera, performing a common minimization of reprojection errors for estimation of the camera parameters as in Eqs. (3.3) and (3.4). Even though this may result in lower RMS error after calibration, simulations as in Section 3.5.4-C. show that this approach *worsens* the accuracy in the estimations, see Fig. 3.36. Similar to the motivation of earlier work in (Strobl and Hirzinger, 2006), I understand that optimal stochastic estimation by residuals minimization can only be achieved if all significant error sources are minimized (viz. according to their statistical distributions). By the inclusion of manipulator readings that are naturally noisy, substantial deviations appear, and these deviations are of similar effect than image noise.

An approach to optimal hand-eye calibration on noisy manipulator readings was previously presented in (Strobl and Hirzinger, 2006) and Section 3.3; it consists in a minimization of transformation errors of a robotic manipulator. Translational and rotational errors ($\mathcal{O}^{\mathrm{tra}}$ and $\mathcal{O}^{\mathrm{rot}}$) are considered separately, but are minimized at the same time in relation to the precision characteristics of the pose tracking system. Here I extend this formulation for simultaneous intrinsic and extrinsic camera calibration by including reprojection errors in the minimization; furthermore, the algorithm is able to automatically adapt its weighting factors ${}^\star\sigma_{\mathrm{x|y}}$, ${}^\star\sigma_{\mathrm{rot}}$, and ${}^\star\sigma_{\mathrm{tra}}$ according to the precision characteristics of the system iteratively, see Section 3.3.4. The extended optimization problem now reads:

$$\left\{{}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}},\, {}_{\mathrm{B}}\boldsymbol{T}^{0},\, \alpha\right\}^\star = \arg\min_{{}_{\mathrm{T}}T^{\mathrm{C}},\,{}_{\mathrm{B}}T^{0},\,\alpha}\left(\sum_{i=1}^{N}\frac{(\mathcal{O}_i^{\mathrm{im}})^2}{{}^\star\sigma^2_{\mathrm{x|y}}}+\frac{(\mathcal{O}_i^{\mathrm{rot}})^2}{{}^\star\sigma^2_{\mathrm{rot}}}+\frac{(\mathcal{O}_i^{\mathrm{tra}})^2}{{}^\star\sigma^2_{\mathrm{tra}}}\right) \tag{3.33}$$

where $\mathcal{O}_i^{\mathrm{im}}=\sum_{p=1}^{n_i}({}^i_p\Delta^2_{\mathrm{x}}+{}^i_p\Delta^2_{\mathrm{y}})$ accumulates the n_i square reprojection residuals $\Delta^2_{\mathrm{x}}+\Delta^2_{\mathrm{y}}$ in image i.

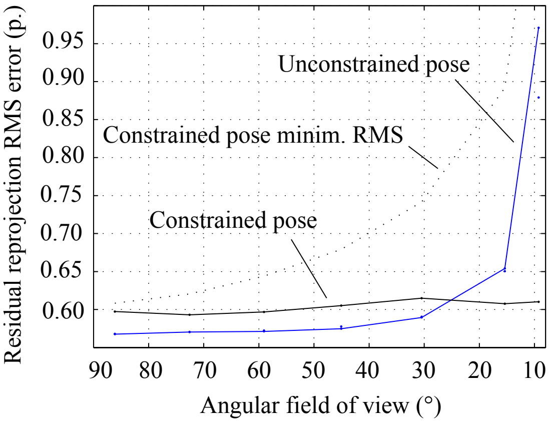


Figure 3.36: Residual reprojection RMS error after calibration *and* further erroneous reprojection for intrinsic calibration supported by the robotic manipulator. The standard intrinsic calibration results of Fig. 3.35 are overlaid.

Next the analogous simulation to Section 3.5.4-C. is performed. In addition, noisy manipulator poses were generated over ground-truth manipulator poses inspired by real calibration scenarios. The error was added to the pose of the end-effector[26] as follows: The angles θ of the angle-axis representation $\{\theta, \boldsymbol{p}\}$ of the added rotational errors follow a zero-mean Gaussian distribution with $\sigma_\theta = 0.1°$ and their axes $\boldsymbol{p}$ are uniformly distributed, *i.e.*, their azimuth and elevation angles ϕ and ψ are $\phi \in [-90°, 90°)$ according to the **p**robability **d**ensity **f**unction $\mathrm{pdf}(\phi) = 180^{-1}\ [°]^{-1}$ and $\psi \in [-90°, 90°)$ with $\mathrm{pdf}(\psi) \propto \sin(\psi/90)\ [°]^{-1}$. The translational errors $\boldsymbol{t}$ also follow a zero-mean Gaussian distribution in range with $\sigma_t = 0.5$ mm and the directions are again uniformly distributed. These relative precision levels are conservative and are readily surpassed by most commercial robotic manipulators.

In Fig. 3.36 the results of the intrinsic calibration aided by the robotic manipulator (*constrained pose*) are superimposed on the former results of the standard intrinsic approach in Fig. 3.35 (*unconstrained pose*). **The proposed constrained approach is insensitive to the narrowness of the AOV** but reaches slightly worse intrinsic accuracy than optimal due to the noise in the manipulator readings. This very low error level seems preferable to the dangers of using affine camera models. The figure also shows the level of narrowness at which this method should be preferred to standard perspective camera calibration ($\sim 25°$). For bigger AOV this positioning aid should not be used. The figure also shows the accuracy reached by minimizing only RMS reprojection errors, *i.e.*, not considering errors in the manipulator readings; the algorithm performs even worse than standard intrinsic camera calibration at these positioning accuracy levels.

[26] Translational residual errors at the end-effector S_T are more realistic than at the base S_B because of the rigid base of manipulators; note that, for generality, the minimization algorithm assumes translational errors both at S_B and at S_T. Orientational errors in S_B and in S_T are, of course, equivalent.

3.5.6 Summary

This section 3.5 considers the issue of camera calibration for computer vision applications with the particularity of narrow angular fields of view. I reveal deficiencies not in the validity of the pinhole camera model, but in the ability of standard camera calibration algorithms to accurately parameterize it. Narrow angular fields of view make it difficult to obtain the required evidence on perspectivity in images; this compromises algorithms that rely on this evidence and furthermore consider several images at the same time, like camera calibration.

I start out with an overview on camera calibration for computer vision applications. This justifies clearly why there is need to address this problem. Crucially, a significant number of major application areas are listed. Next, I descriptively explain the roles of focal length and camera pose for the achievement of perspectivity richness in images. I also demonstrate the consequences of critical evidence on perspectivity for exemplary computer vision applications as well as for standard camera calibration, and I lay emphasis on the detrimental effects for image-based estimation of images taken perpendicular to planar objects.

Since perspective distortion is primarily defined by the pose of the camera, it will be difficult for any algorithm to accurately discern pose on the basis of insufficient evidence on perspectivity; the same holds for pose *and* focal length estimation, *i.e.*, standard camera calibration. For this reason I propose an alternative method that uses positioning information from a robotic manipulator in order to support intrinsic camera calibration. Experiments show that the direct insertion of this extrinsic information in the optimization problem, still by minimizing reprojection residuals only, does not support intrinsic camera calibration but compromises it. This is due to the naturally noisy readings of the robotic manipulator. I introduce a novel method that optimizes the intrinsic and extrinsic parameters by the minimization of a *hybrid residual term*; it consists of translational and rotational errors in the kinematic transformation of the robot as well as image reprojection errors. This method extends my former work on accurate hand-eye calibration in (Strobl and Hirzinger, 2006).

Concluding, accuracy assessments compare this formulation with current intrinsic camera calibration approaches, and prove its better performance for narrow angular fields of view.

This section was adapted from the original publication in (Strobl *et al.*, 2009b).

3.6 Calibration of the DLR Laser Stripe Profiler

3.6.1 Introduction

As introduced in Section 2.2.2, the DLR Laser Stripe Profiler (LSP) is a hybrid sensor halfway between the stereo camera introduced in Section 2.2.1 and the pure range sensor DLR Laser Range Scanner (LRS) in Section 2.2.3. As explained in Section 2.2.2, I opt to explicitly keep the camera model in Section 2.2.1 within the model of the LSP; this detailed layout of the geometry of the sensor will help us to increase accuracy during calibration as well as to simplify operation in Section 4.3.

As a consequence of the abovementioned, the calibration of the LSP bases on the calibration of the stereo camera as explained in Section 3.2—the latter calibration remains, however, unaltered as it is not influenced by the results of the LSP calibration. Having the stereo camera accurately calibrated, the only remaining parameters to fully parameterize the LSP model in Eq. (2.36) is the pose of the laser plane w.r.t. the camera reference frame S_C in Eq. (2.35), which solely features 3 DoF.

3.6.2 State of the Art

This type of structured light sensors have been historically calibrated using precision calibration targets, or rather the precise positioning of these targets w.r.t. the imaging camera (Chen and Kak, 1987; Khadraoui *et al.*, 1996; DePiero and Trivedi, 1996; Reid, 1996). If the scene structure is precisely known, it is easy to identify these world points with their projected counterpart points on the image, and then, out of these correspondences, reconstruct the relative pose of the laser plane. Nonetheless, it is expensive to build such reference artifacts and, what is more, potential errors committed during their construction cannot be eventually considered neither in the LSP calibration results nor in eventual measurement error estimations. Similar limitations apply in the case of precisely tracked calibration targets (in pose).

Other approaches rely on known features within calibration objects (McIvor, 1999; Wang *et al.*, 2001); the calibration object has to be shifted w.r.t. the sensor (McIvor, 2002). Neither fiducial marks because of their problematic perspective projection, nor corner features because these are a chronic problem for triangulation based on structured light projectors (Curless and Levoy, 1995), seem to be appropriate for a calibration stage. The use of laser peak-detection algorithms to calculate illuminated coordinates is more convenient (Trucco *et al.*, 1998). Alternatively, other approaches use stereo vision for locating the stripes in the scene (Taylor *et al.*, 2002); these may not suffer from the abovementioned problems, but they do strongly rely on the accuracy of both camera calibrations and their relative calibration obtained in advance.

One option in order to avoid very precise calibration objects and calibration methods is to include a calibration refining method after an inaccurate attempt of the former methods. Jokinen in (Jokinen, 1999) brings forward a formulation to refine inaccurate calibrations by matching multiple 3-D profile maps.

Another option to avoid expensive and inconvenient calibration procedures is to devise a novel calibration algorithm that works without precise calibration objects or precise pose tracking in the first place. Inspired by the self-calibration method in (Jokinen, 1999), in (Strobl *et al.*, 2004) I proposed a novel method for laser plane self-calibration based on the assessment of the deformations due to miscalibration of the laser plane. The proposed method allows for rapid, accurate calibration of the laser plane on the sole basis of an unknown, planar surface.

3.6.3 Laser Plane Calibration

Laser plane calibration is the process of determining the relative pose of the laser plane w.r.t. S_{C} (or S_{T}). Eventual operation of the LSP in Section 4.3 is extremely sensitive to miscalibration leading to misalignments and warpage effects in scanned surfaces.

I propose a novel method for laser plane self-calibration based on the assessment of the deformations the miscalibration leads to. In (Jokinen, 1999), Jokinen's calibration approach focuses on matching maps, searching for shape correspondences—the method is based on previous research in registration algorithms. Here I propose a method that focuses on *correcting* the resulting maps, rather than on *matching* them, reaching in this way a much simpler and swifter formulation. Moreover, this method does not require any complicated calibration target but a planar surface.

When locating the laser plane, *i.e.*, estimating the parameters ${}_{\mathrm{T}}\boldsymbol{n}$ and ${}_{\mathrm{T}}d$ of the Hessian normal form of the plane in Eq. (2.35), there are three *independent* DoF to be identified. Here the spherical-polar coordinates in S_{C} are used for orientation: the angles roll ${}_{\mathrm{C}}\alpha$ and pitch ${}_{\mathrm{C}}\beta$, and there is the minimum distance ${}_{\mathrm{C}}d$ between the laser plane and S_{C}. I bring these three parameters together in ${}_{\mathrm{C}}\boldsymbol{\Omega} = \{{}_{\mathrm{C}}\alpha,\ {}_{\mathrm{C}}\beta,\ {}_{\mathrm{C}}d\}$. For any roughly estimated $(\widehat{\ })$ laser plane pose ${}_{\mathrm{C}}\widehat{\boldsymbol{\Omega}}$, estimation errors ${}_{\mathrm{C}}\varepsilon_{\alpha}$, ${}_{\mathrm{C}}\varepsilon_{\beta}$, and ${}_{\mathrm{C}}\varepsilon_{d}$ occur. These errors eventually cause deformations in the estimated surface for every scanning motion of the hand-guided DLR 3D-Modeler. These deformations range from simple scaling errors (typically when having high ${}_{\mathrm{C}}\varepsilon_{d}$) to convex/concave, warped deformations (typically when having high ${}_{\mathrm{C}}\varepsilon_{\beta}$, cf. Fig. 3.37), or even irregularly warped results (both with high ${}_{\mathrm{C}}\varepsilon_{\alpha}$ and with a mixture of them all).

The proposed self-calibration method is as follows: The reconstruction process in Section 4.3 runs with some a priori laser plane calibration parameters ${}_{\mathrm{C}}\widehat{\boldsymbol{\Omega}}_{\mathrm{initial}}$, which have been roughly estimated in advance. The proposed method exploits the distortions caused in the reconstruction process when scanning surfaces. As calibration surface I use *a plane of unknown pose*, both in order to avoid the construction of a complex calibration target and due to the fact that a plane has the geometrical shape that can be straightened out in the easiest way. In the process, all N measured $(\widetilde{\ })$ sensor poses ${}_{0}\widetilde{\boldsymbol{T}}^{\mathrm{T}_i}$, $\forall i \in \mathbb{N}_1,\ i \leq N$, are stored in Υ, and two image points nearby the ends of every stripes ${}_{\mathrm{M}}\boldsymbol{p}_i^{\{\mathrm{left,right}\}}$ are stored in Φ for every image i. Both, poses Υ and image projections Φ, yield $2N$ 3-D points ${}_{0}\hat{\boldsymbol{p}}_i^{\{\mathrm{left,right}\}}$ in the world frame S_0 that also depend on the pose ${}_{\mathrm{C}}\widehat{\boldsymbol{\Omega}}$

of the laser plane w.r.t. S_{C} following Eq. (2.37). The reconstructed pointcloud shows unevenness when scanning from very different points of view. Subsequently, the best fitting calibration target plane for these points ${}_{0}\hat{\boldsymbol{p}}_i^{\{\text{left,right}\}}$ is estimated in closed form by Singular Value Decomposition (SVD) because it is an overdetermined problem. The calibration plane can be parameterized by its normal vector ${}_{0}\hat{\boldsymbol{n}}_{\angle}$ w.r.t. the origin of S_0 as well as its minimum distance ${}_{0}\hat{d}_{\angle}$ to it. Finally, optimal ($\star$) laser plane parameters ${}_{\mathrm{C}}\widehat{\boldsymbol{\Omega}}_{\star}$ are estimated off-line by optimization: the goal is to minimize the mean squared distance $\sigma_{\angle}^2$ of every 3-D reconstructed point ${}_{0}\hat{\boldsymbol{p}}_i^{\{\text{left,right}\}}$ to the best fitting calibration plane. This can be mathematically expressed as:

$$ {}_{\mathrm{C}}\widehat{\boldsymbol{\Omega}}_{\star} = \arg\min_{{}_{\mathrm{C}}\widehat{\boldsymbol{\Omega}}} \sigma_{\angle}^2 \left(\Upsilon,\, \Phi,\, {}_{\mathrm{C}}\widehat{\boldsymbol{\Omega}} \right) \quad , \tag{3.34} $$

$$ \sigma_{\angle}^2 = \sum_{i=1}^{2N} \left({}_{0}\hat{d}_{\angle} - {}_{0}\hat{\boldsymbol{n}}_{\angle}^{\top}\, {}_{0}\hat{\boldsymbol{p}}_i^{\{\text{left,right}\}} \right)^2 . \tag{3.35} $$

The Nelder-Mead Simplex method for numerical optimization was chosen (Nelder and Mead, 1965). To recapitulate, the laser plane parameters ${}_{\mathrm{C}}\widehat{\boldsymbol{\Omega}}$ are adapted in such a way that, in the end, the scanned surface becomes as flat as possible, see Fig. 3.37.

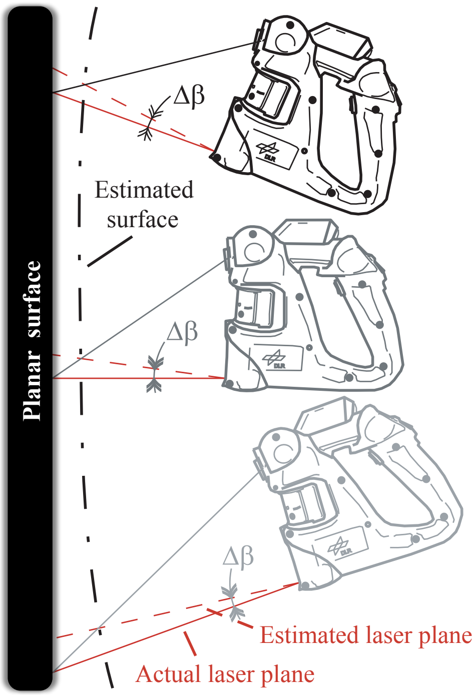


Figure 3.37: Reconstruction consequences of miscalibration of the laser pitch orientation.

Fig. 3.38 shows the 2-D cost matrix representing, in the ordinate axis, the achieved flatness $\sigma^2_\angle$ of the reconstructed plane, and in the abscissas axes different laser plane orientations ${}_{\mathrm{C}}\alpha$ and ${}_{\mathrm{C}}\beta$ (the third parameter of ${}_{\mathrm{C}}\boldsymbol{\Omega}$, the minimum distance ${}_{\mathrm{C}}d$ of the laser plane w.r.t. S_{T}, has been fixed to its optimum value ${}_{\mathrm{C}}d_{\mathrm{c}}$ for the sake of clarity). Indeed, ${}_{\mathrm{C}}\boldsymbol{\Omega}_{\mathrm{c}} = \{{}_{\mathrm{C}}\alpha_{\mathrm{c}},\, {}_{\mathrm{C}}\beta_{\mathrm{c}},\, {}_{\mathrm{C}}d_{\mathrm{c}}\}$ are the a$\underline{\mathrm{c}}$tual laser pose parameters. The figure shows the robustness of the method for any reasonable ${}_{\mathrm{C}}\widehat{\boldsymbol{\Omega}}_{\mathrm{initial}}$, provided the laser plane intersects the image rays in the camera view direction. It is worth noting that for distances ${}_{\mathrm{C}}\hat{d} \neq {}_{\mathrm{C}}d_{\mathrm{c}}$ the optimal orientation parameters ${}_{\mathrm{C}}\hat{\alpha}_\star$ and ${}_{\mathrm{C}}\hat{\beta}_\star$ do vary slightly from the actual ones ${}_{\mathrm{C}}\alpha_{\mathrm{c}}$ and ${}_{\mathrm{C}}\beta_{\mathrm{c}}$ (particularly ${}_{\mathrm{C}}\hat{\beta}_\star$). This is due to the fact that a variation in ${}_{\mathrm{C}}\hat{\beta}_\star$ compensates for an erroneous ${}_{\mathrm{C}}\hat{d}$ e.g. when scanning with constant orientation and distance of the profiler w.r.t. the calibration plane. Owing to the very different poses made during the calibration process, this does not yield any problem for the optimization algorithm, and this compensation mechanism does not get the flatness (lower $\sigma^2_\angle$) that the actual parameters do.

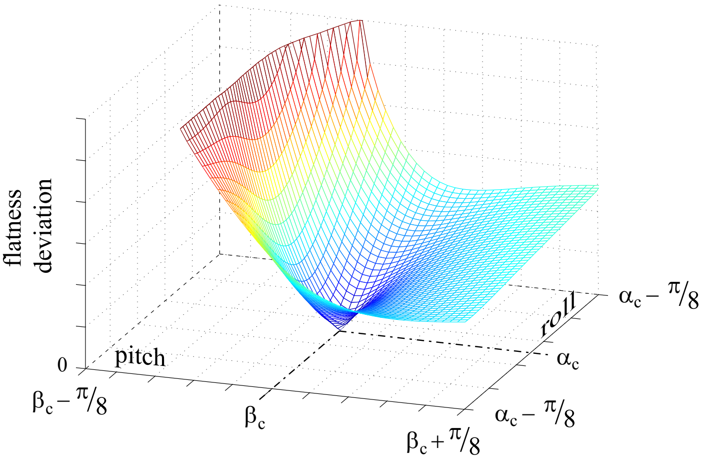


Figure 3.38: Flatness deviation of the reconstructed plane with ${}_{\mathrm{C}}d = {}_{\mathrm{C}}d_{\mathrm{c}}$.

3.6.4 Experiments

In order to show the importance of the accuracy of the laser plane calibration process, as well as to get an idea of the actual precision after calibrating the laser plane with the method presented in Section 3.6.3, I next present an typical scanning result that clearly shows reconstruction errors by miscalibrated systems.

Fig. 3.39 shows two pointclouds focused orthogonally to one of the corners of a small cardboard box. Each pointcloud consists of three scans over the cardboard box: the first time the cardbox was scanned orthogonally to its top (1), the second time orthogonally to a side (2), and the third time orthogonally to the corner edge and in the direction of the bisector of the angle formed between these two sides (3). This particular procedure facilitates the assessment of the reconstruction errors if represented orthogonally to the corner of the cardbox.

Fig. 3.39 (a) shows the profile under correct calibration results following the calibration procedure in Section 3.6.3. The profiler exhibits a precision in submillimeter domain. Fig. 3.39 (b) shows the effect of slightly modified laser plane parameters to the latter—the laser plane pose has been given erroneously with $\varepsilon_\beta = 1°$. Misalignments and warpage appear for this small calibration error. This result gives an idea of both, the achieved calibration accuracy and its precision.

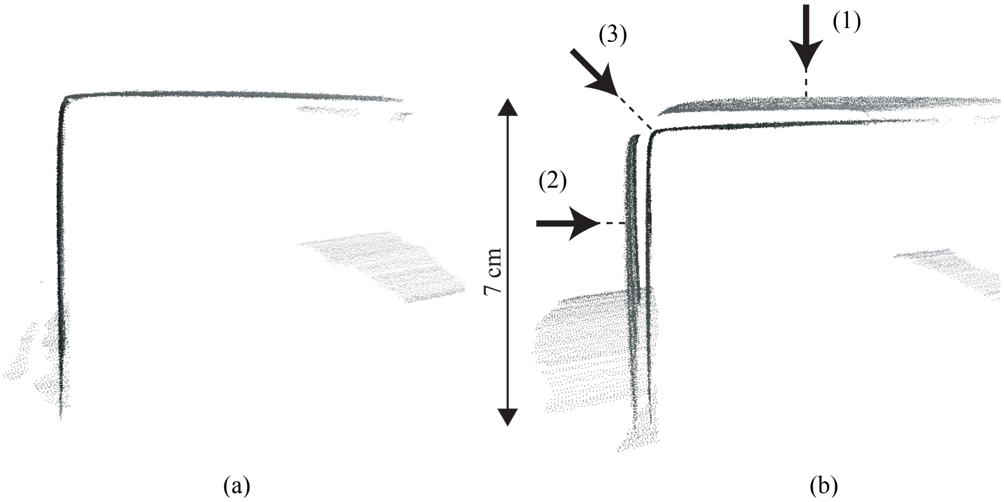


Figure 3.39: Two sides of a cardboard box scanned with optimized (a) and slightly erroneous (b) laser plane calibration parameters ${}_C\boldsymbol{\Omega}$.

3.6.5 Summary

In this section 3.6 I propose a novel calibration method for the DLR Laser Stripe Profiler (LSP). The proposed method takes place after regular intrinsic and extrinsic calibration of the stereo camera of the DLR 3D-Modeler as explained in Sections 3.2 and 3.3. Consequently, only 3 DoF of the pose of the laser plane w.r.t. the stereo camera remain to be estimated.

Instead of using precision calibration targets of precise positioning of the calibration target and the laser plane, I opt for a self-calibration approach that proceeds by correcting deformations caused by miscalibration of the laser plane. The chosen calibration object is a planar surface of unknown pose. The procedure is highly unlabored and yields very accurate results: it consists in scanning the planar surface with adequate scan motions and refining the parameters of the laser plane by flattening the resulting 3-D pointcloud representing the calibration plane.

This section was adapted from the original publication in (Strobl *et al.*, 2004).

3.7 Calibration of the DLR Laser Range Scanner

3.7.1 Introduction

As stated in Section 2.2.3, the DLR Laser Range Scanner (LRS) sensor is similar to the above-addressed LSP sensor. On second sight, however, the sensor features particularities that affect both, their calibration process as well as the geometric information that eventually will be obtained during operation in Section 4.4.

In this work I leave the intrinsic calibration of the inner components of the LRS aside; in this respect the reader can refer to original work in (Hacker *et al.*, 1997; Kielhöfer, 2003).

3.7.2 Calibration of the Origin of the Rotatory Laser Beam

In line with the rationale in the last section 3.6.3, I categorically refuse to use precision targets for the extrinsic calibration of the LRS. In (Suppa and Hirzinger, 2004) the authors use a small sphere of known radius to calibrate the LRS; even though the method did deliver accurate results, it was decided to implement a novel method inspired by the above method in Section 3.6.3 for the sake of convenience.

The same calibration procedure explained in the last section holds: a planar surface is scanned with adequate scan motions and the model parameters of the LRS are refined by flattening the resulting 3-D pointcloud representing the calibration plane. A crucial difference is, however, the number of parameters required to parameterize the model of the LRS. As explained in Section 3.6.3, the calibration of the LSP was supported by the previous calibration of the stereo camera in Section 3.2. As a consequence, the reference frame for the LSP model could be fixed at the camera reference frame, *i.e.*, $S_{\mathrm{LSP}} \triangleq S_{\mathrm{C}}$, so that the sole remaining parameters to be estimated are the 3 DoF of the pose of the laser plane w.r.t. S_{C}. In the case of the LRS, however, the stereo camera does *not* form part of the range sensor and cannot be used. Instead, the LRS features its own reference frame S_{LRS}, see Fig. 3.40. Consequently, 6 DoF of the *pose* of the LRS (instead of 3 DoF previously) have to be estimated out of the very same calibration data as in Section 3.6.3.

As a consequence of the abovementioned extension of sensor parameters to be estimated during optimization, the success of the calibration procedure of the LRS becomes more sensitive to the completeness of the dataset than in the case of the LSP. In the case of complete datasets deaturing all distinct motions explained in Section 3.6.3, the original optimization equations in Eqs. (3.34) and (3.35) can be directly adopted. The 6 DoF of the optimal ($_\star$) pose ${}_{\mathrm{LRS}}\widehat{\boldsymbol{\Omega}}_\star = {}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{LRS}}$ of the LRS w.r.t. the TCP reference frame S_{T} is estimated as follows:

$$ {}_{\mathrm{LRS}}\widehat{\boldsymbol{\Omega}}_\star = \arg\min_{{}_{\mathrm{LRS}}\widehat{\boldsymbol{\Omega}}} \sigma^2_\angle \left(\Upsilon,\, \Psi,\, {}_{\mathrm{LRS}}\widehat{\boldsymbol{\Omega}}\right) \tag{3.36} $$

where

$$ \sigma^2_\angle = \sum_{i=1}^{2N} \left({}_0\hat{d}_\angle - {}_0\hat{\boldsymbol{n}}_\angle^\top \, {}_0\hat{\boldsymbol{p}}_i^{\{\mathrm{left,right}\}}\right)^2 . \tag{3.37} $$

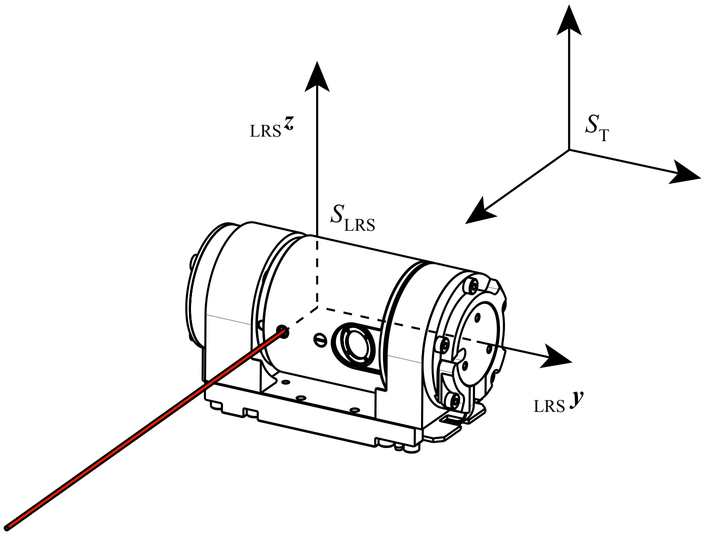


Figure 3.40: The reference frame of the LRS S_{LRS} and the reference frame of the TCP S_{T}.

The 3-D data ${}_{0}\hat{\boldsymbol{p}}_i^{\{\text{left,right}\}}$ stem from 3-D range measurements ${}_{\text{LRS}}\tilde{\boldsymbol{p}}_i^{\{\text{left,right}\}}$ in the LRS reference frame S_{LRS}, viz. two measurements within every ith measurement cycle, along with the rigid body transformations ${}_{0}\widetilde{\boldsymbol{T}}^{\text{T}}$ and ${}_{\text{T}}\widehat{\boldsymbol{T}}^{\text{LRS}}$ so that ${}_{0}\hat{\boldsymbol{p}}_i^{\{\text{left,right}\}} = {}_{0}\widetilde{\boldsymbol{T}}^{\text{T}}\,{}_{\text{T}}\widehat{\boldsymbol{T}}^{\text{LRS}}\,{}_{\text{LRS}}\tilde{\boldsymbol{p}}_i^{\{\text{left,right}\}}$. The parameter Ψ in Eq. (3.36) comprises all these local range measurements ${}_{\text{LRS}}\tilde{\boldsymbol{p}}_i^{\{\text{left,right}\}}$ $\forall i \in \mathbb{N}_1,\ i \leq N$. The parameter Υ gathers all N measured $(\widetilde{\ })$ sensor poses ${}_{0}\widetilde{\boldsymbol{T}}^{\text{T}_i}$, $\forall i \in \mathbb{N}_1,\ i \leq N$. $\sigma_{\angle}$ represents the mean squaared fistance of every 3-D reconstructed point ${}_{0}\hat{\boldsymbol{p}}_i^{\{\text{left,right}\}}$ to the best fitting calibration plane.

In cases of incomplete datasets where one or more required motions could not be performed, the optimization might not be well-defined due to the higher number of DoF to be estimated. The following optimization constraints the problem promoting its robustness:

$$ {}_{\text{LRS}}\widehat{\boldsymbol{\Omega}}_{\star} = \arg \min_{{}_{\text{LRS}}\widehat{\boldsymbol{\Omega}}} \sigma_{\perp}^2 \left(\Upsilon,\ \Psi,\ {}_{\text{LRS}}\widehat{\boldsymbol{\Omega}} \right) \tag{3.38} $$

where

$$ \sigma_{\perp}^2 = \sum_{i=1}^{2N} \left({}_{0}\hat{d}_{\perp} - {}_{0}\boldsymbol{z}^{\mathsf{T}}\ {}_{0}\hat{\boldsymbol{p}}_i^{\{\text{left,right}\}} \right)^2 . \tag{3.39} $$

Here it is not the flatness of the resulting pointcloud that is being minimized but $\sigma_{\perp}$ represents the height variation in ${}_{0}\boldsymbol{z}$ of the resulting pointcloud, *i.e.*, the former pointcloud flatness $\sigma_{\angle}$ when ${}_{0}\hat{\boldsymbol{n}}_{\angle} \triangleq {}_{0}\boldsymbol{z}$. This is on the condition that the calibration target is horizontal w.r.t. S_0, which now conveys precise extrinsic information of the calibration scene. This fact is, however, easy to achieve as the user can readily make sure that the calibration target is level in S_0.

3.7.3 Summary

In this section 3.7 I present a novel method for calibration of the DLR Laser Range Scanner (LRS) that was introduced in order to simplify the calibration procedure compared to the former method in (Suppa and Hirzinger, 2004). The proposed method is in line with Section 3.6 but has been adapted in accordance with the nature of the problem:

- Unlike in the case of the LSP, the stereo camera does not take part of the LRS sensor. Consequently, the location of the LRS reference frame S_{LRS} (3 DoF) has to be estimated along with the orientation of the LRS (3 Dof). The total number of 6 DoF compromises optimal estimation if the dataset is deficient. The modification of the residual function in Eqs. (3.37) and (3.39) provides a more robust formulation to this end.

- The LRS provides full 3-D range measurements instead of mere 2-D directions in the case of the uncalibrated LSP. Consequently, local measurements ${}_{\mathrm{LRS}}\tilde{\boldsymbol{p}}_i^{\{\mathrm{left,right}\}}$ do not require laser plane triangulation on estimated laser plane parameters, which in turn further robustifies my approach.

The proposed self-calibration method proceeds by correcting deformations caused by miscalibration of the 6 DoF pose of the LRS. The chosen calibration object is a planar surface of unknown pose; if Eqs. (3.38) and (3.39) are used, the planar surface has to be leveled w.r.t. the ${}_0\boldsymbol{x}-{}_0\boldsymbol{y}$ plane of S_0. The procedure is highly unlabored and yields very accurate results: it consists in scanning the planar surface with adequate scan motions and refining the pose parameters of the LRS by flattening the resulting 3-D pointcloud representing the calibration plane.

3.8 Calibration of the Inertial Measurement Unit

The inertial measurement unit (IMU) is a complex sensor with numerous components, viz. linear accelerometers, gyroscopes, and magnetometers—refer to Section 2.2.4.

The intrinsic calibration of the geometry of IMUs is normally performed at the manufacturing company. Additionally, a valid model of their biases and noises is provided. More often than not IMUs are packaged with electronics that autonomously compensate for that biases e.g. on the basis of the magnetometer readings. In our case, the DLR 3D-Modeler features the AscTec AutoPilot IMU; its manufacturer Ascending Technologies GmbH already provided an accurate intrinsic calibration as well as electronic compensation of intrinsic biases.

The only remaining parameters for direct usage of the IMU readings are the extrinsic pose of the IMU reference frame S_{IMU} w.r.t. the TCP reference frame S_{T}. As explained in Section 2.2.4, visual pose tracking in Chapter 5 will only make use of differential rotational readings of the IMU, so that the only required transformation is the relative orientation ${}_{\mathrm{C}}\boldsymbol{R}^{\mathrm{IMU}}$ betwenn S_{IMU} and S_{C}. Luckily, it is easy to rigidly mount the IMU aligned w.r.t. the TCP frame S_{T} so that

$${}_{\mathrm{T}}\widehat{\boldsymbol{R}}^{\mathrm{IMU}} = \boldsymbol{I}(3) \;\Rightarrow\; {}_{\mathrm{C}}\widehat{\boldsymbol{R}}^{\mathrm{IMU}} = {}_{\mathrm{C}}\widehat{\boldsymbol{R}}^{\mathrm{T}} \tag{3.40}$$

holds and can be eventually substituted in Eq. (2.50).

It is worth noting that the fact that the pose tracking algorithm following Section 5.4.2 does not require any translational readings of the IMU is hugely useful to decrease the calibration complexity of the whole DLR 3D-Modeler and avoid human calibration mistakes at that, as explained in Section 1.5.2. A parallel investigation by my colleagues in (Fleps *et al.*, 2011) clearly illustrates the inconvenience of the translational extrinsic calibration of IMUs.

3.9 Extrinsic Recalibration of Sensor Components

The statement in this section is perhaps an obvious one to the attentive reader. It is, however, an extremely useful one in the context of the calibration of the component sensors of the DLR 3D-Modeler, hence may I lay stress on the following.

In Section 2.3 a series of pose tracking devices has been listed. Some pose tracking devices are more convenient to particular applications than others. In the end, the optimal pose tracking device to be attached to the DLR 3D-Modeler largely becomes an application-dependent decision. Being the DLR 3D-Modeler a device with a multitude of sensors, the estimated ($\widehat{\ }$) transformations of all these sensors w.r.t. the TCP of the chosen tracking system (e.g. ${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$, ${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{LSP}}$, ${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{LRS}}$, ${}_{\mathrm{T}}\widehat{\boldsymbol{T}}^{\mathrm{IMU}}$, etc.) are all required for dutiful representation of 3-D data e.g. in the world reference frame S_0.

It is worth noting that, if the sensor components of the DLR 3D-Modeler were already extrinsically calibrated w.r.t. a particular pose tracking device #1, the following transformations ${}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$, ${}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{LSP}}$, ${}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{LRS}}$, and ${}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{IMU}}$ w.r.t. its TCP frame S_{T1} are indeed accurately known; it then holds:

$$\begin{aligned}
{}_0\overline{\boldsymbol{p}\mathbf{1}} &= {}_0\widetilde{\boldsymbol{T}}^{\mathrm{T1}}\,{}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{C}}\qquad {}_{\mathrm{C}}\overline{\boldsymbol{p}\mathbf{1}}\\
{}_0\overline{\boldsymbol{p}\mathbf{2}} &= {}_0\widetilde{\boldsymbol{T}}^{\mathrm{T1}}\,{}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{LSP}}\,{}_{\mathrm{LSP}}\overline{\boldsymbol{p}\mathbf{2}}\\
{}_0\overline{\boldsymbol{p}\mathbf{3}} &= {}_0\widetilde{\boldsymbol{T}}^{\mathrm{T1}}\,{}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{LRS}}\,{}_{\mathrm{LRS}}\overline{\boldsymbol{p}\mathbf{3}}\\
{}_0\overline{\boldsymbol{p}\mathbf{4}} &= {}_0\widetilde{\boldsymbol{T}}^{\mathrm{T1}}\,{}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{IMU}}\,{}_{\mathrm{IMU}}\overline{\boldsymbol{p}\mathbf{4}}
\end{aligned}\tag{3.41}$$

where ${}_0\widetilde{\boldsymbol{T}}^{\mathrm{T1}}$ are pose readings from the acquainted pose tracking device #1 and ${}_{\mathrm{C}}\boldsymbol{p}\mathbf{1}$, ${}_{\mathrm{LSP}}\boldsymbol{p}\mathbf{2}$, ${}_{\mathrm{LRS}}\boldsymbol{p}\mathbf{3}$, and ${}_{\mathrm{IMU}}\boldsymbol{p}\mathbf{4}$ are local range data in their respective reference frames, in homogeneous ($\bar{\ }$) coordinates.

In the case of a new pose tracking device #2 with TCP reference frame S_{T2} that was not calibrated w.r.t. the sensor components of the DLR 3D-Modeler, it similarly holds:

$$\begin{aligned}
{}_0\overline{\boldsymbol{p}\mathbf{1}} &= {}_0\widetilde{\boldsymbol{T}}^{\mathrm{T2}}\,{}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{C}}\qquad {}_{\mathrm{C}}\overline{\boldsymbol{p}\mathbf{1}}\\
{}_0\overline{\boldsymbol{p}\mathbf{2}} &= {}_0\widetilde{\boldsymbol{T}}^{\mathrm{T2}}\,{}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{LSP}}\,{}_{\mathrm{LSP}}\overline{\boldsymbol{p}\mathbf{2}}\\
{}_0\overline{\boldsymbol{p}\mathbf{3}} &= {}_0\widetilde{\boldsymbol{T}}^{\mathrm{T2}}\,{}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{LRS}}\,{}_{\mathrm{LRS}}\overline{\boldsymbol{p}\mathbf{3}}\\
{}_0\overline{\boldsymbol{p}\mathbf{4}} &= {}_0\widetilde{\boldsymbol{T}}^{\mathrm{T2}}\,{}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{IMU}}\,{}_{\mathrm{IMU}}\overline{\boldsymbol{p}\mathbf{4}}
\end{aligned}\tag{3.42}$$

where ${}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$, ${}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{LSP}}$, ${}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{LRS}}$, and ${}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{IMU}}$ are unknown rigid body transformations still to be estimated, see Fig. 3.41.

Note, however, that

$$\begin{aligned}
{}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{C}} &= {}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}}\,{}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{C}}\\
{}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{LSP}} &= {}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}}\,{}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{LSP}}\\
{}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{LRS}} &= {}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}}\,{}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{LRS}}\\
{}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{IMU}} &= {}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}}\,{}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{IMU}}
\end{aligned}\tag{3.43}$$

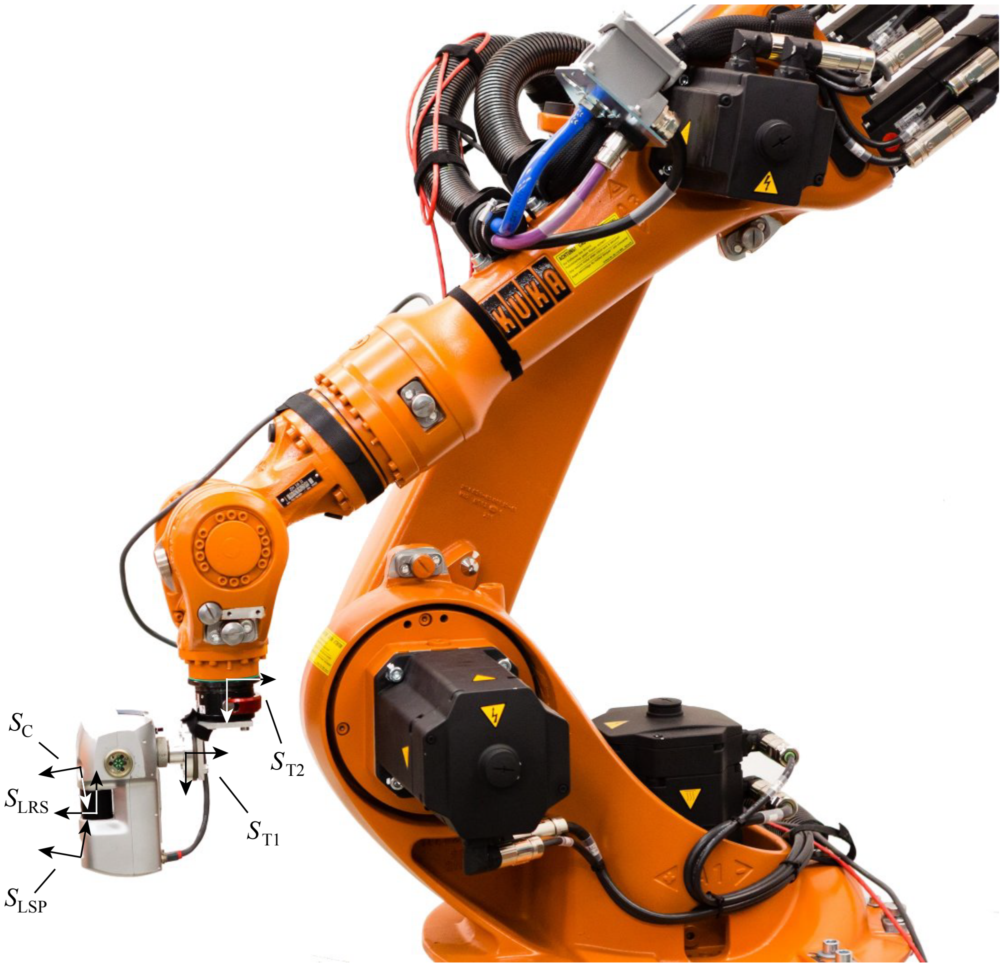


Figure 3.41: The rigid body transformations between the sensor reference frames S_{C}, S_{LSP} and S_{LRS} and the original TCP reference frame S_{T1} were known; now a second reference frame S_{T2} fixed to the Kuka KR 16 is used so that all former reference frames have to be referenced to it.

because the component sensors are rigidly attached to each other, thus the laws of rigid body motion hold. Consequently:

$$\begin{aligned}
{}_{0}\overline{\boldsymbol{p1}} &= {}_{0}\widetilde{\boldsymbol{T}}^{\mathrm{T2}}\ {}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}}\ {}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{C}}\ \ {}_{\mathrm{C}}\overline{\boldsymbol{p1}} \\
{}_{0}\overline{\boldsymbol{p2}} &= {}_{0}\widetilde{\boldsymbol{T}}^{\mathrm{T2}}\ {}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}}\ {}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{LSP}}\ {}_{\mathrm{LSP}}\overline{\boldsymbol{p2}} \\
{}_{0}\overline{\boldsymbol{p3}} &= {}_{0}\widetilde{\boldsymbol{T}}^{\mathrm{T2}}\ {}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}}\ {}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{LRS}}\ {}_{\mathrm{LRS}}\overline{\boldsymbol{p3}} \\
{}_{0}\overline{\boldsymbol{p4}} &= {}_{0}\widetilde{\boldsymbol{T}}^{\mathrm{T2}}\ {}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}}\ {}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{IMU}}\ {}_{\mathrm{IMU}}\overline{\boldsymbol{p4}}
\end{aligned} \tag{3.44}$$

where the only unknown transformation is ${}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}}$. It is easy and fast to estimate the latter transformation e.g. out of a sole additional extrinsic camera calibration w.r.t. the newer pose tracking device #2, *i.e.*, w.r.t. the reference frame S_{T2}, leading to ${}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{C}}$ (refer to Section 3.3). Eventually:

$${}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{T1}} = {}_{\mathrm{T2}}\widehat{\boldsymbol{T}}^{\mathrm{C}} \left({}_{\mathrm{T1}}\widehat{\boldsymbol{T}}^{\mathrm{C}} \right)^{-1} \tag{3.45}$$

and all transformations on the left-hand side in Eqs. (3.43) can be recalculated by homogeneous matrix multiplication.

It is worth noting that, if it is the camera that is being recalibrated w.r.t. the pose tracking device #2, it is crucial to fix the intrinsic parameters of the camera to the former ones during the intrinsic camera calibration step required for eventual hand-eye calibration. In the contrary, the principal point of the camera may shift, which translates into a different camera reference frame S_{C}, refer to (Tsai, 1987), so that the above Eqs. (3.43) do not hold anymore.

This approach is, of course, much faster than performing separate extrinsic calibration for all sensor components of the DLR 3D-Modeler when mounting it on a newer pose tracking device.

3.10 Summary

Based upon the sensor models presented in the last chapter 2 and in line with the good practice guidelines in chapter 1, in this chapter 3 I presented a number of novel methods to accurately and easily calibrate all sensor components of the DLR 3D-Modeler.

Starting out, I motivate the development of accurate and yet simple calibration methods. First, sound methods are required that minimize actual residual errors according with the system models for the sake of statistical optimality. Second, the user ought to choose an appropriate sensor model in order to avoid overparametrization—the calibration method can support the user at that. Third, a simple calibration method that reduces requirements e.g. on the calibration object should be preferred as it avoids human mistakes otherwise bound to occur. Last, great care should be taken to provide valid calibration data that contain enough evidence for the calibration method to be able to infer the correct parametrization.

In Section 3.2 I cope with the intrinsic calibration of the main sensor of the DLR 3D-Modeler: the **stereo camera**. Digital cameras are the principal perception systems in many areas like computer vision for robotic applications, hence convenient models and methods already exist that indeed should be used. Most academic users, however, find it difficult to obtain adequate software that implements these methods. For this reason I developed the calibration software DLR CalLab that implements the abovementioned methods (among others) and is freely distributed among academia. The software is main part of the well-known calibration toolbox DLR CalDe and DLR CalLab; the interested reader can find its documentation in Appendix C.

Most camera measurements have to be represented in general reference systems beyond the camera reference frame S_{C} e.g. for robots to steer for measured items (the most significant exception being visual servoing). When transferring local data in S_{C} into the world reference frame S_0 e.g. by using a robotic manipulator, the static pose of the camera w.r.t. the end-effector (or tool center point) of the manipulator ${}_{\mathrm{C}}\boldsymbol{T}^{\mathrm{T}}$ is required. External camera calibration or **hand-eye calibration** aims precisely at that transformation. When implementing former hand-eye calibration methods, however, I astonished at their lack of statistical soundness when identifying the appropriate residuals for optimization by non-linear minimization. Hence I devised a novel hand-eye calibration method that minimizes Euclidean transformation errors of the robotic manipulator for statistically optimal estimation, decoupling this process from the former intrinsic camera calibration method, see Section 3.3. Ever since its original presentation in (Strobl and Hirzinger, 2006), the method very much became standard in academia and industry and it has been included in the calibration toolbox DLR CalDe and DLR CalLab (Strobl *et al.*, 2005).

Besides, I identified **two major shortcomings** of both, the standard camera calibration method presented in (Zhang, 2000; Sturm and Maybank, 1999) as well as of my own method for hand-eye calibration in Section 3.3. On the one hand, both methods demand accurate geometrical knowledge of the calibration object (viz. a planar, precision checkerboard target). In Section 3.4 I note that it is very difficult for off-the-shelf printers to achieve the required accuracy and that most users fail to accurately measure the resulting patterns by hand. I provide two alternative methods that yield optimal parametrization of the camera and of its pose w.r.t. an external pose tracking system irrespective of the actual dimensions of the calibration target (Strobl and Hirzinger, 2008, 2011). On the other hand, I became aware of the unsuitability of the standard methods for intrinsic and extrinsic camera calibration in the case of cameras with narrow angular field of view; consequently, in Section 3.5 I bring forward an alternative method merging standard camera calibration in (Zhang, 2000; Sturm and Maybank, 1999) with standard hand-eye extrinsic calibration in (Strobl and Hirzinger, 2006), which improves calibration performance in the case of cameras featuring narrow angular field of view.

In Section 3.6 I address the calibration of the **light stripe profiler** (LSP) of the DLR 3D-Modeler. Again, calibration methods relying on precision, complex hardware are avoided by devising a novel method that leverages the prior, accurate intrinsic and extrinsic calibrations of the stereo camera of the DLR 3D-Modeler (Strobl *et al.*, 2004). As a consequence, only 3 DoF are left for calibration, viz. the pose of the laser plane w.r.t. the stereo camera reference frame S_{C}. This self-calibration approach proceeds by correcting deformations when scanning a planar surface of unknown pose caused by miscalibration of the laser plane.

A variant of the last method is presented in Section 3.7 for the calibration of the **laser range scanner** (LRS) of the DLR 3D-Modeler. I note that the optimization problem is harder than in the case of the LSP because the origin of the LRS is not known in advance (in the case of the LSP its origin is coincident with the origin of S_{C}). For this reason the calibration by optimization is more sensitive to the completeness of the datasets for calibration. In the case that the user is restricted when gathering calibration data, I propose an alternative residual function that robustifies the solution at the expense of the former unnecessity to determine the pose of the calibration plane.

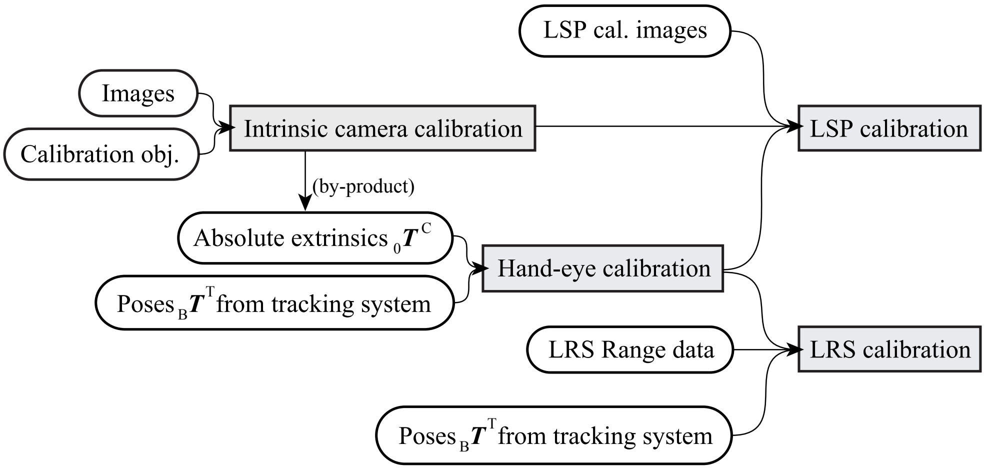

LSP cal. images
Images
Intrinsic camera calibration
LSP calibration
Calibration obj.
(by-product)
Absolute extrinsics $_0\boldsymbol{T}^C$
Hand-eye calibration
Poses $_B\boldsymbol{T}^T$ from tracking system
LRS Range data
LRS calibration
Poses $_B\boldsymbol{T}^T$ from tracking system

Figure 3.42: Functional interaction between the calibration procedures (Fig. 1.2 reprint).

Fig. 3.42 is a reprint of Fig. 1.2 showing the functional interactions between the abovementioned calibration procedures.

In Section 3.8 I shortly notice that in this work I am not concerned with a complex calibration of the **inertial measurement unit** (IMU) in its position but only in its orientation as, in Section 5.4.2, I shall relax requirements precisely in order to keep the complexity of the whole calibration procedure for the DLR 3D-Modeler in practical terms.

Finally in Section 3.9 I draw the reader's attention to the possibility for **rapid extrinsic recalibration** of all sensor components of the DLR 3D-Modeler by one-time extrinsic calibration for the calibrated stereo camera.

"It's hardware that makes a machine fast.
It's software that makes a fast machine slow."
—Craig Bruce, 1990

4 Robust Operation of Sensors

4.1 Introduction

Modern sensors are constituted by software and hardware in equal proportions. It is by software (e.g. image processing and algebraic calculations) that accurate range data can be eventually delivered. In turn, the basis for correct range sensing is appropriate sensor modeling (Chapter 2) and accurate parametrization of these models (Chapter 3). Still, the computation of range data has to be correct in order to preserve the utility of the former models and their parametrizations. In this chapter 4 the required computations for the component sensors of the DLR 3D-Modeler will be presented—perhaps with the exception of the virtual pose sensor that will be separately presented in the following chapter 5.

In detail, in this chapter I will mainly address the three main sensor components of the DLR 3D-Modeler. First, depth computation by stereo vision will be shortly explained—the fundamental implementation of depth range estimation stems from Heiko Hirschmüller and is detailed in (Hirschmüller, 2008). Second, I address depth computation by structured light using the laser stripe profiler in (Strobl *et al.*, 2004); in detail, robust image processing serves data to the triangulation methods in Eqs. (2.36) and (2.37). Third, I explain how to register and filter depth values from the laser range scanner, and I elaborate on their convenient allocation for further processing.

Moreover, final depth computation will allow us to determine the precision of the range estimations as well as the field of view of the different sensors. As explained in Section 2.2, the characteristics of the sensor will be eventually compared in order to choose the optimal sensor for a specific task or rather to evade or clear sensor weaknesses increasing robustness of the overall approach at that.

In the end, depth computation will allow a variety of applications in the context of 3-D scanning that will be shortly listed in Section 4.6 as well as in Appendix B.

4.2 The Stereo Camera

4.2.1 Introduction

In Section 2.2.1 the inner geometry of single and stereo cameras have been presented in detail. By rigidly joining two cameras to each other, a novel realm of geometric dependencies between both images and the 3-D scene emerges. In the end, a range image of the scene can be generated (*i.e.*, a 2.5-D image), provided the stereo camera is calibrated as in Section 3.2. In this section I shall, first, describe these relationships in mathematical terms and, second, present the current dense stereo vision algorithm at use at the DLR 3D-Modeler.

4.2.2 The Geometry of Two Views

The geometry of two views, *i.e.*, the geometry of stereo vision, is also called epipolar geometry. It lays down the constraints that apply when searching for correspondences between their component cameras and it is central to reconstruct projections into their original 3-D locations using stereo vision algorithms.

The search for projection correspondences starts out from detected features on the image of the main camera. As illustrated in Fig. 4.1, that projection necessarily corresponds to a 3-D point contained in its 1-D projection ray to the main focal point. If the 3-D point is being imaged by the second camera, its projection ray has to intersect the focal point of the second camera as well. The collection of all possible projection rays unto the second camera from all potential ranges of the original 3-D point w.r.t. the first camera clearly constitutes a plane that contains both focal points as well as the original projection ray. This plane is called the epipolar plane, and its intersection with the image plane of the second camera is called the epipolar line of the original projection and will be useful for correspondence search.

Note that the intersection of the line joining both focal points will be always contained in all possible epipolar planes, *i.e.*, in the pencil of potential epipolar planes; these two points are, of course, static, and they are called epipoles.

Since the inner geometry of the stereo camera is precisely known by using the methods presented in Section 3.2, it ought to be possible to estimate the epipolar line corresponding with any original projection (if it exists), thus constraining the search for valid correspondences on the second image plane to a linear region (to be more precise, to a segment).

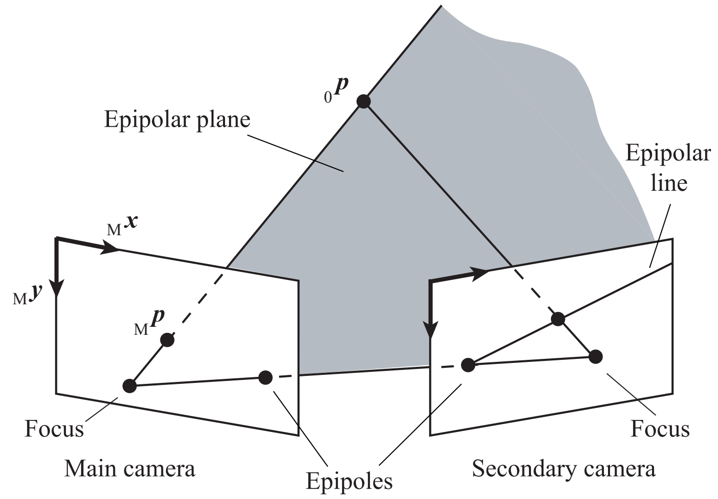


Figure 4.1: Epipolar geometry of two views.

In the presence of lens distortion, however, any projected line (that does not contain the principal point) will be inevitably curved; this is inconvenient for computational correspondence search algorithms. As a consequence, prior undistortion of the image is usually performed. In addition, general, oblique lines are not optimal for computational correspondence search algorithms. Ideally the orientation of the epipolar line for correspondence search ought to be horizontal or vertical. The good news is that, in the absence of lens distortion, planar transformations under perspective projection are linear in homogeneous coordinates (cf. Section 2.2.1) so that images can be easily warped to their ideal stereo configuration where both camera frames are on the same plane and orientation so that epipolar lines horizontally correspond with their original projections in the main camera frame, see Fig. 4.2. This step is called rectification and allows for more efficient image processing when searching for stereo correspondences (Trucco and Verri, 1998).

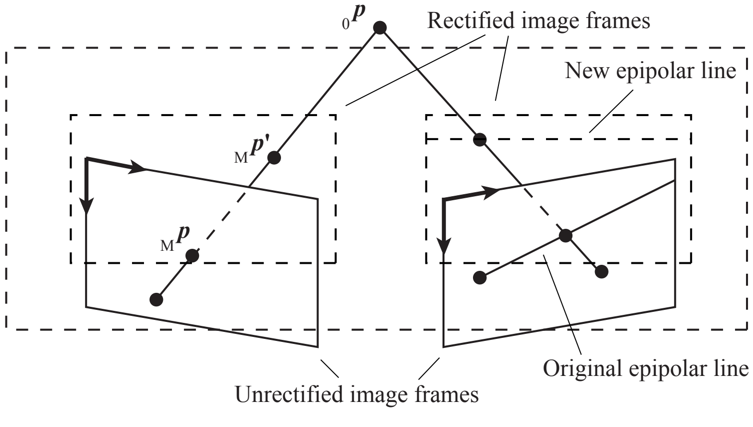


Figure 4.2: Reprojection of original projections unto rectified image frames.

It is clear that, at infinity, feature projections coincide in both projections. In the case of closer points, however, an horizontal mismatch between both images' projections appears that is inversely proportional to the absolute distance R (range) of the feature w.r.t. the stereo camera. This distance is called disparity D and is directly proportional to the focal length f as well as to the baseline between cameras B as follows:

$$D = f\,\frac{B}{R} \quad . \tag{4.1}$$

Since f and B are constant parameters and only R varies, disparity results can be considered as raw output of stereo vision reconstruction algorithms e.g. representing them on the original image frame. On occasions, these images can be further processed to Euclidean depths (2.5-D images) taking the intrinsic parameters of the cameras into account.

4.2.3 The Semiglobal Matching Algorithm

The Semiglobal Matching (SGM) stereo vision method by Heiko Hirschmüller computes disparity images in a dense way—as opposed to sparse, feature-based methods as in Section 5.4.1. Dense methods allow for advanced 3-D reasoning as well as 3-D modeling, hence the method suits the DLR 3D-Modeler.

At higher baseline distance or closer range to the scene, radiometric differences on input stereo images appear. A pixelwise, mutual information (MI)-based matching cost has been introduced to adjust these differences along with smoothness constraints. Still, SGM is highly sensitive to the accuracy of the camera calibration process in Section 3.2.

Since this method is not part of my work within this thesis, I refer the reader to (Hirschmüller, 2008) for further information.

4.2.4 Data Representation

It is of central importance for efficient 3-D data representation and eventual treatment to comply with one of the pre-defined range data representation types. These data instances are delivered at high rate, *i.e.*, normally at the camera's rate e.g. 25 Hz. In the context of the DLR 3D-Modeler four types of data instances are supported:

- Cartesian type: A 2-D array of ranges that are orthogonal to the sensing plane and sequentially registered in an equally spaced 2-D sensor grid.

- Perspective type: A 2-D array of ranges that complies with the perspective model presented in Section 2.2.1 on the basis of a normalized, rectified pinhole camera model, *i.e.*, $\alpha \triangleq \beta$, $\gamma \triangleq 0$, $u_0 \triangleq v_0 \triangleq 0$, and in the absence of optical distortion; all ranges are sequentially registered in an equally spaced, projective 2-D sensor grid.

- Cylindrical type: Ranges are registered in 2 DoF, viz. a single rotatory axis and its origin is translated on the same axis, on equally spaced distances and angles.

- Spherical type: Ranges are registered in 2 DoF, viz. two rotatory axis with fixed origin. The array of ranges is distributed in the two equally-spaced angles.

In the case of the stereo camera (and especially if image rectification has been previously performed), the natural representation of range data is the perspective one. First, the size and the resolution of the normalized sensor grid is defined with the parameters N_{rows} (number of equally spaced rows), N_{cols} (number of equally spaced columns), u_{initial} (initial value of u), v_{initial} (initial value of v), Δu (distance between adjacent columns), and Δv (distance between adjacent rows). By doing so, the explicit 2-D coordinates of single range instances are silent as they can be sequentially computed from the above values as follows:

$$u_i = u_{\text{initial}} + i \cdot \Delta u \quad \forall i \in \mathbb{N}_0, \quad i < N_{\text{rows}} \quad , \tag{4.2}$$

$$v_j = v_{\text{initial}} + j \cdot \Delta u \quad \forall j \in \mathbb{N}_0, \quad j < N_{\text{cols}} \quad . \tag{4.3}$$

Apart from the metadata mentioned above, the only data being streamed are $N_{\text{rows}} \cdot N_{\text{cols}}$ real values

$$\tilde{d}_n \quad \forall n \in \mathbb{N}_0, \quad n < (N_{\text{rows}} \cdot N_{\text{cols}}) \tag{4.4}$$

corresponding to the measured ($\tilde{}$) Euclidean depth of the triangulated features, in row-major order. These are usually supplied in the form of a large, real vector size $N_{\text{rows}} \cdot N_{\text{cols}}$.

From these data the local, 3-D position of all range data in S_{C} can be rapidly computed as follows:

$$_{\text{C}}\boldsymbol{p}_n = \begin{bmatrix} \tilde{d}_n \cdot u_i \\ \tilde{d}_n \cdot v_j \\ \tilde{d}_n \end{bmatrix} \quad . \tag{4.5}$$

In addition, the 6 DoF of the tracked pose of the stereo camera S_{C} w.r.t. some world coordinate frame S_0 can be delivered for every dense depth image. In the end:

$$_0\boldsymbol{p}_n = {_0\widetilde{\boldsymbol{T}}}^{\text{C}}_{3\times4} \begin{bmatrix} \tilde{d}_n \cdot u_i \\ \tilde{d}_n \cdot v_j \\ \tilde{d}_n \end{bmatrix} \quad . \tag{4.6}$$

4.2.5 Operating Range

Because of its wide scope sensing range and the (low) speed of acquisition (a typical runtime of 1 to 2 seconds for dense 2.5-D range images on typical scenarios), stereo vision on the DLR 3D-Modeler is deployed as a middle- to far-range sensor. The stereo baseline of 5 cm (see Section 2.2) has been chosen for a typical operating range between 30 cm and 2 m. Since the sensor yields less range accuracy than the LRS or the LSP, it is best suited for exploration or obstacle avoidance scenarios. Still, in controlled environments—e.g. for the humanoid robot "Justin" in (Borst *et al.*, 2009), precise scene recognition for on-table manipulation can be also achieved.

4.2.6 Range Estimation Accuracy

A Monte Carlo simulation of feature-based stereo vision triangulation at a typical depth of 25 cm is represented in Fig. 4.3. We can see that the potential accuracy in this case is millimetric in range, which would suffice for 3-D modeling. Stereo vision by SGM does not, however, operate on punctual features but in a dense manner that, unfortunately, is prone to invalid solutions especially at contours or repetitive patterns.

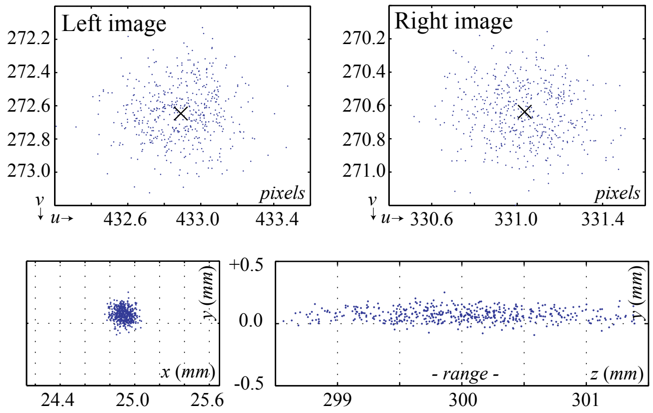


Figure 4.3: Monte Carlo analysis on the expected confidence in feature-based stereo vision. A feature located at {24.9, 0, 300} mm w.r.t. the left camera (basis distance between cameras 50.1 mm) is erroneously detected in both images (top, 1000 samples) with $\sqrt{\sigma_{\mathrm{u}}^2+\sigma_{\mathrm{v}}^2}=0.25$ pixels. The triangulation results show a bias of +0.004 mm in range (z), and a standard deviation σ_{z} of 0.6 mm. In x and y: $\sigma_{\mathrm{x}}=\sigma_{\mathrm{y}}=0.05$ mm.

The general formula for range precision bases on Eq. (4.1) above. The estimation of range R depends on two constant parameters (focal length f and stereo baseline B) as well as on an inaccurate parameter owing to image processing: the disparity D. Under the conservative assumption of a standard deviation $\sigma_{\mathrm{D}}=1$ pixels and according to the perturbation theory in (Haralick, 1998; Matthies, 1992), we can infer the error characteristics in reconstructed 3-D coordinates from the image processing noise by error propagation. We first compute the partial derivative

$$\frac{\partial R}{\partial D}=-\frac{fB}{D^2} \tag{4.7}$$

that can be easily linearized; after linear covariance computation we obtain:

$$\sigma_{\mathrm{R}}=\frac{\partial R}{\partial D}\,\sigma_{\mathrm{D}} \quad , \tag{4.8}$$

substituting Eq. (4.1) into the last equation:

$$\sigma_{\mathrm{R}}=\frac{fB}{D^2}\,\sigma_{\mathrm{D}} \quad , \tag{4.9}$$

or, in other words

$$\sigma_{\mathrm{R}}=\frac{R^2}{fB}\,\sigma_{\mathrm{D}} \quad . \tag{4.10}$$

In the case of the DLR 3D-Modeler featuring a stereo baseline $B=50$ mm and focal length $f=750$ pixels, Fig. 4.4 shows the expected precision in range R. Note that this noise level matches experimental results in Section 3.4.4.

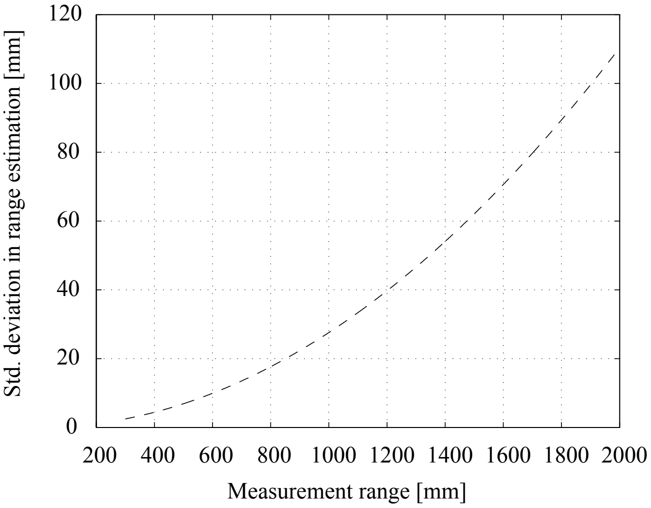


Figure 4.4: Expected range precision w.r.t. the triangulation range.

4.3 DLR Laser Stripe Profiler

4.3.1 Introduction

In Section 2.2.2 the geometry of the DLR laser stripe profiler (LSP) has been addressed. In addition, the triangulation formulae for the crosshair, dual LSP have been presented as they were needed for laser plane self-calibration in Section 3.6.

In this section I present the required algorithms that will feed in data to the triangulation solution in Eq. (2.46). The concerned image processing methods are dominated by the difficulty that we chose during conception of the DLR 3D-Modeler in the first place: Neither of the cameras mounted on the DLR 3D-Modeler can be filtered to laser light in order to simplify the detection of laser projections by image processing methods. As mentioned in Sections 1.5 and 2.2.2, filtered cameras would prevent multisensory operation of the DLR 3D-Modeler e.g. when using stereo vision in Section 4.2, texturing of 3-D models in Section 5.4.6, or pose tracking from images in Chapter 5. This decision is in contrast with most 3-D modeling systems based on laser stripe profiling, refer to Section 1.4. On the other hand, however, image processing is now much harder than in the case of the latter methods, requiring much more robust operation than before. The novel image processing approach presented here was originally introduced in (Strobl *et al.*, 2004).

In addition, the convenient representation of its output data as well as the operating range of the LSP and experiments on its range precision are also being addressed.

4.3.2 Robust Image Processing

In this section the laser plane projections ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_i = [{}_{\mathrm{M}}\tilde{x}_i, {}_{\mathrm{M}}\tilde{y}_i]$ are to be delivered out of raw, unfiltered footage as in Fig. 4.5. These results—along with the calibration results from Sections 3.2 and 3.6—will be fed in the triangulation Eqs. (2.36), (2.44) and (2.45).

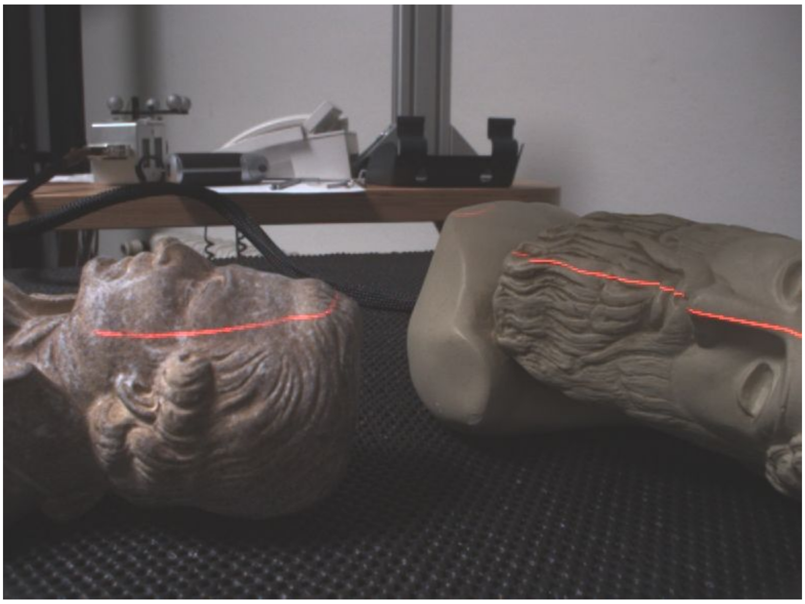


Figure 4.5: Unfiltered, raw image as input to the stripe segmentation algorithm.

In order to widen the range of objects that can be scanned, the image processing algorithms presented in this sections do not have any a priori knowledge about the final shape of the laser outlines. Admittedly, this fact may yield erroneous results. For instance, in the case of specular reflections or red light sources, a simple segmentation algorithm would erroneously detect them as potential laser projections. Consequently, different measures must be introduced in orders to reduce these type of failures to a minimum. To make matters worse, the cameras on the DLR 3D-Modeler do not have any optical filter to laser light.

To start with, the camera settings ought to be considered. For instance, in the case of older CCD interlaced cameras, I choose to process only the even fields of the images in order to avoid both, desyncronizations and radiometric distortions (Kamberova, 1997). Of course, the camera focus and its aperture have to be chosen appropriately to get sharp, bright images—bright images are inconvenient for laser stripe segmentation but then they are a requirement for online texturing or visual pose tracking in Chapter 5. In addition, in order to maximize both, the range of view and the range precision of the LSP, camera and laser have to be arranged in such a way that the projected stripes become approximately horizontal, see Fig. 4.5. In the case of the dual, crosshair LSP, a crossed layout has to be chosen, refer to Section 2.2.2.

The implemented approach presented next is based on four image processing stages. The approach is as follows: image $\mathcal{I}$ is processed column by column. For each column $\mathcal{C}$, different laser stripes may be detected after *Stage #1*. *Stages #2* and *#3* validate the original results and, in the affirmative, *Stage #4* estimates the pth center point of the stripe ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_p$.

Stage #1: Detection of the edges of laser stripes

To start with, the upper and lower edges of the laser stripe are to be detected. To this end I choose a variant of the Sobel filter (gradient operator); it approximates absolute gradient magnitudes at each image point ${}_{\mathrm{M}}\boldsymbol{p}$ and filters potential noise. First, the red component of the image $\mathcal{I}$ is extracted. Second, the convolution of the Sobel kernel unto the resulting image emphasizes the horizontal edges of the laser stripe owing to the tight focusing and brightness of laser light, see Fig. 4.6.

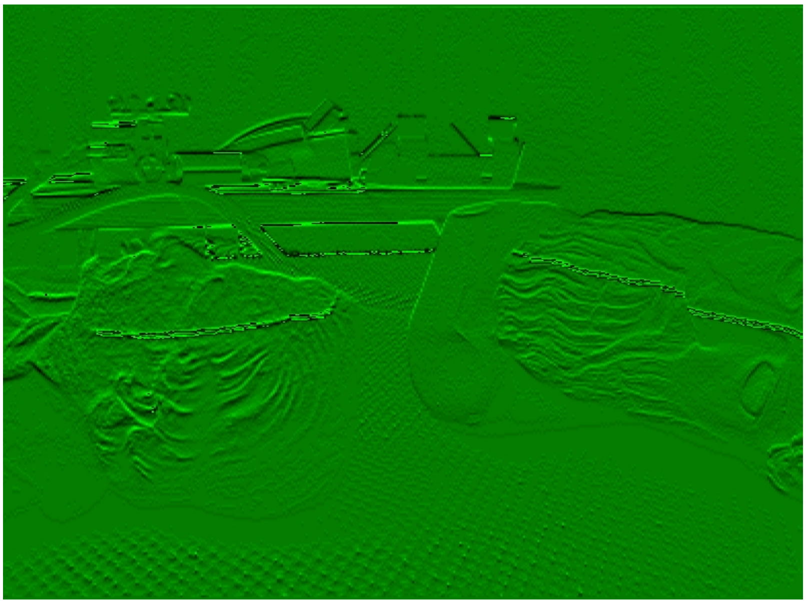


Figure 4.6: Result of the convolution of the Sobel kernel unto the original image Fig. 4.5.

On the basis of the resulting image, upper (brighter) and lower (darker) edges of potential stripes are selected at every column $\mathcal{C}$. Note that, subject to the structure of the scene, several stripes can be projected on the very same column $\mathcal{C}$. The detection of edges is performed by setting appropriate upper and lower brightness thresholds to the image in Fig. 4.6, as brightness correspond to absolute vertical derivatives of the original image in Fig. 4.5. In Fig. 4.7 the detected edges are overlayed on the original image for optimal threshold levels. In Fig. 4.7 I show other results with non-optimal threshold levels.

Stage #2: Color validation

The potential stripes detected in Fig. 4.7 can contain e.g. white stripes as white is also strong on its red image channel. To validate the detected stripe, the color values among adjacent edges are evaluated. For this purpose, a **L**ook-**U**p **T**able (LUT) of image colors was previously generated; the LUT will be used to decide whether a color value belongs to the background colorsubspace or to the laser stripe one. The LUT was generated prior to the scanning process as follows: a series of pictures $\mathcal{I}_i$ are taken from the natural scene without laser projection.

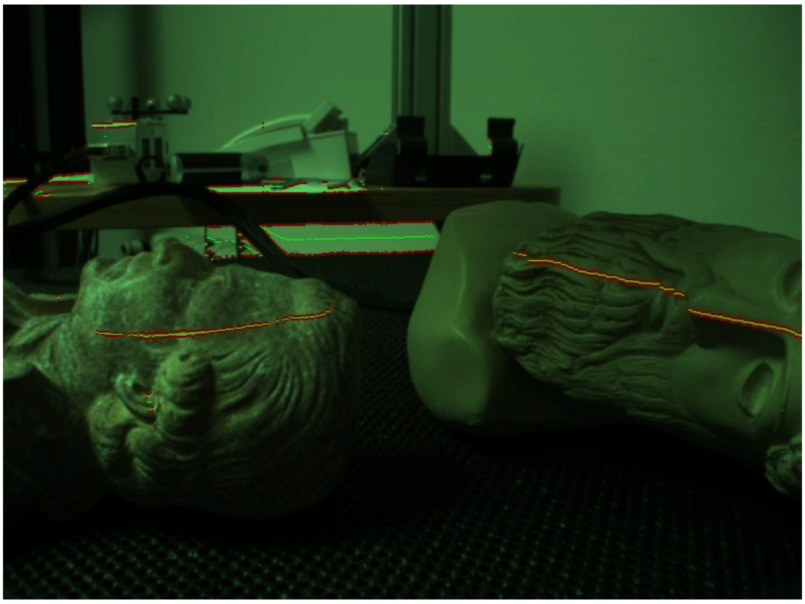


Figure 4.7: Detected edges after convolution of the Sobel kernel unto the original image Fig. 4.5 for optimal threshold levels.

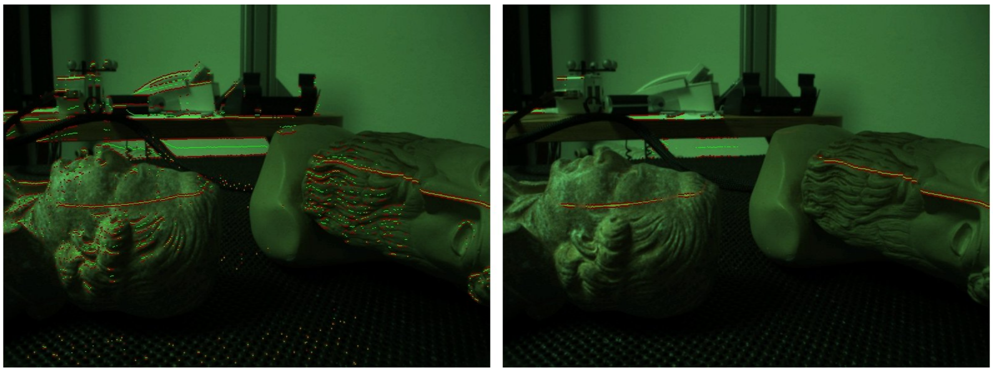


Figure 4.8: Detected edges after convolution of the Sobel kernel unto the original image Fig. 4.5 for non-optimal threshold levels.

Here $\mathcal{I}_i = \{\boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_J\}$ where $\boldsymbol{c}_j = (\mathcal{R}, \mathcal{G}, \mathcal{B}) \in [0,31][0,63][0,31]$ are the color values of every pixel j, $\forall j \in \mathbb{N}_1$, $j \leq J$; J is the number of pixels in the image, e.g. $J = 780 \times 580$. All perceived color values are stored online in a preliminary background LUT

$$\boldsymbol{bcs}_{\text{temp}}(\mathcal{R}, \mathcal{G}, \mathcal{B}) = \begin{cases} 1 & \text{if} \quad \exists\, i : (\mathcal{R}, \mathcal{G}, \mathcal{B}) \in \mathcal{I}_i \ \cup \ \mathcal{R} = 0 \\ 0 & \text{if} \quad \text{else} \end{cases} . \tag{4.11}$$

The indices of $\boldsymbol{bcs}_{\text{temp}}$ that are set match up then with background (including objects) color values. However, this assertion may not be reciprocal, *i.e.*, there may be background color values not yet set in the former indices in relation to the limited diversity of images. In order to ensure completeness, an LUT named $\boldsymbol{bcs}$ is created based on the assumption that laser stripe color values are expected to hold higher red components as follows:

$$\boldsymbol{bcs}(\mathcal{R}, \mathcal{G}, \mathcal{B}) = \begin{cases} 1 & \text{if} \quad \exists\, \mathcal{R}' \geq \mathcal{R} - \mathcal{R}_{\text{offset}} : \boldsymbol{bcs}_{\text{temp}}(\mathcal{R}', \mathcal{G}, \mathcal{B}) = 1 \\ 0 & \text{if} \quad \text{else} \end{cases} , \tag{4.12}$$

where $\mathcal{R}_{\text{offset}}$ represents an offset in red values; this offset is required because of the diffuse radiation that the laser naturally sends out to the environment due to impurities in the spreading cylindrical lens. Fig. 4.9 shows an example of the set $\boldsymbol{bcs}_{\text{temp}}$ indices. The resulting stripes of *Stage #1* are accepted whenever they show pixel color values corresponding to laser (*i.e.*, *not* to the natural scene according to their respective $\boldsymbol{bcs}$ entry) within their upper and lower edges. This method copes very well with the problem of bright reflections. In addition, this method supports robustness and flexibility against changing lighting conditions.

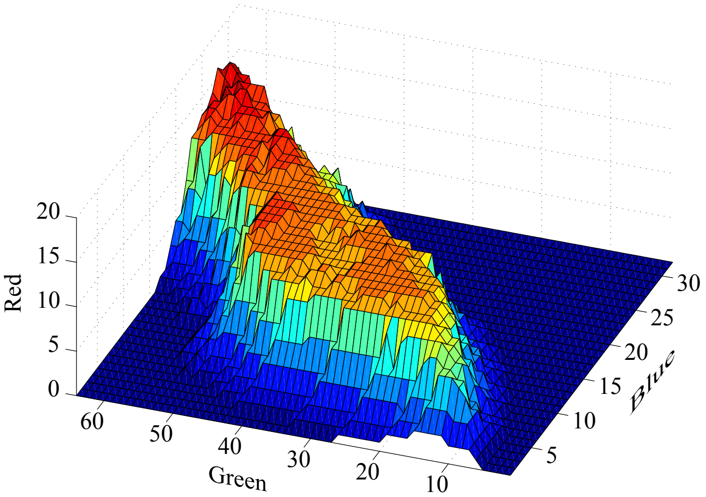


Figure 4.9: *RGB565* Look-Up Table. *Stage #2* is fulfilled for red values above this mesh.

Fig. 4.10 illustrates the significance of the offset on laser red values $\mathcal{R}_{\text{offset}}$. In Fig. 4.11 the valid LUT results in the right-hand side of Fig. 4.10 are used to validate laser stripe edges in the output image of *Stage #1* (Fig. 4.7).

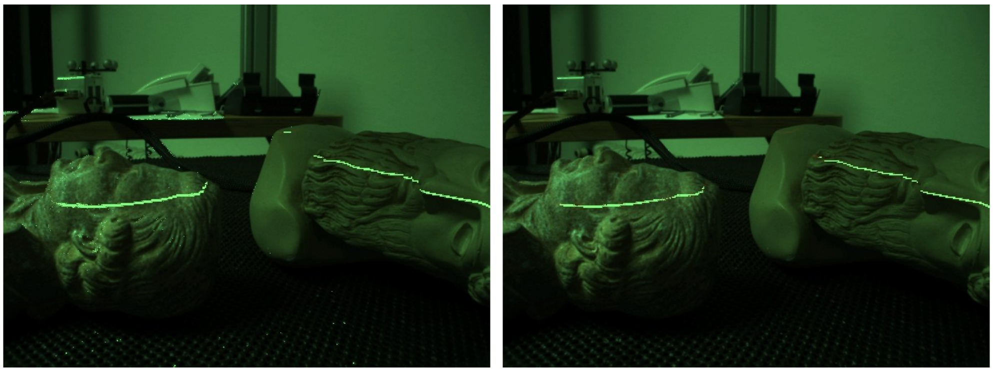


Figure 4.10: On the lefth-hand side, pixel colors that fulfill the LUT with $\mathcal{R}_{\text{offset}} = 0$; on the right-hand side, pixel colors that fulfill the LUT and an appropriate offset $\mathcal{R}_{\text{offset}}$. Note that the former are trapped by specular, white reflections at corners.

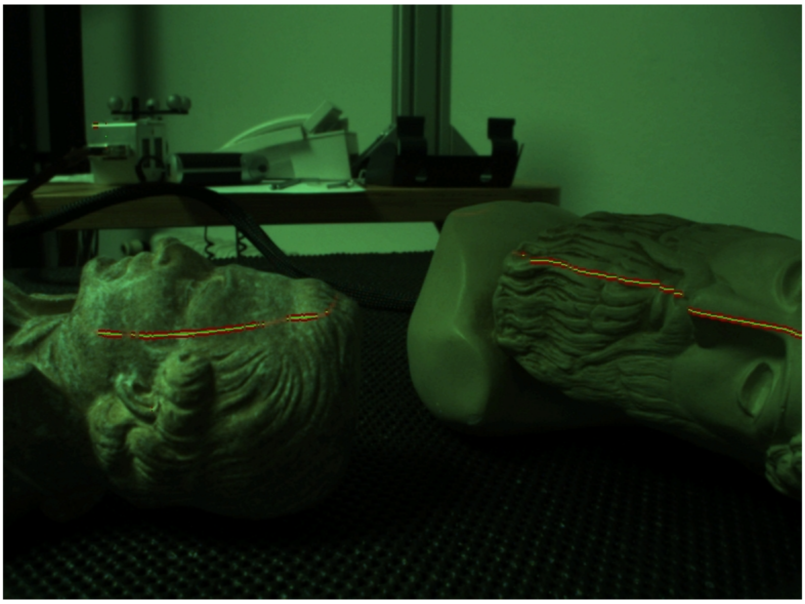


Figure 4.11: The detected laser stripe edges in Fig. 4.6 have been crosschecked w.r.t. the LUT in Fig. 4.9.

Note that the transformation of the native red-green-blue (RGB) color space of the images into the hue-saturation-lightness (HSL) color space has been considered in (Sladczyk, 2008); using the same methods presented here, it has been proven to be of no advantage at the cost of additional computations.

Stage #3: Width validation

The second validation stage addresses the problem of specular reflections caused by the laser itself. This type of reflections would be most likely recognized as laser stripes by the first two stages.

In a nutshell, *Stage #3* accepts laser stripes whenever the pixel width of the stripe is within a certain range. The feasible width of the laser stripe depends on the measuring distance, the laser projection angle as well as on surface reflection characteristics. In order not to be conservative when defining this interval, experiments have been performed offline in order to identify the biggest and the smallest stripe widths for every projected stripe in S_{M}, refer to Fig. 4.12.

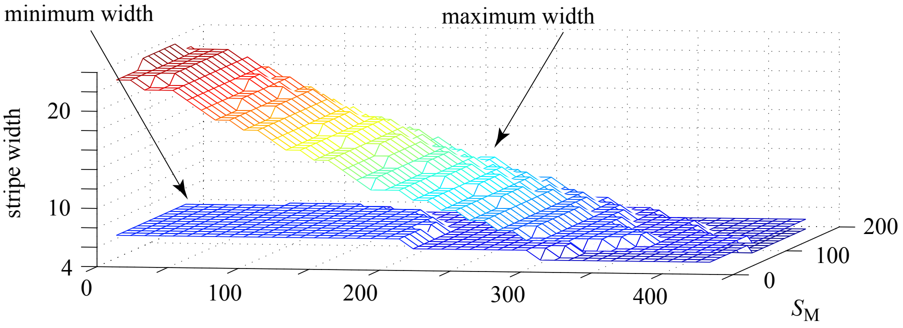


Figure 4.12: Maximum and minimum allowed widths for every projected stripe in S_{M}.

These extreme widths are stored in an LUT and remain valid as long as the LSP is not rearranged. In addition, this method copes with some of the corner reconstruction artifacts addressed more in detail in (Curless, 1997).

In Fig. 4.13 the resulting, final image where stripe detection has gone through *Stages #1* to *#3* is shown.

Estimation of the center of the stripe

When valid laser stripe projections have been found, their projection center have to be determined. I proceed with sub-pixel precision (*i.e.*, to within a fraction of a pixel) by means of the center of mass method over the red channel of the image. The precision hereby achieved are similar to the precision by other methods like Gaussian approximation (Trucco *et al.*, 1998). In fact, the latter method often delivers erroneous or invalid results in the presence of saturated brightness values; these may naturally happen on our images, not so on the images of their original publication *because they opt for filtered cameras.* Whenever the central image projection ${}_{\mathrm{M}}\tilde{\boldsymbol{p}}_p$ of the laser plane is available, by using Eq. (2.37) we can obtain their 3-D coordinates ${}_{0}\hat{\boldsymbol{p}}_p$.

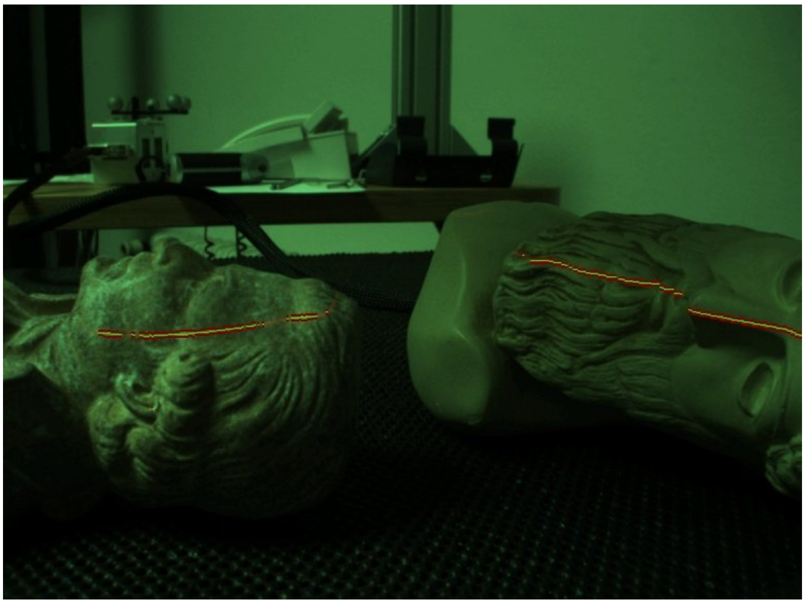


Figure 4.13: The detected laser stripe edges in Fig. 4.6 have been crosschecked w.r.t. the LUT in Fig. 4.9 as well as w.r.t. the laser width LUT in Fig. 4.12.

4.3.3 Data Representation

It is of central importance for efficient 3-D data representation and eventual treatment to comply with one of the pre-defined range data representation types. These data instances are delivered at high rate, *i.e.*, normally at the camera's rate e.g. 25 Hz. In the context of the DLR 3D-Modeler four types of data instances are supported:

- Cartesian type: A 2-D array of ranges that are orthogonal to the sensing plane and sequentially registered in an equally spaced 2-D sensor grid.

- Perspective type: A 2-D array of ranges that complies with the perspective model presented in Section 2.2.1 on the basis of a normalized, rectified pinhole camera model, *i.e.*, $\alpha \triangleq \beta$, $\gamma \triangleq 0$, $u_0 \triangleq v_0 \triangleq 0$, and in the absence of optical distortion; all ranges are sequentially registered in an equally spaced, projective 2-D sensor grid.

- Cylindrical type: Ranges are registered in 2 DoF, viz. a single rotatory axis and its origin is translated on the same axis, on equally spaced distances and angles.

- Spherical type: Ranges are registered in 2 DoF, viz. two rotatory axis with fixed origin. The array of ranges is distributed in the two equally-spaced angles.

Since the range data of the LSP are limited to its laser plane, I choose the cylindrical type of depth data representation on equally spaced angles, *i.e.*, with 1 sole DoF. First, the angular size and the resolution of the normalized sensor grid is defined with the parameters N_{LSP} (number of equally spaced angles), $\vartheta_{\text{initial}}$ (initial value of the cylindrical angle ϑ), and $\Delta\vartheta$ (distance between adjacent angles ϑ). The silent angles of all measured ($\tilde{}$) range data $\tilde{d}_n$ are:

$$\vartheta_n = \vartheta_{\text{initial}} + n \cdot \Delta\vartheta \quad \forall n \in \mathbb{N}_0, \quad i < N_{\text{LSP}} \quad . \tag{4.13}$$

Apart from the metadata mentioned above, the only data being streamed are N_{LSP} real values

$$\tilde{d}_n \quad \forall n \in \mathbb{N}_0, \quad n < N_{\text{LSP}} \tag{4.14}$$

corresponding to the Euclidean depth of the triangulated features using both, the laser plane and the camera. These are usually supplied in the form of a large, real vector size N_{LSP}.

From these data the local, 3-D position of all range data in can be rapidly computed as follows:

$${}_{\text{LSP}}\boldsymbol{p}_n = \begin{bmatrix} \tilde{d}_n \cdot \sin\vartheta_n \\ 0 \\ \tilde{d}_n \cdot \cos\vartheta_n \end{bmatrix} \quad . \tag{4.15}$$

In addition, the 6 DoF of the tracked pose of the main camera S_{C} w.r.t. some world coordinate frame S_0 can be delivered for every dense depth image. In the end:

$${}_0\boldsymbol{p}_n = {}_0\tilde{\boldsymbol{T}}^{\text{LSP}}_{3\times 4} \begin{bmatrix} \tilde{d}_n \cdot \sin\vartheta_n \\ 0 \\ \tilde{d}_n \cdot \cos\vartheta_n \end{bmatrix} \quad . \tag{4.16}$$

4.3.4 Operating Range

The LSP may be considered as the Swiss army knife of the DLR 3D-Modeler as it can be easily configured for varied operating ranges. The parameters that determine its operating range are:

- whether it is mounted on crosshair or regular configuration,
- the angular field of emission of the laser plane,
- the intrinsic parameters of the camera (*i.e.*, its scaling factor and its AOV), and
- the pose of the laser plane w.r.t. its respective camera.

For instance, in Fig. 4.14 the potential range of view of the LSP in relation to the inclination of the laser plane is depicted, for constant basis distance between the main camera and the laser plane of 10 cm. Within the highlighted admissible area, I chose the a range of view of 15 to 100 cm. In Fig. 4.15 the resulting line projections in S_{I} are shown. Similar investigations—although more involved—can be conducted in the case of the dual, crosshair LSP.

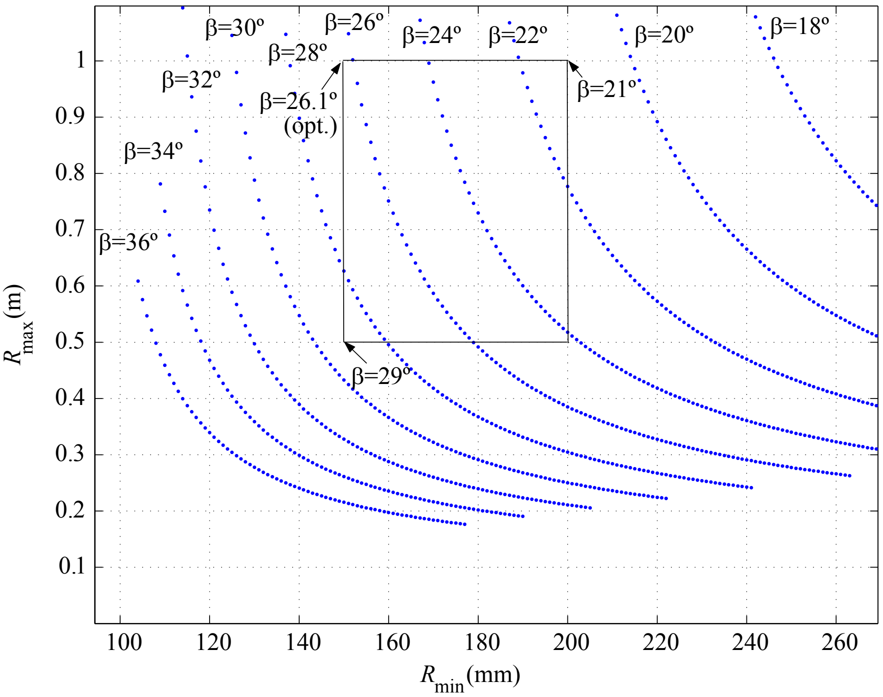


Figure 4.14: Minimum and maximum ranges R_{min} and R_{max} achieved by the LSP in relation to the inclination β of the laser plane, for a constant base distance $d = 12.8$ cm. Note that $\beta = \arctan\left(\frac{d}{2}\left(\frac{1}{R_{\mathrm{min}}} - \frac{1}{R_{\mathrm{max}}}\right)\right)$. I choose an optimal inclination of $\beta^{\star} = 26.1°$.

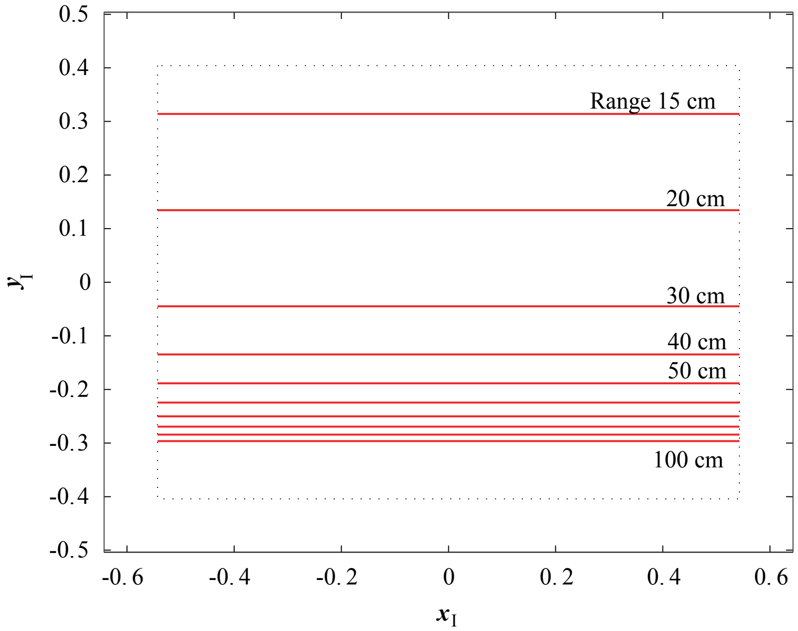


Figure 4.15: Expected range projections unto the normalized image frame for the laser plane inclination highlighted in Fig. 4.14.

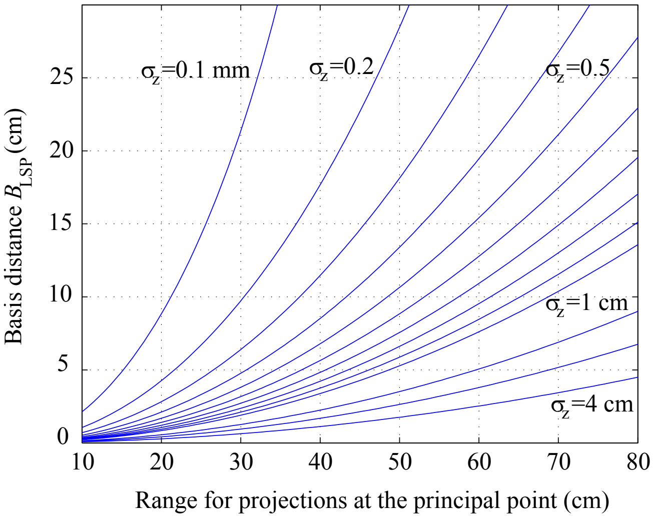


Figure 4.16: Potential ratio between the adopted laser-to-camera basis distance B_{LSP} and the range to the reconstructed feature location at the principal point, for 13 different target range accuracy levels.

In the case of the DLR 3D-Modeler, the baseline distance B_{LSP} between the laser plane and the main camera S_{C} is largely prespecified by the mechanical construction of the multisensory head. It is still possible, however, to conduct studies on the appropriate baseline in relation to the required accuracy. In Fig. 4.16 I show the expected range precision at a particular range to the scene, dependent on the base distance B_{LSP} between the laser plane and the main camera. As in the case of the stereo camera, we can estimate the expected range precision out of noisy image processing and the geometry of the LSP by using perturbation analysis (Haralick, 1998). In detail, the estimation of range R using the triangulation Eqs. (2.36) can be differentiated w.r.t. the projection location ${}_{\mathrm{M}}\boldsymbol{x}$; the linearized system relates now range precision σ_{R} with image processing noise σ_{I}. We solve the equations for the baseline distance B_{LSP} as follows:

$$B_{\mathrm{LSP}} = 2 \cdot \frac{f \cdot \sigma_{\mathrm{R}} - \sqrt{-R^2 \cdot \sigma_{\mathrm{I}}^2 + f^2 \cdot \sigma_{\mathrm{R}}^2}}{\sigma_{\mathrm{I}}} \tag{4.17}$$

where f relates to the actual focal length of the calibrated camera. Still, this formula should be handled with care, as it assumes that the intrinsic geometry of the LSP (e.g. from Sections 3.2 and 3.6) is perfectly known. If calibration is not dutifully performed, these levels of precision cannot be achieved. Note that potential inaccuracies by pose registration e.g. by the FaroArm Gold, the ARTtrack2 or the Kuka KR 16, are not included in these expectations.

In the end, I chose to lay out the LSP for it to bridge the operating range between the LRS and the stereo camera, *i.e.*, in the range from 15 to 50 cm. In detail, the LSP features 10 cm basis distance between the camera and the laser plane, 6 mm objectives on the camera and, consequently, 58° AOV of the LSP.

In the following figures I detail the spatial resolution of the final implementation of the LSP within the DLR 3D-Modeler; the data stems from actual measurements. In Fig. 4.17 the range (depth) obtained for every pixel projection in S_{M} is shown; the illustration takes radial lens distortion into account.

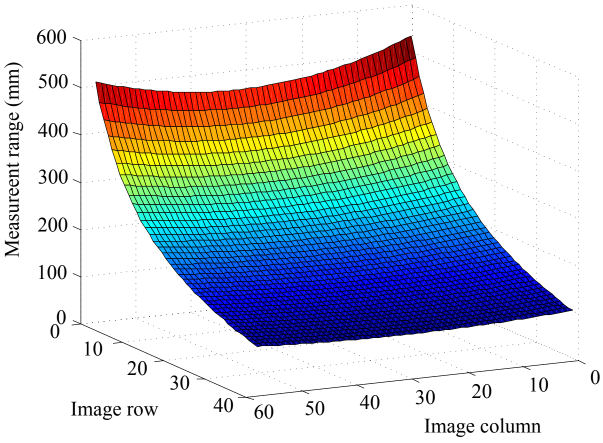


Figure 4.17: Measurement depth for every potential laser projection in the image memory frame S_{M}.

In Fig. 4.18 the azimuth angle (*i.e.*, the horizontal yaw angle or angular field of view AOV) for every pixel projection within S_M is shown.

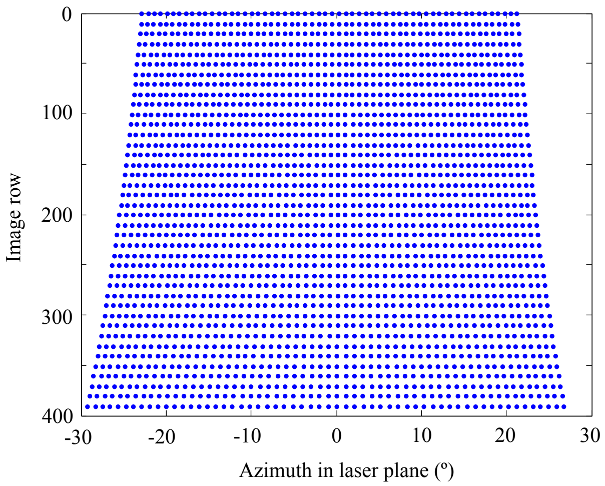


Figure 4.18: Azimuth horizontal angle (*i.e.*, yaw) for all potential laser projections in the image memory frame S_M.

In Fig. 4.19 the horizontal angular resolution (*i.e.*, in the azimuth angle) for every pixel projection within S_M is shown.

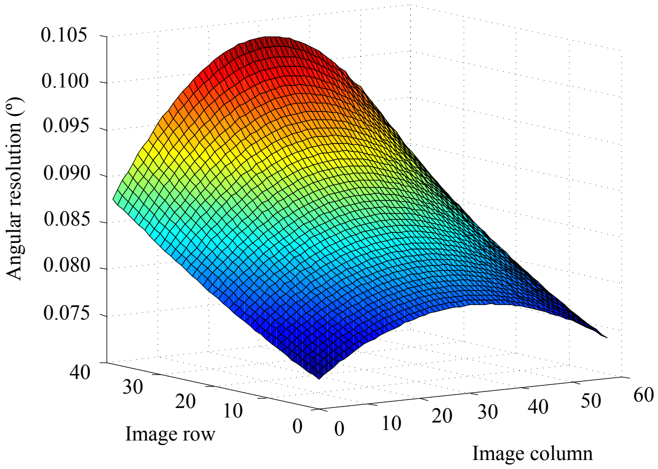


Figure 4.19: Angular resolution in the azimuth horizontal angle for all potential laser projections in the image memory frame S_M.

4.3.5 Range Estimation Accuracy

Experiments have been conducted to confirm expectations raised in Eq. (4.17) and Fig. 4.16. The results are presented in Fig. 4.20. Range precision is in sub-millimetric domain ranging from 0.3 mm at 15 cm range to 0.6 mm at 50 range; these are the standard deviations of Gaussian distribution resulting from repeated, independent measurements. Experiments at 1 m distance still yield high precision with standard deviation 1.5 mm.

Note that these results are in line with the expected accuracy of 0.3 mm in Fig. 4.16, with base distance of 10 cm and laser projection range between 30 and 40 cm at the principal point of the image. At peripheral image areas the noise model may not exactly hold, e.g. at close range (15 to 20 cm) precision seems to slightly worsen; this may be due to the bigger size of the projected stripe (as the laser plane is not really a plane) as well as to residual lens distortion effects at the peripheral region of the image.

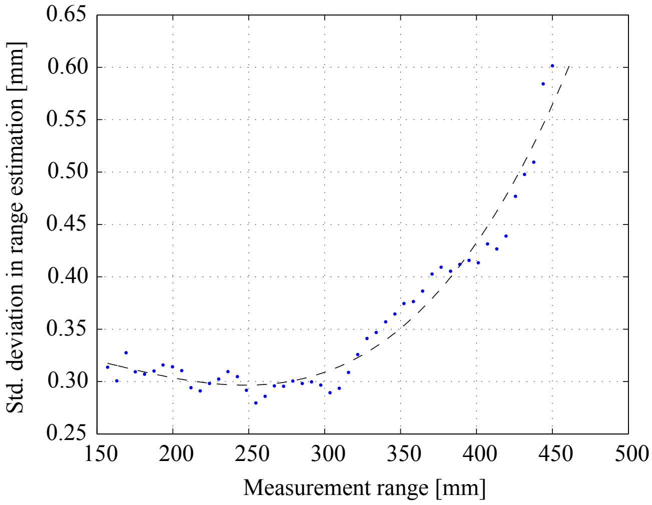


Figure 4.20: Measured range precision of the LSP. The regression function, *i.e.*, the best fitting polynomial model, reads $\sigma_{\mathrm{LSP}}(z) = 1.3639 \cdot 10^{-8} \cdot z^3 - 6.3523 \cdot 10^6 \cdot z^2 + 0.00063736 \cdot z + 0.32112$ mm. Refer to Fig. 4.22 for the joint representation of these data with the precision characteristics of the LRS and the stereo camera.

4.4 DLR Laser Range Scanner

4.4.1 Introduction

In Section 2.2.3 the geometry of the DLR laser range scanner (LRS) has been presented. In that section, the dissimilarities of both, the LSP and the LRS, have been addressed. The most significant dissimilarity is that the LRS works without external cameras, hence is self-contained. This allows for embedded, dedicated electronics that compute ranges out of position sensitive devide (PSD) projections. I will explain some particularities of these computations in the next section.

In addition, the convenient representation of its output data as well as the operating range of the LRS and experiments on its range precision are also being addressed.

4.4.2 Robust Operation

As mentioned above, the rotor of the LRS does not only contain the laser diode and the imaging optics, but it also contains computational electronics as well as components for external communication.

The laser reflection is detected by the PSD. It generates two electrical currents within the PSD, which relative intensity is proportional to the projection location of the laser light, *i.e.*, indirectly to the distance to the actual laser spot. The characteristic curve describing this dependency is obtained during the intrinsic calibration phase (Kielhöfer, 2003). In reality, this relationship is subject to other factors like the background brightness of the scene and the correct detection of the laser light in the first place. In the case that the laser spot cannot be detected, the laser beam intensity is logaritmically being risen in a closed loop with the readings of the PSD. It is clear that the precision of such an analog system is dependent on the eventual intensities as well as on the reflection properties and range of the scene.

Since range calculation is here embedded and is not open to external observation, the constructors deliver a value on the expected range precision at a particular distance and laser power—they coin it "quality value." This value takes some of the abovementioned limitations into account. They are delivered along with every single range measurement. What is more, void range measurements where laser projections have not been found are indeed marked as negative measurements. This is a very useful feature e.g. in the realm of robotic exploration for the robot to be positive on the absence of obstacles.

4.4.3 Data Representation

It is of central importance for efficient 3-D data representation and eventual treatment to comply with one of the pre-defined range data representation types. These data instances are delivered at high rate, *i.e.*, normally at the camera's rate e.g. 25 Hz. In the context of the DLR 3D-Modeler four types of data instances are supported:

- Cartesian type: A 2-D array of ranges that are orthogonal to the sensing plane and sequentially registered in an equally spaced 2-D sensor grid.
- Perspective type: A 2-D array of ranges that complies with the perspective model presented in Section 2.2.1 on the basis of a normalized, rectified pinhole camera model, *i.e.*, $\alpha \triangleq \beta$, $\gamma \triangleq 0$, $u_0 \triangleq v_0 \triangleq 0$, and in the absence of optical distortion; all ranges are sequentially registered in an equally spaced, projective 2-D sensor grid.
- Cylindrical type: Ranges are registered in 2 DoF, viz. a single rotatory axis and its origin is translated on the same axis, on equally spaced distances and angles.
- Spherical type: Ranges are registered in 2 DoF, viz. two rotatory axis with fixed origin. The array of ranges is distributed in the two equally-spaced angles.

Since the range data of the LRS are naturally limited to the plane containing the rotating laser beam, I choose the cylindrical type of depth data representation on equally spaced angles, *i.e.*, with 1 sole DoF. First, the angular size and the resolution of the normalized sensor grid is defined with the parameters N_{LRS} (number of equally spaced angles), ϱ_{initial} (initial value of the cylindrical angle ϱ), and $\Delta\varrho$ (distance between adjacent angles ϱ). The silent angles of all measured ($\tilde{}$) range data $\tilde{d}_n$ are:

$$\varrho_n = \varrho_{\text{initial}} + n \cdot \Delta\varrho \quad \forall n \in \mathbb{N}_0, \quad i < N_{\text{LRS}} \quad . \tag{4.18}$$

Apart from the metadata mentioned above, N_{LRS} real values corresponding to the Euclidean depths of the triangulated features are streamed:

$$\tilde{d}_n \quad \forall n \in \mathbb{N}_0, \quad n < N_{\text{LRS}} \quad . \tag{4.19}$$

These data are usually supplied in the form of a large, real vector size N_{LRS}. Additionally, quality values referring to the expected precision of LRS range data are delivered:

$$q_n \quad \forall n \in \mathbb{N}_0, \quad n < N_{\text{LRS}} \quad . \tag{4.20}$$

From the depth data in Eq. (4.19), the local, 3-D position of all range data in can be rapidly computed as follows:

$$_{\text{LRS}}\boldsymbol{p}_n = \begin{bmatrix} \tilde{d}_n \cdot \sin \varrho_n \\ 0 \\ \tilde{d}_n \cdot \cos \varrho_n \end{bmatrix} \quad . \tag{4.21}$$

In addition, the 6 DoF of the tracked pose of the LRS S_{LRS} w.r.t. some world coordinate frame S_0 can be delivered for every dense depth image. In the end:

$$_0\boldsymbol{p}_n = {_0}\widetilde{\boldsymbol{T}}^{\text{LRS}}_{3\times 4} \begin{bmatrix} \tilde{d}_n \cdot \sin \varrho_n \\ 0 \\ \tilde{d}_n \cdot \cos \varrho_n \end{bmatrix} \quad . \tag{4.22}$$

4.4.4 Operating Range

Quite different from the flexibility of the LSP in Section 4.3.4, the LRS is innately limited to the operating range predefined during its design and construction.

Its operating range limits in actual experiments are 80 and 250 mm, although the manufacturers (in lab conditions) claim to detect laser reflections between 50 and 500 mm range. This discrepancy is surely related to the abovementioned dependency of the LRS on the surface properties as well as on the ambient illumination. On the other hand, the field of view of the LRS is still extended due to its broad opening angle of 270°.

4.4.5 Range Estimation Accuracy

Average accuracy values on white paper are provided in Fig. 3.4 within (Kielhöfer, 2003). Depending on the range, its ranging accuracy is given by a standard deviation between 0.1 and 2.5 mm at 25 cm range, cf. Fig. 4.21. This accuracy levels at close range are remarkable with a base distance between the laser emitter and the PSD (receiver) of only 20 mm.

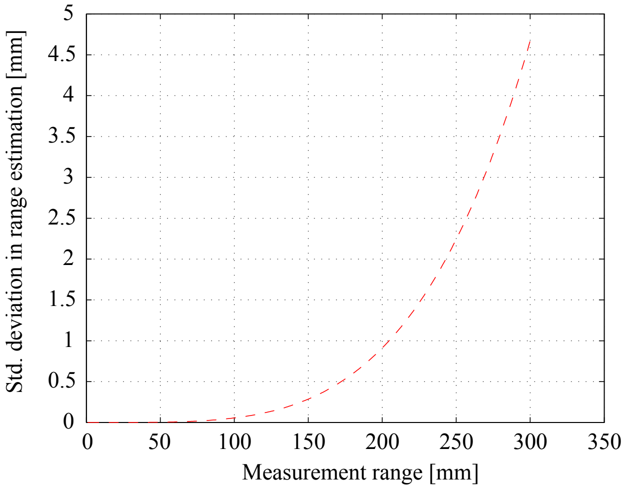


Figure 4.21: Regression function from experiments on the range precision of the LRS. The best fitting polynomial model, reads $\sigma_{\text{LRS}}(z) = 5.52 \cdot 10^{-10} \cdot z^{4.01}$ mm, see Eq. (3.2) in (Kielhöfer, 2003). Refer to Fig. 4.22 for the joint representation of these data with the precision characteristics of the LSP and the stereo camera.

4.5 Global Range Estimation Accuracy

In this section I aim at validating the assertion in Section 2.2 where I stated that the DLR 3D-Modeler combines sensors that complement each other, evading weaknesses that arise e.g. when the laser beam of the LRS gets too weak to be measured by its PSD and it ends up delivering very noisy data. In this section I only discuss the precision characteristics of the sensor components.

Fig. 4.22 unifies the accuracy curves for stereo vision in Fig. 4.4, for the LSP in Fig. 4.20, and for the LRS in Fig. 4.21. Note that, for more convenient representation, the scale of the current plot is logarithmic.

It can be observed from this joint plot of accuracies that the accuracy levels achieved by the LRS are unmatched, viz. in the order of a tenth of a millimeter (without consideration of the absolute pose tracking system errors nor of its extrinsic calibration w.r.t. it). The accuracy of range data by the LRS, however, decays rapidly with its distance to the scene owing to its short baseline of only 20 mm. Further, the LSP has been configured to fairly constant accuracy in the order of 0.3 mm between 15 and 35 cm. This setup for the LSP perfectly bridges the gap between the LRS operating range and the operating range of stereo vision. The latter may start out at 30 cm range but then its precision is an order de magnitude worse than the precision of the LSP. Stereo vision by SGM, however, delivers dense depth images at extended ranges of up to 2 meters, thus properly finishes off the desired scanning range of the DLR 3D-Modeler.

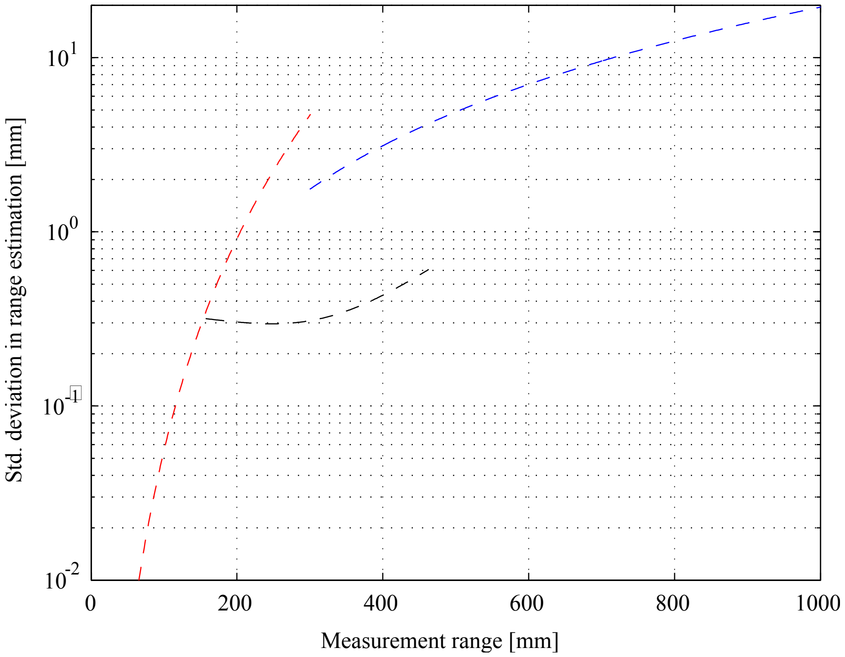


Figure 4.22: Joint representation of expected range accuracy of the LRS, the LSP, and the stereo vision method SGM, in relation to their range to the scene.

4.6 Supplementary Procedures

At this point we are in a position to deliver accurate, raw 3-D data of the scene in front of the DLR 3D-Modeler in the form of pointclouds. This is, however, not the whole story, as pointclouds are useless by themselves. In this section I go on to the application level, mentioning three extensions of the presented work (by my colleagues Tim Bodenmüller, Michael Suppa, Simon Kriegel, Wolfgang Sepp, and Ulrich Hillenbrand) that build on accurate 3-D data in the form of pointclouds in order to fulfill major tasks:

- *Streaming surface reconstruction from real-time 3-D measurements* in the form of e.g. triangle meshes is perhaps the closest extension that is indeed demandable within the context of 3-D modeling with the DLR 3D-Modeler. As mentioned above, in most applications 3-D pointclouds are not valid 3-D models. In the case of hand-guided 3-D modeling, online suface generation supports the user's experience when digitizing complex objects if it is deployed for visual feedback. The rendering of raw 3-D measurement data is, of course, also possible, but it is inadequate because it does not allow for correct shading nor for the user's to be able to assess the quality of the eventual model. Real-time surface reconstruction is challenging because a priori knowledge of the shape or size of the object are missing. In addition, surface reconstruction fixes the varying density issue, as in the case of hand-guided systems the user frequently choses to revisit past areas (aiming at completeness due to past occlusions or concavities), which yields irregularities in the density of points at the surface.

Streaming surface reconstruction can also exploit the expected continuity of surfaces in order to cancel out sensor noise. A convenient surface model is a triangle mesh, which involves the reduction of the pointcloud's density to 2-D homogeneity. In addition, this method does not only allow for virtual shading but for realistic model texturing as will be presented in Section 5.4.6 (Bodenmüller, 2009).

- Many applications would benefit from automatization whenever 3-D modeling is a repeated task, especially in industry. Robotic manipulators like the Kuka KR 16 or the DLR Lightweight Robot III can be used to move the DLR 3D-Modeler around the unknown object—delivering accurate motion information at that. The problem of estimating the optimal motion of a sensor for gathering 3-D data is known as the best-next-view problem. Of course, to optimally solve that problem it is critical to know the exact geometry of the object; this is, however, absurd if the purpose of the procedure is precisely to obtain the object's geometry. Solving the next-best-view problem in this context is a challenging task that has been addressed using the DLR 3D-Modeler in (Suppa, 2008; Kriegel *et al.*, 2012).

- Ever since the advent of widespread 3-D sensors like Microsoft's Kinect$^{\mathrm{TM}}$, it has become clear that 3-D information allows for a new realm of applications compared to 2-D sensors like simple cameras. In (Sepp, 2008) the author used the DLR 3D-Modeler to obtain 3-D pointclouds that allow him to track known objects in a novel, more robust way, using 2-D cameras, as 3-D information of the object readily allows for 6 DoF pose tracking.

- The DLR 3D-Modeler is mounted at the top of the humanoid robot "Justin" as its perception head, see (Borst *et al.*, 2009) and Section B.2.1 within Appendix B. In (Hillenbrand, 2008) one of the best known robotics demonstration at my lab is addressed: "Justin" manages to recognize known objects on a table (carafes, bottles, jars, and glasses) out of raw, 3-D data from stereo vision; after that, "Justin" is able to manipulate these objects to autonomously prepare drinks. Pose clustering on a 6 DoF parameters (pose) space on raw, 3-D data does the trick to accomplish the task of object recognition as well as the estimation of their relative poses.

4.7 Summary

In this chapter I focused on the software computations that are regularly being performed in realtime out of raw data (e.g. passive images or laser projections), for the different sensor components of the DLR 3D-Modeler in order to deliver depth information that, together with external pose tracking, lead to 3-D pointclouds.

First, I address stereo vision by the two cameras of the DLR 3D-Modeler. I give an outline of the geometry of two views, which is basis for an introduction in the stereo reconstruction algorithm used at the DLR 3D-Modeler: the semiglobal matching algorithm detailed in (Hirschmüller, 2008). Stereo vision is a convenient sensing modality due to its passivity w.r.t. the scene (*i.e.*, it does not affect the scene not even by projecting light unto it). On the other hand, however, its fair range precision may not allow for accurate 3-D reconstruction of the scene in many applications. In the context of the DLR 3D-Modeler, stereo vision is deployed as a long-range sensor for dense exploration where coarse depth accuracy suffices.

After that, the operation of the perhaps main depth sensor of the DLR 3D-Modeler is detailed, the laser stripe profiler (LSP). Depth computation out of laser light projections is in this case dominated by the difficulty of proceeding when the cameras ought not to be filtered to laser light—this fact was chosen as a requirement during the conception of the DLR 3D-Modeler in the first place. I present the novel method originally introduced in (Strobl *et al.*, 2004); it tackles the problem by a cascade of subsequent validation stages that starts out from a stripes detector on the red channel of the images using the Sobel filter.

Next, the operation of the laser range scanner (LRS) is explained. The LRS is a self-contained sensor that works without external cameras. Embedded electronics compute ranges based on a previous, in-house intrinsic calibration process. It is neccesary to know about its internal operation when choosing the desired sensor component for a particular task as well as in order to realize the expected accuracy levels of the sensor.

As a matter of fact, I present experiments that show the expected accuracy levels of the three sensors mentioned above. It comes as no real surprise that sensors operating at closer range excel in accuracy compared to sensors operating at longer range. Still, I sensibly combine optimal operating ranges of the different sensor components of the multisensory DLR 3D-Modeler in order to facilitate the required prior choice of sensor principles, see Section 4.5.

I conclude this chapter with a degree of humility: It is clear that 3-D pointclouds only represent the starting point for more advanced algorithms capable of tasks like 3-D modeling, object recognition, or scene exploration. I realize that my task ends at the beginning of other tasks. With this in mind, an standardized representation of depth data is provided for every single sensor component. Additionally, in Section 4.6 I list higher-level algorithms developed by my colleagues at DLR in the context of the DLR 3D-Modeler.

"We see because we move; we move because we see."
—James J. Gibson, The Perception of the Visual World, 1950

5
Visual Pose Tracking

5.1 Introduction

Several factors like object self-occlusion, object size, or limited field of view make it impossible for a 3-D modeling system to acquire a complete model in a single measurement step; this is especially true in close-range. Multiple views (or multiple sensors) are required to subsequently merge data to a single 3-D model. The prevalent approach is to measure the position and orientation (pose) of the sensor while acquiring range data, thereby registering multiple views, potentially in realtime, see Section 1.4.3. A range of tracking systems, robotic manipulators, passive arms, turntables, CMMs, or electromagnetic devices are commonly deployed for this purpose, see Section 5.2. Indeed, the DLR 3D-Modeler has been only deployed together with robotic manipulators or infrared tracking systems so far. These options are inconvenient for three reasons: *First*, they limit user's mobility; *second*, they are subject to accurate synchronization and extrinsic calibration, which are cumbersome, error-prone processes (Bodenmüller *et al.*, 2007; Strobl and Hirzinger, 2006), and what is more they cannot be rearranged; *last*, it turns out that external positioning systems almost always represent the largest and most expensive part of the 3-D modeling system.

In this work I present the required algorithms for robust and accurate pose tracking of close-range 3-D modeling devices at a high data rate, by the use of the images captured by their own cameras; cameras are already present in most of these devices after all. In this way, the three limitations mentioned above are lifted. Note that this potential extension is just another neat by-product of our original decision to omit laser-light filters on the cameras.

Cameras are preferred sensors in many areas because they are light, affordable, consume less energy, allow for a very accurate parametrization of its simple operating model, and still they gather a plethora of information (both radiometric and geometric) within a single, rapid measurement. Further benefits exist: cameras are non-contact sensors, thus free-floating, and passive since they do not need to project or exert action on the environment. Note that potential pose tracking from its own images would inherently become calibrated and synchronized with further image-based sensing.

And yet visual pose tracking is a hard problem as geometric information becomes entangled in radiometric and perspective geometric issues. Following distinct regions of interest (**feature-based tracking**) is a popular and efficient technique to overcome this problem. Still, images produce many features that have to be handled frugally if we want to operate in realtime. In addition, feature tracking in close-range is especially demanding because features move faster than in medium- or long-range because they are also affected by camera translation. To make matters worse, highest accuracy is necessary as cameras feature small angular fields of view, which call for the concatenation of relative measurements so that errors readily accumulate.

In order to alleviate difficulties in feature tracking, I propose two novel schemes: either leveraging an inertial measurement unit (IMU), calibrated and synchronized with respect to (w.r.t.) our system to complement visual tracking (Strobl *et al.*, 2009a; Fleps *et al.*, 2011), or adopting the Active Matching paradigm for more efficient tracking (Davison, 2005; Strobl *et al.*, 2011). In order to increase accuracy, **graph-based, nonlinear optimization** (keyframe-based bundle adjustment) on *relative* transformations and measurement constraints, **parallel computing** of front-end, back-end and other sub-tasks, feature-based stereo vision, as well as **loop-closing detection** for dead reckoning error compensation are employed. Even in the case that everything else fails, **appearance-based recognition** of older features is provided so that pose tracking can be resumed.

Finally, since manual 3-D scanning requires visual feedback to the user in realtime, a streaming surface reconstruction method is presented that delivers realistic 3-D models *in-the-loop* during scanning as well as refined models promptly after loop-closing corrections.

The remainder of this section is as follows: An extended survey on related 3-D modeling devices, their pose tracking techniques, and visual pose tracking in general is delivered in Section 5.2. In Section 5.3 I motivate the layout of the approach to visual pose tracking that will be presented in Section 5.4; its algorithms allow now for the DLR 3D-Modeler to track its own pose in 6 DoF, in realtime. The approach will be validated with experiments in Section 5.5.

5.2 State of the Art

This section extends the State of the Art Section 1.4, now focusing on 3-D modeling work with regard to their 3-D data registration concept—provided the system meets our requirements, *i.e.*, is non-contact and relatively lightweight. I shall focus on mature, commercial systems; I only mention research work in the areas where commercial systems are missing. When addressing 3-D data registration by visual pose tracking, due to the novelty of the approach I elaborate on its real-time variants irrespective of their potential application to these types of systems.

5.2.1 3-D Data Registration by Scan Alignment

Dense depth sensors that e.g. provide 2-D range images (2.5-D images) may yield rich surfaces that allow for raw 3-D data registration by 3-D matching, without the necessity for explicitly estimating sensor motion. *This is not possible, however, in the case of 1-D range images e.g. by laser stripe triangulation.*

3-D matching is a computationally demanding task because correspondence search is on higher dimensionality compared to traditional 2-D image registration. Additionally, data overlapping is required, which has to be detected in advance out of raw depth data and perhaps some probabilistic priors. For these reasons, scan alignment is nearly always being performed off-line, often in an interactive way with the user. The estimation usually involves an optimization in the form of the minimization of a particular distance metric between scans, being the ICP method in (Besl and McKay, 1992) the reference work. Different metrics and modifications of the original algorithm have been proposed for improved robustness against noise as well as efficiency (Coudrin *et al.*, 2011). With the recent advent of general-purpose computing on GPUs, real-time implementations of ICP have been presented, e.g. sequential multiscale ICP on RGB-D data (Kinect) in (Newcombe *et al.*, 2011a). In the same context, other authors opt for bootstrapping ICP by feature-based pose tracking for more robust scan alignment, see (Henry *et al.*, 2012) and Section 5.2.4.

It is worth mentioning recent work by Coudrin *et al.* for the company Noomeo SAS, see (Coudrin *et al.*, 2011). Even though the authors realize the convenience of visual pose tracking for online data registration, in their approach visual pose tracking merely serves as an initial estimation for subsequent ICP optimization because they use active 3-D modeling by densely projected patterns, which precludes concurrent visual feature matching. They can only ressort to interleaved stereo frames where the projected pattern is switched off, so that 3-D modeling and pose tracking are desynchronized. In the end, half of the images serve 3-D modeling whereas the other half merely serves as an initialization step for ICP.

5.2.2 3-D Data Registration by External Pose Tracking

Raw 3-D data registration poses an overdetermined problem where the space of unknowns comprises 6 degrees of freedom (DoF). It is common practice to take subsets of 3-D data to simplify the estimation problem, but it still remains a demanding one. In addition, its convergence is subject to a high degree of unpredictability as it is strictly dependent on the particular geometry being acquired. We would benefit from an independent procedure yielding an equivalent solution to the original 6-D matching problem. It is well known that the relative sensor pose estimation problem (6 DoF) yields that same solution, although represented in the camera reference frame instead of in the object reference frame.

The use of traditional absolute positioning systems attached to a 3-D sensor is arguably the most straightforward approach for solving this problem. Due to their robustness and accuracy, the systems listed below became widespread and are the dominant (commercial) 3-D modeling devices in close-range:

• *External, optical (infrared mostly) tracking systems* are used by Northern Digital Inc., Metris NV, and Steinbichler Optotechnik GmbH. Optical tracking systems detect and track artificial (e.g. infrared-reflecting) markers attached to the 3-D sensor. They seem convenient to hand-held operation due to the absence of a rigid positioning contact to the sensor. On second sight, however, the user eventually feels strongly limited because of their small tolerance to sensor rotation owing to visibility constraints (occlusion). Furthermore, since the spatial distribution of the markers is limited, the accuracy of orientation estimation is generally poor.

• *Passive arms* are used by FARO Technologies Inc., KREON Technologies, RSI GmbH, Metris NV, and ShapeGrabber Inc. The use or passive arms, or even robotic manipulators, is, of course, inconvenient to manual operation of the sensor. However, they are the most accurate option for pose tracking—subject to their accurate synchronization and extrinsic calibration w.r.t. the sensor. Price and size are prohibitive in many applications.

• *Electromagnetic positioning systems* are implemented by Polhemus Inc. These tracking devices resemble optical tracking in operation, but now it is not required for the sensor to maintain a free line of sight to any marker. However, accuracy is dependent on the distance to the electromagnetic emitter and its signal can be affected by e.g. metallic structures.

• *Turntables* used by Cyberware Inc. and Polygon Technology GmbH. These allow for inexpensive systems, but are limited to small, light objects and rarely allow for the generation of complete models.

External pose tracking does allow for accurate 3-D data registration in realtime, but all of the above absolute positioning systems have in common that they *represent the bulkiest and most expensive part of the eventual* 3-*D modeling systems*. Furthermore, they limit the system in mobility and flexibility, and are subject to accurate external calibration and synchronization. These strong limitations apply especially in the realm of robotics, where sensors are precisely meant to promote autonomy without imposing additional constraints.

5.2.3 3-D Data Registration by Visual Pose Tracking

Since digital videocameras are already present in most close-range 3-D modeling systems, the estimation of the sensor motion from its own video footage is highly desirable to avoid using external systems. Camera motion estimation is feasible because, on a static scene, camera motion is the only factor that accounts for varying perspective projection of the 3-D scene onto 2-D images. In addition, since visual pose tracking is in the camera frame, an external calibration step of the tracking system w.r.t. the camera is no longer required. Similarly, estimations become inherently synchronized with further camera-based sensing, dispensing with the need for meticulous synchronization. From this idea two variants emerged:

- *Low-rate, visual pose tracking* is used by Noomeo SAS in the Optinum™ scanners, probably as an initialization stage to scan alignment from dense range images.

- *High-rate, visual pose tracking* is achieved by the HandyScan 3D scanners of Creaform Inc. (also marketed as ZScanner® by Z Corporation).

The latter implementation lies close to our goal of high-rate pose tracking from a video stream. However, the necessity to adhere reflective markers to the objects is inconvenient. In fact, in a number of applications it is prohibited or impossible. Being one of the main motivations for using cameras the fact that they are non-contact, free-floating sensors, *i.e.*, effectively passive to the scene, it is counterproductive to rely on this type of adhesive markers. Furthermore, their dependency on active infrared (IR) illumination also entails limitations.

The DAVID-Laserscanner is a commercially available, very simple scanner that works without an external positioning system. The pose *of the laser projector* is estimated from images of a static camera that, at the same time, estimates projections depths by triangulation. The approach is fundamentally limited to a single view with potential, subsequent scan alignment.

For the remainder we concentrate on research work.

In Refs. (Hébert, 2001) and (Khoury, 2006) a self-referenced, hand-held crosshair laser stripe profiler was presented. Its stereo camera makes use of fixed marker points, actively projected onto the scene, and localizes itself continuously by stereo triangulation w.r.t. these points. Actively projecting marker points onto a scene is inconvenient and, furthermore, limits flexibility since the cameras must see the markers the entire time. In addition, both laser profiler operation and texturing are influenced by active illumination. The algorithm seems to lack robustness, and efficiency considerations are not reported.

Actual image-based, *passive* localization approaches for 3-D modeling do exist:

The approach in (Pollefeys *et al.*, 2004) uses projective reconstruction jointly with posterior self-calibration to estimate metric—yet unscaled—motion in uncalibrated image sequences. After that, bundle adjustment is used to refine the results. A similar approach in (Roth and Whitehead, 2000) does make partial use of a previous camera calibration for metric reconstruction. The approach is intended for dense stereo vision applications and is not real-time. Accuracy analyses are missing even though non-stochastic approaches to self-calibration compromise it.

It is worth mentioning the instant Scene Modeler iSM device by MDA Ltd., Space Missions in (Se and Jasiobedzki, 2008). The system produces 3-D models from hand-held stereo vision by registering views with scaled poses from visual pose tracking. In contrast to the objectives in this work, the system aims at mid-range operation using dense stereo vision. Stereo is computationally expensive and, therefore, frame-rate is low, which in turn makes pose tracking under unknown motion harder and essentially different from a high-rate variant. The problem is solved using SIFT features—which again are computationally expensive—as well as lower resolution footage.

We presented in (Strobl *et al.*, 2009a) the first hand-held 3-D modeling device for close-range applications that localizes itself passively from its own images in realtime, at a high data rate. In that work, pose tracking was optionally supported by an on-board IMU for more efficient feature tracking. In (Strobl *et al.*, 2011) we present an alternative tracking method that, inspired by the Active Matching paradigm (Chli and Davison, 2008), achieves remarkable tracking resilience without the need for inertial readings.

Finally, I mention a major development by Newcombe and Davison on 3-D modeling from dense images by concurrent simultaneous localization and mapping (SLAM), so-termed DSLAM, in (Newcombe and Davison, 2010; Newcombe *et al.*, 2011b). DSLAM aims at considering every single pixel of the video stream for structure estimation and interleaved pose tracking, maximizing information gathering and overall performance at that. It is hard to explicitly do without distinct features (cf. Section 5.2.4-I.) as features are, by definition, invariant under several aspects and, therefore, can be better discriminated than other areas of the image. Consequently, the method is currently limited to confined viewpoint areas and constant lighting conditions as it assumes brightness constancy (surface smoothing priors are introduced to partly relieve of this limitation). Still, viewpoint limitation is certainly unsuitable if full-body 3-D modeling is intended. The current implementation is computationally very costly, leveraging on general-purpose computing on GPUs to achieve real-time performance. Despite all that, DSLAM already reached improved performance concerning resilience to erratic camera motion, pose tracking accuracy (albeit unproven in experiments) and, most importantly, concerning its low hardware requirements, namely a single camera and a commodity computer featuring a GPU.

5.2.4 Visual Pose Tracking in Realtime

Visual pose tracking is a hard problem because, in geometric terms, images merely convey 2-D information that originally stems from a higher dimensional space, e.g. 6 DoF of camera pose, full geometry of the scene structure as well as the intrinsic geometry of the camera(s). It is often just one among the latter parameters that we are interested in, yet *still have to infer them all* from 2-D images. This dimensionality reduction renders the problem often unsolvable using a single image. It is by increasing the dimensionality of gathered data—either taking separate measurements or taking them successively in time (as model parameters change)—that we can draw a distinction between the original, unknown parameters themselves, and infer their respective values. In doing so, we regularly exploit prior knowledge e.g. on the rigidity of the scene, on Euclidean geometry and on perspective projection.

In particular, there is a prevalent ambiguity in scene structure and camera pose estimation. For instance, in the event of detecting projections of an unknown object, it is impossible to discriminate between object size and camera range to that object. It is indeed a chicken-and-egg problem that also characterizes research in SLAM: motion estimation (localization) is straightforward on known 3-D geometry, whereas 3-D geometry estimation (mapping) in turn asks for known camera motion. As mentioned above, tackling the problem of SLAM is solved by integrating data in time, when some parameters vary (e.g. camera motion, *i.e.*, apparent perspective distortion) to differentiate them from others (e.g. static scene geometry).

To make matters worse, many applications require estimations in realtime, e.g. at 30 Hz. On the one hand, it is important to realize that less applications require a complete solution in realtime, but only part of it—the full solution can be readily delivered delayed in time. On the other hand, parts of the solution are really being required in realtime and, therefore, efficient methods are in demand. Temporal priors e.g. on the dynamics of the system can be of use for improved performance.

In this Section I address three key aspects for designing real-time visual pose tracking algorithms, listing reference works at that.

I. *How to parameterize/interpret scene structure?*

II. *How long to memorize associative visual data?*

III. *Which calculations for real-time performance?*

I. Feature-Based vs. Dense Tracking. A picture might well be worth a thousand words, but then not all visual information is created equal. Depending on the task at hand, some image regions convey more information than others (Brady, 1987; Torr and Zisserman, 1999; Irani and Anandan, 1999). Visual information can then be reduced to regions of interest (points or corners, edges) that still allow for highly accurate inference. In our context, these features represent the *Merkwelt* necessary for pose tracking. As most of these regions of interest can be described in very concise, parametric ways, methods following this paradigm ought to be more efficient than *direct methods*, which compute pixelwise from dense, raw image data.[1] Furthermore, these regions are more invariant to viewpoint location (e.g. concerning light reflection) and varying lighting conditions, which allows wide baseline matching to increase accuracy. Last, estimation on these separate regions is largely uncorrelated, *i.e.*, statistical independence holds (unlike when using direct methods) and, therefore, optimal estimation using maximum likelihood methods is warranted. Admittedly, the feature-based estimation paradigm entails limitations on its own, like the feature selection, scene understanding and data association issues. In general, feature-based methods are being preferred when designing visual pose tracking algorithms.

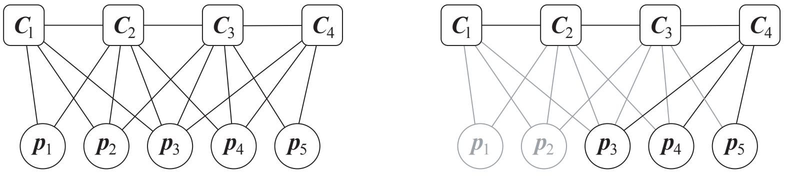
Feature-based methods utilize interest operators to *detect* salient/distinct regions of the images, *i.e.*, fiducial points or features at repeatable, stable locations despite change of viewpoint. Salient regions arise either from texture or from geometry (e.g. object corners). In general, features from (planar) texture are preferred since corner projections are not invariant to viewpoint location e.g. due to self-occlusion. Well-known detectors are: Harris-Stephens (Harris and Stephens, 1988) or Shi-Tomasi (Shi and Tomasi, 1994), the Laplacians LoG, DoG or DoB (Marr, 1982), MSER (Matas *et al.*, 2002), SUSAN (Smith and Brady, 1997), SURF (Bay *et al.*, 2008), FAST (Rosten and Drummond, 2005) and AGAST (Mair *et al.*, 2010a). Additionally, an operator for invariant *description* of these features is needed to be able to discriminate features against each other. Well-known descriptors are: planar, oriented patches (Davison and Murray, 2002), SIFT (Lowe, 1999), GLOH (Mikolajczyk and Schmid, 2005), HOG (Dalal and Triggs, 2005), SURF (Bay *et al.*, 2008), CenSurE (Agrawal *et al.*, 2008), BRIEF (Calonder *et al.*, 2010), BRISK (Leutenegger *et al.*, 2011), FREAK (Alahi *et al.*, 2012) and KAZE (Alcantarilla *et al.*, 2012). We speak of feature tracking when these descriptions are being matched in time, either starting from the anonymous output of a feature detector or based on camera/feature motion priors. In the former case, a current description is compared with a database of past descriptions, whereas in the latter case the current description is compared with a subset of that database (potentially just one description) within a reduced area of the image (Neira and Tardós, 2001; Chli and Davison, 2009a; Strobl *et al.*, 2011). Of course, the matching method is descriptor-specific, e.g. normalized cross-correlation for planar patches or computing Hamming distances for BRIEF descriptors.

Setting an optimal framework for detection, description and matching of features is subject to trade-offs: a *general* descriptor is expected so that it is invariant to change of viewpoint or illuminance; at the same time, feature descriptions have to be distinctive and, therefore, *specific* to particular features. Moreover, specific descriptors call for an exhaustive representation of features, which is in contradiction with the omnipresent requirement for *compactness and efficiency.*

[1] This conventional view is much-debated since the introduction of graphics units specializing in parallel computation.

Dense methods are less invariant to change of viewpoint or illuminance; however, they are alleged to be potentially more accurate and locally robust than feature-based methods because their representations (whole images) are more informative than just features. For instance, they allow for dense reconstruction of the environment (Newcombe *et al.*, 2011b). However, is a pixel-wise, perspectively-projected and rasterized 2-D abstraction of the scene the best possible representation of both 3-D scene and 6-D motion, really? On top of that, direct methods are being complemented with simplifying assumptions like brightness constancy. Does not a selective set of distinct points or edges, along with their robust associative information, make up a more informative representation for pose tracking? In any case, the implementation of dense methods on current hardware is demanding both on computational and electric power, which keeps them away from cheaper, widespread implementations and especially from constrained environments like space. It is worth noting recent research work that leverages feature-based methods in order to bootstrap, accelerate and robustify direct methods, see (Comport *et al.*, 2011; Henry *et al.*, 2012).

II. Visual Odometry vs. Visual SLAM. Both visual odometry by dead reckoning and visual SLAM (V-SLAM) incrementally estimate camera motion from video streams in realtime. For that purpose visual odometry exclusively uses the last subsequent image frames—potentially more than two,[2] but then critically the total number of images considered is limited. If an image gets outside this scope, its associated information will not be used for motion estimation anymore (Nistér *et al.*, 2004; Cheng *et al.*, 2006; Konolige *et al.*, 2007). On the other hand, V-SLAM may accumulate *all* information from past images, representing it either in the form of a graph of camera motions and measurements or in the form of a map, continuously updating them using present visual information (Fig. 5.1). A graph of motions and measurements is a light and exact way to accumulate raw data, whereas generating an actual map is more involved although potentially closer to the targeted estimations (typically feature and camera locations as well as their statistical moments). Visual odometry does not maintain this type of representations of the environment.



(a) Measurements used by SLAM (b) Measurements by visual odometry

Figure 5.1: Graph on the measurements potentially being used for pose estimation at $\boldsymbol{C}_4$ by SLAM (a) and by visual odometry (b).

[2] Using two frames for sequential motion estimation is subject to drift in absolute scale. It is only by using at least three overlapping frames of matched features that estimations may anchor in the original scale.

Considering older information (either in the form of a map or of a graph) is convenient in two respects: *First*, the relative pose estimation accuracy is essentially improved. Since the graph or the map relates to older camera stations, relative pose estimations w.r.t. that older stations will be more accurate than performing repeated, relative pose estimation over unrelated frames. In addition, virtual parallax will be bigger, therefore relative pose estimation more accurate on the assumption of constant image noise level. *Second*, the very existence of a map or a graph makes it possible to find older features again (loop closing), based either on their relative locations w.r.t. the camera or merely on visual descriptions; this is critical to further increase pose estimation accuracy. Indeed, it is only through closing loops that consistent graphs and drift-free camera motion estimation can be achieved in the presence of noise.[3] A downside to maintaining a map also exists: it is computationally expensive, as complex calculations are involved e.g. to obtain statistically optimal estimations. In addition, a considerable amount of memory is used.

When performing visual odometry the data quantity is limited to recent camera frames, which renders the estimation problem tractable; it is not necessary to make use of approximations like e.g. when using filtering to solve V-SLAM in realtime, see Section 5.2.4-III. Still, some tricks are used to boost performance and ensure robustness against outliers. For instance, it is common practice to compute minimal relative motion solutions from either 3 (Grunert, 1841; Wolfe *et al.*, 1991; Haralick *et al.*, 1994; Nistér, 2004a), 5 (Nistér, 2004b), 6, 7 or 8 (Stewénius *et al.*, 2006) feature points (depending on our knowledge of the structure and the camera), which are rapidly computed in closed-form, in order to obtain ballpark motion estimates. After that, the best solution may bootstrap a least squares optimizer minimizing reprojection errors (iterative refinement), potentially using more than two images (sliding window optimization yields optimal *motion* estimation, see (Nistér *et al.*, 2004; Nistér *et al.*, 2006; Mouragnon *et al.*, 2006)). In this context, scene structure is usually unknown, and consequently feature matching may be erroneous. In order to detect outliers, the latter minimal solutions to the relative motion problem are often within a geometric hypothesize-and-test framework like RANSAC (Nistér, 2003; Mouragnon *et al.*, 2009; Konolige *et al.*, 2007). The final least squares solution may also concern a robustified residual function.

From an operational point of view, the essential difference between visual odometry and V-SLAM can be summarized as follows: Whereas visual odometry estimates camera motion from (feature) correspondences between selected images, V-SLAM estimates camera motion from a conceptual matching between current image features and a representation of the accumulated system state, which in turn stems from past feature tracking. Since recently, it is generally acknowledged that hybrid solutions, running both processes potentially at different rates, are most effective as they complement one another (Klein and Murray, 2007; Konolige and Agrawal, 2008; Mei *et al.*, 2009; Williams and Reid, 2010; Strasdat *et al.*, 2011).

[3] Visual odometry systems may leverage IMU or GPS devices fusing data to overcome this problem.

III. The Back-End of SLAM. The SLAM problem can be divided in two tasks: front-end and back-end. The motivation for this division is the unfeasibility of achieving overall optimal estimation in realtime. Front-end calculations essentially deal with image processing and the intelligent arrangement of input data, and should run in realtime. Note that an intelligent arrangement of data may include the solution to the data association problem and that local pose tracking (or visual odometry) in realtime may be of necessity to that end. It is expected that front-end calculations are rather exact even if performed in realtime. On the other hand, back-end calculations concern the consistent representation of the data arranged by the front-end in the form of a graph of associated measurements or of a map. As the map grows and becomes interconnected, the complexity of this sub-task naturally grows—eventually becoming the bottleneck to optimally solving SLAM. Consequently, back-end methods dominated research on SLAM for the last decade.

To ameliorate the situation, methods that compute approximate solutions in realtime have been historically preferred. In recent years, however, a pertinent observation led to a different type of algorithms delivering far more accurate results: *Global geometric representation is rarely being required in realtime* (Klein and Murray, 2007), even though this assertion is of course subject to the final application.[4] More accurate estimations can be readily delivered *at a lower rate*, which suits present hardware developments just fine, paving the way to a plethora of methods trading off efficiency against accuracy (leveraging parallel computing hardware at that). As a side note, the geometric representation delivered by the back-end can in turn be used to support the front-end regarding e.g. data association (local loop closing). In the remainder, the most noted back-end methods are being addressed.

As a consequence of V-SLAM being preceded by SLAM, initial research adapted existing SLAM techniques (mainly using 2-D scanners) to visual input data, without actually realizing the two main challenges of V-SLAM w.r.t. traditional SLAM: First, digital cameras feature a *narrower field of view* than 2-D scanners, which makes direct triangulation harder and the time window for feature tracking shorter; it is now more vital than ever to be *accurate* in local, relative feature-based estimations, as many of them will have to concatenate for extended motion estimation. Second, visual data spreads now in 3-D, stacking up *larger amounts of data* than former SLAM methods in 2-D.

In fact, the first, best-known approach to V-SLAM by Davison in (Davison, 1999) used an Extended Kalman Filter (EKF), which delivered good, fast results if the map size was kept small concerning both, the number of features and the overall number of measurements. Early adopters rapidly noted this limitation, along with inconsistency in the estimations due to linearization errors and potential inadequacy of the Gaussian error models (Julier and Uhlmann, 2001). The preferred measure to ameliorate effects has been the decomposition of maps

[4] For instance, in our case of visual pose tracking for 3-D modeling with online visual feedback of the scanned object, a fairly accurate estimation of the whole motion history for timely 3-D scan display is of course required in realtime.

into submaps that become strictly uncorrelated from one another (Leonard and Feder, 2000; Guivant and Nebot, 2001; Eade and Drummond, 2007), which is at the cost of map accuracy.

The second major method for back-end estimation in V-SLAM is the Particle Filter (PF) (aka sequential Monte Carlo method) (Qian and Chellappa, 2004). A PF aims at more accurate and consistent estimations by representing estimation distributions as well as model noise by sets of particles. However, the size of the map that is manageable is still limited as the number of required particles grows exponentially with the number of features and their dimensions. A variant of the PF was proposed called Rao-Blackwellized PF (e.g. FastSLAM) (Sim and Little, 2006; Montemerlo and Thrun, 2007). The authors observe that feature measurements are naturally uncorrelated if they are conditioned to a particular path estimate of the camera. Consequently, feature maps can be efficiently computed using sparse EKFs associated to their respective pose particles. The principal drawback of PFs and its variants is the resampling step, which is introduced to eliminate improbable particles (that would otherwise naturally spread) in order to keep computational costs low; regrettably, the resampling step causes the lost of essential, small correlation densities (depletion problem) and consequently a loss of accuracy as well as eventual inconsistency.

As mentioned before, the two main drawbacks of exclusively using filtering methods (EKF, PF) for the back-end of V-SLAM are both their computational cost when dealing with a large number of features (map size) as well as their limited potential accuracy and inconsistency. In actual fact, the latter limitation can be effectively attenuated by increasing the number of measurements, but this is in turn unsuitable due to the former limitation in the size of the map (Strasdat *et al.*, 2010b). This limitation is inherent to filtering approaches for the following reason: Filtering is about maintaining a compact state-space estimation of currently useful parameters by marginalizing out past estimations (e.g. past camera locations) so that less computations and memory are required. In doing so, artificial correlations between parameters, e.g. estimated feature positions, have to be produced since their current position estimations depend on common past camera locations (when they were measured in the first place) that now have been removed from memory. Note that these correlations were non-existent at the moment of measurement, refer to the filtering graph in Fig. 5.2. Even though these correlations can be rapidly processed if the number of features is low, the complexity of the algebra of non-sparse matrices (full of correlations) is cubic in the number of features, which rapidly renders filtering approaches ineffective as cameras gather many more features than 2-D scanners. This could be avoided if the original measuring locations were still being considered, leading to a sparse graph of constraints. It is precisely the algebra of sparse matrices that is fast to solve after all.

From this, a different paradigm for the back-end of V-SLAM arose: graph-based nonlinear optimization in near-realtime. The authors of the seminal work PTAM in Ref. (Klein and Murray, 2007) utilize the well-known optimal algorithm for concurrent estimation of scene structure and camera motion called bundle adjustment (BA) (Triggs *et al.*, 1999). The basic idea was first formulated by Lu and Milios in (Lu and Milios, 1997), by which all motion data and measurements can be represented as a stochastic graph of nodes and edges (in V-SLAM: camera and feature locations and measurements, respectively). The goal is to find an optimal spatial configuration of the nodes that agrees with the constraints provided by the edges, by means of probabilistic inference (usually a nonlinear optimization). BA is known to be unsuitable for real-time estimation. However, the novel nature of off-the-shelf hardware featuring multiple cores for parallel computing gives the opportunity to perform BA in a real-time context: By computationally separating front-end and back-end calculations, BA can readily perform at lower rate without affecting local tracking performance at the front-end. It turns out that BA is less affected by both of the limitations of the aforementioned filtering methods. Still, its complexity linearly increases with the number of measurements and is cubic with the number of frames, which can quickly become prohibitive. It has been shown that, in the context of real-time SLAM, gathering many features per frame is preferable to processing many frames with less features, close in time (Strasdat *et al.*, 2010b). Therefore, the authors proposed a variant of BA called keyframe-based BA (kBA) (Klein and Murray, 2007; Mouragnon *et al.*, 2006), which selects, in a heuristic way, the most informative frames to consider, see Fig. 5.2 (c). kBA can be considered as an outstreched sliding window BA approach aiming at a fair distribution of computing power in space. If the number of keyframes is low, its complexity is effectively quadratic in the number of frames.

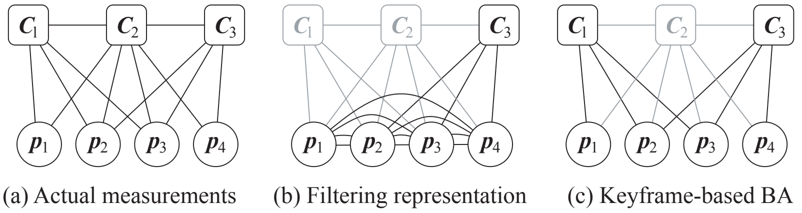


Figure 5.2: Filtering approaches (b), motivated by the Markov property, marginalize out past measurements (a) producing artificial correlations; keyframe-based approaches (c) avoid doing so discarding frames with lower information content.

Of course, static, regularly-spaced keyframes do not sort well with the heterogeneous nature of V-SLAM in mobile systems. In the spirit of kBA, more flexible approaches arised that focus resources on different parts of the state space. Since there are many more features than frames, pose-to-pose graph-based optimizations like FrameSLAM perform well *in large-scale* by marginalizing out feature locations (Konolige and Agrawal, 2008; Strasdat *et al.*, 2010a).

Marginalization may come at a cost of lower estimation accuracy if the optimized poses deviate too much from their initial estimations where marginalization took place. By formulating the problem in terms of relative transformations, the authors alleviate some of these effects. Another successful approach, called RSLAM, avoids computation by sticking with a topological representation of the localization problem (Mei *et al.*, 2010), leaving metric reconstruction aside. By using a continuous, relative representation of the camera's trajectory, BA computation becomes largely sparse (see RBA in (Sibley *et al.*, 2009)), which is especially efficient e.g. *when closing large loops*. In general, V-SLAM for mobile systems is a broad area where engineers ought to set up a task-oriented, hybrid algorithm combining different methods featuring e.g. local metric accuracy in realtime and robust loop closing on a topological representation, see Refs. (Strasdat *et al.*, 2011; Clipp *et al.*, 2010; Lim *et al.*, 2011). It is worth mentioning that filtering methods are not out of the race as they are believed to have a niche in systems with low resources *and* smaller map size. They can also take part in hybrid algorithms during Euclidean feature initialization or local tracking within the front-end, where by the way their explicit covariances can be of use to improved feature matching, see (Chli and Davison, 2008).

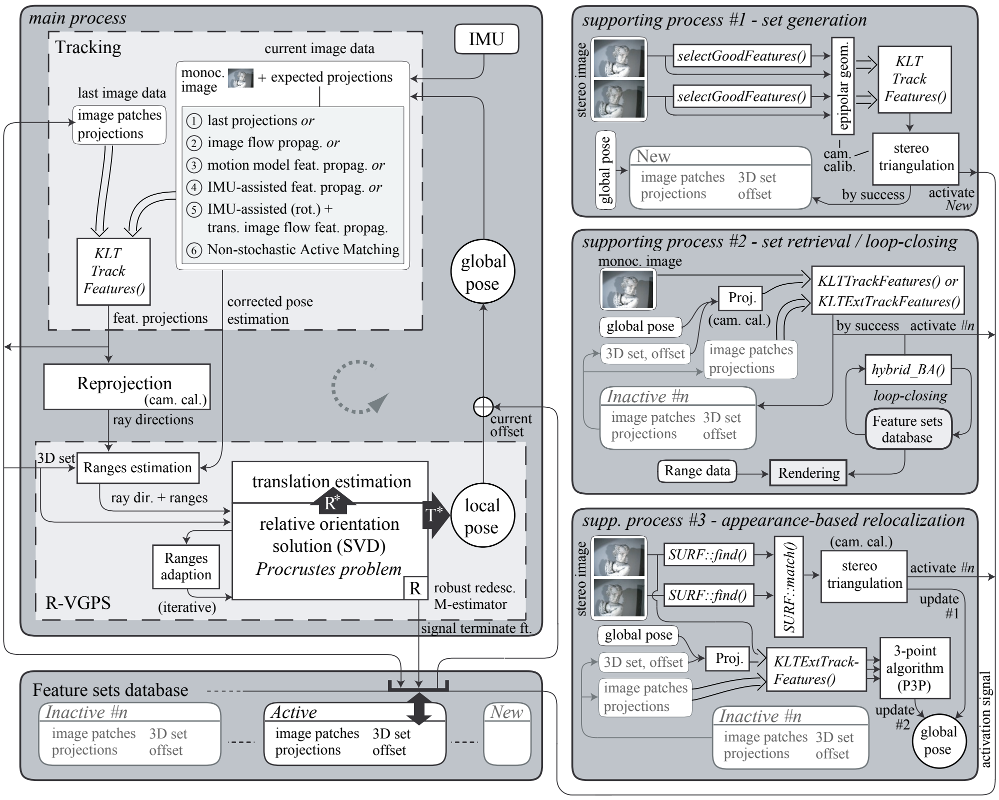


Figure 5.3: Block diagram for visual pose tracking. Four processes are differentiated. The feature sets database serves them storing data.

5.3 Design Considerations for Visual Pose Tracking in Realtime

Three major requirements arise for 3-D modeling using the self-referenced DLR 3D-Modeler: *1) real-time capability* for the methods to supply motion estimation, *2) high positioning accuracy* as required for 3-D modeling (compared to robotic manipulators or tracking systems *plus* their corresponding hand-eye transformations),[5] and *3) time-invariant estimations*, meaning that repeated scans should provide the same (high) accuracy irrespective of the scanning time.

In the light of these requirements, three major consequences follow: First, real-time capability implies both that motion estimations should be regularly performed within e.g. 40 ms (25 Hz) *and* that this should hold all the time, *i.e.*, irrespective of the motion history; we support this requirement on efficiency by the choice of a *feature-based approach* where the algorithm processes naturally salient, local regions of the images—recall Section 5.2.4-I. Furthermore, the requirement on constant efficiency irrespective of motion history merges with the requirement on time-invariant precision mentioned before, and points at the selection of a *non-filtering approach* for sequential pose tracking—refer to Section 5.2.4-III. Stochastic filtering approaches use knowledge of modeling errors (e.g. noise in image processing or uncertainty in the motion model) in order to increase precision. This feature is most relevant *if* that extra accuracy is really required; in fact, requirement #2 demands high accuracy for the system. However, it turns out that we are capable of highly accurate 3-D reconstruction of features on the scanning area by *feature-based stereo vision*, which yields highly accurate structure; this in turn allows for accurate pose tracking without the need for stochastic filtering. By using feature-based stereo vision, the algorithm only processes the strictly required 3-D structure information for accurate 6-D localization—extensive 3-D modeling is left for concurrent operation of the other sensors. The hereby achieved efficiency sorts well with the present paradigm of multithreaded, efficient computing.

This rationale (cf. Fig. 5.4) leads to the development of a feature-based, non-filtering pose tracking algorithm that requires occasional stereo initialization of natural features and monocular tracking of these features over time.[6] Monocular tracking yields 2-D motion of salient features in the image stream. Since stereo vision provides the 3-D geometry of these features, their 2-D motion is now solely dependent on perspective projection, *i.e.*, the (static and known) magnifying characteristics of the camera *and* its motion in 6 degrees of freedom (DoF). In order to extract camera motion I opt for an efficient solution to the relative pose estimation problem: the Visual-GPS method first presented in (Burschka and Hager, 2003), see Fig. 5.5. In addition, feature ini-

[5] Typical accuracies for robotic manipulators are $\sigma_\theta < 0.1^\circ$ and $\sigma_p \approx 0.5\,mm$; for IR tracking systems $\sigma_\theta \approx 0.25^\circ$ and $\sigma_p > 0.5\,mm$. The accuracy of the IR tracking system in orientation depends on the constellation of markers and is very limited.

[6] The features are supposed to be in rigid coupling; thus, in general, deformable objects or dynamic scenes are prohibited. In reality, moving objects are tolerated as a by-product of the robustified approach that will be presented in Section 5.4.3.

tialization and loop closing have to be governed by a data management scheme at a higher level, see Section 5.4.3 and Fig. 5.3. Crucially, feature tracking data are being stored, which enables intensive nonlinear optimizations at eventual loop closures, see Section 5.4.5.

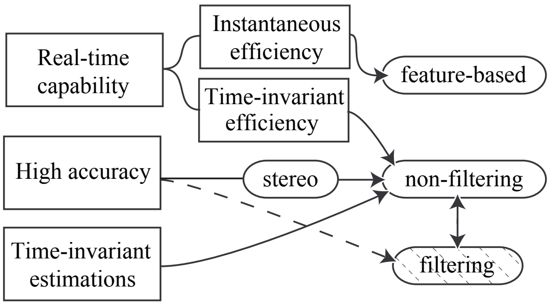


Figure 5.4: Requirements, implications, and consequences.

Note our accordance with the graph-based optimization paradigm in SLAM of reducing DoF in high-rate pose tracking for better performance, see Section 5.2.4-III. PTAM reduced them from $6+3\cdot N$ in general SLAM (N is the number of features) to 6 in PTAM (local pose tracking), estimating further DoF (mapping) and absolute motion in a concurrent thread, at lower rate, from selected keyframes. In our case mapping also relies on keyframes, but substitute repeated bundle adjustment by accurate, feature-based stereo vision. The latter is computationally cheaper and, furthermore, contributes absolute scaling—a prerequisite in 3-D modeling. Of course, in the event of loop closures, structure can be globally optimized by graph-based nonlinear optimization techniques, see Section 5.4.5.

5.4 Visual Pose Tracking with the DLR 3D-Modeler

In this section I present novel methods required for visual pose tracking of the DLR 3D-Modeler from its own images, in realtime. By doing this, concurrent 3-D data acquisition and registration is possible without the need for external reference systems, which signifies a remarkable improvement in flexibility and cost of the system. Taking the multisensory capabilities of the DLR 3D-Modeler into account, the methods have been specially tailored not to actively affect the scene nor, by implication, other 3-D sensors. In order to ensure mobility, the computational complexity of the algorithms has to be especially low for unrestricted concurrent operation of the other 3-D sensors on the same hardware.

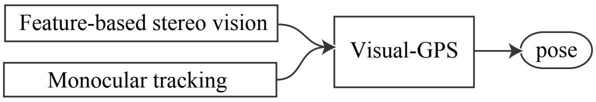


Figure 5.5: Feature-based stereo vision and monocular tracking serve Visual-GPS, which pays out with camera pose estimations.

5.4.1 Accurate Structure Estimation by Stereo Vision

Accurate knowledge of the sparse scene structure is a grounding pillar of this approach, as it increases efficiency and accuracy of local pose tracking. In addition, 3-D modeling requires accurate knowledge of the scene scale as well as passivity w.r.t. the scene, *i.e.*, the inclusion of artificial landmarks in the scene should be avoided. These requirements point at *feature-based stereo vision* for accurate scene structure estimation.

Conventional, sparse feature-based stereo matching relies on computationaly expensive Harris affine and DoG feature detectors that deal with affine transformations (Marr, 1982). In our case of parallel cameras on a short baseline, however, affine distortion can be neglected, which leads to the same assumptions of Shi and Tomasi in (Shi and Tomasi, 1994): *Good features to track* are extracted from the main camera image. Next, a larger number of features are extracted from the second image. Correspondence search is now restricted to a few locations within the epipolar band, which is also limited in disparity to obtain useful features on the near scene, refer to Section 4.2.2. Gradient descent optimization yields sub-pixel accurate feature matching, and the match with smallest difference in gradient patches is chosen. Feature triangulation is then performed by linear least squares and tested for consistency. Expected accuracy levels by stereo vision in our application domain are shown in (Strobl *et al.*, 2009a) and in Section 4.2.6.

Note that this feature initialization process cannot be performed in real-time; I opt for using a separate computing thread while concurrently tracking, in monocular, already initialized features in the former thread (in monocular) so that pose tracking is not interrumpted. Of course, at the very first initialization step no features are available and pose cannot be delivered. Here it is still necessary to monocularly track the potential features until their corresponding projections in the stereo image are found and triangulated, in order to seamlessly bootstrap the feature tracking algorithm presented next.

5.4.2 Efficient Monocular Tracking of Distinctive Features

The pose tracking algorithm basically compares a known set of 3-D features (result of last section) with their current *monocular* projections—with due regard to correct feature-to-projection correspondences. In order to correctly establish correspondences, two approaches are possible: *global feature tracking* searches for their appearance (e.g. a 2-D descriptor patch) within the whole image, whereas *local, sequential feature tracking* looks for them locally, in particular spots of the image after tracking them ever since their 3-D stereo initialization. I opt for the latter option, which is on the premise that features *slightly* move in successive images, which holds if the camera motion is moderate.

Both the already presented stereo-based feature initialization step and monocular tracking in this section are based on the KLT feature tracker (Tomasi and Kanade, 1991) owing to its potential efficiency and robustness. The implementation at (Birchfield) was extended for more efficient and robust operation in resource-limited platforms Mair *et al.* (2010b): First, convolutions (smoothing and gradients calculation) are applied locally around expected feature locations, which speeds up tracking especially in high-resolution footage. Second, good features to track (Shi and Tomasi, 1994) are selected on sub-regions of the image so that features are better spread. Third, the image processing functions within the multithreading library Integrated Performance Primitives (IPP) by Intel® are being extensively used. The fourth improvement concerns the predictive estimation of feature search areas in order to minimize computation and support tracking robustness.

Sequential feature tracking is a predictive feature search method that exploits probabilistic priors on their expected image projections in order to know where to focus processing resources in each image. These prior distributions ultimately depend on the 3-D location of the features and on the camera motion. Camera motion can be estimated from past measurements and further predicted using e.g. a motion model. 3-D structure, camera past motion estimation as well as its present motion model may however differ from reality to some extent, translating into “gated” image regions where each feature is expected to lie according to priors, see Fig. 5.6. The feature tracker seeks feature appearance matches within these bounded regions, hereby delivering temporal image displacements of features—keeping track of correct data association.

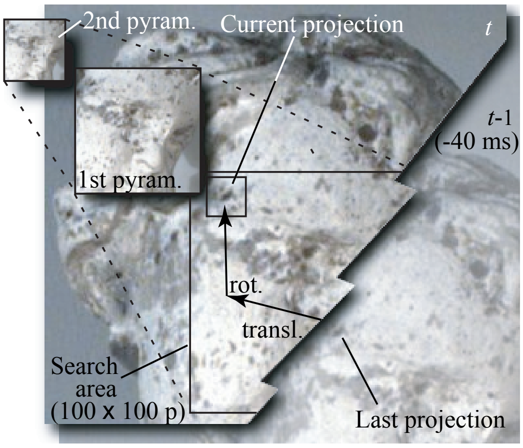


Figure 5.6: KLT tracker with big search area due to large expected displacements. Two levels of pyramidal representation are shown.

At close range, translation and rotation potentially cause projection displacements of similar size (the rotational component is dominant at long range). Displacements may add up to long distances (e.g. search areas of 100×100 pixels) that are beyond the real-time capabilities of the regular KLT tracker, even in its pyramidal implementation (coarse-to-fine matching, see (Bouguet, 2000) and Fig. 5.6). The pyramidal implementation applies the original implementation's gradient descent search also to coarser resolutions (higher pyramidal levels) of the original image pair—for conveniency, pyramidal levels differ in size at least by powers of two (octave steps). Matching at lower-resolution helps in the predominant case where, at original resolution, the search region is bigger than the 'basin of attraction' of the matching function, which depends on the chosen size of the feature template (typically between 7×7 and 11×11 pixels).

The use of similar-sized patches in lower-resolution images implies bigger, virtual 'basins of attraction' at the original resolution, which aim at the size of the original search region. By sequentially searching into lower pyramidal levels (higher-resolution images), absolute convergence is in theory guaranteed if the matching precision at the higher level is higher than the limits of the 'basin of attraction' at the lower level search—for all pyramid levels. The algorithm ideally ends up matching correctly at the lowest level—and matching is finished. Two significant limitations may apply:

- The bigger the search range, the more pyramidal levels have to be created. This can render tracking computationally too expensive—especially with a large number of features.

- Features following the Shi-Tomasi criterion are good features to track at the resolution where they were selected in the first place. At lower resolutions this does not necessarily hold anymore (distinctive small corners will attenuate and potentially disappear).

In fact, hand-held operation of the DLR 3D-Modeler can be highly dynamic and a simple motion model will not be able to narrow down feature search areas to admissible sizes. Therefore, both limitations apply.

In general, it is possible to lay out *optical flow prediction schemes* at different complexity levels (see Fig. 5.3):

1. Extrapolation of the last measured ($\tilde{}$) feature projections $\tilde{\boldsymbol{f}}$ in time; current estimated ($\hat{}$) projections are their last measured projections, *i.e.*, $\hat{\boldsymbol{f}}^t \triangleq \tilde{\boldsymbol{f}}^{t-1}$.

2. Extrapolation of the 2-D displacement (optical flow) of the last projections, *i.e.*, $\hat{\boldsymbol{d}}^t \triangleq \tilde{\boldsymbol{d}}^{t-1} = \tilde{\boldsymbol{f}}^{t-1} - \tilde{\boldsymbol{f}}^{t-2}$.

3. Extrapolation of the last camera 6-D motion, assuming either constant velocity or acceleration.

4. IMU-assisted camera motion prediction at time t using the last estimated camera motion at $t-1$ together with the integrated IMU outputs (rotational rate and linear accelerations) from time $t-1$ to t.

5. IMU-assisted camera *orientation* prediction together with optical flow-based translational extrapolation.

6. Individual feature displacement prediction by sequential Active Matching (Chli and Davison, 2008).

The optical flow prediction schemes that best perform in the case of general motion of the DLR 3D-Modeler are #4, #5 and #6. Since schemes #5 and #6 deliver similar accuracy to scheme #4 *at lower cost*, the formers are to be preferred and will be explained next.

Hybrid feature displacement prediction leveraging an IMU (optical flow prediction scheme #5)

In order to improve accuracy in feature displacement estimations (and as a result reduce search areas in size), the user may opt for improving camera motion estimation by rigidly attaching an IMU (scheme #4). Reduced search areas make KLT tracking more efficient, hence feasible in the presence of broader motion bandwidth.

Inertial sensors are being extensively implemented in vision systems because they perfectly complement off-the-shelf cameras both in measuring rate and in temporal precision: On the one hand, regular cameras take about 25 images per second and estimations from their images are, in principle, equally accurate all the time; on the other hand, IMUs yield data at kHz rates but their readings drift in time—they are only accurate for the short term. I am using interleaved IMU data to support the prediction step of feature projections.

Introducing an IMU in the system entails, however, costs and extra weight. In addition, careful synchronization and extrinsic calibration are critical. I propose an alternative scheme #5 that deskills this problem.

Even though translations and rotations are not commutative in general, their effects on feature projections are clearly differentiated: camera rotations cause projection flow irrespective of their range w.r.t. the camera, whereas camera translations only have a measurable effect at close range. Since the DLR 3D-Modeler operates at close range, both motion aspects will affect their optical flow. In fact, experiments show that jerky handling of the device entails speeds of up to 75 °/s (interframe 3 ° at 25 Hz) and 0.5 m/s (interframe 2 cm), which may yield 40 rotational and 50 translational pixels interframe optical flow at a typical range of 30 cm. Both components are potentially about the same size and neither of them should be neglected, see Fig. 5.7.

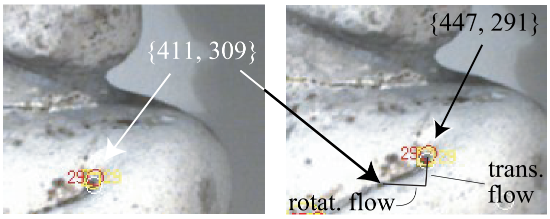


Figure 5.7: Feature displacement in the same image area in two consecutive images. A feature moves from {411, 309} to {447, 291} a distance of 40.2 pixels within 40 ms. 37 pixels are due to rotation, whereas 17 pixels stem from translation; some pixels cancel out.

Nonetheless, it is not the amplitude of the optical flow that is critical for successful tracking, but the certainty of its prediction. This is why informed, IMU-assisted predictions are more helpful than mere (uninformed) motion extrapolations. It turns out that hand-held operation allows for much better uninformed *translational* camera motion prediction than rotational. This is because of the bulky body of the sensor: It is easier for the user to rotate it with a facile twist movement than to linearly accelerate the whole sensor in any direction. This fact translates into the following relevant circumstance:

Translational feature displacements may be as big as the rotational ones, but they vary much less in time or, in other words, they allow for more accurate prediction by mere extrapolation.

In addition, many factors discourage developers from accessing *translational* readings of the IMU: First, translation is given in the form of its second derivative, which calls for repeated integration and for maintaining a camera motion model as well as an IMU model for its own drifts. Second, acceleration values are noisier than rotational rate ones, even after integration. Third, the gravity vector has to be estimated and subtracted all the time. Last, translational camera motion depends both on the rotational rate and on the linear acceleration readings of the IMU, whereas camera rotation directly corresponds to the integrated rotational rate at the IMU because it is rigid body motion. What is more, these relationships are defined by an external IMU-to-camera calibration process, which can be very simple for the rotational component but complex and prone to errors for the translational one (Fleps *et al.*, 2011).

These considerations led me to a novel, *hybrid* optical flow prediction scheme where the predicted displacements $\hat{\boldsymbol{d}}^t$ decompose into rotational and translational components: The rotational part $\tilde{\boldsymbol{d}}^t_{\text{rot}}$ is an informed estimation from the (integration of the) rotational rate of the IMU, whereas the translational part $\hat{\boldsymbol{d}}^t_{\text{tra}}$ is an extrapolation of the last *translational* optical flow $\tilde{\boldsymbol{d}}^{t-1}_{\text{tra}}$, for each feature. The latter stems from the subtraction of the last, informed rotational optical flow $\tilde{\boldsymbol{d}}^{t-1}_{\text{rot}}$ from the last, actually tracked displacement $\tilde{\boldsymbol{d}}^{t-1}$ as follows:

$$\hat{\boldsymbol{d}}^t_{\text{tra}} \triangleq \tilde{\boldsymbol{d}}^{t-1}_{\text{tra}} = \tilde{\boldsymbol{d}}^{t-1} - \tilde{\boldsymbol{d}}^{t-1}_{\text{rot}} = \tilde{\boldsymbol{f}}^{t-1} - \tilde{\boldsymbol{f}}^{t-2} - \tilde{\boldsymbol{d}}^{t-1}_{\text{rot}}\,. \tag{5.1}$$

To round off, there exists another appeal for this approach: Feature flow prediction now only depends on the last tracking results $\tilde{\boldsymbol{f}}^{t-1}$ and $\tilde{\boldsymbol{f}}^{t-2}$ as well as on IMU rotational rate readings (both $\tilde{\boldsymbol{d}}^{t-1}_{\text{rot}}$ and $\tilde{\boldsymbol{d}}^t_{\text{rot}}$); contrary to pose tracking or motion model estimations, these *measurements* are characterized by *low noise level.*

Of course, this hybrid scheme cannot be applied to temporarily lost features (e.g. due to occlusions, blur or limited field of view). In this case, the prediction scheme #3 takes over for that particular features until enough optical flow information is accumulated—and the hybrid scheme seamlessly recovers control.

Feature displacement prediction by Active Matching (optical flow prediction scheme #6)

Active Matching (AM) is a recent paradigm to feature tracking that yields considerable advantages w.r.t. traditional methods. It follows from the crucial observation that feature matching does not necessarily have to be a monolytic 2-D process, but might as well incur higher level estimations *during* operation, see Refs. (Davison, 2005; Chli and Davison, 2008, 2009a) and Fig. 5.8. In short, AM is putting feature matching *into the loop* of e.g. SLAM, performing feature by feature matching search while updating the system state as well as predicting measurement projections after every single feature matching process. This is a "dynamic" optical flow prediction scheme that can be compared to covert attention in humans, quickly browsing for informative locations within the image—contrary to overt attention linked to sluggish sensory saccades.

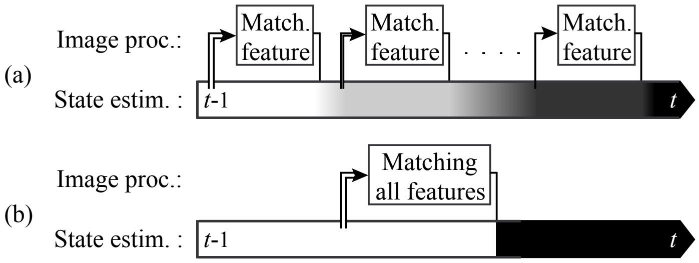


Figure 5.8: Traditional methods (b) take priors on feature projections once, where image projections are most uncertain. Active Matching (a) recursively updates (a representation of) the state after single feature matching so that feature projection priors can be more accurately estimated before a matching attempt starts (represented by the thickness of the arrows).

• *Built-in global consensus.* Instead of hypothesizing on correct data association after a monolytic feature matching process (Neira and Tardós, 2001), AM can readily walk down the sole correct hypothesis *during* feature matching by alternation of single feature matching and subsequent state update—for all features. In doing so, AM puts image processing *into the loop* of the search for global consensus by not processing areas of the image where features are not really expected in the first place. Feature matching will then be trapped in far less matching ambiguities. In order to cope with residual mismatches due to unavoidable image ambiguity, in (Chli and Davison, 2008, 2009a) the original authors make use of a dynamic Mixture of Gaussians (MoG) representation.
• *Less computation through less image processing.* AM leads feature tracking to process far smaller areas of the image. This is because of the paradigm shift from matching between images to matching between an image and the state, which is a far more informative description of the system history.
• *Less computation through guided search.* A stochastic representation along with the use of information theory make it possible for AM to quantify potential information gain. Some feature measurements will be more informative than others; by taking them first, the overall, eventual computational cost will be further reduced. Furthermore, this allows for anticipated termination of feature matching at a point of diminishing returns.
• *Estimation accuracy.* Real-time algorithms are usually tuned to perform at full capacity at the expense of e.g. a larger number of features being tracked, or more accurate feature matching or pose estimation results. Therefore, in real-time vision more efficient algorithms generally imply more accurate estimations.

The aforementioned aspects allow for more effective feature matching, but this is not the whole story because they are not for free as two important calculations must be repeatedly performed: First, the system state must be invariably updated after every single attempt of feature matching.[7] Second, making guided search decisions on information-theoretic grounds is associated

[7] A *joint* distribution on the expected feature projections may be used instead for efficiency reasons (Chli and Davison, 2008, 2009a).

with substantial computational costs. It is appropriate to question whether the alleged information gain really merits the extra calculations involved, finally yielding an overall more efficient algorithm.

It is worth noting that the aforementioned aspects are potential but not compulsory; it is readily possible to take advantage of some aspects and not of the others. For instance, since a large number of features skyrocket the cost of making exact decisions on guided search, an alternative algorithm called Fast AM (FAM) was proposed in (Chli and Davison, 2009a). FAM is cheaper than AM by making approximate, non-optimal decisions. On the other hand, the works in (Solà, 2007; Kaess and Dellaert, 2009) precisely employ guided search for selective feature matching but do not update the state after single feature matching.

In (Strobl *et al.*, 2011) we make out our case for a *non-stochastic version of AM* using accurate 3-D knowledge of the scene. We point to the fact that accurate knowledge of the scene (Section 5.4.1) allows for higher accuracy in pose estimation with the aid of even less features, which in turn renders the overall stochastic treatment of AM and local pose tracking insignificant and ineffective. In this section and in Section 5.4.3 I present feature and pose tracking algorithms tailored to this observation, aiming at more efficiency whilst accurate in their estimations.

Traditional AM in the context of SLAM is severely limited by the difficulty in distinguishing 3-D scene and camera motion, with the result that many feature measurements are necessary to discriminate between them. What is more, the expected projection areas remain uncertain, *i.e.*, big in size. This is quite the contrary in our case of accurate 3-D scene knowledge where immediate inference of camera motion from a few feature displacements is possible. Furthermore, all remaining displacements will only depend precisely on that newly estimated motion. For this reason I assert that non-stochastic AM with known structure is a *best case scenario for AM*, where eventual search regions can be reduced outright.

The approach is as follows: We aim at rapid, full (6-D) camera motion *preliminary* estimation using a minimal set of features thanks to our prior 3-D knowledge of them. This estimation will be used to update priors on feature projections yet to be measured. Only now very small residuals will allow for extensive KLT feature tracking in a highly efficient way.

The minimal set of known features for unconstrained motion prediction in 6 DoF comprises 3 perspective projection correspondences of non-collinear 3-D points—this was first described by Grunert in (Grunert, 1841). One of the principal findings in Section 5.4.2 was that, at close range, translational and rotational effects in feature displacements are potentially of similar size, but *translational effects can be accurately predicted using a constant camera velocity model.* From this I now propose reducing the required number of correspondences for full motion estimation from 3 to 2 (1.5 actually) by predicting camera translation from the motion model, so that only rotation (3 DoF) remains to be estimated. We expect that potential translation prediction errors will not corrupt this preliminary rotation estimation. My intention is that, after applying this preliminary motion estimation, the projections of the $N-2$ remaining features will fall within their respective 'basins of attraction' of regular KLT feature matching, e.g. 5 pixel radius. This rules out using expensive pyramidal representations for $N-2$ matching processes. Note the potential significance of such an achievement: By using AM, projection estimation errors are reduced, for every remaining $N-2$ features, from potentially more than 50 to less than 5 pixels after two sole feature matching results, see Fig. 5.9. These last unavoidable minor residuals are consequences of the approximation concerning translation propagation.

The dramatic reduction of image processing is here the biggest appeal for using AM: Only two *active* features have to be extensively searched for (the second one less extensively) and the remaining features are easy prey for the regular KLT tracker. A second major appeal exists: Guided feature search based on information theory is, together with state update, main overhead in potential AM-related calculations (Chli and Davison, 2009a). Sensible guidance of feature search is indeed advantageous in SLAM because some 3-D feature locations are more correlated than others (Chli and Davison, 2009b). However, in our case of full map knowledge, all feature locations are equally (totally) correlated in $SE(3)$, so that fair preliminary motion estimation can be achieved by *any* feature pair used. Third, we expect small residuals within their 'basins of attraction' for $N-2$ features; therefore, most features are clear of data association issues. These arguments consolidate my view of non-stochastic AM with known structure as a best case scenario for AM.

• **The KLT feature tracker with larger search regions.** The KLT feature tracker is able to cope with larger feature search regions using pyramidal representations of image patches, refer to (Bouguet, 2000; Mair *et al.*, 2009) and Section 5.4.2. However, this poses difficulties in efficiency and matching robustness at higher pyramidal levels. In order to avoid the two limitations mentioned in Section 5.4.2, I constrain the KLT tracker in the context of non-stochastic AM: First, the height of the pyramidal representation is limited to one sole subsampled level with decimation by a factor of two. Second, at the subsampled level, I perform exhaustive template search by sum of absolute differences (SAD) to half a pixel accuracy, followed by standard, sub-pixel accurate gradient descent search at the original resolution; I do not perform gradient descent search at the subsampled level because the search region, even at that level, is still much bigger than the template size, potentially leading to local minima. Sequential, exhaustive template search by SAD using bigger templates may be expensive, but it is very robust to ambiguities. These modifications only apply to the initial set of two *active* features with big search areas.

• **Preliminary motion estimation from two features.** I present an algorithm for interframe rotation estimation from two sequentially tracked features at the current image frame $\mathcal{I}^t$. This estimation $\widehat{\boldsymbol{R}}_{\text{ptr}}$, together with translational propagation following a constant velocity motion model similar to Eq. (5.1), yields tight priors on all other feature projections.

The algorithm is detailed in Alg. 2 and Fig. 5.9. The choice of the *active* features $\boldsymbol{p}$ and $\boldsymbol{q}$ is quite immaterial—provided they were sequentially tracked in $\mathcal{I}^{t-1}$ and $\mathcal{I}^{t}$, and their templates at lower resolution are distinctive. I choose the two most distant features in the image to avoid noise in the estimation of the ***roll*** rotation $\widehat{\boldsymbol{R}}_{\mathrm{r}}$. The ***pan+tilt*** rotation $\widehat{\boldsymbol{R}}_{\mathrm{pt}}$ is estimated ($\widehat{\ }$) from the first *active* feature $\boldsymbol{p}$. Together with $\widehat{\boldsymbol{R}}_{\mathrm{r}}$ they form $\widehat{\boldsymbol{R}}_{\mathrm{ptr}}$.

Algorithm 2 Pose correction from two features.

```
Require: Last tracked features and last camera translation $\boldsymbol{t}^{t-1}$.
  repeat
    Pick first active feature $\boldsymbol{p}$
    Apply translation propagation: $\hat{\boldsymbol{p}}^{t}_{\mathrm{tra}} = \mathrm{proj}\big({}_{\mathrm{C}}\tilde{\boldsymbol{p}}^{t-1} - \boldsymbol{t}^{t-1}\big)$
    Exhaustive template match around $\hat{\boldsymbol{p}}^{t}_{\mathrm{tra}}$                {wide search}
  until reliable match $\tilde{\boldsymbol{p}}^{t}$                                        {normally 1×}
  Estimate minimal rotation (2 DoF): $\widehat{\boldsymbol{R}}^{t-1,\,t}_{\mathrm{pt}}$                        {Eq. (5.2)}
  ..........................................................................................
  repeat
    Pick second active feature $\boldsymbol{q}$
    Apply translation propagation: ${}_{\mathrm{C}}\hat{\boldsymbol{q}}^{t}_{\mathrm{tra}} = {}_{\mathrm{C}}\tilde{\boldsymbol{q}}^{t-1} - \boldsymbol{t}^{t-1}$
    Apply minimal rotation: $\hat{\boldsymbol{q}}^{t}_{\mathrm{tra+pt}} = \mathrm{proj}\big(\widehat{\boldsymbol{R}}^{t-1,\,t}_{\mathrm{pt}} \cdot {}_{\mathrm{C}}\hat{\boldsymbol{q}}^{t}_{\mathrm{tra}}\big)$
    Exhaustive template match around $\hat{\boldsymbol{q}}^{t}_{\mathrm{tra+pt}}$             {narrow search}
  until reliable match $\tilde{\boldsymbol{q}}^{t}$                                        {normally 1×}
  Estimate remaining rotation (1 DoF): $\widehat{\boldsymbol{R}}^{t-1,\,t}_{\mathrm{r}}$                       {Eq. (5.3)}
  ..........................................................................................

  Pick random validation set e.g. ${}^{1..5}\boldsymbol{v}$
  Apply translation propagation: ${}^{1..5}_{\mathrm{C}}\hat{\boldsymbol{v}}^{t}_{\mathrm{tra}} = {}^{1..5}_{\mathrm{C}}\tilde{\boldsymbol{v}}^{t-1} - \boldsymbol{t}^{t-1}$
  Apply rotation: ${}^{1..5}\hat{\boldsymbol{v}}^{t}_{\mathrm{tra+ptr}} = \mathrm{proj}\big(\widehat{\boldsymbol{R}}^{t-1,\,t}_{\mathrm{pt}} \cdot \widehat{\boldsymbol{R}}^{t-1,\,t}_{\mathrm{r}} \cdot {}^{1..5}_{\mathrm{C}}\hat{\boldsymbol{v}}^{t}_{\mathrm{tra}}\big)$
  Validation by regular KLT tracker on ${}^{1..5}\hat{\boldsymbol{v}}^{t}_{\mathrm{tra+ptr}}$                {else restart}
  ..........................................................................................

  Apply transl. propagation to ${}^{i}_{\mathrm{C}}\hat{\boldsymbol{f}}^{t}_{\mathrm{tra}} = {}^{i}_{\mathrm{C}}\tilde{\boldsymbol{f}}^{t-1} - \boldsymbol{t}^{t-1}$, $\forall\, {}^{i}\boldsymbol{f} \in \mathcal{I}^{t}$
  Apply rotation: ${}^{i}\hat{\boldsymbol{f}}^{t}_{\mathrm{tra+ptr}} = \mathrm{proj}\big(\widehat{\boldsymbol{R}}^{t-1,\,t}_{\mathrm{pt}} \cdot \widehat{\boldsymbol{R}}^{t-1,\,t}_{\mathrm{r}} \cdot {}^{i}_{\mathrm{C}}\hat{\boldsymbol{f}}^{t}_{\mathrm{tra}}\big)$
  return updated feature projections ${}^{i}\hat{\boldsymbol{f}}^{t}_{\mathrm{tra+ptr}}$ for regular KLT.
```

From the discrepancy between the translationally propagated $\hat{\boldsymbol{p}}^{t}_{\mathrm{tra}}$ and the first exhaustive matching result $\tilde{\boldsymbol{p}}^{t}$, the minimal rotation potentially responsible for that displacement reads, in axis-angle representation,

$$\big(\boldsymbol{\omega} = \underline{\hat{\boldsymbol{p}}}^{t}_{\mathrm{tra}} \times \underline{\tilde{\boldsymbol{p}}}^{t},\ \theta = \pm \arccos\big(\underline{\hat{\boldsymbol{p}}}^{t}_{\mathrm{tra}} \cdot\, \underline{\tilde{\boldsymbol{p}}}^{t}\big)\big), \tag{5.2}$$

where $\underline{\boldsymbol{p}}^t = {}_{\mathrm{C}}\boldsymbol{p}^t / |{}_{\mathrm{C}}\boldsymbol{p}^t|$ and ${}_{\mathrm{C}}\boldsymbol{p}^t$ is the 3-D location of $\boldsymbol{p}$ in the camera reference frame S_{C} at time t, thus $\underline{\tilde{\boldsymbol{p}}}^t$ is the direction in S_{C} of the 2-D, actually tracked feature $\tilde{\boldsymbol{p}}^t$. Eventually in SO(3): $\widehat{\boldsymbol{R}}_{\mathrm{pt}}^{t-1,\,t} = \exp([\boldsymbol{\omega}]_\times,\, \theta)$ where $[\boldsymbol{\omega}]_\times$ is the skew-symmetric cross product matrix of $\boldsymbol{\omega}$.

From the second match $\tilde{\boldsymbol{q}}^t$ the only remaining DoF can be estimated: the *roll* rotation around the axis $\underline{\tilde{\boldsymbol{p}}}^t$ that relates the planes containing the estimated projection $\hat{\boldsymbol{q}}^t_{\mathrm{tra+pt}}$ and the actual projection $\tilde{\boldsymbol{q}}^t$ as follows:

$$\left(\boldsymbol{\omega} = \underline{\tilde{\boldsymbol{p}}}^t,\ \theta = \pm \arccos\left(\left(\underline{\tilde{\boldsymbol{p}}}^t \times \underline{\hat{\boldsymbol{q}}}^t_{\mathrm{tra+pt}}\right) \cdot \left(\underline{\tilde{\boldsymbol{p}}}^t \times \underline{\tilde{\boldsymbol{q}}}^t\right)\right)\right) \tag{5.3}$$

and the rotation matrix $\widehat{\boldsymbol{R}}_{\mathrm{r}}^{t-1,\,t}$ is calculated as above.

Both *pan+tilt* and *roll* rotations yield:

$$\widehat{\boldsymbol{R}}_{\mathrm{ptr}}^{t-1,\,t} = \widehat{\boldsymbol{R}}_{\mathrm{pt}}^{t-1,\,t} \cdot \widehat{\boldsymbol{R}}_{\mathrm{r}}^{t-1,\,t}\,, \tag{5.4}$$

which is good estimate of the interframe camera rotation between $t-1$ and t. Together with the last camera translation $\boldsymbol{t}^{t-1}$ it can be used to recompute further feature projections. Note that $\widehat{\boldsymbol{R}}_{\mathrm{pt}}$ obtained from feature $\boldsymbol{p}$ was also used for improved tracking of feature $\boldsymbol{q}$.

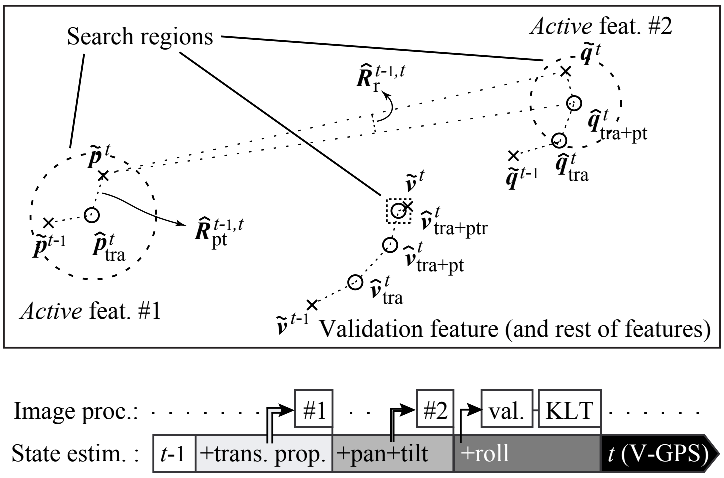


Figure 5.9: Top: Pictorial schematic on the 2-D estimations involved. Two *active* features $\boldsymbol{p}$ and $\boldsymbol{q}$ as well as the resulting estimation steps on a further feature $\boldsymbol{v}$ are detailed. The latter is tracked using the regular KLT feature tracker. Bottom: Time evolution of state estimation w.r.t. the image processing steps.

After successful tracking of features $\boldsymbol{p}$ and $\boldsymbol{q}$ I opt for tracking a random subset of five of the remaining features in order to validate the rotation hypothesis in case of mismatches or inaccurate translational motion propagation. The validation features ${}^{1..5}\boldsymbol{v}$ are rapidly tracked using the standard, gradient descent KLT tracker at the original resolution only. Valid hypotheses prompt rapid, local matching of all remaining features ${}^{i}\boldsymbol{f}$ (typically 20 to 50), as in the validation step.

Two types of errors may appear when searching for *active* features: *First*, indistinctive matching templates *at lower resolution* or corrupted projections (e.g. occlusions). The frequency of these errors is minimized owing to sequential matching, as they are best detected during matching itself—they are signalized in order not to be further used as *active* features. *Second*, image ambiguity may cause incorrect data association (false positives) even though exhaustive search and sequential matching minimize that risk. The validation step mentioned above detects this by checking consistency w.r.t. the state history (Neira and Tardós, 2001; Kaess and Dellaert, 2009). If there is a discrepancy, the used *active* features are signalized as unsuitable for AM; it is not worth the effort involved in maintaining multiple hypothesis on this event, as ambiguity is recurrent on particular features and we only require two valid *active* features after all. Both types of errors are however rare in regular operation. Since hypothesis generation (tracking of $\boldsymbol{p}$ and $\boldsymbol{q}$) is expensive, I opt for rigorous preemption: one sole hypothesis will be generated unless the aforementioned errors appear. In exceptional cases of multiple errors at the same image, the computational overhead may exceed the time budget for matching (e.g. 20 ms). These occasional peaks can be filtered out by making use of an image buffer, e.g., of the last two images. This implies a latency of e.g. 80 ms, which is admissible in most applications.

The sizes of the search areas for the two *active* features are empirically based on worst case experiments at 25 Hz. They amount to circles of 50 and 25 pixels radii respectively. The second search area is smaller because $\widehat{\boldsymbol{R}}_{\mathrm{pt}}$ is known. Typical matching times on a 2008 notebook equipped with an Intel® Core™ 2 Duo P8700 processor are: for *active* feature #1 3.2 ms (50 p. radius) or 2.3 ms (40 p.), for *active* feature #2 1.3 ms (25 p. radius) or 1.0 ms (20 p.), standard matching of 5 validation features takes 0.6 ms, and the remaining features require 1.7 ms (15 features) or 2.7 ms (20).

It is worth noting that this approach scales well with increasing frame rate since it facilitates tracking through smaller *active* search areas—at constant target motion bandwidth.

5.4.3 Real-Time Pose Tracking from Features Flow

In this section I present the methods used for real-time pose tracking of the DLR 3D-Modeler, which rely on tracking of feature projections (last section), together with their known 3-D geometry (Section 5.4.1). Assuming a rigid set of 3-D points and static camera geometry, the feature projections flow is solely caused by varying perspective projection, *i.e.*, by varying pose of the camera w.r.t. the scene. In this context, pose tracking basically works out camera poses that match these feature displacements (optical flow), see Fig. 5.10.

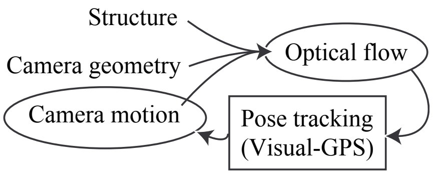


Figure 5.10: Structure, camera geometry, and camera motion determine optical flow.

The robust Visual-GPS

Visual-GPS (V-GPS) is an algorithm that solves the *relative orientation problem* iteratively, but efficiently (Burschka and Hager, 2003). After the determination of the orientation, the translation can be also estimated. The method assumes a set of known 3-D points related to the initial camera reference frame S_0 (a set of n points ${}_0P_i$, $i \in \{1..n\}$). This sparse set can be constructed in an arbitrary way—I use feature-based stereo as in Section 5.4.1. The *exterior orientation problem* of the estimation of the following *monocular* camera poses S_t w.r.t. that reference set ${}_0P_i$ is now equivalent to the original relative pose estimation problem—provided the correspondences between the points P_i and their projections are known.

In order to solve the exterior orientation problem of S_t w.r.t. the set of points ${}_0P_i$, an additional, tentative 3-D model ${}_t\widehat{P}_i$ is generated at the current frame S_t using both, the current 2-D projections of P_i as well as approximated ranges to that points (from preceding estimations). The problem now reduces to solving the *absolute orientation problem* between these two 3-D sets of points ${}_0P_i$ and ${}_t\widehat{P}_i$, which is an approximate, orthogonal Procrustean problem that can be solved in closed form using the singular value decomposition (SVD). As relative translation and rotation are estimated separately, I first set the origins of the sets to their respective centers of mass without modifying their orientations, which yields ${}_0P'_i$ and ${}_t\widehat{P}'_i$. The relative rotation between the sets corresponds to the relative rotation between camera reference frames S_0 and S_t and can be optimized (*) by maximizing the trace of the inertia matrix of the matched set:

$$ {}_t\boldsymbol{R}^* = \arg\max_{\boldsymbol{R}} \operatorname{trace}({}_t\boldsymbol{R}^{\mathrm{T}}\,{}_t\boldsymbol{M}) \,, \quad {}_t\boldsymbol{M} = \sum_{i=1}^{n} {}_t\widehat{P}'_i \; {}_0P'^{\mathrm{T}}_i \quad . $$

Let $(\boldsymbol{U}, \boldsymbol{\sigma}, \boldsymbol{V})$ be the SVD of ${}_t\boldsymbol{M}$, that is $\boldsymbol{U}\boldsymbol{\sigma}\boldsymbol{V}^{\mathrm{T}} = {}_t\boldsymbol{M}$:

$$ {}_t\boldsymbol{R}^* = \boldsymbol{U}\boldsymbol{V}^{\mathrm{T}} \quad , $$

and the translation by subtracting centers of mass:

$$ {}_t\boldsymbol{t}^* = \frac{1}{n}\sum_{i=1}^{n} {}_t\widehat{P}_i - {}_t\boldsymbol{R}^* \, \frac{1}{n}\sum_{i=1}^{n} {}_0P_i \quad . $$

Since the tentative 3-D pointset ${}_t\widehat{P}_i$ may differ from reality (${}_0P_i$), the final solution is found iteratively, by concurrently optimizing the ranges of the tentative model. The algorithm terminates whenever sufficient consistency with the original set of points ${}_0P_i$ is achieved.

The specifics of the implementation are:

- V-GPS is being sequentially applied, see Section 5.4.3.
- The set of 3-D points ${}_0P_i$ is obtained by stereo vision at the initial reference frame S_0 (Section 5.4.1) and is not updated due to its high accuracy. Nevertheless, the robust formulation presented below can also suppress erroneous instances.

- Gross outliers in the estimation of the 3-D set of points or in their 2-D monocular tracking may indeed occur, especially at close range (due to repetitive background patterns, blur, occlusions, shadows, etc.). In order to disregard them, I make novel use of a redescending M-estimator on the residual Euclidean distances between matched 3-D points. Indeed, when tracking rigid body motion, it is indicated to perform robustification wherever the most complete model of the system be used, *i.e.*, in our case during 3-D pose tracking instead of 2-D feature tracking. I use the biweight function of Tukey because of its continuous derivatives and its handy weights. The modification concerns weighting the contribution of each point to the inertia matrix of the matched set of points with

$$\begin{array}{rcll} {}_tw_i & \propto & (1 - {}_tR_i \cdot {}_tR_i)^2 & \text{if } \; |{}_tR_i| < 1 \\ {}_tw_i & = & 0 & \text{if } \; |{}_tR_i| \geq 1 \end{array}$$

where ${}_tR_i = ({}_t\boldsymbol{R}\,{}_0P_i - {}_t\widehat{P}_i)/s$ is the estimated normalized matching residual for object point i at instant t before performing SVD, and s is the scale of the inlier noise. In the end:

$${}_t\boldsymbol{R}^* = \arg\max_{\boldsymbol{R}} \operatorname{trace}({}_t\boldsymbol{R}^{\mathrm{T}}\,{}_t\boldsymbol{M}^{R})\,, \;\; {}_t\boldsymbol{M}^{\mathrm{R}} = \sum_{i=1}^{n} {}_tw_i\,{}_t\widehat{P}'_i\,{}_0P'^{\mathrm{T}}_i .$$

Further, the robustified method (RVGPS) not only neutralizes the effects of outliers, but also signalizes them so that they can be removed from memory.

- I use an efficient termination policy determined by a threshold on absolute orientation correction.

Sequential, relative pose tracking

Following the concept in Section 5.3, an efficient pose-tracking algorithm should adopt a frugal policy when taking advantage of stereo vision. This consideration led me to treat 3-D structure for pose tracking as separate sets of 3-D points. These are being triangulated once (cf. Section 5.4.1) and are used for local, *monocular* pose tracking thereafter, until a new set of points takes over. This is in contrast to tracking methods that dynamically triangulate new points, e.g. (Cheng *et al.*, 2006). In our case, it is only when closing loops that we reutilize past sets of points. My approach is similar to visual odometry in (Nistér *et al.*, 2006), if only using AM instead of RANSAC, V-GPS instead of the 3-point algorithm, and a more precise feature matching algorithm.

It turns out that the abovementioned "limitation" suits two present trends in SLAM back-end design just fine: *First*, the reference frame for representation of structure and motion is neither camera-attached nor absolute, but local in the vicinity of the camera; this yields an advantage in terms of efficiency, see (Moore *et al.*, 2009). *Second*, the sets of triangulated points now represent a natural way of spacing keyframes in the context of keyframe-based BA, refer to

Section 5.2.4-III. *Third*, the relative pose estimation algorithm (V-GPS) regularly makes use of the most representative images that offer biggest parallax for triangulation—first and last feature projections, disregarding redundant data in between (Strasdat *et al.*, 2010b).

Fig. 5.11 depicts the standard operation of local pose tracking. In the case of advanced flow prediction schemes, the tracked feature set is not always unique—two sets are being tracked, in parallel, during handover frames in order for the latter to accumulate feature displacement information that will in turn facilitate feature displacement prediction in the following frames.

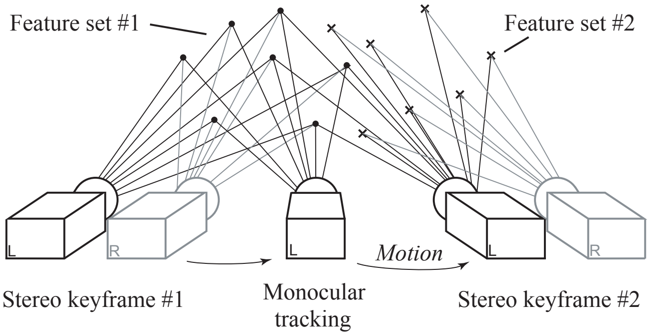


Figure 5.11: Standard operation of local pose tracking. Stereo vision is used in keyframes #1 and #2, and monocular tracking elsewhere.

Of course, individual feature losses may appear, and features regularly get out of sight and void areas take their place. I treat short- and long-term losses separately: *Short-term losses* are features that are lost by tracking but maintain several fellow points of the 3-D set in track so that camera pose can still be estimated. Monocular tracking will repeatedly try to recover these features with the aid of the current pose estimation—unless RVGPS marked them as invalid. *Long-term losses* are features that are deliberately abandoned because their associated 3-D set of points becomes inadequate to the current vantage point. In this event, either an inactive set of features is retrieved, or a new set of 3-D features has to be generated:

Generation of new features sets: I command initialization as in Section 5.4.1 whenever too few features of the current set are being tracked or the set's center of mass drifts outside the central area of the image and no other inactive set lies within the field of view. As soon as generation is successfully completed (and feature flow information is accumulated), I give up on tracking past features. Since this will take some frames, initialization proceeds concurrently in a separate computing thread. After that, only the new feature set #2 is being tracked and the current, local relative pose estimation ${}_{\mathrm{now}}\boldsymbol{T}^{\mathrm{l}_2}$ will add to the relative pose ${}_{\mathrm{l}_2}\boldsymbol{T}^{\mathrm{l}_1}$ between the last keyframes (offset) as follows: ${}_{\mathrm{now}}\boldsymbol{T}^{\mathrm{l}_1} = {}_{\mathrm{now}}\boldsymbol{T}^{\mathrm{l}_2}\,{}_{\mathrm{l}_2}\boldsymbol{T}^{\mathrm{l}_1}$.

Retrieval of inactive features sets (local loop closure): Whenever the projection of the centroid of an inactive set of points is more central than the current one, and their view direction and range did not vary too much, all its features within the current field of view are to be tracked again. Their predicted projections are now solely based on the *current* pose estimation ${}_{\mathrm{now}}\boldsymbol{T}^{\mathrm{l}_1}$, thus no extrapolation is needed, cf. Fig. 5.3. It is worth mentioning that, by referring back to previous sets, potential pose drifts that may have arisen by leaping onto newer sets necessarily disappear, which is another appealing property of this approach compared to sequential dead reckoning (visual odometry) or even filtering approaches, as they usually depend on the particular path history, or even become inconsistent (Julier and Uhlmann, 2001).

Local, hybrid bundle adjustment

It is a peculiarity of 3-D modeling that new areas are continuously being explored and loop-closing events are rare. In this section, I focus on optimal motion estimation without closing large loops, *i.e.*, by dead reckoning (refer to Section 5.2.4-II); in Section 5.4.5 I shall present a more complete optimization in the event of final loop closing, e.g. after scanning all around an object.

While robust V-GPS in Section 5.4.3 provides a robust, fast relative motion estimation from monocular footage by dead reckoning in realtime, it is still advisable to perform optimal motion and structure estimation by minimization of reprojection errors (*i.e.*, BA) at handover stereo keyframes to further increase accuracy. Following e.g. Refs. (Klein and Murray, 2007; Mouragnon *et al.*, 2006; Strasdat *et al.*, 2010b), I opt for an efficient BA optimization disregarding image frames in between selected keyframes (*i.e.*, kBA). It is worth noting that some information contained in these intermediate images is still being used by sequential tracking for correct data association. Since in my approach all 3-D features are measured locally, *i.e.*, on a unique static reference frame defined at keyframes, the global optimization of the covered dead reckoning motion can be decomposed into independent sub-optimizations exclusively concerning one reference frame along with its feature set. In detail, the information required for a sub-optimization is confined to the stereo keyframe ${}_{\mathrm{l}}\mathcal{I}^i \cup {}_{\mathrm{r}}\mathcal{I}^i$ that initialized the i-th feature set by stereo triangulation, along with the **f**inal, left monocular image ${}_{\mathrm{f}}\mathcal{I}^i$ that both, tracks the i-th feature set last, and coincides with the left frame ${}_{\mathrm{l}}\mathcal{I}^{i+1}$ of the next keyframe (from which the following feature set $\#i{+}1$ will be initialized, see Fig. 5.12).[8] This frugal, hybrid keyframe selection policy does deliver high accuracy as both, initial and last tracking vantage points, are being considered for every feature, maximizing their projected parallax. In addition, the inclusion of stereo images serves to anchor global scale.

[8] As kBA is only concerned with keyframes and not with timestamps, the right superscript on variables refers now to the feature set being tracked and not to timestamps as before.

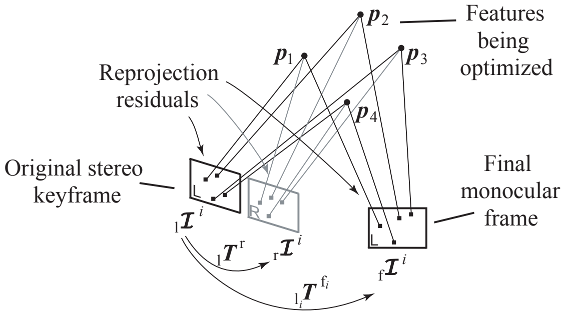

$\boldsymbol{p}_2$
Features being optimized
$\boldsymbol{p}_1$
$\boldsymbol{p}_3$
Reprojection residuals
$\boldsymbol{p}_4$
Original stereo keyframe
L
R
Final monocular frame
L
${}_{\mathrm{l}}\mathcal{I}^i$
${}_{\mathrm{l}}\boldsymbol{T}^{\mathrm{r}}$
${}_{\mathrm{r}}\mathcal{I}^i$
${}_{\mathrm{f}}\mathcal{I}^i$
${}_{\mathrm{l}_i}\boldsymbol{T}^{\mathrm{f}_i}$

Figure 5.12: Data concerned in local, hybrid BA on feature set $\#i$.

The novel formulation minimizes the sum of squared reprojection residuals as follows:

$$\begin{aligned}\widehat{\boldsymbol{\Omega}}^i_\star = \arg\min \sum_{p=1}^{M_i} \Big(& ||{}_{\mathrm{l}}\tilde{\boldsymbol{m}}^i_p - {}_{\mathrm{l}}\hat{\boldsymbol{m}}^i_p({}_{\mathrm{l}}\hat{\boldsymbol{p}}^i_p)||^2 \\ & + ||{}_{\mathrm{r}}\tilde{\boldsymbol{m}}^i_p - {}_{\mathrm{r}}\hat{\boldsymbol{m}}^i_p({}_{\mathrm{l}}\boldsymbol{T}^{\mathrm{r}}; {}_{\mathrm{l}}\hat{\boldsymbol{p}}^i_p)||^2 \\ & + ||{}_{\mathrm{f}}\tilde{\boldsymbol{m}}^i_p - {}_{\mathrm{f}}\hat{\boldsymbol{m}}^i_p({}_{\mathrm{l}_i}\widehat{\boldsymbol{T}}^{\mathrm{f}_i}; {}_{\mathrm{l}}\hat{\boldsymbol{p}}^i_p)||^2 \Big) \end{aligned} \tag{5.5}$$

where the optimized ($\star$) parameters $\boldsymbol{\Omega}^i_\star$ include the 3-D coordinates ${}_{\mathrm{l}}\boldsymbol{p}^i_p = [{}_{\mathrm{l}}x^i_p, {}_{\mathrm{l}}y^i_p, {}_{\mathrm{l}}z^i_p]^{\mathrm{T}}$, $\forall p \in \mathbb{N}_1$, $i \leq M_i$ of the i-th set of M_i features w.r.t. the **l**eft camera at keyframe $\#i$, and the inter-keyframe transformation ${}_{\mathrm{l}_i}\boldsymbol{T}^{\mathrm{f}_i}$ of the **l**eft camera frame between keyframes $\#i$ and $\#i{+}1$. The residual is composed of estimated ($\hat{}$) reprojections ${}_{\mathrm{l}}\hat{\boldsymbol{m}}^i_p = [{}_{\mathrm{l}}\hat{u}^i_p\ {}_{\mathrm{l}}\hat{v}^i_p]^{\mathrm{T}} = \mathrm{proj}({}_{\mathrm{l}}\hat{\boldsymbol{p}}^i_p)$ and ${}_{\mathrm{r}}\hat{\boldsymbol{m}}^i_p = [{}_{\mathrm{r}}\hat{u}^i_p\ {}_{\mathrm{r}}\hat{v}^i_p]^{\mathrm{T}} = \mathrm{proj}({}_{\mathrm{r}}\boldsymbol{T}^{\mathrm{l}}\,{}_{\mathrm{l}}\hat{\boldsymbol{p}}^i_p)$ onto the **l**eft and the **r**ight frames at the initial keyframe of feature set $\#i$, respectively, as well as their last, **f**inal feature projections ${}_{\mathrm{f}}\hat{\boldsymbol{m}}^i_p = [{}_{\mathrm{f}}\hat{u}^i_p\ {}_{\mathrm{f}}\hat{v}^i_p]^{\mathrm{T}} = \mathrm{proj}({}_{\mathrm{f}_i}\widehat{\boldsymbol{T}}^{\mathrm{l}_i}\,{}_{\mathrm{l}}\hat{\boldsymbol{p}}^i_p)$ at the left frame (remember that ${}_{\mathrm{f}}\mathcal{I}^i \triangleq {}_{\mathrm{l}}\mathcal{I}^{i+1}$). These estimations are being subtracted from the actual measurements ${}_{\mathrm{l}}\tilde{\boldsymbol{m}}^i_p$, ${}_{\mathrm{r}}\tilde{\boldsymbol{m}}^i_p$ and ${}_{\mathrm{f}}\tilde{\boldsymbol{m}}^i_p$. The transformation ${}_{\mathrm{r}}\boldsymbol{T}^{\mathrm{l}}$ stems from the epipolar geometry of the stereo camera by camera calibration (Strobl *et al.*, 2005). Note that the projection function proj() does not include lens distortion; for efficiency reasons, I opt for minimizing undistorted reprojection errors and have to undistort actual, **d**istorted projections beforehand, e.g. ${}_{\mathrm{l}}\hat{\boldsymbol{m}}_p = \mathrm{undist}({}_{\mathrm{l}}\hat{\boldsymbol{m}}^{\mathrm{d}}_p)$.

Note that global scale could also be anchored even if the projections ${}_{\mathrm{r}}\hat{\boldsymbol{m}}^i_p$ had not been included in the residual function, but considered ground truth instead. However, I stress that the structure of the 3-D features $\boldsymbol{p}^i_p$ does not stem from selected, ground truth projections into an image (e.g. ${}_{\mathrm{r}}\tilde{\boldsymbol{m}}^i_p$) in the context of stereo vision, but from their rigid body geometry alone. In this way, by releasing all three key projections the optimal 3-D solution will be solely constrained by both, the rigid body assumption along with perspective geometry. In addition, it is well known that full BA turns out to be faster than any attempts to eliminate e.g. the structure parameters (Nistér *et al.*, 2006).

The hybrid optimization utilizes the nonlinear least squares optimization function `dlevmar_der()` (Lourakis, Jul. 2004), which implements the Levenberg-Marquardt method. I am providing analytic Jacobians for improved performance, see Eq. (5.6). Even though the Jacobians are always sparse, the small size of the system of equations renders sparse optimization methods unnecessary. It is worth noting that minimal representations are used for unknown rotations, specifically differential perturbations of Euler angles. In addition, the residual function has been robustified in case of outliers.

$$\frac{\partial(\tilde{\boldsymbol{m}}^i - \hat{\boldsymbol{m}}^i)}{\partial\boldsymbol{\Omega}^i} = \begin{bmatrix} \vdots \\ \dfrac{\partial\boldsymbol{\Delta}\boldsymbol{m}_p^i}{\partial\boldsymbol{\Omega}^i} \\ \vdots \end{bmatrix}_{6M_i\times(6+3M_i)} = \begin{bmatrix} \vdots \\ \dfrac{\partial\boldsymbol{\Delta}_{\mathrm{l}}\boldsymbol{m}_p^i}{\partial\boldsymbol{\Omega}^i} \\ \dfrac{\partial\boldsymbol{\Delta}_{\mathrm{r}}\boldsymbol{m}_p^i}{\partial\boldsymbol{\Omega}^i} \\ \dfrac{\partial\boldsymbol{\Delta}_{\mathrm{f}}\boldsymbol{m}_p^i}{\partial\boldsymbol{\Omega}^i} \\ \vdots \end{bmatrix} = \cdots$$

$$= \begin{array}{r} \\ \\ {}_{\mathrm{l}}\boldsymbol{m}_p^i \rightarrow \\ {}_{\mathrm{r}}\boldsymbol{m}_p^i \rightarrow \\ {}_{\mathrm{f}}\boldsymbol{m}_p^i \rightarrow \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} \overbrace{\qquad}^{\boldsymbol{\Delta T}_{\mathrm{l}_{i+1}}} & \overbrace{\qquad}^{\boldsymbol{p}_1^i\cdot\cdot\boldsymbol{p}_{p-1}^i} & \overbrace{\qquad}^{\boldsymbol{p}_p^i} & \overbrace{\qquad}^{\boldsymbol{p}_{p+1}^i\cdot\cdot\boldsymbol{p}_{M_i}^i} \end{array} \\ \left[\begin{array}{c|c|c|c} & & \vdots & \\ \mathbf{0}_{2\times6} & \mathbf{0}_{2\times3(p-1)} & \dfrac{\partial\boldsymbol{\Delta}_{\mathrm{l}}\boldsymbol{m}_p^i}{\partial\boldsymbol{p}_p^i} & \mathbf{0}_{2\times3(M_i-p)} \\ \mathbf{0}_{2\times6} & \mathbf{0}_{2\times3(p-1)} & \dfrac{\partial\boldsymbol{\Delta}_{\mathrm{r}}\boldsymbol{m}_p^i}{\partial\boldsymbol{p}_p^i} & \mathbf{0}_{2\times3(M_i-p)} \\ \dfrac{\partial\boldsymbol{\Delta}_{\mathrm{f}}\boldsymbol{m}_p^i}{\partial\boldsymbol{T}} & \mathbf{0}_{2\times3(p-1)} & \dfrac{\partial\boldsymbol{\Delta}_{\mathrm{f}}\boldsymbol{m}_p^i}{\partial\boldsymbol{p}_p^i} & \mathbf{0}_{2\times3(M_i-p)} \\ & & \vdots & \end{array}\right] \end{array} =$$

$$= \left[\begin{array}{c|c|c|c} & & \vdots & \\ \mathbf{0}_{2\times6} & \mathbf{0}_{2\times3(p-1)} & \begin{array}{c}▨□▨ \\ □▨▨\end{array} & \mathbf{0}_{2\times3(M_i-p)} \\ \mathbf{0}_{2\times6} & \mathbf{0}_{2\times3(p-1)} & \begin{array}{c}▨▨▨ \\ ▨▨▨\end{array} & \mathbf{0}_{2\times3(M_i-p)} \\ \begin{array}{c}▨▨▨▨□▨ \\ ▨▨▨□▨▨\end{array} & \mathbf{0}_{2\times3(p-1)} & \begin{array}{c}▨▨▨ \\ ▨■▨\end{array} & \mathbf{0}_{2\times3(M_i-p)} \\ & & \vdots & \end{array}\right] \tag{5.6}$$

* White boxes correspond to zero elements;
gray or black boxes to non-zero ones.

By way of illustration, I go into detail about the calculation of the **black** Jacobian element above:

$$\frac{\partial({}_{\mathrm{f}}\tilde{v}_p^i - {}_{\mathrm{f}}\hat{v}_p^i)}{\partial\,{}_{\mathrm{l}}\hat{y}_p^i} = -\frac{\partial{}_{\mathrm{f}}\hat{v}_p^i}{\partial\,{}_{\mathrm{f}}\hat{y}_p^i}\frac{\partial\,{}_{\mathrm{f}}\hat{y}_p^i}{\partial\,{}_{\mathrm{l}}\hat{y}_p^i} - \frac{\partial{}_{\mathrm{f}}\hat{v}_p^i}{\partial\,{}_{\mathrm{f}}\hat{z}_p^i}\frac{\partial\,{}_{\mathrm{f}}\hat{z}_p^i}{\partial\,{}_{\mathrm{l}}\hat{y}_p^i}$$

where

$$\left\{\begin{array}{l} {}_{\mathrm{f}}\hat{v}_p^i = (\,\beta\,\dfrac{{}_{\mathrm{f}}\hat{y}_p^i}{{}_{\mathrm{f}}\hat{z}_p^i} + v_0) \\ \begin{bmatrix} {}_{\mathrm{f}}\hat{x}_p^i \\ {}_{\mathrm{f}}\hat{y}_p^i \\ {}_{\mathrm{f}}\hat{z}_p^i \\ 1 \end{bmatrix} = \underbrace{\boldsymbol{\Delta}\widehat{\boldsymbol{T}}^{-1}_{\mathrm{l}_{i+1}}}_{\text{perturbation}}\ {}_{\mathrm{f}_i}\boldsymbol{T}^{\mathrm{l}_i} \begin{bmatrix} {}_{\mathrm{l}}\hat{x}_p^i \\ {}_{\mathrm{l}}\hat{y}_p^i \\ {}_{\mathrm{l}}\hat{z}_p^i \\ 1 \end{bmatrix} \end{array}\right. \quad ; \tag{5.7}$$

β and v_0 are part of the intrinsic parameters of the left camera, and $\Delta\widehat{\boldsymbol{T}}_{\mathrm{l}_{i+1}}$ represents the estimated rigid body perturbation on the **l**eft camera pose at keyframe $i+1$.

This method yields sub-millimetric corrections w.r.t. V-GPS on 3-D feature locations $\boldsymbol{p}_p^i$ and the relative pose ${}_{\mathrm{l}_i}\boldsymbol{T}^{\mathrm{f}_i}$ for every keyframe or feature set. Millimetric differences may arise on eventual loop closures after many keyframes, e.g. when scanning all around an object. On balance, it turns out that this method does not substantially improve the already accurate dead reckoning motion estimation by V-GPS. On the other hand, its computational cost remains low (2 to 5 ms)—roughly twice as long as V-GPS.

5.4.4 Appearance-Based Relocalization

Whenever

1. saccadic motion precludes sequential tracking,
2. the user browses outside a proximate scene, or
3. the cameras return to an area used before (loop closing) that has not been tracked for a long time,

pose tracking accuracy gets too low for consistent KLT tracking to be warranted anymore—even in its AM variant. Due to the richness of visual data, cameras are ideally suited for recognizing similarity; appearance-based relocalization can help to continue scanning *on the original reference frame.*

As mentioned in Section 5.2.4-I, there exist a number of operators, called descriptors, that concern about the visual appearance of features in order to be distinctive between features and invariant to viewpoint pose. In this work, I choose the performant SURF features (Bay *et al.*, 2008) in its original implementation, on stereo images. By using stereo images, the 3-D position of SURF features w.r.t. the camera ${}_{\mathrm{left}}\boldsymbol{T}^{\mathrm{SURF}}$ can be triangulated *at the same frame* during stereo initialization of the KLT feature set, where we obtained ${}_{\mathrm{left}}\boldsymbol{T}^{\mathrm{KLT}}$, see Section 5.4.1 and Fig. 5.13. By doing so, whenever 3 or more of these SURF features (and consequently ${}_{\mathrm{now}}\boldsymbol{T}^{\mathrm{SURF}}$) are found again, the location of the stereo camera w.r.t. the original KLT feature set can be roughly estimated as follows:

$${}_{\mathrm{now}}\widehat{\boldsymbol{T}}^{\mathrm{KLT}} = {}_{\mathrm{now}}\boldsymbol{T}^{\mathrm{SURF}} \left({}_{\mathrm{left}}\boldsymbol{T}^{\mathrm{SURF}}\right)^{-1} {}_{\mathrm{left}}\boldsymbol{T}^{\mathrm{KLT}}\,.$$

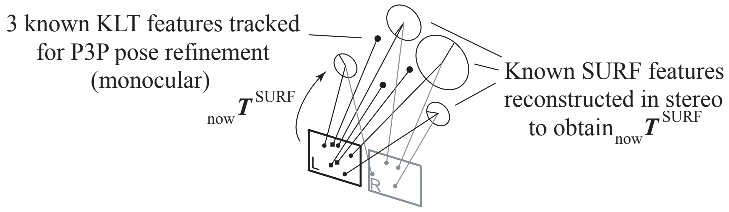

3 known KLT features tracked for P3P pose refinement (monocular)
$_{\text{now}}\boldsymbol{T}^{\text{SURF}}$
Known SURF features reconstructed in stereo to obtain $_{\text{now}}\boldsymbol{T}^{\text{SURF}}$
L
R

Figure 5.13: SURF features are robustly detected in stereo and triangulated. From this fair pose estimation $_{\text{now}}\boldsymbol{T}^{\text{SURF}}$, I estimate $_{\text{now}}\boldsymbol{T}^{\text{KLT}}$ to support monocular 2-D tracking of known KLT features. These SURF features bootstrap P3P pose estimation on KLT features to increase pose precision (Grunert, 1841); for improved robustness, three different sets of three features are being used. The best pose estimation allows seamless monocular KLT tracking as in Section 5.4.2.

This estimation is far less accurate than sequential pose tracking using V-GPS, compromising seamless transition to KLT tracking. I opt for using interleaved, *monocular* three point perspective (P3P) pose estimation on KLT features for increased accuracy. Feature matching is now on extended search regions due to inaccurate SURF-based pose estimation, thus requires exhaustive template matching similar to *active* features in Section 5.4.2. The procedure can be also interpreted as a validation stage on candidate SURF-based hypothesis. In the end, regular KLT tracking takes on sequential pose tracking *on the original reference frame*—not without prior scaling and affine distortion of the features' templates according to the current pose.

In detail, the P3P problem deals with the estimation of the positions of three known points out of their *monocular* perspective projections (in our case KLT features), *i.e.*, the pose estimation of the set of points w.r.t. the current vantage point. The direct solution to this problem is in the form of a fourth-degree polynomial (Grunert, 1841). Solving for the polynomial roots involves the computation of the eigenvalues of its companion matrix, which delivers up to four real solutions (rigid body transformations). From these, I pick the one that, in translation, is most consistent with the original SURF-based stereo solution. I choose not to use a RANSAC scheme, both for efficiency reasons (feature matching is expensive) and because we already have a fair pose estimation in the first place. My heuristic approach is to obtain three independent P3P solutions (out of three different triplets) and, again, choose the one that is most coherent with the SURF-based stereo solution.

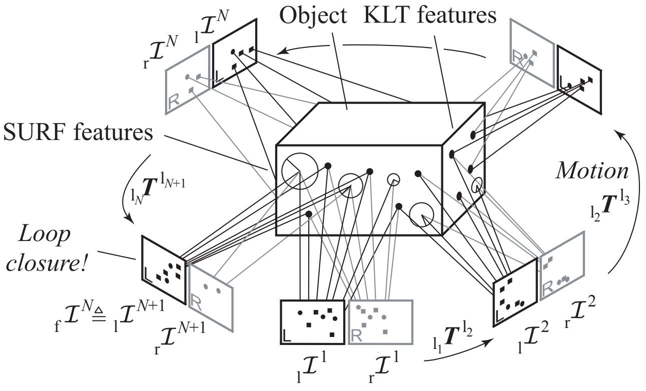
5.4.5 Global, Relative and Hybrid BA on Loop Closures

Loop closure events occur whenever former scene features that have not been recently tracked are being revisited. These events present the opportunity to greatly increase present and past pose tracking accuracy.

I distinguish between two types of loop closures: local loop closures can still take advantage of metric information for improved performance (Section 5.4.3), whereas global, large-scale loop closure ought to be independent of motion estimation precisely because its main objective is to correct inaccurate motion estimation in the first place. Global, large-scale loop closing may instead concern about the projected appearance of features, which are still discriminative in the face of unknown localization, see Sections 5.2.4-I. and 5.4.4.

Whatever the nature of the loop closure, it is indicated to subsequently optimize structure and motion estimations in the light of the discrepancies between expected and actually matched loop-closing features.

In the absence of loop closures, current measurements (projections) only depend on their initial stereo keyframe and on the current relative pose w.r.t. that frame, see dead reckoning in Section 5.4.3. In the event of loop closure, however, current projections also depend on the camera motion history, *i.e.*, on all relative transformations and stereo feature triangulations even since the creation of the newly regained features, see Fig. 5.14.



Object
KLT features
$_{\mathrm{r}}\mathcal{I}^N$
$_{\mathrm{l}}\mathcal{I}^N$
SURF features
$_{\mathrm{l}_N}\boldsymbol{T}^{\mathrm{l}_{N+1}}$
Loop closure!
$_{\mathrm{f}}\mathcal{I}^N \triangleq {}_{\mathrm{l}}\mathcal{I}^{N+1}$
$_{\mathrm{r}}\mathcal{I}^{N+1}$
Motion
$_{\mathrm{l}_2}\boldsymbol{T}^{\mathrm{l}_3}$
$_{\mathrm{l}_1}\boldsymbol{T}^{\mathrm{l}_2}$
$_{\mathrm{l}}\mathcal{I}^1$
$_{\mathrm{r}}\mathcal{I}^1$
$_{\mathrm{l}}\mathcal{I}^2$
$_{\mathrm{r}}\mathcal{I}^2$
L
R

Figure 5.14: Skeleton of stereo keyframes 1..N when browsing around an object. During monocular tracking of feature set #N, feature set #1 can be retrieved at images $_{\mathrm{l,r}}\mathcal{I}^{N+1}$. Depending on the distance traveled, loop closing occurs either by monocular tracking of KLT features or with the help of stereo SURF features.

As a consequence, the optimal solution by nonlinear optimization consisting in the minimization of squared reprojection residuals presents higher complexity than the local optimization in Eq. (5.5). Now:

$$\begin{aligned}\widehat{\boldsymbol{\Omega}}_\star = \arg\min \sum_{i=j}^{N}\sum_{p=1}^{M_i}\Big(& ||{}_{\mathrm{l}}\tilde{\boldsymbol{m}}_p^i - {}_{\mathrm{l}}\hat{\boldsymbol{m}}_p^i({}_{\mathrm{l}}\hat{\boldsymbol{p}}_p^i)||^2 \\ & + ||{}_{\mathrm{r}}\tilde{\boldsymbol{m}}_p^i - {}_{\mathrm{r}}\hat{\boldsymbol{m}}_p^i({}_{\mathrm{l}}\boldsymbol{T}^{\mathrm{r}}, {}_{\mathrm{l}}\hat{\boldsymbol{p}}_p^i)||^2 \\ & + ||{}_{\mathrm{f}}\tilde{\boldsymbol{m}}_p^i - {}_{\mathrm{f}}\hat{\boldsymbol{m}}_p^i({}_{\mathrm{l}_i}\widehat{\boldsymbol{T}}^{\mathrm{f}_i}, {}_{\mathrm{l}}\hat{\boldsymbol{p}}_p^i)||^2\Big) \\ + \sum_{p\in\mathcal{R}} & ||{}_{\mathrm{l}}\tilde{\boldsymbol{r}}_p^j - {}_{\mathrm{l}}\hat{\boldsymbol{r}}_p^j({}_{\mathrm{l}_j}\widehat{\boldsymbol{T}}^{\mathrm{f}_j}, \ldots, {}_{\mathrm{l}_N}\widehat{\boldsymbol{T}}^{\mathrm{f}_N}, {}_{\mathrm{l}}\hat{\boldsymbol{p}}_p^j)||^2 \end{aligned} \tag{5.8}$$

where the parameters to be optimized $\boldsymbol{\Omega}_\star = [\boldsymbol{\Omega}^j .. \boldsymbol{\Omega}^N]$ include all history of 3-D features between the older feature set #j being found again, and the last tracked feature set #N (*i.e.*, $N-j+1$ feature sets in total), as well as the $N-j$ relative, inter-keyframe transformations between their respectives keyframes and the final local pose ${}_{1_N}\widehat{\boldsymbol{T}}^{\mathrm{f}_N}$ where the loop was closed (included in $\boldsymbol{\Omega}^N$). In total, this amounts to $\sum_{i=j}^{N}(3{\cdot}M_i{+}6)$ parameters, compared to $3{\cdot}M_i{+}6$ in Eq. (5.5). Note that, due to the non-convexity of the regular BA problem, I am optimizing over (differential perturbations of) non-privileged, relative transformations in order to avoid local minima (Strasdat *et al.*, 2010a). Consequently, feature locations and camera motions are both locally Euclidean, but globally topological; the global Euclidean representation remains as a separate task, left aside e.g. for the realtime meshing application to consistently visualize it in realtime, perhaps augmenting it with the live image stream in Section 5.4.6.

The residual in Eq. (5.8) is composed of the accumulation of residuals $\boldsymbol{\Delta m}_p^i$ in Eq. (5.5), now for every feature set i within the loop, as well as for the subset $\mathcal{R}$ of features contained in feature set j that have been found again in projections ${}_1\tilde{\boldsymbol{r}}_p^j = [{}_{1_{N+1}}\tilde{u}_p^j \; {}_{1_{N+1}}\tilde{v}_p^j]^{\mathrm{T}}$, see Fig. 5.14. In matricial form, the number of equations amounts to $\sum_{i=j}^{N}(2{\cdot}3{\cdot}M_i){+}2{\cdot}\mathrm{size}(\mathcal{R})$, compared to just $2{\cdot}3{\cdot}M_i$ in the case of local BA for dead reckoning in Eq. (5.5).

Optimization processes with system equations of this magnitude clearly benefit from sparse optimization methods if their Jacobians are sparse. Indeed, zero elements are pervasive in the Jacobian of this system of equations w.r.t. the abovementioned parameters:

$$\begin{array}{ccc} & \begin{array}{ccc} \overbrace{\quad\boldsymbol{\Omega}^j\quad} & \cdots & \overbrace{\quad\boldsymbol{\Omega}^N\quad} \end{array} & \\ \dfrac{\partial(\tilde{\boldsymbol{m}}-\hat{\boldsymbol{m}})}{\partial\boldsymbol{\Omega}} = & \left[\begin{array}{cc} \dfrac{\partial\boldsymbol{\Delta m}^j}{\partial\boldsymbol{\Omega}^j} & \mathbf{0} \\ & \ddots \\ \mathbf{0} & \dfrac{\partial\boldsymbol{\Delta m}^N}{\partial\boldsymbol{\Omega}^N} \\ \hline \multicolumn{2}{c}{\dfrac{\partial\boldsymbol{\Delta r}^j}{\partial\boldsymbol{\Omega}}} \end{array}\right] & \begin{array}{l} \leftarrow \boldsymbol{m}^j \\ \\ \leftarrow \boldsymbol{m}^N \\ \leftarrow \boldsymbol{r}^j \end{array} \end{array} \tag{5.9}$$

where

$$\frac{\partial\boldsymbol{\Delta r}^j}{\partial\boldsymbol{\Omega}} = \left[\begin{array}{ccccc} & & \vdots & & \\ \dfrac{\partial\boldsymbol{\Delta r}_p^j}{\partial\boldsymbol{\Omega}^j} & \cdots & \dfrac{\partial\boldsymbol{\Delta r}_p^j}{\partial\boldsymbol{\Omega}^k} & \cdots & \dfrac{\partial\boldsymbol{\Delta r}_p^j}{\partial\boldsymbol{\Omega}^N} \\ & & \vdots & & \end{array}\right]$$

$$\frac{\partial\boldsymbol{\Delta r}_p^j}{\partial\boldsymbol{\Omega}^j} = \left[\begin{array}{cccccc|c|ccc|c} ▨ & ▨ & ▨ & ▨ & \square & ▨ & \mathbf{0}_{2\times3(p-1)} & ▨ & ▨ & ▨ & \mathbf{0}_{2\times3(M_i-p)} \\ ▨ & ▨ & ▨ & \square & ▨ & ▨ & & ▨ & ▨ & ▨ & \end{array}\right]$$

$$\frac{\partial\boldsymbol{\Delta r}_p^j}{\partial\boldsymbol{\Omega}^k} = \left[\begin{array}{cccccc|c} ▨ & ▨ & ▨ & ▨ & ▨ & ▨ & \mathbf{0}_{2\times3M_i} \\ \blacksquare & ▨ & ▨ & ▨ & ▨ & ▨ & \end{array}\right]$$

$$\frac{\partial\boldsymbol{\Delta r}_p^j}{\partial\boldsymbol{\Omega}^N} = \left[\begin{array}{cccccc|c} ▨ & ▨ & ▨ & ▨ & ▨ & ▨ & \mathbf{0}_{2\times3M_i} \\ ▨ & ▨ & ▨ & ▨ & ▨ & ▨ & \end{array}\right] \tag{5.10}$$

∗ White boxes correspond to zero elements;
gray or black boxes to non-zero ones.

I now go into detail about the calculation of the black Jacobian element highlighted above concerning features within $\mathcal{R}$ that have been tracked again.

The estimated reprojection ${}_{\mathrm{l}}\hat{\boldsymbol{r}}_p^j$ of these features from feature set $\#j$ onto the **l**eft camera frame at keyframe $\#N{+}1$ is a function of both, the 3-D structure ${}_{\mathrm{l}}\hat{\boldsymbol{p}}_p^j$ of the original set $\#j$ and the current left camera pose ${}_{\mathrm{l}_j}\widehat{\boldsymbol{T}}^{\mathrm{f}_N}$ that, in turn, is a function of all local transformations by dead reckoning lying between keyframes $\#j$ and $\#N+1$. Here the calculation of its partial derivative w.r.t. the first Euler angle ${}_{\mathrm{l}}\alpha^k$ of the differential perturbation $\Delta\widehat{\boldsymbol{T}}_{\mathrm{l}_k}^{-1}$ at the **l**eft camera frame of keyframe $\#k$ is detailed:

$$\frac{\partial({}_{\mathrm{l}_{N+1}}\tilde{v}_p^j - {}_{\mathrm{l}_{N+1}}\hat{v}_p^j)}{\partial\, {}_{\mathrm{l}}\alpha^k} = -\frac{\partial {}_{\mathrm{l}_{N+1}}\hat{v}_p^j}{\partial\, {}_{\mathrm{l}_{N+1}}\hat{y}_p^j}\,\frac{\partial\, {}_{\mathrm{l}_{N+1}}\hat{y}_p^j}{\partial\, {}_{\mathrm{l}}\alpha^k} - \frac{\partial {}_{\mathrm{l}_{N+1}}\hat{v}_p^j}{\partial\, {}_{\mathrm{l}_{N+1}}\hat{z}_p^j}\,\frac{\partial\, {}_{\mathrm{l}_{N+1}}\hat{z}_p^j}{\partial\, {}_{\mathrm{l}}\alpha^k}$$

where

$$\begin{cases} {}_{\mathrm{l}_{N+1}}\hat{v}_p^j = (\,\beta\,\dfrac{{}_{\mathrm{l}_{N+1}}\hat{y}_p^j}{{}_{\mathrm{l}_{N+1}}\hat{z}_p^j} + v_0) \\ \begin{bmatrix} {}_{\mathrm{l}_{N+1}}\hat{x}_p^j \\ {}_{\mathrm{l}_{N+1}}\hat{y}_p^j \\ {}_{\mathrm{l}_{N+1}}\hat{z}_p^j \\ 1 \end{bmatrix} = {}_{\mathrm{f}_N}\widehat{\boldsymbol{T}}^{\mathrm{l}_{k+1}}\,\underbrace{\Delta\widehat{\boldsymbol{T}}_{\mathrm{l}_{k+1}}^{-1}}_{\text{perturbation}}\,{}_{\mathrm{f}_k}\widehat{\boldsymbol{T}}^{\mathrm{l}_j} \begin{bmatrix} {}_{\mathrm{l}_j}\hat{x}_p^j \\ {}_{\mathrm{l}_j}\hat{y}_p^j \\ {}_{\mathrm{l}_j}\hat{z}_p^j \\ 1 \end{bmatrix} \end{cases} . \qquad (5.11)$$

These few features are of extreme importance, as they produce the only residuals bringing about loop-closing information—else global optimization equals repeated local optimization by dead reckoning in Eq. (5.5).

In reality, the formulation explained above corresponds to the ideal case where all features tracked at loop closure have also been tracked at their triangulation frame, *i.e.*, ${}_{\mathrm{f}}\boldsymbol{m}_p^j$ exists and is included in both Eqs. (5.8) and (5.9); however, features that were not succesfully tracked until keyframe $\#j+1$ can readily be found again when closing the loop. In that case (approximately 15% of the detected features), the residual Eq. (5.8), the optimization parameters $\boldsymbol{\Omega}$, as well as the Jacobian in Eq. (5.9) have to be extended to include their initial projections ${}_{\mathrm{l}}\boldsymbol{m}_p^j$ and ${}_{\mathrm{r}}\boldsymbol{m}_p^j$ as well as their 3-D locations.

My hybrid optimization utilizes the nonlinear, least squares sparse optimization function `sparselm_dercrs()` detailed in (Lourakis, 2010), as well as supernodal sparse Cholesky factorization by CHOLMOD (Chen *et al.*, 2008) and graph partitioning by METIS (Karypis and Kumar, 1999) to observe both primary and secondary sparsity structures of the Jacobian in Eq. (5.9), see (Konolige, 2010). I am providing the abovementioned, full analytic Jacobian in CRS fomat for improved performance. Of course, common derivative components are being stored instead of recalculated. By way of example, timekeeping improves from 94 sec (standard BA *with* full analytic Jacobian) to between 750 ms and 1.4 sec using its sparse variant. Not providing analytic Jacobians proves slower by a factor of 2 or 3. Global BA is performed in a separate computing thread in order not to disrupt concurrent real-time pose tracking and 3-D modeling. In Section 5.5 I show extended loop closure experiments where global BA compensates for substantial dead reckoning errors of several cm in the course of obtaining consistent topology of the map.

Apart from the novel, hybrid nature of my approach to anchor global scale by stereo vision in selected keyframes (which incidentally deskills local pose tracking), my work differs from similar relative implementations in the general SLAM literature in Refs. (Mei *et al.*, 2010; Sibley *et al.*, 2009; Konolige and Agrawal, 2008; Strasdat *et al.*, 2010a; Strasdat *et al.*, 2011; Clipp *et al.*, 2010; Lim *et al.*, 2011) since accurate motion tracking is here required globally, for the whole motion history, whereas in SLAM metric accuracy is encouraged only locally, as global topological integrity suffices (Sibley *et al.*, 2009).

5.4.6 Real-Time Surface Reconstruction and Correction

Manual 3-D scanning requires visual feedback to the user for timely and successful acquisition of whole 3-D models. For this purpose we designed a streaming surface reconstruction method that delivers realistic 3-D models *in-the-loop*, concurrently with 3-D acquisition of unorganized range data as well as pose tracking.

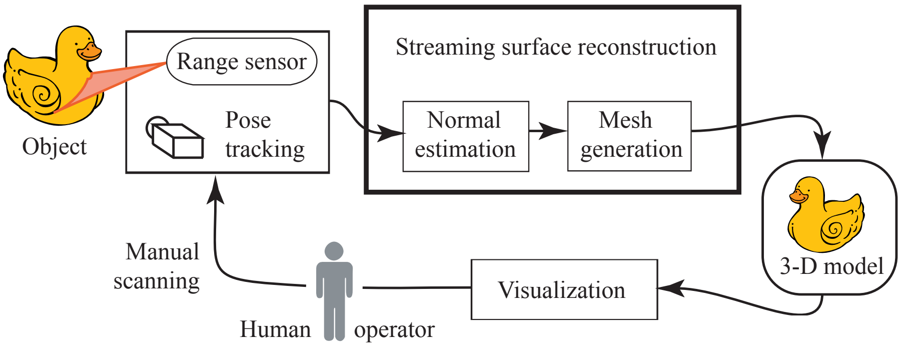
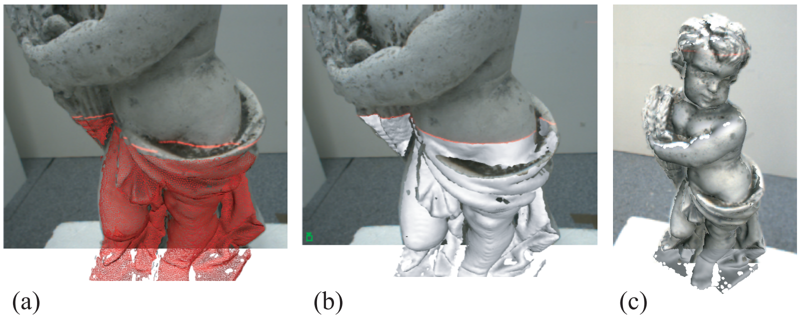


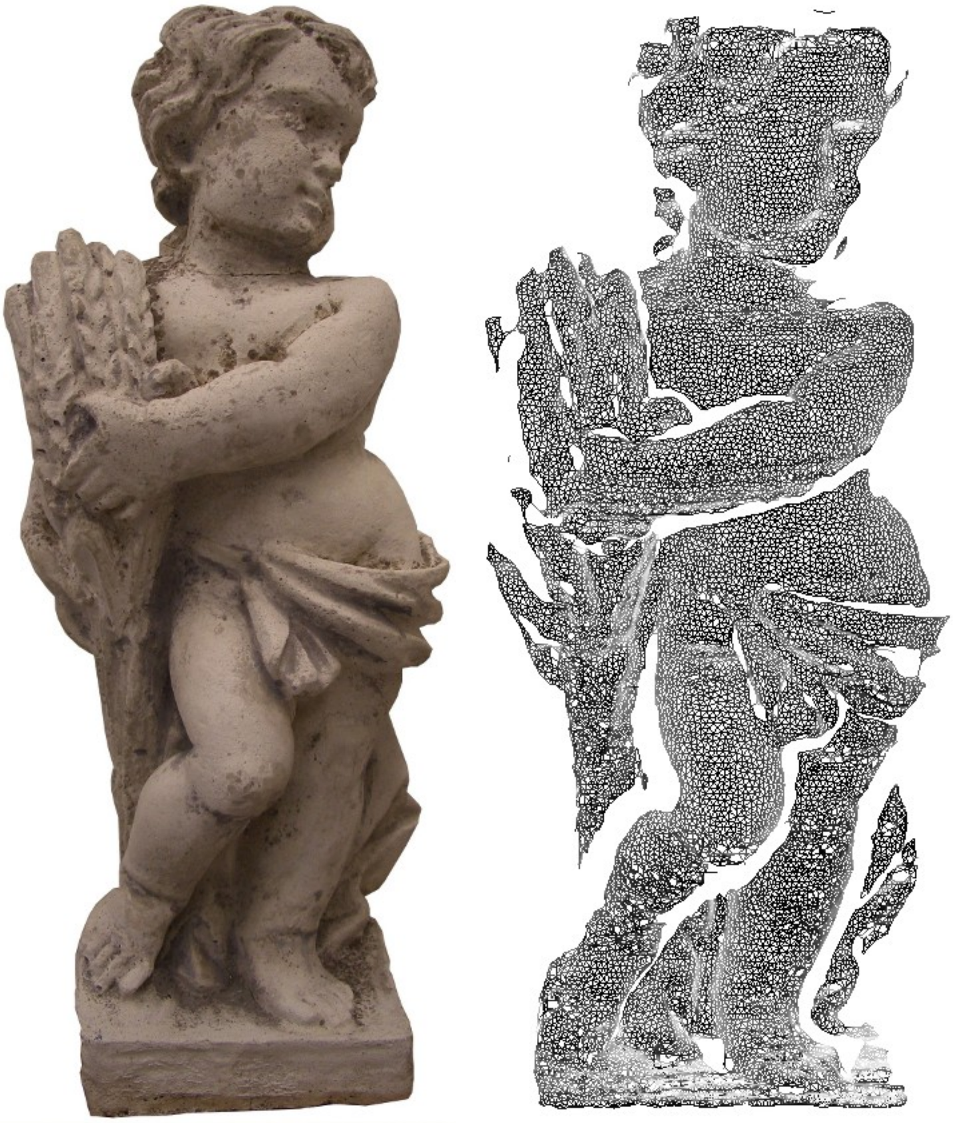
Figure 5.15: 3-D modeling pipeline including fusion of range and pose data, online surface reconstruction by normal estimation and mesh generation, and 3-D model rendering for visualization.

In detail, the real-time method iteratively generates a dense and homogeneous triangle mesh in Euclidean space by inserting sample points from real-time data streams and motion readings from e.g. visual pose tracking, refining the surface model locally around each new sample point, see Fig. 5.15. A dynamic spatial data structure using an extendable octree ensures prompt access to growing pointsets as well as continuously updated meshes without restrictions to object size or number of sample points, see (Bodenmüller, 2009). The generated model can then be accessed at any time, e.g. for visualization and optional live image stream registration, see Fig. 5.16.



(a) (b) (c)

Figure 5.16: Online visualization by augmented triangle mesh (a) or surface model (b), leading to a textured model (c).



(a) Picture (b) Front triangle mesh

Figure 5.17: Resulting mesh from a front scanning sweep.

5.4.7 Summary

This section presented the required algorithms to provide pose tracking estimations in realtime out of monocular tracking of salient features by using a robust V-GPS algorithm in the context of an extended KLT feature tracker; the features have been accurately reconstructed in 3-D by one-time stereo vision. Perhaps the major challenge in this context is tracking features in close-range; unlike in the case of long-range tracking, close-range feature tracking is affected by translations to a similar extent as by rotations. This is aggravated by the fact that the DLR 3D-Modeler is a hand-guided device prone to jerky motion. My novel optical flow prediction scheme #5 (Strobl *et al.*, 2009a) leverages the rotational readings of an IMU for improved estimation of the displacement of features in between frames. An alternative method, coined optical flow prediction scheme #6 (Strobl *et al.*, 2011), works without IMU readings by casting the KLT feature tracker unto the Active Matching (AM) paradigm, achieving robust feature tracking at an even higher motion bandwidth.

3-D modeling by browsing an object is largely an exploratory task where loop closures are rare. Hence pose tracking in realtime largely relies on dead reckoning, which is inconvenient as visual odometry invariably accumulates drifts. The ultimate goal of 3-D modeling is, however, the complete reconstruction of objects e.g. by scanning all around the object. This event always involves (at least) one loop closure that provides the opportunity to greatly increase present and past pose tracking accuracy so that the final 3-D model will excel in accuracy irrespective of prior motion drifts by visual odometry. It is worth noting that visual odometry is still perfectly useful during the browsing period as it is mainly used to provide live image augmentation and timely meshing results for the user to be fully aware of the fulfillment of the 3-D modeling task, and secondarily to support rapid, local loop closures. For these reasons, in this work I also introduce graph-based nonlinear optimization of the tracked pose by minimization of residual reprojection errors. In the context of the current localization framework, I opt for a hybrid, keyframe-based bundle adjustment (kBA) algorithm on stereo keyframes and monocular views because kBA is allegedly the most accurate and efficient option to tackle this problem in the face of higher number of features and keyframes (Strasdat *et al.*, 2010b). It is by the relative/topological nature of the approach that the eventual optimization will be sparse, yielding rapid optimization of the whole tracking history. Loop closing as well as relocalization in the case of interrumpted feature tracking are supported by the use of appearance-based SURF descriptors and rapid, coarse relocalization using a bank of parallel three-point-perspective pose solvers. In the end, highly accurate motion history is delivered to the meshing algorithm that is able to refine the whole object model using both, accumulated range data and newly optimized motion, within a second.

I learned that bundle adjustment (BA) for dead reckoning estimation hardly improves accuracy compared with V-GPS. In the case of loop closures, however, the use of BA makes large pose corrections possible. In addition, I learned that it is crucial to consider the sparsity of the pose-graph optimization problem for BA to perform in a timely manner.

5.5 Experimental Validation

In this section I first describe the operation of the proposed visual pose tracking methods by detailing on a challenging sequence. Second, the accuracy of the approach is addressed by assessing the consistency of loop closures as well as by pre-defined motions in concert with a rigidly attached robotic manipulator that acts as ground truth. Third, the computational efficiency is evaluated. For a more descriptive demonstration please refer to the supplementary videos—during scanning, hectic movements were intentionally performed to prove the robustness of the system.

5.5.1 Operation

I illustrate the operation of the proposed methods both, with the assistance of a synchronized and calibrated IMU for resilient feature tracking, and without it, by following the Active Matching paradigm, refer to Section 5.4.2. May I suggest that the reader retrieves the processed video streams from the Internet that show the look and feel of visual pose tracking of the DLR 3D-Modeler at:

- `http://www.robotic.de/Klaus.Strobl/iros2009`,
- `http://www.robotic.de/Klaus.Strobl/icra2011`.

The challenging sequence is composed of 625 images acquired at 25 Hz for a period of time of 25 s. The hand-held 3D-Modeler targets a 40 cm tall sculpture at a range of approximately 35 cm, sweeping up and down the figure three times similar to scanning it. Both the distance to the sculpture and the rough view direction to it are maintained. However, during that time the camera suffers from very strong, saccadic movements, which create an optical flow the size of 40 pixels. The IMU readings state maximal orientation changes of 2.5° and translations of up to 1 cm (*i.e.*, 62°/s and 0.25 m/s) between images (*i.e.*, within 40 ms).

The visual tracking method presented in Section 5.4.2 sequentially localizes the camera w.r.t. eight different sets of points in realtime. The sequence is initialized by a set of 3-D points *Set*#1, which is composed of 25 points and this is also the average number of features in the following sets. Fig. 5.18 (a) shows *Set*#1. The camera moves downwards, see Fig. 5.18 (b), and five further sets of points are initialized, one after another. Then the camera reaches its lowest position and starts moving back to the top. Here the algorithm does not create new sets of points but detects former ones following the policies in Section 5.4.3, see Fig. 5.18 (c), and leaps onto them. Fig. 5.18 (d) traces these changes during the entire sequence; note two additional sets at images number #298 (*Set*#7) and #349 (*Set*#8). In the end, the camera returns to the initial area where the algorithm refers back to *Set*#1.

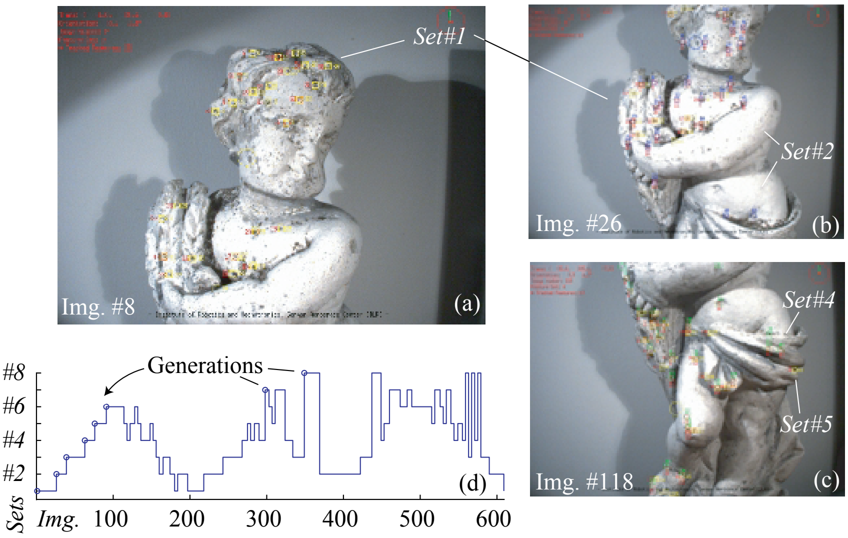


Figure 5.18: (a) Image #8 tracking *Set*#1. (b) Image #26 after generation of *Set*#2, changing reference. (c) Image #118 while retrieving *Set*#4. (d) Reference sets history during the experiment.

The behavior defined by the policies in Section 5.4.3 yields successful tracking all the time. It seamlessly leaps from current reference sets onto former ones (local loop closure), which implies bias-free round-scanning, *i.e.*, the positioning accuracy at the end of the sequence equals the accuracy at the beginning.

The tracking method based on the Active Matching paradigm presented in Section 5.4.2 does a similar job *without* the help of an IMU. Fig. 5.19 displays a typical frame highlighting both active features, the validation set, as well as remaining features. The authors suggest that the reader retrieves the processed sequence from the Internet for in-depth examination.

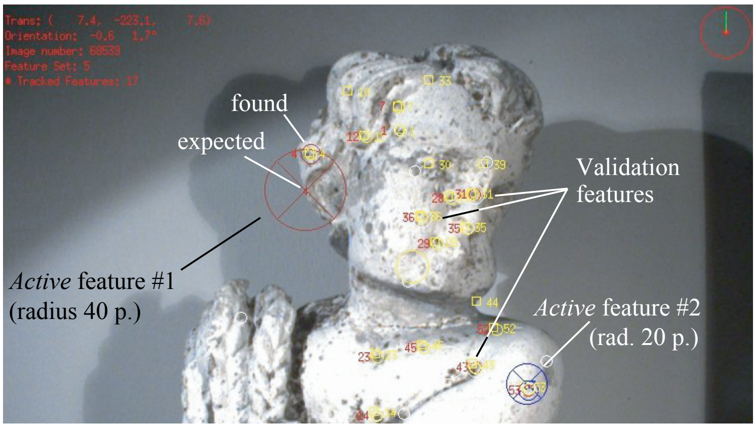


Figure 5.19: Image frame including two *active* features, three validation features, and a number of current and past regular features.

5.5.2 Positioning Accuracy

Loop closing is the most natural option for assessing pose tracking accuracy, as pose estimation is possible w.r.t. both, original and present features, immediately after detection of the closure. Subject to the original and the present vantage points w.r.t. the original features, pose estimation w.r.t. that features is truly very accurate, which virtually acts as ground truth to ongoing dead reckoning estimations.

Fig. 5.20 depicts a complete scanning procedure around a 50 cm tall sculpture. A natural browsing procedure asks for prolonged scanning sweeps and is characterized by the absence of loop closure events (neither local nor global), *i.e.*, only dead reckoning estimation is possible. The video at `http://goo.gl/1Bx6eE` shows 4 sweeps featuring a roll angle of 90° between them, a total length of 320 cm and an accumulated rotation of 360°, which certainly bring about dead reckoning errors higher than the tolerated for accurate 3-D modeling. In this event, we close the motion loop as explained in Section 5.4.5, which corrects current and former pose estimation within a second, and subsequently the whole mesh of the 3-D model as well.

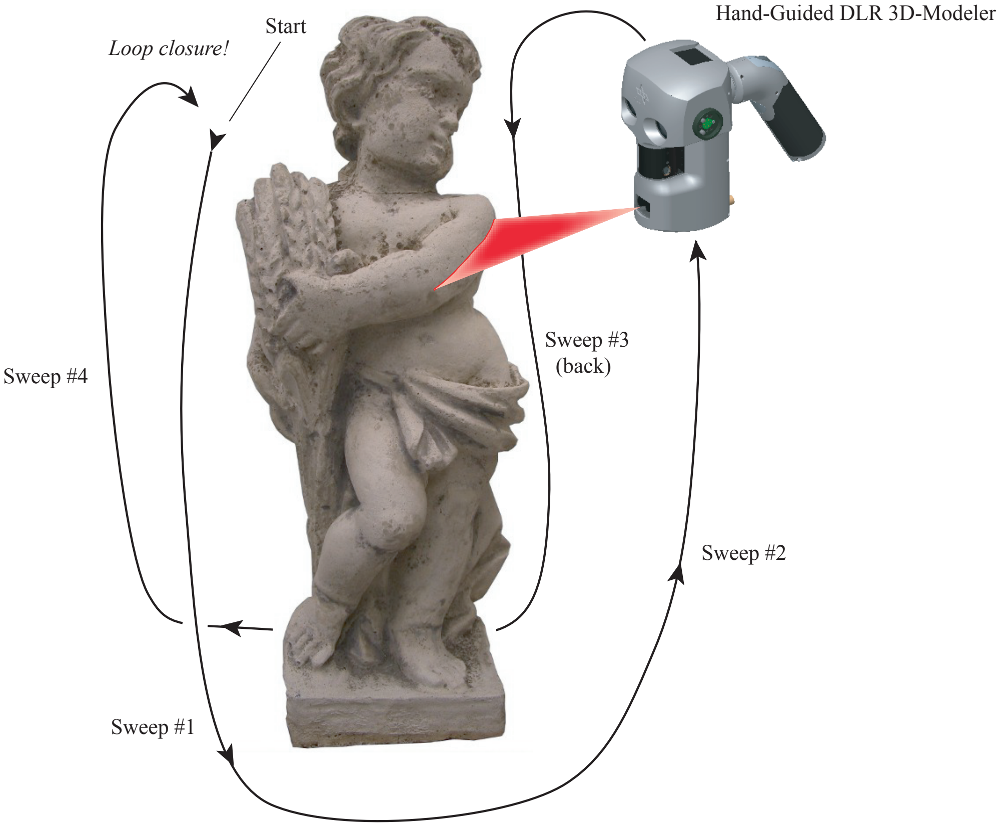


Figure 5.20: The hand-guided DLR 3D-Modeler browsing all around the sculpture.

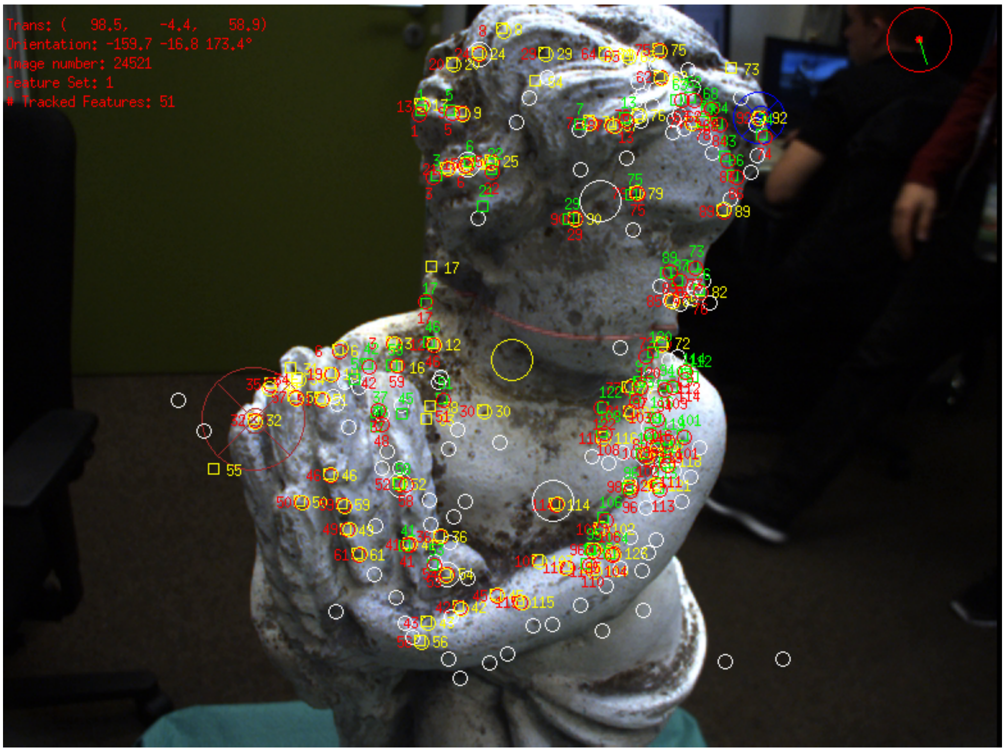
The video sequence starts tracking salient features in frame #23033, featuring 4 sweeps in 90° relative yaw angles, prior to loop closing in frame #24521 for a total motion length of 320 cm. During the whole trayectory 44 feature sets are initialized by feature-based stereo vision.

As can be seen by the drift of the white circles corresponding to the features of the two first datasets, dead reckoning errors accumulate to an extent that precludes seamless KLT tracking when trying to retrieve these sets based on their expected relative pose to the camera—even in its AM implementation. Appearance-based relocalization on stereo images (triggered on a sensible basis based on the rough pose of the camera) may detect older SURF features, but their positioning accuracy by stereo vision is still insufficient. It is only by the inclusion of the intermediate stage concerning P3P pose estimation on KLT features with larger search regions (see Section 5.4.4) that we achieve the required pose accuracy for seamless KLT tracking of 55 features pertaining to the feature set #1. After that, pose refinement by global, hybrid BA as explained in Section 5.4.5 takes place. Of course, relocalization on SURF features and subsequent P3P pose estimation happen at an older image frame because these prior stages run in a separate computing thread. After successful pose refinement by P3P pose estimation, the AM implementation of the extended KLT tracker presented in Ref. Strobl *et al.* (2011) takes over, tracking as many features of the original feature set #1 as possible, cf. Fig. 5.21. These 55 features in turn trigger the global, hybrid BA process explained in Section 5.4.5 in a separate computing thread. It is only by image frame #24535 that pose refinement on the whole pose graph is finished, updating all 43 relative transformations ${}_{1_i}\boldsymbol{T}^{\mathrm{f}_i}$, $\forall i \in \mathbb{N}_1$, $i \leq 44$ along with the 3-D pose of all 1816 features $\boldsymbol{p}_p^i$, $\forall p \in \mathbb{N}_1$, $p \leq M_i$. This step cannot be appreciated in the sequence because, ever since the successful tracking in frame #24521, data is already being represented in the local reference frame of feature set #1.

Using a dated notebook equipped with an Intel® Core$^{\text{TM}}$ 2 Duo P8700 processor, the robustified nonlinear optimization takes 870 ms. The parameters vector contains all features and relative poses involved in the loop closure; its size amounts to 5769. The size of the residuals vector is 11090 including past and current *hybrid* residuals on stereo and monocular images.

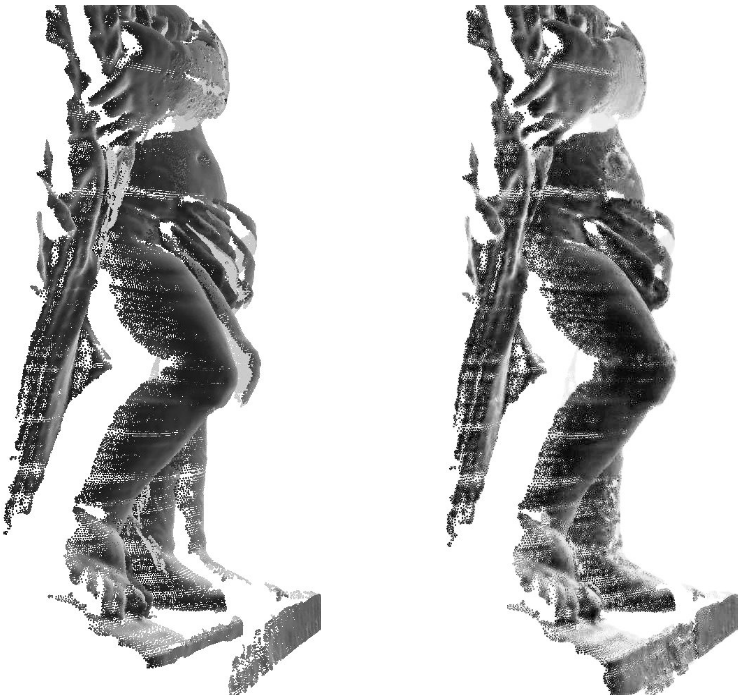
The final pose correction after 320 cm of dead reckoning estimation amounts to 2.5 cm and 6.5°. The appearance-based stage in Section 5.4.4 misses the point by 7.5 mm and 1.5°, which is still adequate for successful tracking by the AM implementation of KLT tracking in (Strobl *et al.*, 2011). Figs. 5.22 and 5.23 show typical corrections of the resulting 3-D pointcloud and full mesh after successful closure of the loop.

Note that the LSP is active for a second process to segment laser stripe projections and subsequently triangulate range data (Strobl *et al.*, 2004). A third process performs online meshing of 3-D data on the original camera frame (*i.e.*, at the camera frame at image frame #23033). Refer to the older videos in the above links to learn more about these concurrent processes.



```
Trans: (   98.5,    -4.4,    58.9)
Orientation: -159.7 -16.8 173.4°
Image number: 24521
Feature Set: 1
# Tracked Features: 51
```

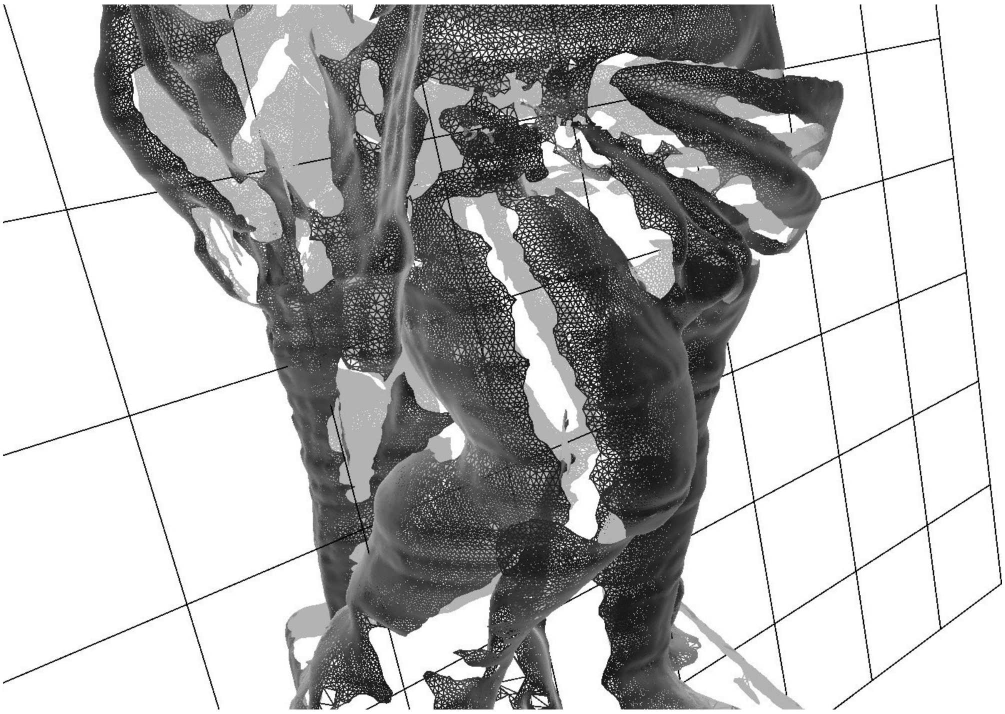
Figure 5.21: Parallel tracking of feature set #43 and loop-closing set #1 at the loop-closing image frame #24521. Parallel tracking is necessary to build up feature drift information required for robust tracking following (Strobl *et al.*, 2011). **Please find the high-resolution sequence at:** `http://goo.gl/1Bx6eE`.



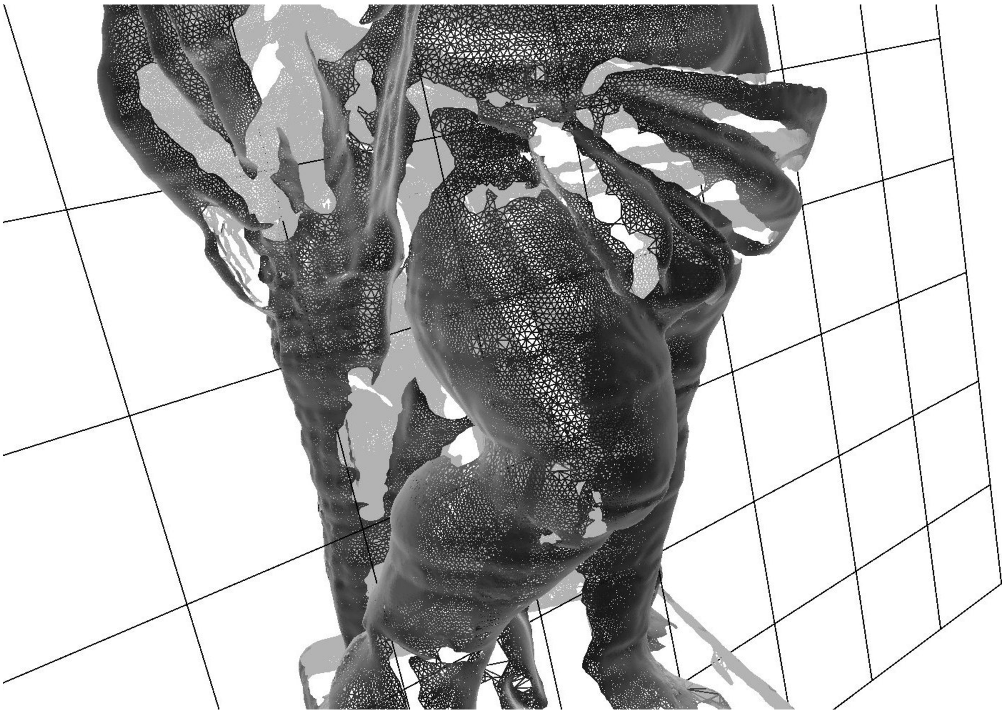
(a) Prior to loop closure

(b) After loop closure

Figure 5.22: Pointcloud correction after successful closure of the loop.



(a) Prior to loop closure



(b) After loop closure

Figure 5.23: Mesh correction after successful closure of the loop.

I also compare the pose tracking accuracy of my method with an **external positioning system**: the Kuka KR 16 robotic manipulator featuring ∼0.1 mm and less than 0.1° accuracy. The DLR 3D-Modeler was attached at the TCP of the manipulator. Still, translational readings are on a different reference frame and have to be calibrated w.r.t. the camera reference frame, see (Strobl and Hirzinger, 2006; Strobl *et al.*, 2005), which is subject to errors. As a consequence, *the results of this comparison ought to be considered a worst-case estimation of their accuracy.*

A motion around an object is performed, total length of 125 cm and 55°, featuring 710 stereo frames. The images are synchronized with the robot's motion following Ref. (Bodenmüller *et al.*, 2007). Fig. 5.24 shows residual errors in translation and rotation. Additionally, motion estimation by non-robustified V-GPS is also shown in order to realize the significance of the robustified variant introduced in Section 5.4.3. Pose tracking error by dead reckoning increases up to 3 mm and 0.4° at the turning point; on its way back, the error is removed thanks to the retrival of former sets of points. It is worth noting that the vanilla, non-robustified V-GPS in (Burschka and Hager, 2003) cannot provide reliable position due to the outliers (up to 2 cm error even though its orientation accuracy is still fair). These results featuring less than 1% error in range match former visual odometry results in (Nistér *et al.*, 2004; Cheng *et al.*, 2006).

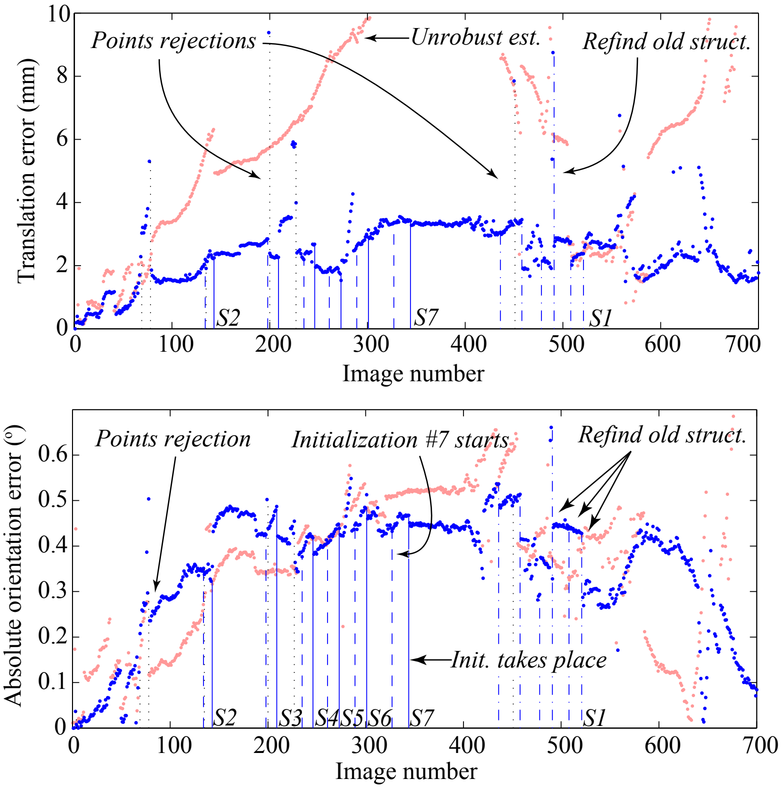


Figure 5.24: Residual translation (*upper*) and rotation (*lower*) errors w.r.t. the robotic manipulator using RVGPS (blue) and using V-GPS (pink); in the latter case, translation error rises to more than 2 cm.

5.5.3 Performance

Last I recap on typical processing times of visual pose tracking on the DLR 3D-Modeler. Recall that these computations are in parallel to LSP triangulation (Strobl *et al.*, 2004) and surface reconstruction in Section 5.4.6 (Bodenmüller, 2009).

- Feature-based stereo triangulation (Section 5.4.1) takes approximately 300 ms (for 50 features).
- High-rate, 2-D feature tracking:
  - Using an IMU (optical flow prediction scheme #5) it takes approximately 18 ms (for 25 features).
  - Following AM (optical flow prediction scheme #6) it takes $3+1.2+0.6+8 = 12.8$ ms (for 50 features).
- The robustified V-GPS estimation takes 3 ms (for 50 features), refer to Section 5.4.3.
- Local bundle adjustment as in Section 5.4.3 takes 6 ms.
- Large-scale loop closing (global bundle adjustment) as in Section 5.4.5 takes 650 ms (including 48 stereo keyframes and 2100 features).
- Appearance-based relocalization as in Section 5.4.4 takes approximately 600 ms.
- The costs of video stream visualization should not be neglected in the case of weaker devices. Otherwise frame-rate visualization typically needs less than 3 ms.

Most of these tasks are independent threads themselves. Computing times are on an aged Intel® Core™ 2 Duo P8700 processor notebook.

5.6 Summary and Discussion

In this chapter I describe the required algorithms that instantiate the first 3-D modeling device for close-range applications that localizes itself passively from its own images in realtime, at a high data rate. This is an important contribution in order to increase flexibility for this type of devices, doing without external positioning systems that constrain the system in size, mobility, and cost.

A comprehensive review of 3-D modeling systems in Section 5.2 points out the lack of devices able to passively localize themselves at a high data rate. I implemented a visual pose tracking algorithm tailored to 3-D modeling by carefully engineering its key processes: relative motion is delivered at a high data rate from feature tracking on a monocular image stream using a novel, robustified V-GPS algorithm characterized by its efficiency and accuracy, see Section 5.4.3; in turn, feature tracking is based upon an accelerated KLT feature tracker, cast into the Active Matching paradigm for improved performance in close-range (close-range feature tracking is twice as hard as in long range), see Section 5.4.2. In order to detach feature set structure estimation from high-rate tracking at the front-end, feature-based stereo vision is being frugally triggered (at keyframe instants only) to compute accurate 3-D feature sets—providing accurate absolute scale at that, see Section 5.4.1; in case of interrupted pose tracking, contingent appearance-based relocalization on known SURF features is provided, together with a rapid pose refinement using a bank of parallel three-point-perspective pose solvers, see Section 5.4.4; finally, potential loop closures are utilized to increase accuracy in motion estimation performing pose-graph optimization in the form of a hybrid, sparse bundle adjustment by minimization of the reprojection errors in a set of stereo keyframes and monocular views (see Section 5.4.5). In addition, real-time reconstruction and texturing of the 3-D model's surface provides visual feedback during acquisition. In Section 5.4.7 I summarize and discuss the details of the implementation. In the end, pose-graph optimization on loop closures delivers *refined* motion history to the online meshing algorithm for display (see Section 5.4.6). Extended validation experiments including links to videos are provided in Section 5.5.

The presented approach is in a position to deskill current 3-D modeling, providing more flexibility in their usage and driving down prices, such that it might even call for reconsideration in areas that traditionally turned away from 3-D modeling.

"It is the mark of an educated mind to be able to entertain a thought without accepting it."
—Aristotle, Metaphysics (980a), 350 B.C.

6
Conclusion

6.1 Summary

This thesis presents the algorithms required for the production of a multisensory hand-held 3-D modeling system that complies with the specific requirements of service robotics applications. In Chapter 1 these requirements are listed, and it is shown that state of the art 3-D modeling systems do not currently comply with them. The algorithms devised throughout this work led to the development of the DLR 3D-Modeler.

The DLR 3D-Modeler is a multipurpose, multisensory platform for geometric and visual perception. It combines complementary sensors in a compact, generic way. The main approaches for depth acquisition include stereo vision, structured light, and laser scanning. Fig. 4.22 shows that these sensors seamlessly cover the desired range of sensing and, what is more, that their expected accuracy levels sensibly compensate one another. In addition, the DLR 3D-Modeler achieves robustness through data fusion: The sensor principles can be compared, and the best one chosen for a specific task; evading and clearing sensor weaknesses can be also accomplished.

Since accurate geometric reconstruction plays a central role in 3-D modeling, a through understanding of the underlying operational principles of all component sensors is required. In Chapter 2 we focus on compact, non-redundant sensor models that feature general validity, as opposed to extended models that are subject to overfitting.

These valid sensor models are parameterized in Chapter 3. This extended chapter introduces novel calibration methods for the sensor components of the DLR 3D-Modeler. This is a central chapter as it makes possible to accurately operate with the DLR 3D-Modeler's sensors in Chapter 4, as well as to visually track their pose by its own cameras in Chapter 5. As a consequence, the DLR 3D-Modeler has been successfully deployed in many applications, see Appendix B. In my view, two key factors decide on the validity of calibration methods: First, whether the method is mathematically sound (e.g. whether it minimizes actual errors for the sake of statistical optimality). Second, whether the method's requirements are low, yielding a straightforward and flexible method—as opposed to traditional methods that entail severe costs like expensive calibration targets and external mesurements. In this context, I identify wrongdoings in standard calibration methods of both, cameras and range scanners, and bring forward novel methods that amend the shortcomings. In the spirit of the softwarization paradigm introduced in Section 1.2, the proposed calibration methods may be computationally more intensive, but they do yield unforeseen benefits:

- Calibration approaches that reduce hardware requirements tend to avert human or measurement mistakes that were otherwise bound to occur; they also enable faster, automatic calibration of elaborate setups as illustrated in Section B.2.4 within Appendix B.
- Accurately parameterized sensor models will eventually render new methods possible, such as accurate, visual pose tracking even in the absence of closures in the motion loop (Chapter 5), or stereo reconstruction on inexistent textures (Section B.2.1 within Appendix B).

The following novel calibration methods are contributed:

1. A novel calibration method for eye-in-hand systems is presented in Section 3.3. Eye-in-hand systems attaching cameras at the end-effectors of robotic manipulators are the most common approach currently used to promote their autonomy. The method estimates both, the hand-eye transformation ${}_{\mathrm{T}}\boldsymbol{T}^{\mathrm{C}}$ as well as the robot-to-world transformation ${}_{\mathrm{B}}\boldsymbol{T}^{0}$. It minimizes Euclidean transformation errors of the external pose tracking system for statistically optimal estimation. Ever since its original presentation in (Strobl and Hirzinger, 2006), the method very much became standard in academia and industry and it has been included in the calibration toolbox DLR CalDe and DLR CalLab (Strobl *et al.*, 2005).
2. In Section 3.4 I identified and addressed the problem of widespread inaccurate knowledge of the geometry of the pattern imprinted on planar calibration targets; this type of patterns are being predominantly used in the context of camera calibration. I note that highly accurate knowledge of the dimensions of the calibration pattern rarely exists, and furthermore that this violation has negative effects on the proper estimation of the camera parameters. I provide two novel methods that yield optimal parametrization of the camera and of its pose w.r.t. an external pose tracking system, irrespective of the actual dimensions of the calibration target (Strobl and Hirzinger, 2008, 2011).

3. I also became aware of the unsuitability of the standard methods for intrinsic and extrinsic camera calibration in the case of cameras with narrow angular field of view. Consequently, in Section 3.5 I bring forward an alternative method merging standard intrinsic camera calibration with the hand-eye extrinsic calibration method proposed in Section 3.3. This improves calibration performance in the case of cameras featuring narrow angular field of view.

4. The traditional camera calibration methods, together with the abovementioned novel methods, have been implemented in a calibration toolbox called DLR CalDe and DLR CalLab (Strobl *et al.*, 2005). The software is freely available worldwide (for academic purposes only). I am main author of DLR CalLab (the calibration part of the toolbox), and my colleagues Wolfgang Sepp and Stefan Fuchs developed DLR CalDe (the corner detection software). The software is ranked in the top three among the freely-available camera calibration toolboxes worldwide. Beyond learning my lessons on algorithmic and computer programming, I learned a lot about maintaining a software package for an active community of users.

5. Section 3.6 details the novel calibration method for the light stripe profiler (a laser triangulation method), originally presented in (Strobl *et al.*, 2004). The method avoids complex calibraton targets as required by state-of-the-art methods, leveraging the prior, accurate intrinsic and extrinsic camera calibrations so that only 3 DoF are left for calibration. This self-calibration approach proceeds by correcting deformations when scanning a planar surface of unknown pose caused by miscalibration of the laser plane.

More often than not, the abovementioned methods become entangled; Fig. 6.1 illustrates their potential functional interactions.

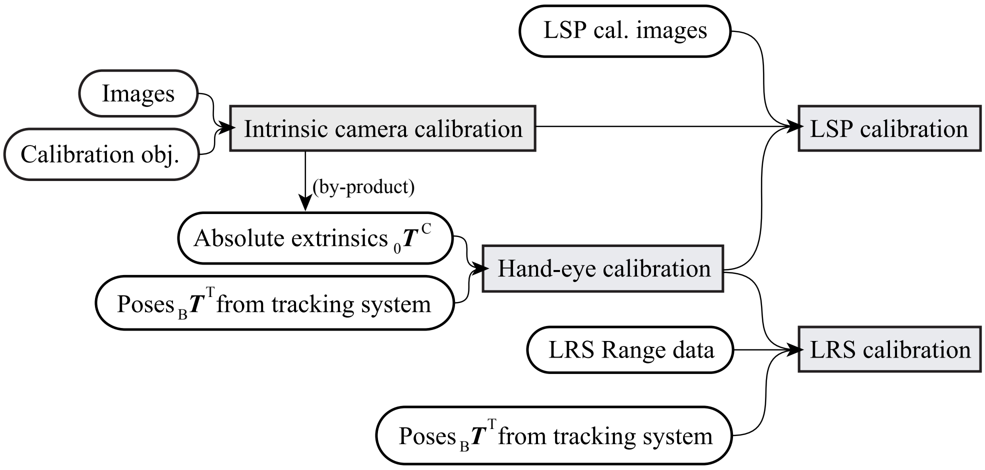

LSP cal. images
Images
Intrinsic camera calibration
LSP calibration
Calibration obj.
(by-product)
Absolute extrinsics $_0\boldsymbol{T}^C$
Hand-eye calibration
Poses $_B\boldsymbol{T}^T$ from tracking system
LRS Range data
LRS calibration
Poses $_B\boldsymbol{T}^T$ from tracking system

Figure 6.1: Functional interaction between the calibration procedures (Figs. 1.2 reprint).

Again in the spirit of the softwarization paradigm, which anticipates operational advantages whenever hardware measures can be successfully substituted by software, in Chapter 4 complex algorithms for sensor data processing are presented. These methods eventually yield unforeseen operational advantages. For instance, the design guidelines for the development or more effective perception systems presented in Section 1.2 mandate multisensory devices that provide multiple data types within a compact, lightweight package. As a consequence, in the case of the light stripe profiler we decided not to use narrow-band optical filters in front of the cameras; filters are otherwise regularly used to filter out non-laser light. This measure makes it harder to obtain robust 3-D data, hence robust methods for software-based segmentation of the laser line are required; I propose a multi-stage method featuring a cascade of detection and validation steps, as well as a dual, crosshair configuration of the sensor. On the other hand, however, unfiltered cameras now allow for concurrent operation of stereo vision, texturing, image augmentation, and even of visual pose tracking based on the same images.

Indeed, this last contribution concerning the possibility to track the pose of the DLR 3D-Modeler by its own images is key to promote autonomy during its operation, and it turned the DLR 3D-Modeler into a worldwide novelty. Due to object self-occlusion, object size, or limited field of view, it is often impossible to acquire a complete 3-D model in a single measurement step. It is common for 3-D modeling devices to revert to external tracking systems in order to represent data in a common reference frame. This option is, however, inconvenient for three reasons: First, they limit the user's mobility; second, they are subject to accurate synchronization and extrinsic calibration, which are cumbersome, error-prone processes; last, it turns out that external tracking systems almost always represent the largest and most expensive part of the 3-D modeling system. In this context, the DLR 3D-Modeler is extended to passive visual pose tracking, yielding the first hand-held 3-D modeling device for close-range applications that localizes itself passively from its own images in realtime, at a high data rate. The approach comprises efficient tracking of distinct features following the active matching paradigm, a frugal use of feature-based stereo triangulation, accurate, relative motion estimation by dead reckoning using a robust V-GPS as well as bundle adjustment, appearance-based relocalization, and real-time reconstruction of the scene. Ideally, objects are completely digitized by browsing around the scene; in the event of closing the motion loop, a hybrid, graph-based nonlinear optimization takes place, which delivers highly accurate motion history to the meshing algorithm that is able to refine the whole object model using both, accumulated range data and newly optimized motion, within a second. The approach has been welcomed by the computer vision community, as it was rated as a finalist to the best paper award at the well-known IROS conference in 2009. Again, inconvenient hardware is hereby substituted by software in the context of the softwarization paradigm.

In Appendix B the methods presented throughout this thesis, as well as the DLR 3D-Modeler, have been successfully applied to a number of scenarios in robotics and beyond.

6.2 Open Directions

“*Dissertations are not finished; they are abandoned.*” —Fred Brooks

This section lists potential extensions of the methods presented in this thesis.

- After so many years maintaining the camera calibration toolbox DLR CalDe and DLR CalLab—fostering researchers especially during failed calibration attempts, I intend to further simplify the camera calibration process by promoting its own degree of autonomy. Novel methods are to be devised to autonomously select the appropriate camera model and optimization method, and even to instruct the user during the data acquisition stage in the first place.

- A straightforward option to simplify hardware concerning stereo cameras is to use a monocular camera instead. In this context, monocular visual pose tracking will be pursued, together with alternative solutions to provide absolute scale.

- An exclusive use of visual pose tracking requires a closer examination of its precision characteristics in 6-D motion. The solution is expected to depend on the camera’s inner geometry, its motion, and the imaged scene. The outcome of this examination ought to enable intrinsic and extrinsic calibration methods that are better suited to the hardware used.

- The DLR 3D-Modeler pushes traditional 3-D modeling forward owing to its flexibility, passivity, and accuracy. It deskills 3-D modeling, driving down prices, such that an extensive review of new applications might enable access to markets that traditionally turned away from 3-D modeling.

"What most experimenters take for granted before they begin their experiments is infinitely more interesting than any results to which their experiments lead."
—Norbert Wiener

A
Homography Estimation in Perspective Projection

In the context of perspective projection using cameras, homographies are linear transformations that geometrically relate *planar* projections of pencils of rays intersecting at the focal point of the camera, on the condition that these projections are represented in homogeneous coordinates. This fact stems from the simple geometric principle of the Thales' theorem. Homographies are widely used in computer vision, where applicable, because they are linear transformations that allow for rapid computation by linear algebra.

In this work we use homographies for initialization of the intrinsic parameters of cameras in advance of their more accurate, nonlinear optimization, see Section 3.2.3. The transformation between image projections ${}_{\mathrm{M}}\boldsymbol{p}_i = [{}_{\mathrm{M}}x_i \; {}_{\mathrm{M}}y_i]^{\mathsf{T}}$ and the actual, 3-D coordinates of the projecting corners ${}_{0}\boldsymbol{x}_i$ can be reduced to the homography between the image projections ${}_{\mathrm{M}}\boldsymbol{p}_i$ and the planar, 2-D coordinates of the corners on the calibration plane as follows:

$$
{}_{\mathrm{M}}\bar{\boldsymbol{p}}_i \propto \begin{bmatrix} {}_{\mathrm{M}}x_i \\ {}_{\mathrm{M}}y_i \\ 1 \end{bmatrix} \propto \boldsymbol{H}_{(3\times3)} \begin{bmatrix} {}_{0}x_i \\ {}_{0}y_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} {}_{0}x_i \\ {}_{0}y_i \\ 1 \end{bmatrix} \tag{A.1}
$$

where ${}_{\mathrm{M}}\bar{\boldsymbol{p}}_i$ are the homogeneous ($\bar{\ }$) coordinates of the image projections. This relationship holds for every single image projection at a particular camera pose w.r.t. the planar calibration object, say I projections. It is worth noting that the relationship only holds in the absence of lens distortion, cf. Section 2.2.1, but then using distorted projections still allows for fairly accurate estimations that will eventually bootstrap final nonlinear optimization of the complete camera model (including lens distortion) in Section 3.2.3.

For every instance $i \leq I$ of Eq. (A.1) we obtain two independent equations

$$ {}_{\mathrm{M}}x_i = \frac{h_{11}\,{}_0x_i + h_{12}\,{}_0y_i + h_{13}}{h_{31}\,{}_0x_i + h_{32}\,{}_0y_i + h_{33}} \tag{A.2}$$

$$ {}_{\mathrm{M}}y_i = \frac{h_{21}\,{}_0x_i + h_{22}\,{}_0y_i + h_{23}}{h_{31}\,{}_0x_i + h_{32}\,{}_0y_i + h_{33}} \quad . \tag{A.3}$$

It is only by accumulating 4 or more instances of these equations that we are in a position to calculate the 8 DoF of the homography matrix $\boldsymbol{H}$. In general, many more instances are available (≈ 100) which makes it impossible to find an exact solution to the system of equations in the face of unavoidable errors either in image processing or in the measurement of the corners of the planar calibration object. Hence we opt for the homogeneous linear least squares solution of the system of all $2{\cdot}I$ equations. To this end, we can rearrange Eqs. (A.2) and (A.3) for every instance i to form the homogeneous linear system of equations

$$\boldsymbol{A}\boldsymbol{h} = \boldsymbol{0} \tag{A.4}$$

where the unknown parameters stack in a vector

$$\boldsymbol{h} = [\, h_{11}\; h_{12}\; h_{13}\; h_{21}\; h_{22}\; h_{23}\; h_{31}\; h_{32}\; h_{33} \,]^{\mathsf{T}} \quad . \tag{A.5}$$

The data matrix $\boldsymbol{A}$ stacks 2 equations for every single instance i seen in the image, *i.e.*, is size $2I \times 9$, as follows:

$$\boldsymbol{A} = [\, \boldsymbol{A}_1 \; \cdots \; \boldsymbol{A}_i \; \cdots \; \boldsymbol{A}_I \,]^{\mathsf{T}} \tag{A.6}$$

where

$$\boldsymbol{A}_i = \begin{bmatrix} {}_0x_i & {}_0y_i & 1 & 0 & 0 & 0 & -{}_{\mathrm{M}}x_i\,{}_0x_i & -{}_{\mathrm{M}}x_i\,{}_0y_i & -{}_{\mathrm{M}}x_i \\ 0 & 0 & 0 & {}_0x_i & {}_0y_i & 1 & -{}_{\mathrm{M}}y_i\,{}_0x_i & -{}_{\mathrm{M}}y_i\,{}_0y_i & -{}_{\mathrm{M}}y_i \end{bmatrix}^{\mathsf{T}} \quad . \tag{A.7}$$

The system of equations in Eq. (A.4) can be solved for the unknowns $\boldsymbol{h}$ following the homogeneous linear least squares method using the singular value decomposition. The right singular vector corresponding to the smallest singular value of the matrix $\boldsymbol{A}$ (ultimately the eigenvector of $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$ that has the eigenvalue closest to zero) contains the solution to $\boldsymbol{h}$ in the least squares sense concerning errors in the null vector in the right-hand side of Eq. (A.4).

It is worth noting that this solution is sensitive to the scale units used in both, the image projections ${}_{\mathrm{M}}\boldsymbol{p}_i$ and the actual, 3-D coordinates of the projecting corners ${}_0\boldsymbol{x}_i$. We suggest in strongest terms to normalize both input data and to relocate them to their respective average values (Hartley, 1997). Of course, the resulting homography will only perform on normalized data, hence it should be scaled back to actual scale units (e.g. pixels against millimeters).

"My mind is made up; don't confuse me with the facts."
—Roy S. Durstine, Advertising & Selling, 1945

B
Experimental Platforms and Applications

B.1 Introduction

In this chapter I focus on major implementations either of the algorithms presented all thougout this thesis or, directly, of the DLR 3D-Modeler. Note that separate experiments that are specific to the contributions of this thesis have been already delivered contiguous to their corresponding descriptions. The successful implementations presented in this chapter are a consequence of the statements in Section 1.1, where I laid out the design guidelines for the algorithms developed in this thesis to be effectively implemented in service robotics applications.

The applications in this chapter range from satellite pose tracking to minimally invasive surgery. The realization of regular camera calibration tasks on innumerable platforms using the algorithms and software provided in this thesis is not included in this chapter, with the exception of special cases like the intrinsic and extrinsic calibration of the 18 cameras mounted on the robotic car RoboMobil, the inverse calibration of a so-called laser pico projector as a virtual (inverted) camera, and the calibration of cameras mounted on a 3-D display by means of a hand-held mirror; this last contribution is the only patent that has been filed in the context of this thesis.

In addition, the camera calibration toolbox DLR CalDe and DLR CalLab (Strobl *et al.*, 2005) will be addressed as a stand-alone application of the methods presented in Chapter 3.

B.2 Experimental Platforms

B.2.1 The Humanoid Robot "Justin"

Introduction

The perception head for the humanoid robot "Justin" is perhaps the most direct application of the DLR 3D-Modeler (Borst *et al.*, 2009). Upon its early completion in 2006 (originally as a torso mounted on a table), "Justin" was topped off with the DLR 3D-Modeler due to its modularity and interfaces. The DLR 3D-Modeler was attached to a pan-tilt unit mounted on the the torso of "Justin," see Fig. B.1, in order to make active perception by saccadic motions possible.

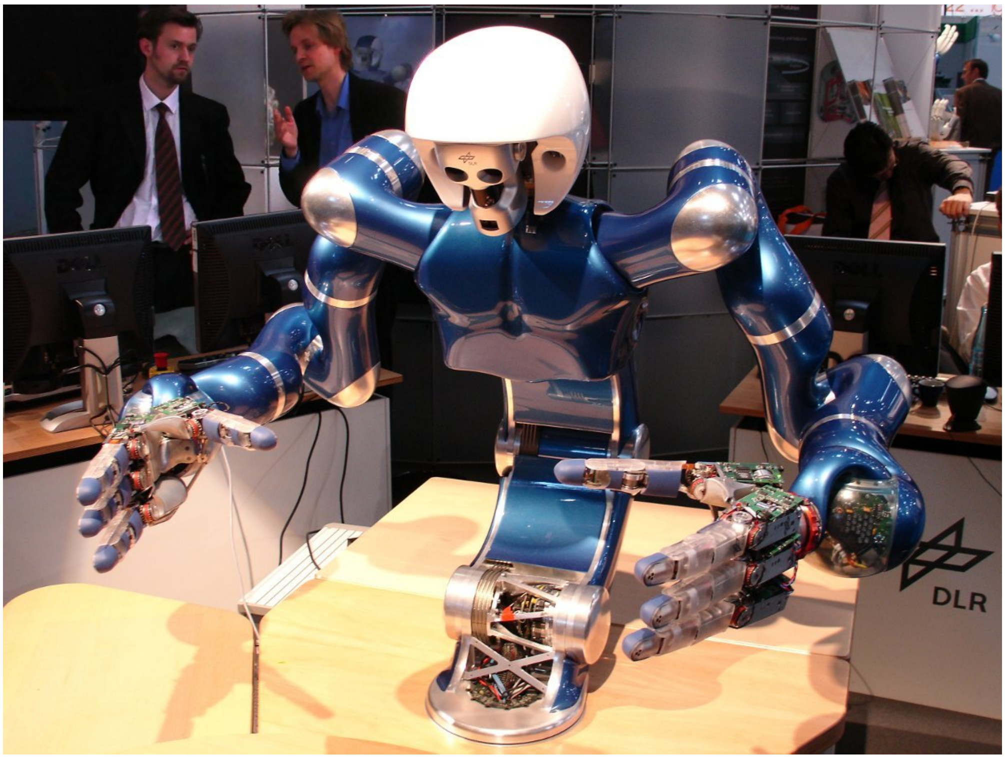


Figure B.1: The DLR 3D-Modeler as the perception head of the original humanoid torso "Justin" at the Automatica Fair 2006.

Extrinsic calibration with respect to "Justin"'s torso

The original "Justin" torso has been conceived for two-handed manipulation using its compliant arms and hands. In order for "Justin" to achieve some degree of autonomy, its perception head is used e.g. to detect and locate objects on its mounting table for it to be able to grasp them. It is therefore essential to deliver range data on robot-related coordinates instead of using e.g. the camera reference frame S_C. This registration process is facilitated if the extrinsic calibration of the DLR 3D-Modeler w.r.t. some reference frame on "Justin" is provided, *i.e.*, if its hand-eye calibration to it is provided, refer to Section 3.3.

Interestingly enough, it turns out that the introduction of a pan-tilt unit becomes essential in order to be able to extrinsically calibrate the DLR 3D-Modeler using the convenient, vision-based methods presented in Section 3.3. Vision-based methods are preferred because reference frames both, at the pan-tilt unit and at the DLR 3D-Modeler, are non-salient locations that have only been defined in theory, e.g. using CAD tools. The method addressed in Section 3.3 is a workaround to avoid direct measurement of these reference frames. It exploits the fact that, if two rotational *motions* of both reference frames (with nonparallel rotation axis) are known w.r.t. a common rigid body (e.g. the scene), it is possible to mathematically estimate the rigid body transformation between the camera frame S_C and the hand or TCP frame S_T without the need for direct measurements on these reference frames. Luckily, one of these motion estimates is exactly what the pan-tilt unit delivers, viz. the pose of its moving head w.r.t. "Justin"'s torso.

The only missing motion data is the motion of the DLR 3D-Modeler w.r.t. the torso. Two options have been considered:

1. To use stereo images of a calibration plate taken by the DLR 3D-Modeler, as performed during intrinsic and extrinsic camera calibration. In actual fact, the stereo camera has been already calibrated in advance. Since using a pan-tilt unit the diversity of perspective projections of a calibration object fixed to the world frame S_0 is naturally limited, the intrinsic parameters of the stereo camera should not be optimized during this extrinsic calibration procedure. The camera calibration toolbox DLR CalDe and DLR CalLab (Strobl *et al.*, 2005) allows for highly-accurate estimation of the pose of the calibration plate w.r.t. the camera (*i.e.*, the absolute extrinsics) similar to an intrinsic calibration, but using fixed intrinsic parameters instead. Additionally, if the other component sensors of the DLR 3D-Modeler have been already calibrated w.r.t. some other extrinsic pose tracking system, the opportunity arises to utilize the method presented in Section 3.9. The disadvantage of this first option is the smaller rotational motions that can be achieved when rotating the stereo camera in front of a fixed calibration plate using a pan-tilt unit. Smaller rotational motions lead to worse conditioning of the hand-eye system of equations in Eqs. (3.6).

2. To use motion readings of an external pose tracking system to which the DLR 3D-Modeler has already been externally calibrated, e.g. the infrared tracking system ARTtrack2. The approach essentially consists in extrinsically calibrating the pan-tilt unit w.r.t. the external pose tracking system by minimizing the discrepancies as explained in Section 3.3; the resulting transformation has to be concatenated with the former hand-eye transformations between the external pose tracking system and e.g. the camera frame S_C. The disadvantage of this option is the necessity to install a tracking system on the working area of "Justin." On the other hand, however, in this case the range of potential rotational motions of the perception head that can be tracked for a better conditioning of the system of hand-eye equations in Eqs. (3.6) is large.

Since the first option is more convenient, I opted to avoid reduced motion ranges by extending the calibration object with multiple origins, see Fig. B.2. The relative position of the origins has been measured in advance of calibration in order to ensure correct association even in the case of partial imaging of the plate. This extension makes it possible to perform larger rotational motions that allow for high-accuracy hand-eye calibration of the DLR 3D-Modeler w.r.t. "Justin"'s pan-tilt unit.

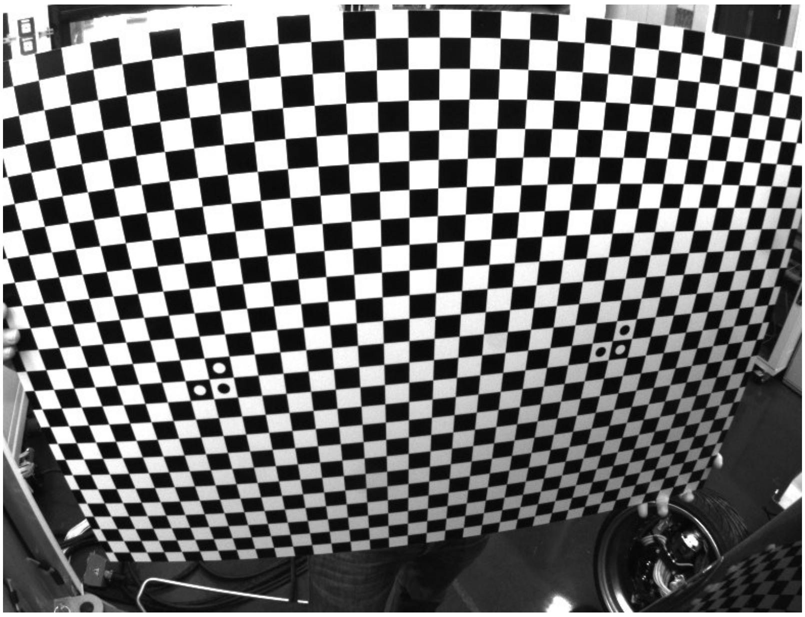


Figure B.2: Calibration plate featuring two origins; it allows for larger rotational motions of the pan-tilt unit compared to the regular calibration plate featuring only sole origin.

The importance of calibration accuracy for stereo vision

In the context of the humanoid robot "Justin," it is indeed crucial to achieve highest accuracy in camera calibration, as illustrated by the following example.

Perhaps the demonstration of "Justin" that is worldwide best known is when servicing on a table, grasping jars and pouring water into glasses, refer to `http://www.youtube.com/watch?v=2tVilONTMfw`. In the scene pictured in Fig. B.3, "Justin" meets with untextured objects like glasses and a carafe. Untextured objects compromise correct association by stereo algorithms. In addition, the objects are transparent so that the stereo algorithm cannot rely on the diffuse reflection of light as e.g. in the case of a planar wall. The stereo algorithm ultimately relies on the accuracy of the intrinsic calibration of the stereo camera by its epipolar geometry in order to find correspondences between both rectified images.

For instance, in the case of the carafe, the most salient features for the stereo algorithm to find correspondences lie on its rim. Due to its circular form, at two points (the top and the bottom projections on the image, see Fig. B.4) these regions project horizontally so that their searches for correspondence using the

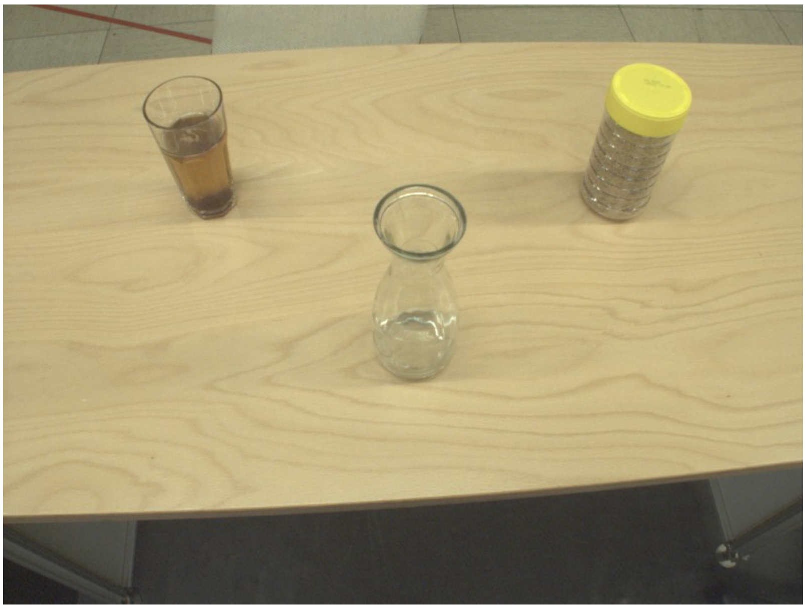


Figure B.3: Typical table scene captured by the main (left) camera of "Justin." Visit `http://goo.gl/XhoCR` to see a video of "Justin" with its hands in the cookie jar.

horizontal epipolar line on rectified images will be very sensitive to erroneous epipolar geometry. In the face of slight calibration errors, the epipolar line corresponding to the top or the bottom projections of the rim either will not meet the borders of the other rectified stereo projection, or it will meet two different areas instead, see Fig. B.4. As a consequence, 3-D artifacts appear that, in the end, yield invalid 3-D reconstruction results as can be seen in Fig. B.5. In that image it is also shown that a precise calibration following the novel method presented in Section 3.4.4 alleviates symptoms compared to an already very precise but standard calibration using DLR CalDe and DLR CalLab.

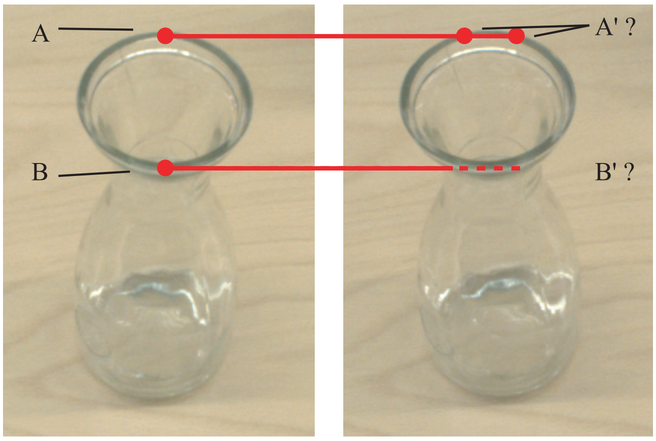
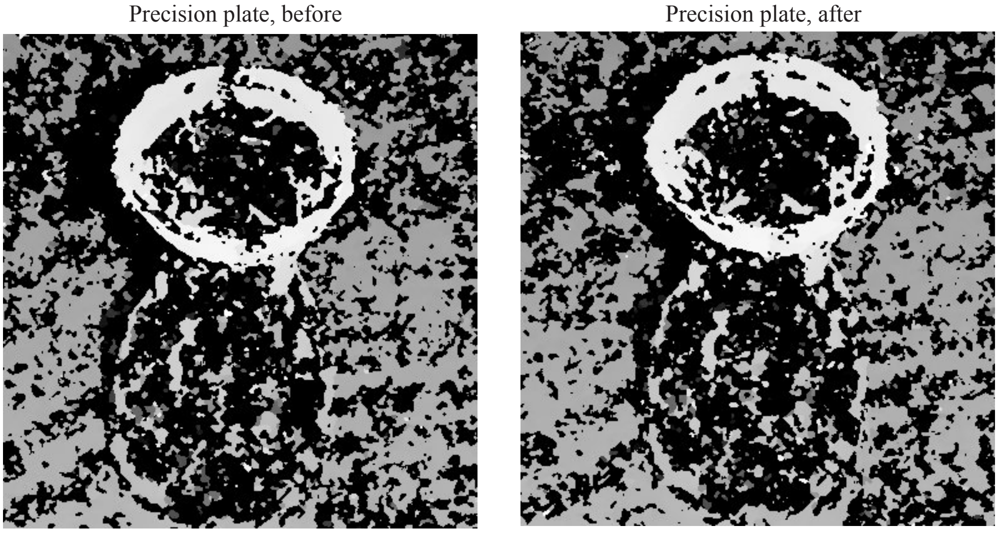


Figure B.4: Search for correspondences between left and right projections of the A and B spots on the rims of the carafe becomes compromised in the case of inaccurate epipolar geometry.



Precision plate, before

Precision plate, after

Figure B.5: Improvement in disparity reconstruction by SGM (Hirschmüller, 2008) of a carafe when opting for the novel method presented in Section 3.4.4 compared to the standard method in Section 3.2 (Fig. 3.25 reprint). The novel method allows for more complete results, particularly in untextured rims parallel to the epipolar line.

Vision-based evaluation of “Justin”’s kinematic accuracy

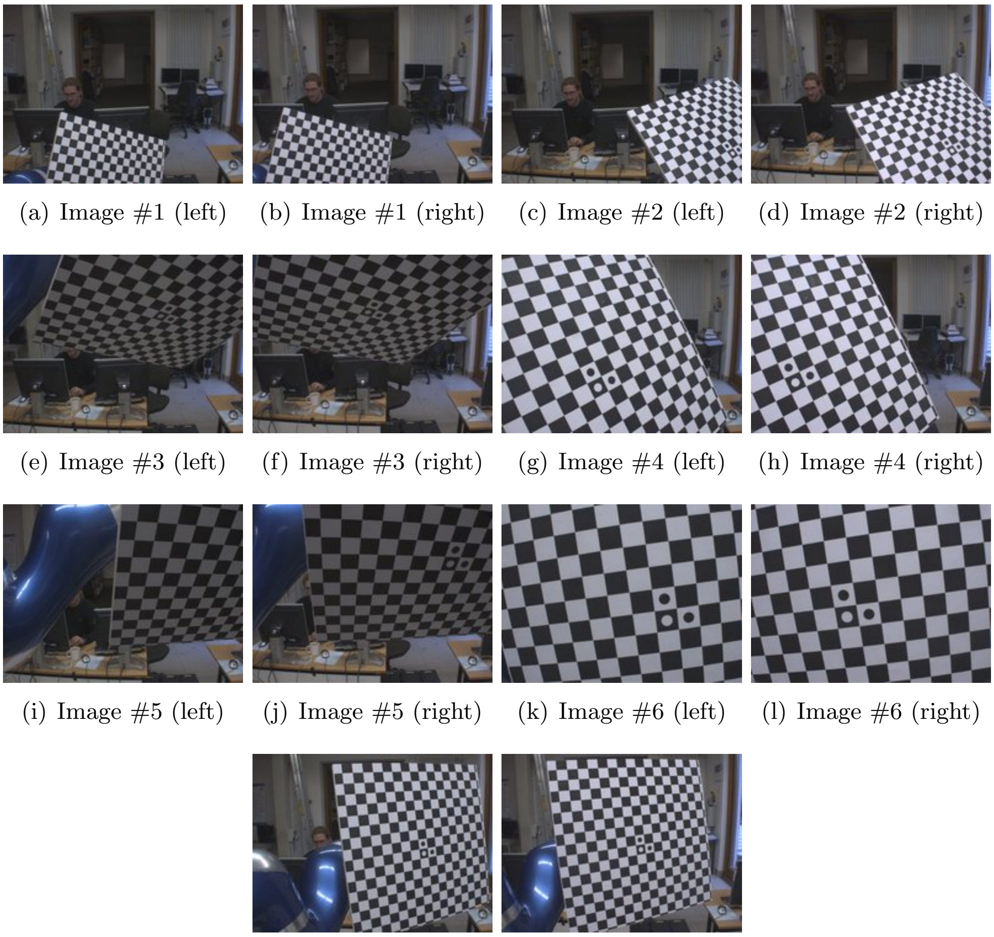
An unexpected use of calibrated cameras on “Justin” has been to assess the accuracy of “Justin”’s kinematics. Since “Justin” is composed of several robotic components, the imprecision in their models and parametrizations accumulate, which is to the detriment of precise manipulation tasks.

Here visual data is used to support a more precise parametrization of the kinematic models of “Justin”. In Fig. B.6 the images used for relative pose estimation are shown.

Two observations have to be considered:

- Image-based pose estimation is more accurate if the calibration plate is *tilted* w.r.t. the principal axis of the camera, refer to Section 3.5 and (Strobl *et al.*, 2009b).
- The configuration of the robotic manipulator ought to differ in every station—irrespective of the orientation of the calibration plate.

In Fig. B.7 augmented reality is used to assess the validity of the kinematic calibration of the whole system.



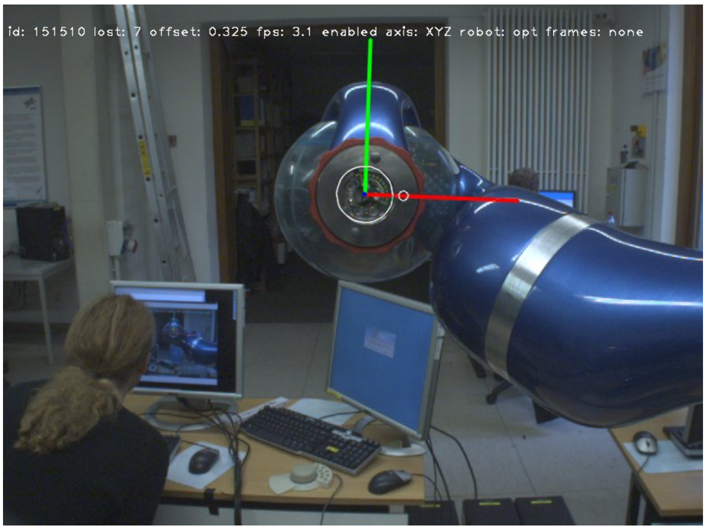
(a) Image #1 (left) (b) Image #1 (right) (c) Image #2 (left) (d) Image #2 (right)

(e) Image #3 (left) (f) Image #3 (right) (g) Image #4 (left) (h) Image #4 (right)

(i) Image #5 (left) (j) Image #5 (right) (k) Image #6 (left) (l) Image #6 (right)

(m) Image #7 (left) (n) Image #7 (right)

Figure B.6: Images used for correction of the kinematic chain of "Justin."



```
id: 151510 lost: 7 offset: 0.325 fps: 3.1 enabled axis: XYZ robot: opt frames: none
```

Figure B.7: Image used for assessment of the validity of the kinematic chain of "Justin."

B.2.2 The HazCam at the ExoMars Rover by the European Space Agency

Introduction

ExoMars (**Exo**biology on **Mars**) is a Mars mission planned by the European Space Agency (ESA) to launch a robotic rover along with stationary and orbiter stations to Mars, in order to search for biosignatures of eventual, past or present life. The mission had been originally conceived as a joint initiative with NASA. Regrettably, starting 2011 and definitely on February 2012, NASA announced that it was compelled to withdraw from the joint mission due to budgetary cuts by the Obama administration. On March 2013, the ESA and the Russian space agency (Roscosmos) agreed to shift NASA's rights and responsibilities to Roscosmos. This option may have saved the mission but, at the same time, it is subject to a number of restraining policies concerning widespread unapproved access to NASA-related original documents by the Russians.

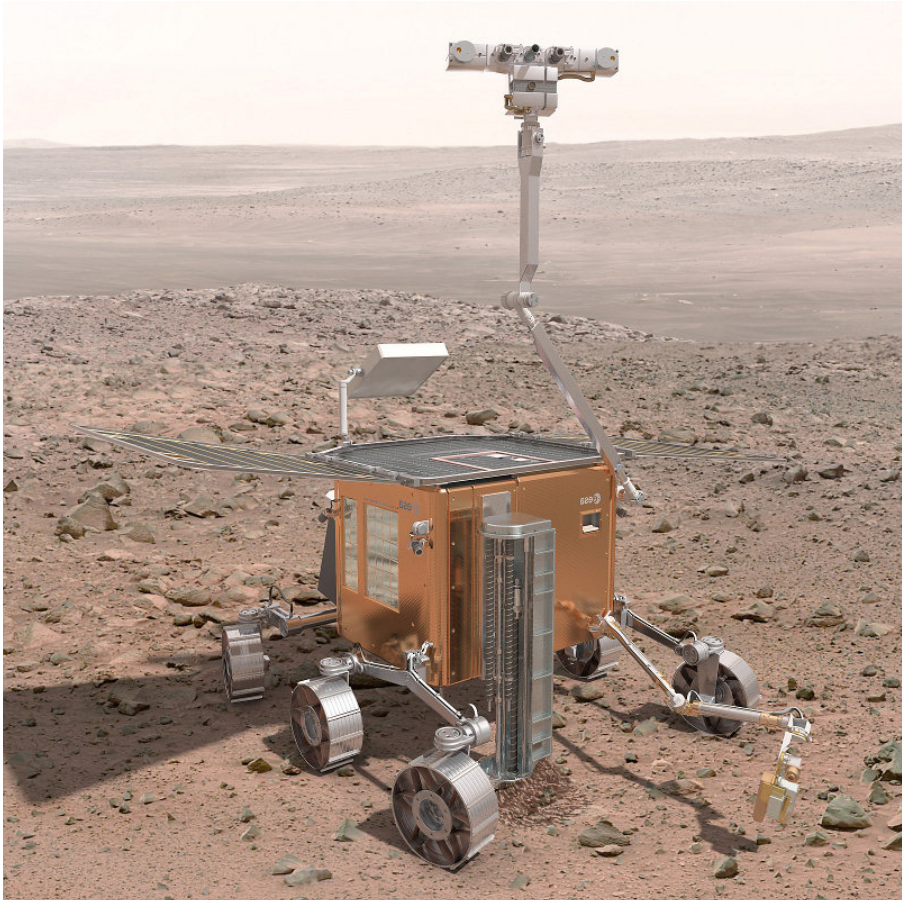


Figure B.8: Photo composition by ESA on the expected Exomars rover. *Courtesy of ESA.*

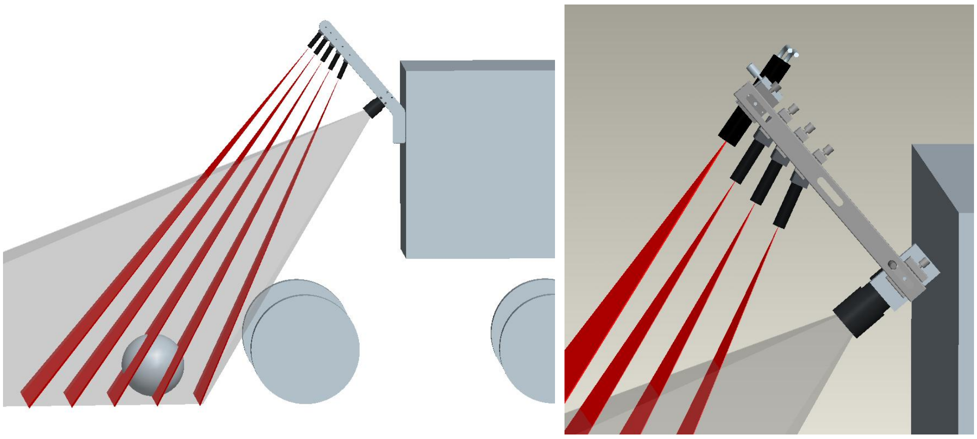
The ExoMars rover features a PanCam (**Pan**oramic **Cam**era System) located at the top of a mast, see Fig. B.8, consisting of two wide AOV cameras for multi-spectral, stereoscopic panoramic mapping along with a high-resolution camera for close-up, color pictures in its search for morphological signatures of life. Furthermore, the rover is meant to autonomously navigate for approximately 100 meters per day. Its main visual sensor to this end is the NavCam (**Nav**igation **Cam**era System), which is a stereo camera mounted on the same housing than the PanCam, featuring a smaller base distance than the latter. The NavCam focuses on long-range navigation, hence does not allow for close-range navigation and obstacle avoidance. Incidentally, a third set of cameras coined OdoCam (**Odo**metry **Cam**era System) are mounted at the front of the rover. The OdoCam yields visual odometry to be fused with wheeled odometry, as slippery conditions are indeed expected. We proposed to take advantage of the OdoCam cameras, making them part of an structured light system including laser projectors in order to create yet another camera system, called HazCam (**Haz**ard Avoidance **Cam**era System). After all, structured light is a more efficient alternative compared to dense stereo vision in close-range obstacle avoidance. This hybrid sensor ought to provide the missing information for local obstacle avoidance.

The project took place in 2008 as a subcontract by the French National Centre for Space Studies (CNES) to the Institute of Robotics and Mechatronics of the DLR. A group of three engineers was dedicated to the project. I led the programming and algorithmic part of the study (data processing and calibration). The study culminated in the experimental validation of the proposed method on an ExoMars prototype at the SEROM facility within the CNES premises in Toulouse, France.

The project suited me just fine in order to put my already developed methods to the proof. First, the required calibration methods had been already developed in the context of the DLR 3D-Modeler, refer to Sections 3.2 and 3.6. Second, the HazCam was not allowed to interfere with the separate OdoCam project, *i.e.*, the OdoCam cameras may not be narrow-band filtered to laser light; this fact also matches with the restrictions that we originally imposed on the DLR 3D-Modeler during its design phase. My robust image processing algorithms presented in Section 4.3.2 were expected to serve that purpose.

The HazCam design

The operating area of the HazCam is the immediate vicinity of the front wheels of the rover, for it not to get stuck by smaller obstacles that could pass undetected by the NavCam. Note that the OdoCams are directed to that region. Since the HazCam is not expected to be actuated and the motion of the rover is slow, a dense, coded structured light approach delivering 2.5-D depth images should be preferred w.r.t. a single light stripe profiler that only yields 2-D information in the form of a 1-D depth vector. It is, however, difficult to convey the desired energy to project a 2-D pattern in daylight conditions—especially on Mars. The generation of a laser plane is much more efficient. For this reason, we opted for the installation of a series of four laser modules that sequentially (*i.e.*, in pulsed mode) project laser stripes on the scene, see Fig. B.9. It is worth noting that, in the event that this option becomes too heavy for final deployment, a workaround with computer-controlled micromirrors can be also provided.



(a) Expected HazCam layout.

(b) Detail of the HazCam.

Figure B.9: Prior design of the HazCam. Note that, in reality, the lasers are pulsed.

The calibration of HazCam

Since the HazCam system matches the DLR 3D-Modeler except for the pose tracking system, I opted for the rapid and accurace methods presented in this thesis both, for the intrinsic calibration of the OdoCam cameras (viz. monochrome uEye UI-1540-M 1.3 MP cameras equipped with Schneider Kreuznach Cinegon 1.4/8-0902 lenses, yielding an AOV of 37° × 45°), refer to Section 3.2, and for the extrinsic calibration of every laser plane w.r.t. the camera frame S_{C}, refer to Section 3.6. Since the latter method needs for the motion of the laser planes, and furthermore for that motion to be tracked by a pose tracking system, I attached the HazCam to the end-effector of a Kuka KR 16 for the calibration process only. In this way, the intrinsic calibration of the whole HazCam system can be readily performed within an hour.

There still remains the extrinsic calibration of the whole HazCam system w.r.t. the ExoMars rover. This method corresponds with the extrinsic calibration of the OdoCam system, which has been already addressed elsewhere. Still, since our prototype did not yet concern the OdoCam cameras, I devised a rapid method for extrinsic calibration of the HazCam by measuring the planar soil in front of the rover.

The image processing algorithms of HazCam

It was my intention to implement the same methods for the segmentation and reconstruction of the laser stripe that I have presented in Sections 4.3.2 and 2.2.2, respectively. Regrettably, I encountered three problems that prevented me from directly using these methods:

1. The projected laser stripes on the images were too weak. The reason is twofold: On the one hand, because the robot had to perform at daylight (on Earth). On the other hand, because the laser projections were more distant than in the case of the DLR 3D-Modeler, e.g. 1 m distant. More

powerful lasers would have make up for this first problem. Even though it is not a problem to use powerful lasers on Mars, and powerful lasers are still energy-efficient, strong payload limitations apply to the HazCam system, which prevented us from using heavier devices.

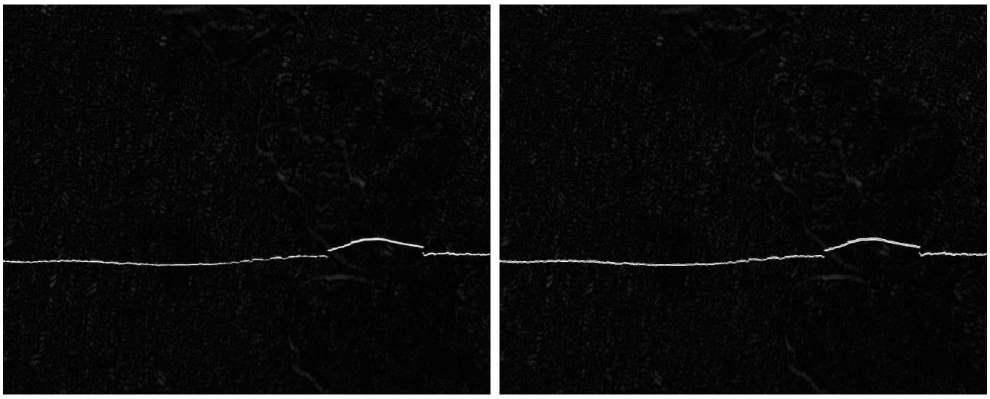
2. The OdoCam system features monochrome cameras. Since the methods presented in Section 4.3.2 partly rely on the red color of laser projections, it is expected that they perform worse when using monochrome cameras.

3. It turns out that the tolerance to erroneous measurements in the ExoMars rover mission totally opposes the assumed tolerance in the case of the DLR 3D-Modeler. Whereas in the latter case false negatives (*i.e.*, non-detected surfaces) are readily tolerated as the DLR 3D-Modeler can repeat measurements from a different vantage point, in the case of the HazCam these errors are strictly prohibited, as their likely consequence is a stranded or damaged rover. Conversely, in the case of the DLR 3D-Modeler false positives (*i.e.*, detection noise from reflections, etc.) ought not to be tolerated in order to obtain clean 3-D models, whereas in the case of the HazCam these measurements are tolerated because they do not compromise the safety of the robot, refer to Table B.1.

Table B.1: Different sensitivity tolerances by the DLR 3D-Modeler and the HazCam.

	DLR 3D-Modeler	HazCam
False negatives (non-detected obstacle)	tolerated	not allowed
False positives (detection w/o obstacle)	not allowed	tolerated

These facts called for a reconception of the detection algorithms in Section 4.3.2. Since the laser modules are being pulsed to activate single modules sequentially, the option presents itself to take 'dark,' reference images where all modules are switched off in between laser images. In this way, the direct subtraction of consecutive images yields very clear laser projections, similar to imaging the laser projection through a narrow-band, laser light filter—subject to the speed of the rover. Fig. B.10 shows a typical differential image at full speed motion of the rover. Due to its speed and the slow triggering of the camera drivers (> 100 ms), some shadow artifacts appear. The resulting images by direct image subtraction, however, already allow for robust, accurate segmentation by simply using *Stage #1* and the center-of-stripe detection stage explained in Section 4.3.2. It plays into our own hands that the rover is slow, or rather static, when performing differential laser light detection.



(a) Raw differential image. (b) Stripe detection on differential image.

Figure B.10: Laser light detection by differential images.

In the case of brighter illumination conditions (above 2000 lux), the former limitation of 100 ms on the triggering speed by the provided camera drivers may actually entail robustness issues. In that event I propose the generation of synthetic images compensating for the rover motion. Unfortunately, this mapping requires the knowledge of the structure of the scene. I propose using the a priori ground plane (as a result of the extrinsic calibration of the HazCam) as supporting structure for this procedure. In addition, in this way the mapping can be reduced to a 2-D-to-2-D (*i.e.*, camera-to-ground) homography, which is a relatively inexpensive transformation to perform to begin with. Admittedly, in the regions where the scene strongly differs from this virtual plane (e.g. rocks, holes, and slopes), the method is not going to perform optimally. Nonetheless, these regions usually correspond to rocks, where this problem is less noticeable than in the highly-frequent texture on the soil in the first place.

The resulting laser stripe projections onto the image are to be undistorted and reprojected on the basis of the intrinsic parameters of the camera, refer to Section 2.2.1. After that, the segmented laser stripe projections are reconstructed in 3-D camera coordinates by triangulation, refer to Section 2.2.2. In addition, the extrinsic transformation makes it possible to represent 3-D data on the rover's ground reference system.

The last issue is the conception of the decision making process responsible for sending the alarm signal on the basis of the surface elevation profiles obtained from the above detection and triangulation steps. This process is open to interpretation in relation to the actual obstacles faced by the rover. In our study, the alarm signal has been sent whenever elevations points were *consistently* detected above a particular threshold (e.g. above 15 cm or below −10 cm), refer to the experimental results below. Furthermore, an elevation representation w.r.t. the ground of the laser projection is continuously overlaid on the image window for representation.

Final experiments at the SEROM facility

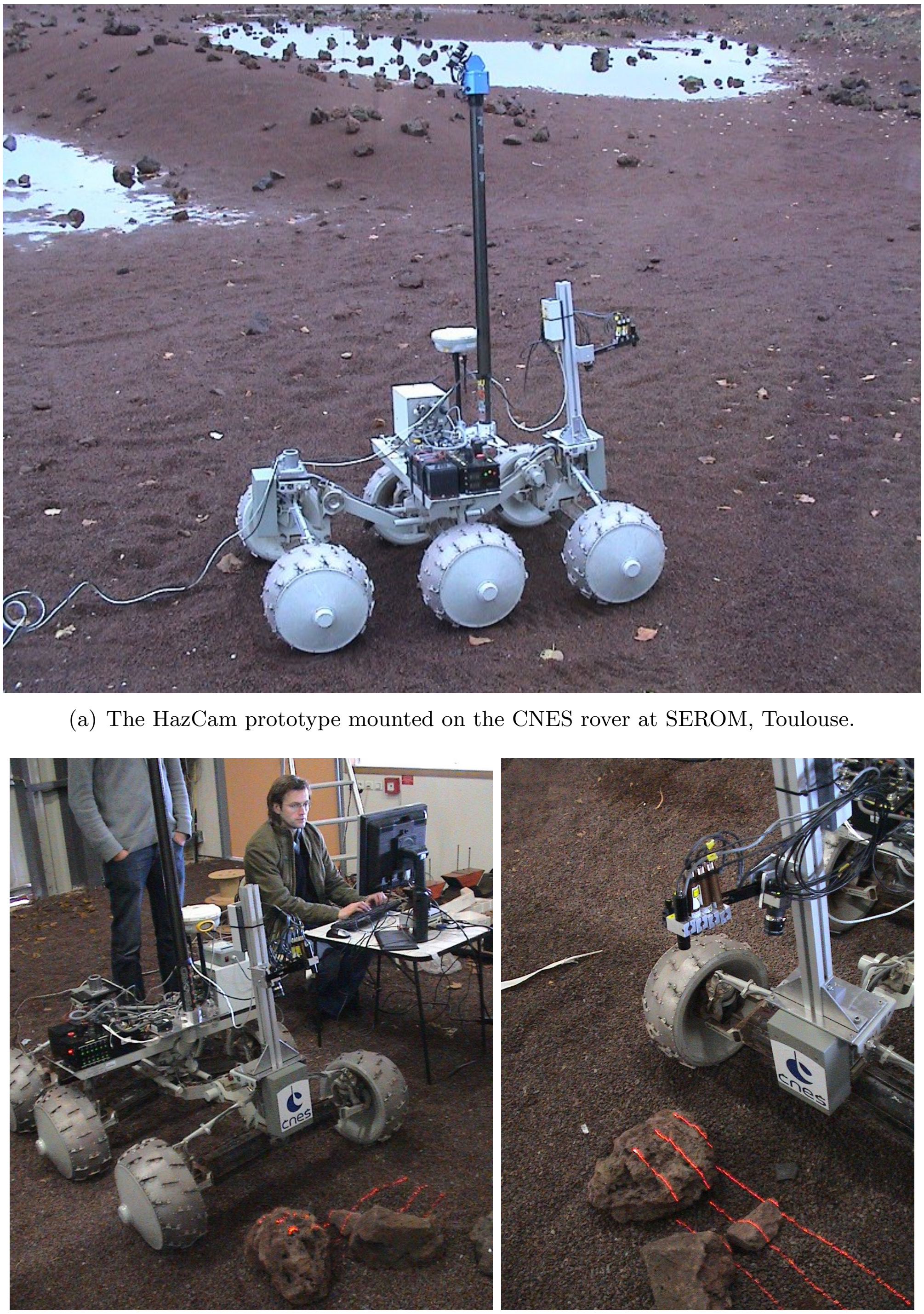
The final on-site verification experiments were conducted on Monday and Tuesday 24th and 25th of November, 2008, at the SEROM facility of CNES in Toulouse, France. The HazCam system was mounted on a prototype rover approximately 70 cm above ground and with an inclination of 20°, refer to Fig. B.11. The HazCam system was operated from an external PC connected by two long USB cables to the system. Experiments were carried out indoors (<200 lux) and outdoors (>2000 lux), both with static and cruising rover.

First, indoor experiments in a hangar were conducted, see Fig. B.11 (b,c). First of all, an online extrinsic calibration process was automatically performed (within a couple of seconds). Then, the Hazcam system delivered accurate, robust data in regular operation both, with static rover and at full speed. Differential images proved indeed extremely robust. The screenshots in Figs. B.12 and B.13 stem from this session. At the top of the images, the reconstructed height profile is overlayed and the hazard detection signal is delivered. In addition, the detected line is highlighted in red.

Last, outdoor experiments at 2000 lux were conducted. At this illuminance level two limitations are noticed:

1. The 5 mW laser light (laser class 2M) is virtually invisible to the naked eye. Differential images, however, do manage to trace the line depending on the scene illumination and the rover speed.
2. The rover motion originates stronger artifacts in differential images due to the brighter background illumination. Critically, these artifacts are aggravated by the fact that the camera drivers do not allow consecutive external triggering faster than approximately 100 ms. In the case of robot motion, the background artifacts are consequently strong. It is future work to fix this limitation to provide seamless robust operation, virtually irrespective of the external illumination level.

In Fig. B.14 successful height profiles are shown when the robot was static. Even though the illumination level was high (2000 lux), the HazCam worked fairly, *i.e.*, within the original precision requirements. In the case of robot motion, however, the robustness in laser detection is strongly compromised mainly due to the abovementioned second reason. This limitation concerns the chosen propietary hardware and software and ought to be lifted for robust outdoors operation.

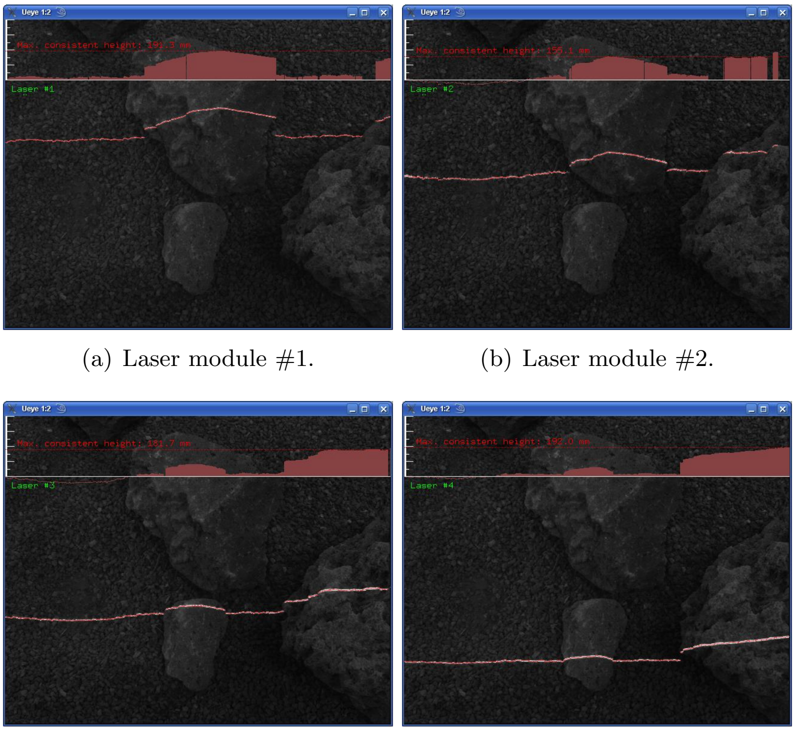


(a) The HazCam prototype mounted on the CNES rover at SEROM, Toulouse.

(b) HazCam control by an external computer.

(c) Detail of the HazCam.

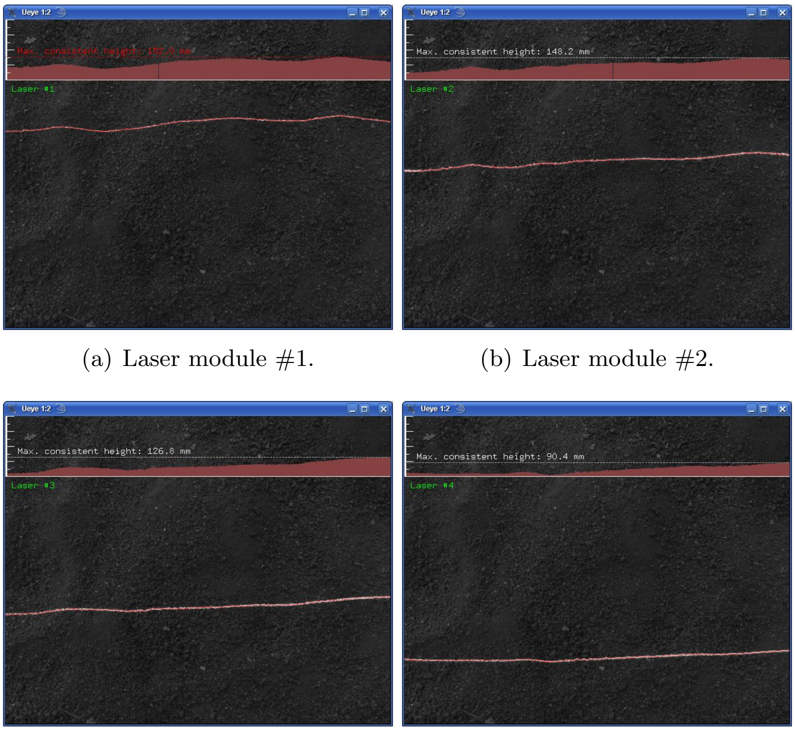
Figure B.11: The HazCam prototype at the CNES rover. The lasers were pulsed in operation. Visit the videos of indoors and outdoors experiments here: `http://goo.gl/jgdFn`, `http://goo.gl/Ik1dt`, `http://goo.gl/mfcVy`, and `http://goo.gl/yG3Ka`.


Ueye 1:2
Max. consistent height: 191.3 mm
Laser #1
Ueye 1:2
Max. consistent height: 155.1 mm
Laser #2
Ueye 1:2
Max. consistent height: 181.7 mm
Laser #3
Ueye 1:2
Max. consistent height: 192.0 mm
Laser #4

(a) Laser module #1. (b) Laser module #2.

(c) Laser module #3. (d) Laser module #4.

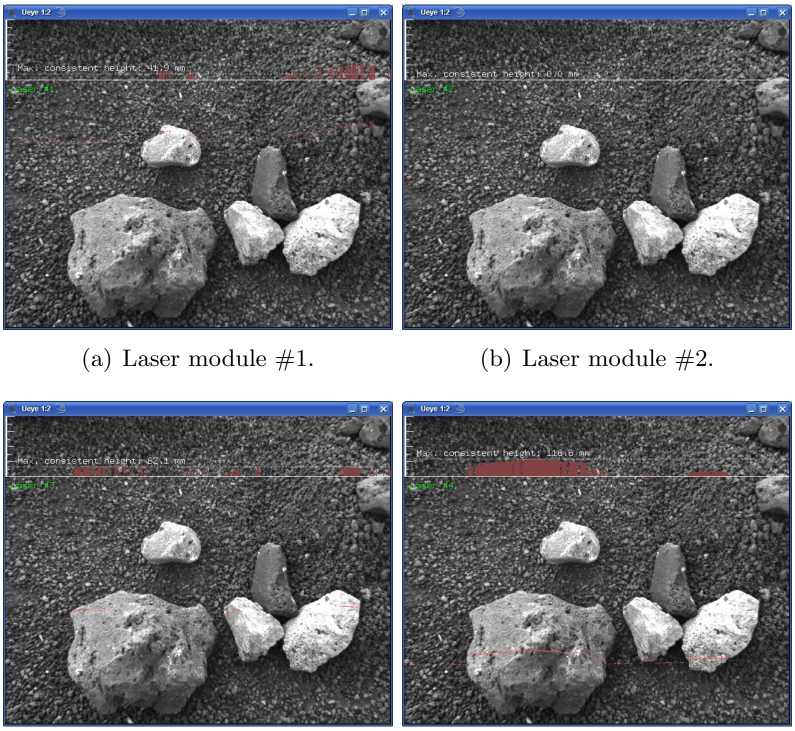
Figure B.12: Sequential HazCam readings in the face of obstacles (indoors).


Ueye 1:2
Max. consistent height: 152.0 mm
Laser #1
Ueye 1:2
Max. consistent height: 148.2 mm
Laser #2
Ueye 1:2
Max. consistent height: 126.8 mm
Laser #3
Ueye 1:2
Max. consistent height: 90.4 mm
Laser #4

(a) Laser module #1. (b) Laser module #2.

(c) Laser module #3. (d) Laser module #4.

Figure B.13: Sequential HazCam readings in the face of a slope (indoors).



```
Max. consistent height: 41.9 mm
Max. consistent height: 0.0 mm
Max. consistent height: 62.1 mm
Max. consistent height: 116.6 mm
```

(a) Laser module #1.

(b) Laser module #2.

(c) Laser module #3.

(d) Laser module #4.

Figure B.14: Sequential HazCam readings in the face of obstacles (outdoors).

B.2.3 Motion Estimation for Free-Flying Satellite Rendezvous

Introduction

The DEOS project (**De**utsche **O**rbitale **S**ervicing Mission) started in 2007 out of the programmatic reorientation of the former TECSAS project by both Germany and Russia. The project focuses on the guidance, navigation and non-destructive capturing of cooperative or non-cooperative tumbling client satellites, to perform maneuvers with the coupled system and, potentially, to de-orbit it in a controlled manner. The project is on behalf of the Space Agency of the German Aerospace Center (DLR), funded by the Federal Ministry of Economy and Technology within the framework of Germany's National Space Program.

In the context of the so-called preliminary design definition phase (Phase B), our institute is assigned with the technical requirements specifications, evaluation, and demonstration of the algorithms required to capture and control the coupled satellite system. A key aspect of the rendezvous and docking phase is the accurate relative motion estimation between unconnected satellites. Since the mockup capturing satellite is equipped with a stereo camera, cf. Fig. B.16, I opted to utilize the relative pose tracking algorithms presented in Chapter 5.

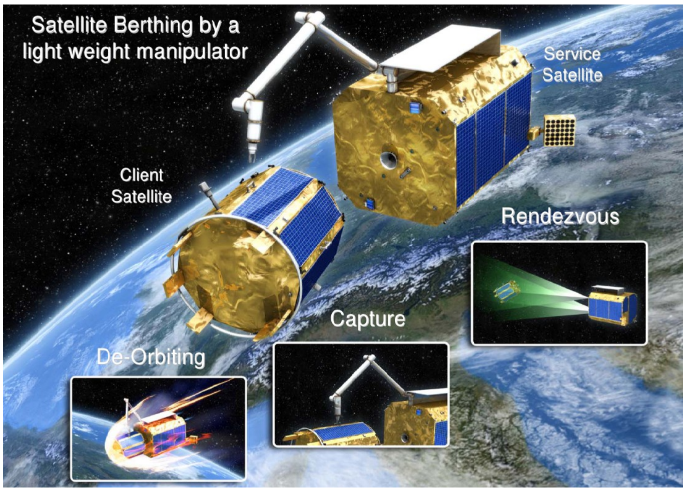


Figure B.15: The DEOS project focuses on the rendezvouz, capture, and controlled de-orbiting of a tumbling target satellite. *Source: DLR.*

The experimental data stems from the European Proximity Operations Simulator (EPOS) facility in Oberpfaffenhofen, Germany, see Fig. B.16. The present setup consists of a fixed-base robotic manipulator holding a stereo camera and a moving robotic manipulator holding a client satellite mockup (size 2.3×1.8 m) with a noozle and thermal foil. The latter satellite is in linear motion as it is mounted on an actuated linear slide. The stereo camera has a sensor chip with 1024×768 pixels and 8 bit resolution. Its full AOV is 124°×109°; the angular resolution of the cameras is slightly worse than the planned CAM-CR. The stereo baseline is 0.12 m. Images are acquired at 1 Hz.

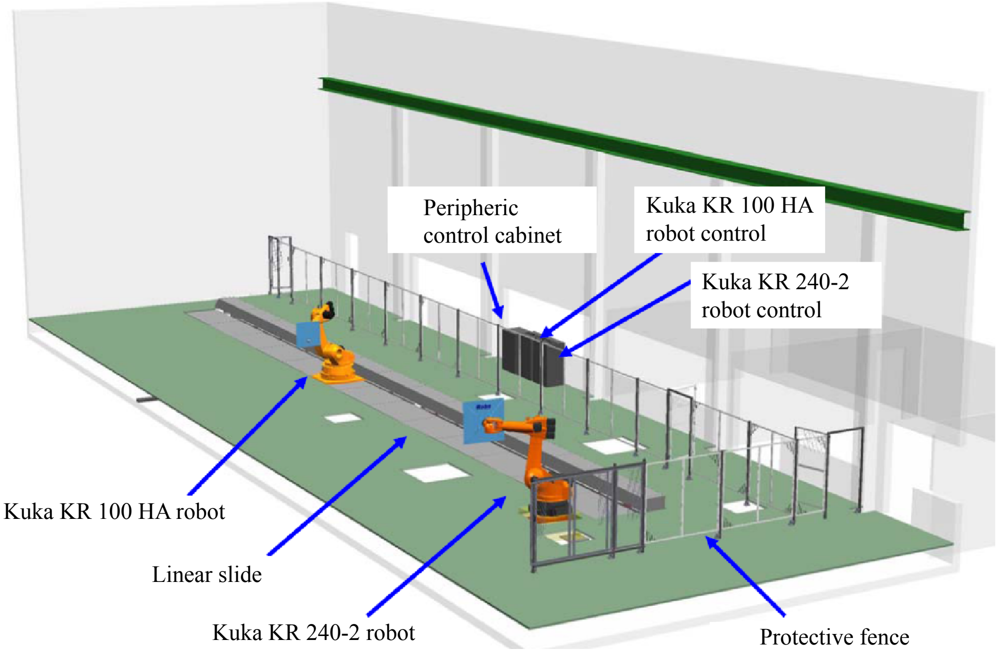

Peripheric control cabinet
Kuka KR 100 HA robot control
Kuka KR 240-2 robot control
Kuka KR 100 HA robot
Linear slide
Kuka KR 240-2 robot
Protective fence

Figure B.16: The EPOS facility; for the present experiment, the moving-base robot held a satellite mockup, the fixed-base robot held a pair of stereo cameras. *Source: DLR.*

Pose tracking method

A prerequisite for accurate pose tracking of the client satellite by the stereo camera of the service satellite is the intrinsic and extrinsic calibration of its stereo camera. This task has been performed following the methods presented in Sections 3.2 and 3.3 and using the software package DLR CalDe and DLR CalLab that will be presented in Section B.3.

The motion of the camera system w.r.t. the target satellite is estimated based on tracked features on the target satellite's surface. Even though most salient features correspond to spurious reflections (cf. Fig. B.17), *i.e.*, they cannot be reliably tracked for longer periods of time, we perform feature tracking with perishable features, dumping features every 4 frames, *i.e.*, every 4 seconds or less. In this way, long-term accuracy is compromised as neither local nor global loop closures are possible, see Section 5.2.4-II., thus motion estimation corresponds with dead reckoning. Nonetheless, the accuracy of this method for visual odometry proved to be sufficient for medium-range satellite pose tracking.

The method presented in Section 5.4.2 selects so-called "good features to track" with distinctive appearance, refer to (Shi and Tomasi, 1994). The upside of using this type of features is their general applicability, as a prior modeling of the (rigid) target satellite is not required anymore—this is useful in the (predominant) case of meeting with an older satellite featuring decommissioned parts. On the other hand, in a satellite-engaging scenario these features frequently correspond to reflections, shadows, or occlusions, which usually are not

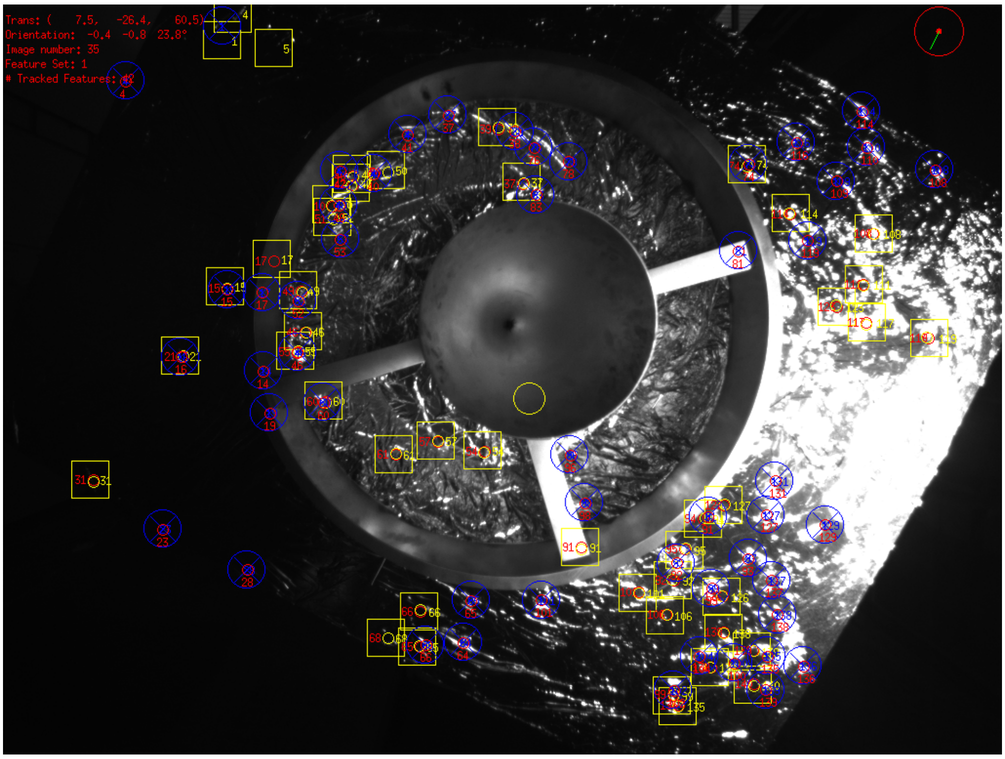


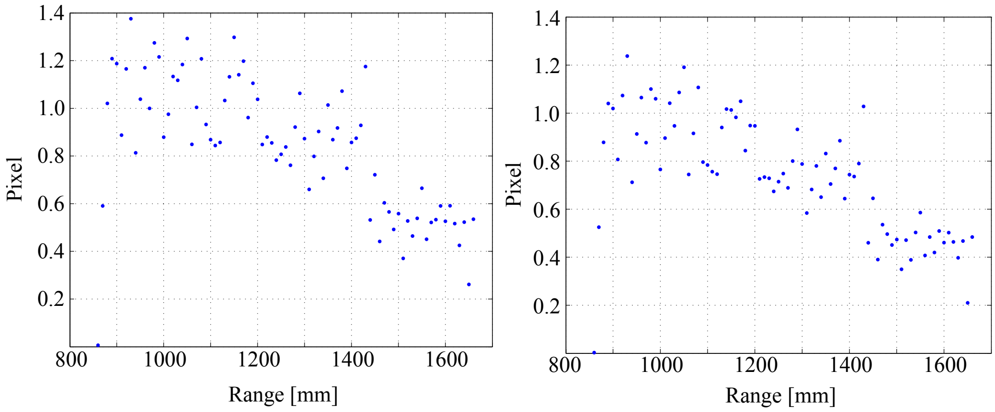
Figure B.17: Sample image from the sequence with tracked image features (yellow: recovered feature points with search regions; blue: new feature points initialized for subsequent tracking). Visit `http://goo.gl/Jhb22` to see the whole sequence in motion.

consistent with the rigid satellite assumption in the long run. To overcome this problem the algorithm regularly reinitializes features, *i.e.*, performs dead reckoning motion estimation. The rigid motion estimator is at its core a robust M-estimator. I used a blunt implementation of the pose tracking method in Section 5.4 and (Strobl *et al.*, 2009a). In a nutshell: Monocular feature tracking and relative pose estimation are performed in parallel with stereo reconstruction of feature depth, see Section 5.4.1. Feature depth in turn makes it possible to deliver scaled, metric pose estimations as well as to sensibly discard inconsistent data.

Experimental results

The tested trajectory is in medium-range, traversing from 0.85 m to 1.65 m *away* from the target satellite, translating it at realistic speed of 1 cm/s away from the service satellite. At the same time, the target satellite is spinning at 4 °/s in the direction of translation.

Fig. B.18 shows the 2-D tracking consistency w.r.t. the virtual projections of the estimated rigid structure as estimated by stereo vision in average, at the optimal pose estimated by the algorithm. The figures evaluate both, the triangulation accuracy and the precision of feature tracking. Fig. B.18 (a) shows the RMS error in relation to the range to the target satellite. Fig. B.18 (b) in turn shows the average absolute error, also in relation to the range to the target satellite. Estimations are consistent to around 1 pixel. The effects of motion in residual reprojection errors naturally decrease at larger distances, leading to an apparently higher consistency with feature detection.

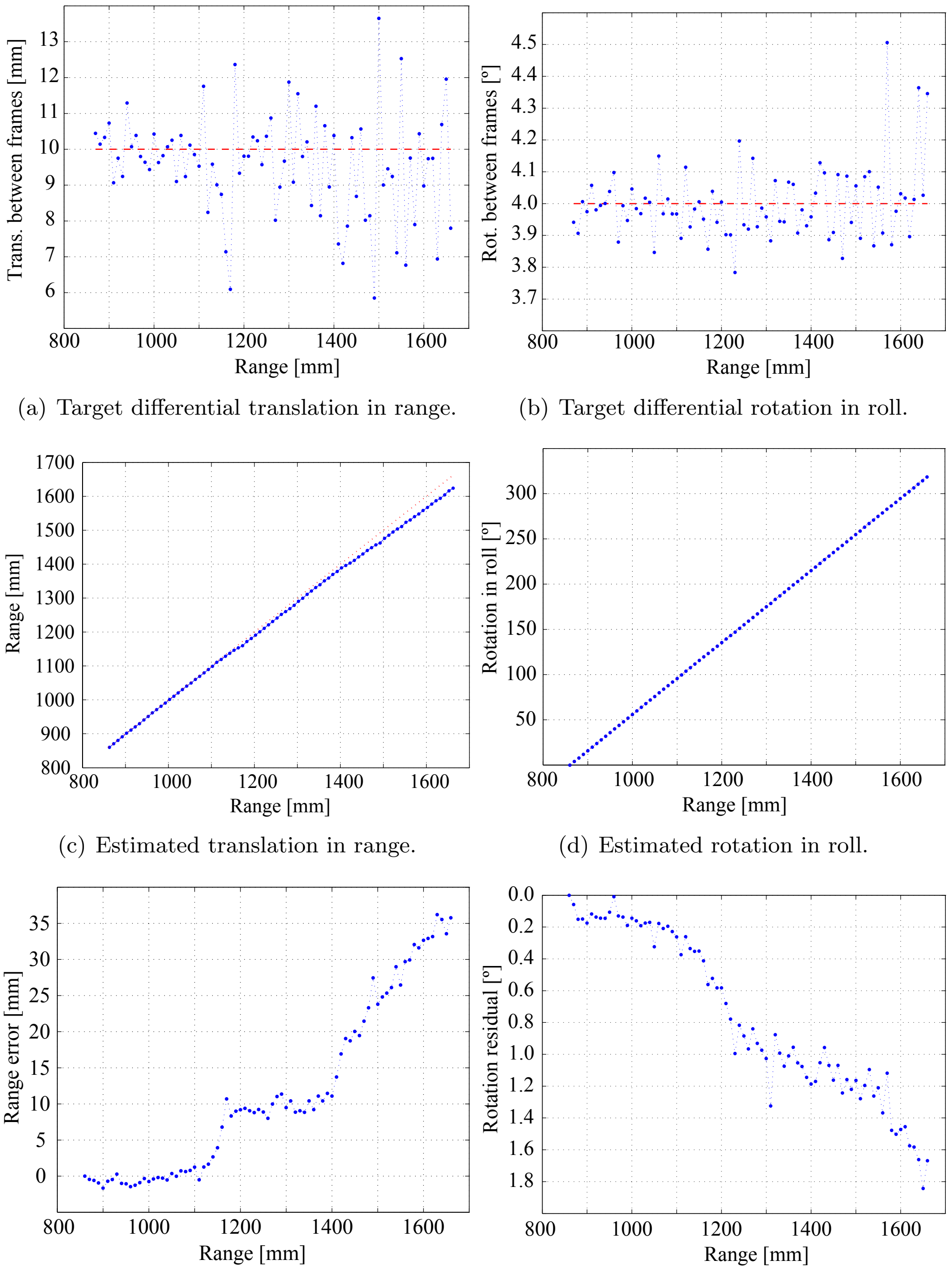


(a) RMS error in range.

(b) Mean absolute error in range.

Figure B.18: Tracking accuracy and consistency in the image memory frame S_{M}.

Next I am assessing the accuracy of the motion estimation. In order to leave potential extrinsic calibration errors aside, I opt to estimate absolute distances and absolute angles, as they are irrespective of the orientation of the reference frames. The actual relative motion between satellites amounts to 1 cm between images and 4°. Fig. B.19 (a,b) show the relative translation depth and roll angle between consecutive frames, in relation to the absolute distance to the satellite. The red horizontal lines indicate the ground truth. Relative roll angle errors stay below 0.1° and relative translation errors within 2 mm for range estimation up to 1.5 m distance. In Fig. B.19 (c,d) the accumulated translational and rotational motions are shown. Whereas orientation estimation in roll angle remains good conditioned, range estimation accuracy abruptly decreases beyond 140 cm, cf. Fig. B.19 (e.f), which is natural consequence of the short basis distance and resolution of the used stereo camera.



(a) Target differential translation in range.

(b) Target differential rotation in roll.

(c) Estimated translation in range.

(d) Estimated rotation in roll.

(e) Accumulated residual error in range.

(f) Accumulated residual error in roll.

Figure B.19: Range and pose estimation accuracy in relative and absolute terms. Note that motion estimation is performed by dead reckoning.

In conclusion, sufficient accuracy in feature tracking and motion estimation w.r.t. the original technical requirements specifications in Phase B is provided at distances up to 1.5 m of the client satellite. It is important, however, to note that these results depend upon the number and distribution of trackable feature points, *i.e.*, on sufficient client satellite texture to be visible in the images.

B.2.4 Rapid Calibration of 18 Cameras on the DLR RoboMobil

Introduction

As distinguished from present efforts of car manufacturers, the RoboMobil project (RoMo) at the Institute of Robotics and Mechatronics of the DLR focuses on autonomous driving from the original conception of the car in the first place, *i.e.*, we aim at an holistic new approach to electricity-enabled autonomous driving from the car rims to its top. And quite literally so: Inspired by the lunar roving vehicles of the Apollo spaceflight missions, mechatronic electric drives are embedded into the rims, as they make for a swifter drive-by-wire control of the car dynamics by the central computer (similar to robotics dynamical simulations in our institute); in addition, they allow for extended steering ranges from $-25°$ up to $95°$. In turn, RoMo's exceptional agility imposes further requirements on its perception system; 18 cameras and further sensors are attached to its roof, as state-of-the-art computers allow for parallel interfacing and computing of loads of independent sensory information. With RoMo we ultimately aim at demonstrating how closely related electric driving and robotics in the near future will be (Brembeck *et al.*, 2011), see Fig. B.20.

Perception is key to increase autonomy in traditional robots as well as in robotic cars, and a consistent representation of the 3-D scene ought to play a central role for this purpose. A number of sensors can be tapped to this end, but a major source are video cameras. They are especially convenient due to

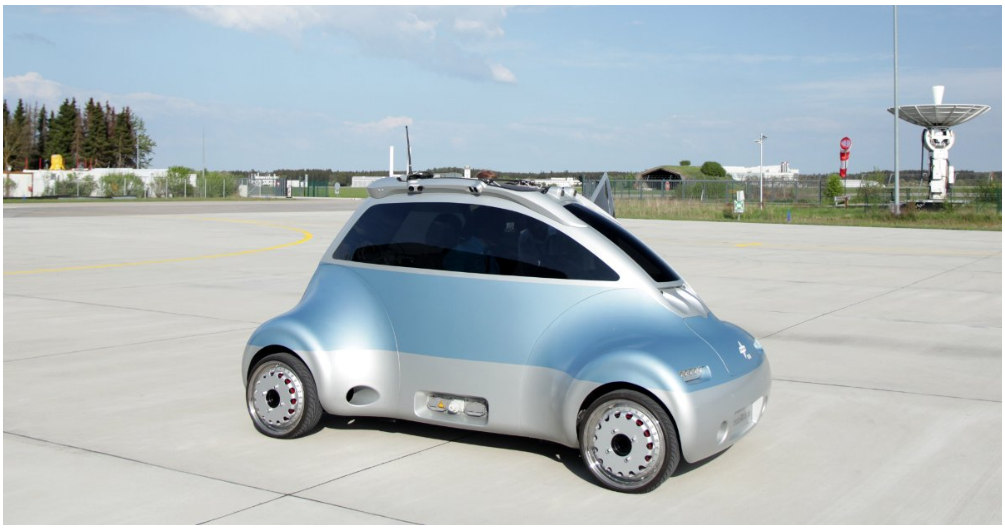


Figure B.20: The DLR RoboMobil (RoMo).

their passivity w.r.t. the scene, *i.e.*, due to their extended operating range, the lack of potential crosstalk with other sensors, and because most road signs and markings are designed for visual recognition in the first place. Their uses range from simple visual feedback for telepresence by humans (mostly in "shared autonomy" mode), to optical flow computation for rapid collision avoidance, lane following, detection of road signs, and 3-D reconstruction by stereo vision for autonomous maneuvering (e.g. parking), refer to Fig. B.21. Their only limitation is the high computational requirements owing to the selected perception method. At RoMo we aim at showing how much computational power is possible to embed into an electric car, taking software and hardware optimization measures into account, e.g. by leveraging GPGPU and FPGA hardware options.

A hard requirement for all of these methods is to calibrate the cameras, both intrinsically and w.r.t. the car's chassis. Due to my extended experience calibrating this type of sensors, see Chapter 3, I assumed the task of intrinsically and extrinsically calibrating its 18 cameras. In this way, a registered representation of depth data from stereo vision as in Fig. B.21 will be rendered possible.

Calibration procedures

In this section the required calibration procedures are detailed. These are largely based on the general methods presented in Chapter 3. Three types of calibration procedures are used:

1. **Intrinsic calibration of monocular cameras and stereo cameras**. The methods and requirements fully correspond with the monocular and stereo methods presented in Section 3.2. The use of the novel methods

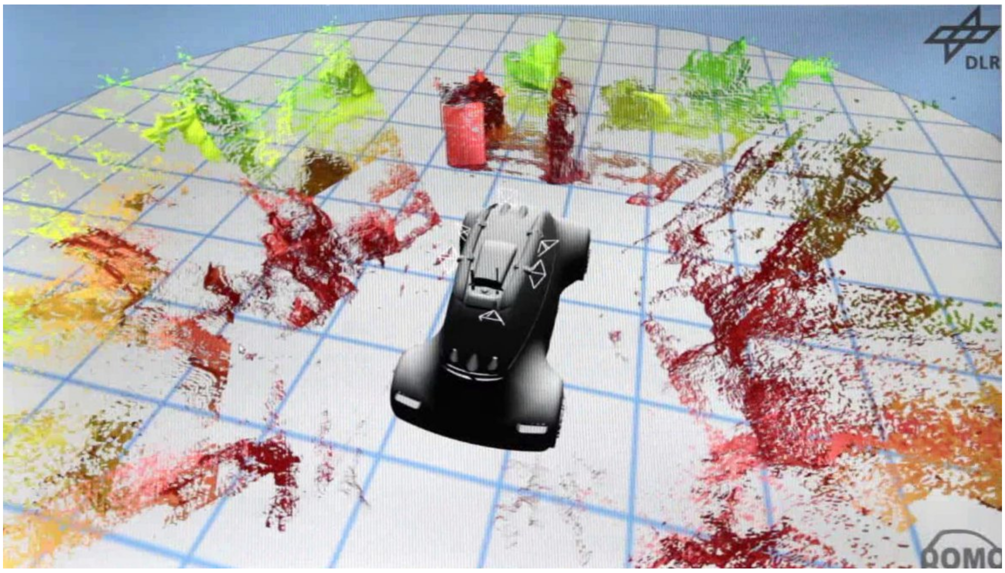


Figure B.21: Registered depth data from stereo vision on 6 concurrent stereo cameras at the RoMo. In this view, all range data except for ground-level range points are represented; their color codes their minimum distance to the car. These data are used e.g. for path planning.

presented in Section 3.4 is discouraged unless the calibration target is perfectly rigid. Since it is difficult for the car to be relocated around the calibration target, I opt for the inverse option, *i.e.*, shifting a calibration target around the car. In detail, I choose a thin metallic plate for the user to be able to easily tilt it by hand in front of the 18 cameras. In this context, sub-millimetric plate deflections are possible, thus a static geometry cannot really be expected, and consequently it is not a good idea to optimize its structure during camera calibration. This observation is, of course, subject to the calibration target used.

2. **Relative, extrinsic calibration between intrinsically calibrated monocular and stereo cameras**. The estimation of the relative rigid body transformation between already intrinsically calibrated cameras is simplified if it is possible for both cameras to aim at the same calibration target. The fields of view (FOV) of some cameras, however, do not overlap (e.g. cameras #1 and #11 in Fig. B.22). Three approaches are possible to overcome this problem: First, it is possible to use two calibration targets that are rigidly attached to each other (in an unknown configuration), and use the hand-eye calibration method presented in Section 3.3 to estimate their relative location. Alternatively, it is possible to build up sparse maps of features with both cameras, rotating the car until their overlapping can be detected and both maps can be registered, refer to (Carrera *et al.*, 2011). A third option is to repeat the absolute extrinsic calibration stage presented below, for every subset of relatively-calibrated cameras. I opt for the last option as it allows for high accuracy without the need for extra hardware, thus it is subject to less potential mistakes by the user.

   Even though it is perfectly possible to estimate relative transformations between cameras in the context of a standard stereo camera calibration, that is an incorrect procedure if the optimal camera parameters have been already estimated during a previous intrinsic calibration stage. As explained in Section 3.2, a different parametrization of the camera goes along with a relocation of its reference frame S_{C}, and consequently the estimated transformation between cameras ${}_{\mathrm{C}_i}\widehat{\boldsymbol{T}}^{\mathrm{C}_j}$ is not exactly valid in

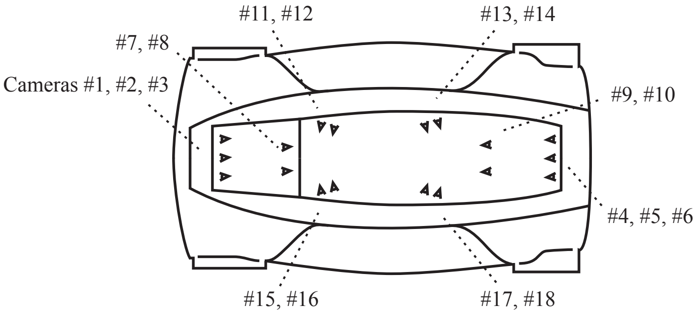


Figure B.22: Layout of the cameras mounted on the DLR RoboMobil. Only cameras #1, #2, #3, #4, #5, and #6 are color cameras.

this context. For this reason, the intrinsic parameters of the camera ought to be fixed during an optimization process that is equivalent to regular stereo camera calibration, where the only parameters to be optimized are either the 6 DoF of the relative transformation between cameras, or the 12 DoF of the relative transformations between the cameras and the common calibration target located in S_0. The camera calibration toolbox DLR CalDe and DLR CalLab allows for the user to load previous optimal solutions to the camera parameters so that only the abovementioned transformations are estimated, with highest accuracy. Depending on the calibration data used (either a file of one stereo camera or two files of two monocular cameras), the desired transformation ${}_{\mathrm{C}_i}\widehat{\boldsymbol{T}}^{\mathrm{C}_j}$ will either be directly obtained using a stereo image (in the case of a stereo calibration), or indirectly using two monocular images in the case of monocular calibrations, by making use of the following equation:

$$ {}_{\mathrm{C}_i}\widehat{\boldsymbol{T}}^{\mathrm{C}_j} = {}_{0}\widehat{\boldsymbol{T}}^{\mathrm{C}_i\,-1}\,{}_{0}\widehat{\boldsymbol{T}}^{\mathrm{C}_j} \quad . \tag{B.1} $$

3. **Absolute, extrinsic calibration of relatively calibrated cameras w.r.t. the car's chassis**. In theory, only one extrinsic calibration of this type is necessary if all 18 cameras have been perfectly registered to each other using the above methods. Since we opted for decoupling subsets of relatively calibrated cameras, we actually require several absolute extrinsic calibration procedures to fully register the cameras of the RoMo. The camera subsets are four, divided in front, right-hand side, left-hand side, and rear camera subsets, cf. Fig. B.23. Thus four absolute extrinsic calibration procedures are required. The overall procedure is as follows:

   (a) Three salient features are located in the FOV of a stereo camera included in each of the four camera subsets (or, if necessary, of a virtual stereo camera composed of relatively calibrated monocular cameras), *i.e.*, a total of 12 features on the scene are required.

   (b) The 3-D coordinates of the features have to be measured w.r.t. the absolute reference frame of the RoMo S_{RoMo}. In order to deskill this procedure, the features should be located on common planes, and two of the features should lie in the direction of one of the main axis of S_{RoMo}, cf. Fig. B.23.

   (c) A single stereo image with sufficient disparity is to be taken of every set of three features. After that, the 3-D location of the features w.r.t. the main camera of the camera subset can be easily estimated by feature-based stereo vision as in Section 5.4.1. Finally, the absolute transformation of every subset of cameras w.r.t. S_{RoMo} can be obtained by 3-D trigonometric calculations and concatenation of rigid body transformations.

Overall procedure

The overall sequence of calibration procedures is as follows:

1. **Intrinsic calibration of monocular cameras and stereo cameras**.
   - 6× stereo camera calibrations for cameras $7 \cup 8$, $9 \cup 10$, $11 \cup 12$, $13 \cup 14$, $15 \cup 16$, and $17 \cup 18$.
   - 6× monocular camera calibrations for cameras 1, 2, 3, 4, 5, and 6.
2. **Relative, extrinsic calibration between intrinsically calibrated monocular and stereo cameras**.
   - 4× between monocular cameras $1 \cup 2$, $2 \cup 3$, $4 \cup 5$, and $5 \cup 6$.
   - 2× between monocular and stereo cameras $2 \cup (7 \cup 8)$ and $5 \cup (9 \cup 10)$.
   - 4× between stereo cameras $(11 \cup 12) \cup (13 \cup 14)$ and $(15 \cup 16) \cup (17 \cup 18)$.

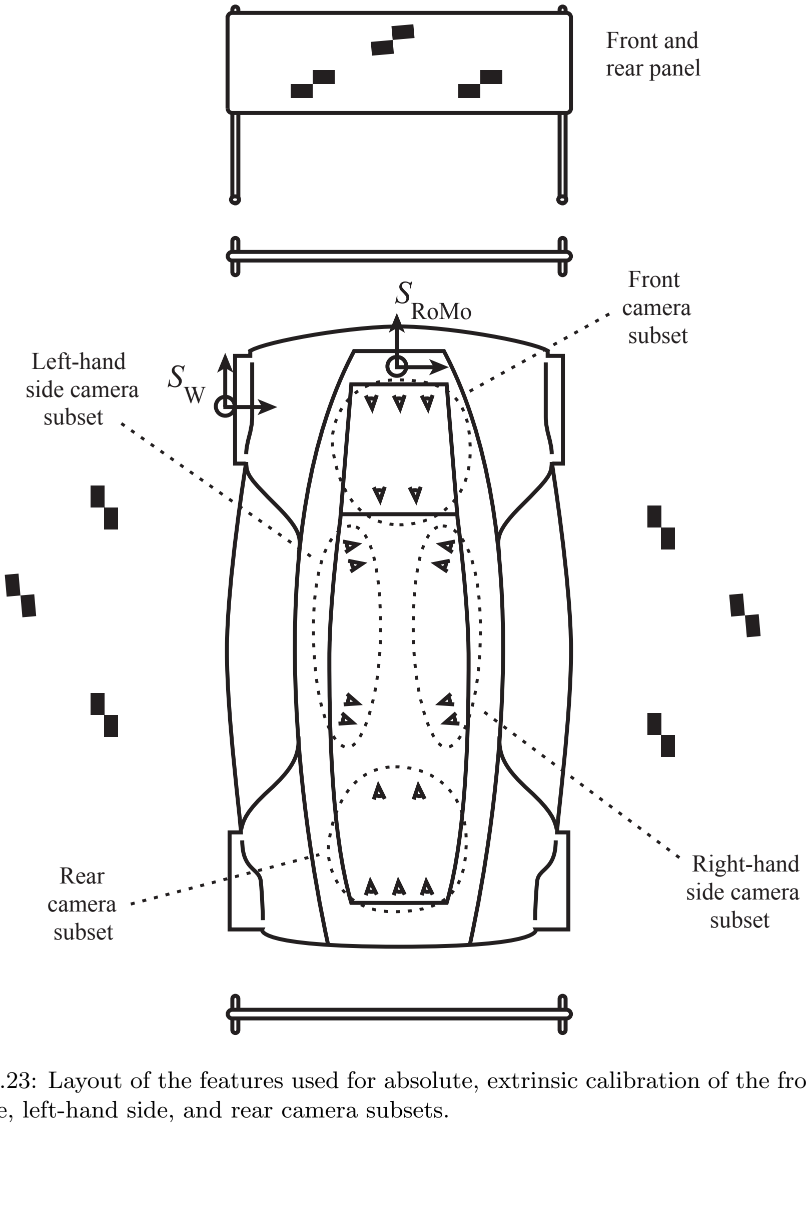


Figure B.23: Layout of the features used for absolute, extrinsic calibration of the front, right-hand side, left-hand side, and rear camera subsets.

3. **Absolute, extrinsic calibration of relatively calibrated cameras w.r.t. the car's chassis**.
    - 4× between 4 camera subsets and 3 known features each, using the stereo cameras $7 \cup 8$ and $9 \cup 10$ as well as the *virtual* stereo cameras $11 \cup 13$ and $16 \cup 18$.

As soon as the calibration images are gathered, the whole process including pattern detection and parameters estimation proceeds automatically. Due to the recent reimplementation of DLR CalDe and DLR CalLab into C++, massive parallelization of image processing is possible. It is worth noting that, due to the flexibility of DLR CalLab, different lens distortion models are supported even between the cameras forming a stereo camera. In the end, a single calibration file listing the intrinsic parameters of all 18 cameras and their relative, as well as their absolute, rigid body transformations is automatically generated.

B.2.5 Retrocalibration of a Pico projector

Introduction

Augmented reality concerns the natural extension of real entities with synthetic, virtual data. One example is the overlay of virtual objects on actual images in a natural way, *i.e.*, taking the actual scene geometry and the vantage point of the camera into consideration. Another example is the active projection of signs or text unto the actual scene. This last example is recurrent in modern medicine e.g. for preoperative planning prior to performing minimally invasive surgery (Konietschke, 2008), see Figs. B.24 and B.25. In 2008 we at the Institute of Robotics and Mechatronics observed the technological development of miniaturized video projectors called pico projectors, and realized that these devices could easily outdo the traditionally deployed laser autopointers that were bulkier and limited to one sole type of pattern projection.

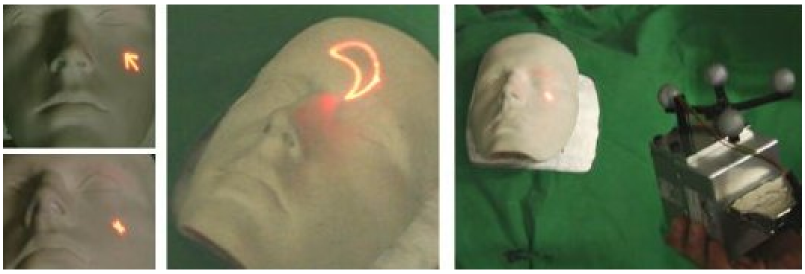


Figure B.24: Using an autopointer, different symbols can be projected onto the patient.

Traditional autopointers feature a limited field of view, therefore have to be shifted to encompass the whole scene. An unknown motion of the autopointer, however, renders it useless as the correct directions of projection cannot be inferred anymore. It is by tracking the motion of the autopointer that the absolute pose of the autopointer in the world reference frame S_0 can be eventually estimated, and hereby the projection directions updated to the actual structure

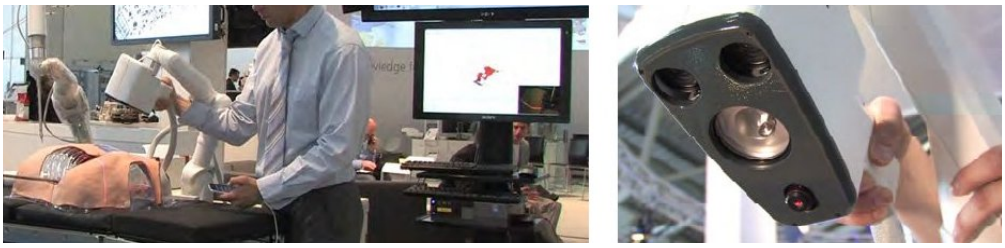


Figure B.25: The VR-Map is a device focusing on augmented reality for robot-assisted surgery including stereo vision, an LSP, and an autopointer, refer to (Schwier *et al.*, 2010).

in front of the autopointer. Regrettably, most tracking systems perform w.r.t. their own reference frame, e.g. the TCP reference frame S_{T}, and therefore it is necessary to extrinsically calibrate the autopointer w.r.t. that external reference system in a similar way as explained in Section 3.3. The inner geometry of the autopointer is usually known ex factory. That is not the case when using pico projectors. They are not really conceived for this applicacion but for visualization only, hence neither their angular amplification is known nor their distortion effects w.r.t. pure perspective projection are totally compensated for. In this work we aimed at intrinsically and extrinsically calibrating a pico projector aiming at advanced augmented reality by projecting complex textures with highly accurate registration w.r.t. the actual scene structure.

We chose the SHOWWX+ laser pico projector of MicroVision, which is size 118 × 60 × 14 mm and projects color, high-resolution images. In order to allow for hand-held motion of the projector, the used external tracking system was an infrared optical tracking system called ARTtrack2. The motion readings of the tracking system together with the intrinsic and extrinsic calibration results of the projector ought to compensate for the random motion of the projector so that projections stay static on the canvas, virtually irrespective of the projector's motion.

Two types of calibrations are required:

1. Extrinsic calibration of the beamer w.r.t. the TCP of the tracking system.
2. Intrinsic calibration of the projector in order to accurately infer 2-D projections out from desired 3-D directions.

Due to my experience in the field of camera calibration, it immediately occurred to me that an active projector ought to be the counterpart of a passive camera, as light rays are emitted from a small region within the optics (similar to the focal point of cameras, see Section 2.2.1), and light rays are emitted nearly constantly distributed in polar coordinates. Furthermore, as the projector contains optics, it is possible that the remaining distortion can be in part compensated by the models presented in Section 2.2.1. Hence I tried to apply the intrinsic calibration method presented in Section 3.2, regarding the projector as a sort of inverse camera—emitting radiation instead of receiving it. Incidentally, if such a calibration is possible, it seems natural to perform extrinsic calibration out of camera absolute extrinsics and readings of the external pose tracking following the method presented in Section 3.3.

The standard camera calibration method in Section 3.2 solely requires image projections corresponding to known planar features on the scene. Nearly without exception, camera calibration is being performed using a common calibration pattern for all images. It is difficult to imagine how to obtain image-to-feature correspondences of a common pattern using a projector. It is critical to realize, however, that using a common calibration pattern is *not* a fundamental requirement for camera calibration as presented in Section 3.2, but it is only for convenience that users do so. By doing so, the calibration pattern only has to be measured once. It is perfectly possible to use a different set of corners at every camera station—as long as the set is flat (for rapid initialization using homographies) and valid correspondences with projector coordinates exist. I propose to project a calibration pattern unto a flat canvas using the projector, from different vantage points, and to measure their projections on the canvas externally, for every projection. This is equivalent to camera calibration when using a different calibration pattern for every projection. In so doing, the correspondence problem can be naturally solved: coordinates in S_0 are identified and measured by an external camera, whereas projections on the projector (the virtual camera) are controlled by the user in the first place.

Related work

Recently in (Gavaghan *et al.*, 2011) the authors intended the same device with a similar approach. Their approach is, however, inadequate in three respects: First, they do not consider the distortion of images due to the use of optics and microelectromechanically-actuated mirrors, which is very strong in such devices. Second, they extrinsically calibrate the projector by measuring the pose of the calibration plane in the world frame S_0 so that only the projector-to-TCP transformation is required, which can be directly estimated from the absolute extrinsics of one sole station included in camera calibration results. My experiments show that such an approach is subject to errors as the plane's pose is estimated from tracking using reflective markers lying on the plane, and the millimetric transformation between the markers and the actual plane has to be ultimately estimated by hand. Third, they use a hand-guided probe to measure the corner projections, for every single image. This task is prone to errors and undeniably dull and inconvenient. For the above reasons, experiments show huge errors of up to ±15 pixels.

On the other hand, the earlier work in (Kimura *et al.*, 2007) fixes their last limitation, using homographies between the projected pattern and images of a calibrated camera unto an unknown plane, instead of measuring projected corners by hand. Still, this implementation falls short because the actual pose of the projector cannot be estimated, and therefore an extrinsic calibration of the projector w.r.t. the tracking system is missing. In addition, the distortion of images by the projector is left unattended. Overall, the approach only yields the apparent scaling of the projector.

The approach presented in the next section fuses both above approaches. In addition, it copes with image distortion and estimates the hand-eye transformation in the context of maximum likelihood estimation using the method presented in Section 3.3.

Proposed method

Fig. B.26 shows the proposed setup for calibration of the laser pico projector.

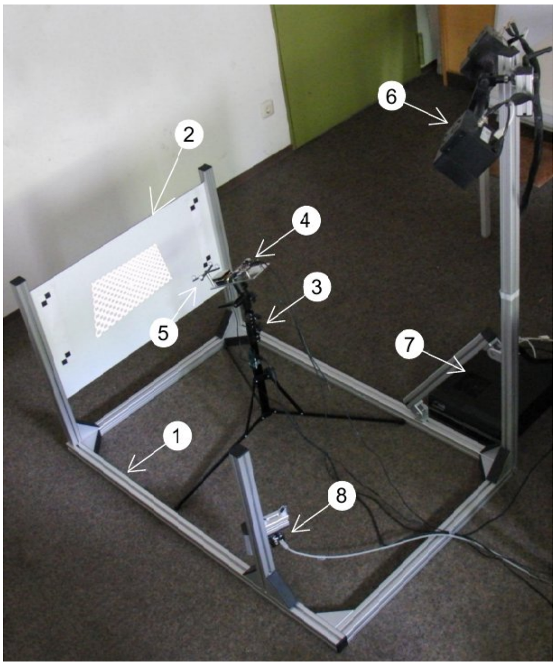


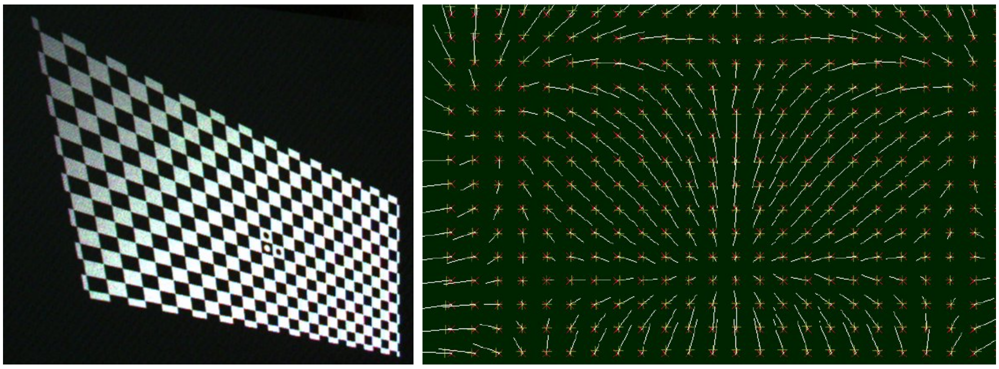
Figure B.26: Experimental setup for intrinsic and extrinsic calibration of the pico projector: aluminum frame (1) where a white canvas (2) is mounted on, including four known features used to estimate the pose of the external camera (8) w.r.t. the canvas by homographies. The external camera is in turn used to measure the projections generated by the pico projector (4) mounted on a tripod (3), which is required for intrinsic and extrinsic calibration of the pico projector (4). The pico projector (4) is tracked by an optical tracking system ARTtrack2 (6) using four IR-reflecting markers (5) mounted on the pico projector (4). The tracking system (6) features its own controller unit (7).

The calibration procedure is as follows:

1. Intrinsically calibrate the camera (8) using the method in Section 3.2.
2. Extimate the camera-to-canvas homography using the method presented in Appendix A and four features on the canvas measured w.r.t. an aleatory reference frame S_{canvas}. This homography, together with the intrinsically calibrated camera, makes highly-accurate, direct metric measurements on the canvas possible.
3. Deploy the projector (4) to project a calibration pattern unto the canvas (2) as performed during intrinsic and extrinsic camera calibration in Sections 3.2 and 3.3, *i.e.*, tilted w.r.t. the canvas and with at least two rotational motions with nonparallel rotation axes. Capture these projections with the external camera (8). Fig. B.27 (a) shows a typical projection unto the canvas. At the same time, record the poses of the TCP (5) of the tracking system attached to the projector (6).
4. Process the images using DLR CalDe, and subsequently convert these pixel projections unto metric coordinates in S_{canvas} using the camera-to-canvas homography obtained above.
5. Modify the text interface files between DLR CalDe and DLR CalLab, substituting pixel coordinates by the projected pattern coordinates in projector coordinates, as well as calibration object coordinates by metric projection coordinates on S_{canvas}, for every image (refer to Section B.3).

6. These files allow now to intrinsically calibrate the projector as a "virtual" camera using DLR CalLab. Regrettably, it turns out that the traditional lens distortion methods presented in Section 2.2.1 do not match the distortion effects of the pico projector. This is due to the fact that distortion is mainly due to the dynamics of the microelectromechanically-actuated mirror, and secondarily to the lenses used. Fig. B.27 (b) shows a typical pattern of projection residuals. Critically, the pattern is identical in every calibrated projection, which proves that the camera can actually be modeled perspectively as long as we manage to cancel out these residual distortion errors. I opt to generate a 2-D LUT on projection coordinates and undistort the projected pattern by direct mapping using bilinear interpolation of the LUT. To that end, we utilize the OpenGL ES library for computer graphics in embedded systems. This method brings distorted projections back to perspectively correct projections and results in sub-pixel accurate reprojections after intrinsic calibration (in the order of 0.25 pixels RMS error), which in turn provide highly-accurate estimation of the absolute extrinsics of the projector w.r.t. the canvas frame S_{canvas}.

7. Proceed with the second stage of DLR CalLab to extrinsically calibrate the projector w.r.t. the external tracking system (hand-eye calibration). This process (explained in Section 3.3) estimates both, the hand-eye transformation between the projector and the TCP of the tracking system, and the rigid body transformation between S_{canvas} and the base reference frame of the tracking system e.g. S_0.



(a) Typical projection captured and measured by the external camera used for calibration of the projector.

(b) Residuals between actual and expected projections using an optimally-parameterized pinhole camera model of the projector (3.7 pixels RMS error).

Figure B.27: Image projections and reprojection residuals during calibration.

This method results in millimetric accuracy in the projected pattern if the projector is static, viz. ±1.5 mm at 40 cm range. If the projector is in motion, the lag of the external motion tracking system and of the interfaces to an embedded computer produce larger deviations in the ballpark estimate of 8 mm in hand-held motion. The estimated delay in pose tracking estimation amounts to 0.1 seconds, refer to (Sollinger, 2012).

B.2.6 Other Platforms

Calibration of eye-tracking cameras on 3-D displays

Manufacturers of display panels introduced the first glassless 3-D flatscreens for close-range applications (*i.e.*, for desktop computers) approximately five years ago. Their technology works from the premise that the human's eyes are located on a particular spot w.r.t. the display. Alternatively, the human's eyes can be tracked e.g. by a display-mounted stereo camera to estimate their pose w.r.t. the camera. If the pose of the stereo camera w.r.t. the display is known, the pose of the human's eyes w.r.t. the display can be finally estimated. Sometimes these cameras are not integrated into the display but attached by the user, therefore a camera calibration method is necessary that estimates both, the intrinsic parameters of the cameras and their pose w.r.t. the display.

In 2010 I and colleagues at DLR filed a patent for a method that solves the abovementioned problem. The **patent specification DE20101004233** at `http://goo.gl/JiQbE` presents a general method for the estimation of the pose of a camera or several cameras w.r.t. an object when the object does not lie within the field of view of the camera(s). To this end, we propose the use of a planar mirror of unknown pose. Put in concrete terms, this general procedure makes it possible to calibrate a display-mounted (stereo) camera in a similar way as presented in Section 3.2, whenever a mirror is used for the camera to visualize a calibration target displayed on the 3-D display, see Fig. B.28. The mirror can be freely moved at the front of the camera and, by varying its orientation, the pattern on the display imaged by the camera, viz. perspectively distorted depending on the poses of the mirror and of the camera. By formulating the intrinsic and extrinsic camera calibration methods w.r.t. a virtual camera behind the mirror, it is possible to estimate both, the pose of the mirror and the pose of the camera—along with its intrinsic camera parameters.

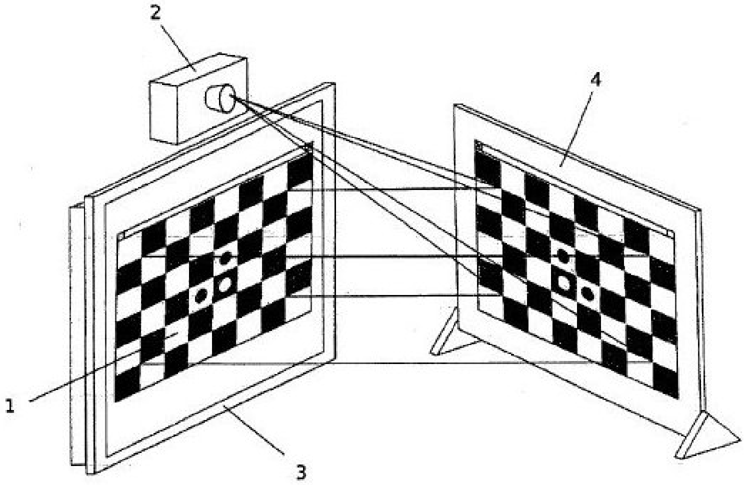


Figure B.28: A calibration pattern (1) is displayed on a 3-D flatscreen (3). A camera (2) is rigidly attached at the top of the 3-D flatscreen (3). A planar mirror (4) is used for the camera (2) to image the calibration pattern on display (1), thus estimating its intrinsic and extrinsic parameters in a similar way as presented in Sections 3.2 and 3.3. *Source: DE20101004233.*

Sensing on automotive assembly lines

The assembly of wheels in the production line of cars is one of the last production stages where human intervention is still necessary. This is due to the fact that the process requires high accuracy whereas the car usually hangs from a conveyor line, and because force-controlled positioning and screwing of the wheel onto the car are required. Furthermore, the front wheels' orientation is unknown and has to be measured. In 2008 our institute led a robotic demonstration of this process, see Fig. B.29.

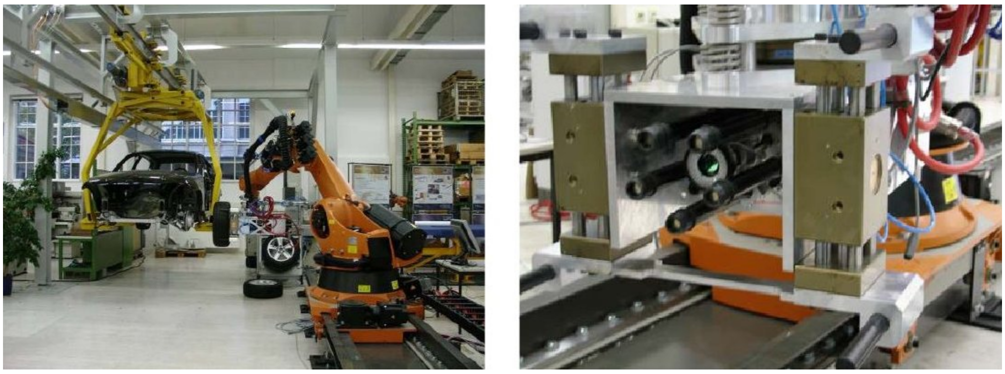


Figure B.29: Demonstration of robotic assembly of wheels onto a BMW car. In the right-hand side: close-up of the impact screwdriver mounted on the end-effector of the robot. Note the camera inbetween the five screwdriver heads.

The position of the conveyor line (see the left-hand side of Fig. B.29) can be only used as a first estimation of the position of the car, as the car may be swinging when hanging from the rails. After that, cameras mounted on the TCP of the robotic manipulator take over perception. To achieve the required accuracy when estimating the pose of the wheel hub w.r.t. the robotic manipulator, the intrinsic and extrinsic calibration methods presented in Sections 3.2 and 3.3 have been employed.

It is, however, hard to estimate the front wheel's rotation in the vertical axis by images from the side of the car. In (Lange *et al.*, 2008) a method was presented to estimate the tilt angle of the front wheels using a laser stripe profiler similar to the used in Sections 2.2.2, 3.6, and 4.3. At close range, the cameras cannot see the wheel hub anymore as their angular field of view is limited by the wheel (the cameras are located at the central hole of the wheel, see Fig. B.29). At close range, contact and force sensors mounted on the robot take over perception to lead control of the robotic manipulator.

Patient registration in minimally invasive surgery

Minimally invasive surgery (MIS) or laparoscopic surgery is a modern surgical technique in which surgical instruments are introduced in the body by small incisions the size of 0.5 to 1.5 cm. In recent years, specialized tools and even robots have been introduced in the field. In robot-assisted MIS surgery it is crucial for the quality of the procedure to accurately localize the patient in a non-intrusive way, as this is a requirement e.g. for preoperative planning of the intervention or augmented reality as presented in Section B.2.5. For instance, MRI or CT data can be overlayed on the patient, and the optimal entry positions for MIS interventions, biopsy needle trajectories, or cutting trajectories for osteotomies can be indicated.

Since humans do not have salient features especially suited for their extrinsic calibration, it is necessary to acquire (part of) their 3-D shape for data registration. A hand-held perception device for 3-D acquisition of the human's body geometry has been developed in the Institute of Robotics and Mechatronics of the DLR, see Fig. B.30. The VR-Map can be considered an adaptation of the DLR 3D-Modeler, see (Schwier *et al.*, 2010). It meets the abovementioned requirements by markerless and contact-free acquisition of 3-D geometry, with highest accuracy. Subsequently, the scan is registered with preoperative data using the Iterative Closest Point algorithm ICP (see (Besl and McKay, 1992) and Fig. B.31 (b)), and represented in the local reference frames of robots for MIS interventions as performed by the MiroSurge robotic system in Fig. B.31 (a).

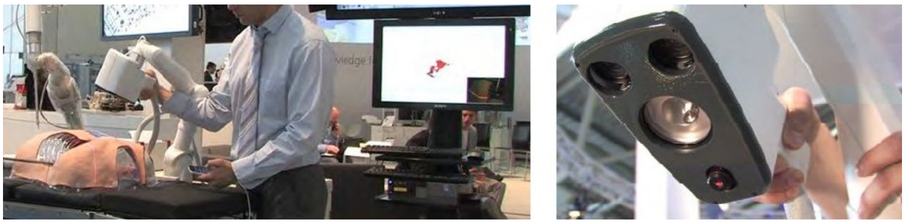
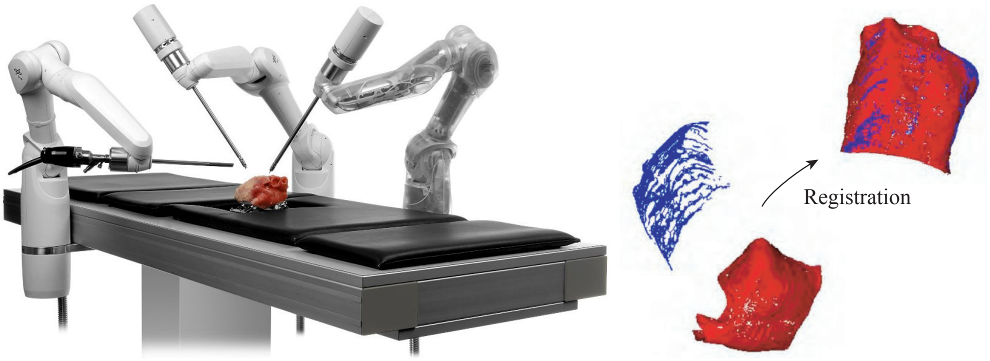


Figure B.30: The VR-Map is a hand-held perception device for robot-assisted surgery including stereo vision, an LSP, and an autopointer, refer to (Schwier *et al.*, 2010). It can be considered an adaptation of the DLR 3D-Modeler to medical applications (Fig. B.25 reprint).



(a) The robotic system for minimally invasive surgery MiroSurge developed at DLR.

(b) 3-D data registration of a torso 3-D scan using the ICP algorithm.

Figure B.31: Patient shape registration in minimally invasive surgery.

B.3 The Camera Calibration Toolbox DLR CalDe and DLR CalLab

DLR CalDe and DLR CalLab emerged late in the year 2005 at the Institute of Robotics and Mechatronics of the DLR. It followed from the strategic purpose of both, upgrading the former CalLab package and to develope a platform-independent application. It was decided that a brand new application—independent but inspired by the old CalLab—had to be produced. Having platform independency in mind, it was chosen to develop in the IDL programming language. In addition to that, this choice yielded reduced development time and boosted performance.

Concerning application design, it was decided to detach the processes of feature detection from the parameters estimation process. The former task is now performed by the program DLR CalDe, which is completely independent of DLR CalLab. The latter task is exclusively performed by DLR CalLab. The sole interface are plain text files.

Figs. B.32 and B.33 give a first impression of the look-and-feel of the calibration toolbox.

Note that recently I translated DLR CalDe and DLR CalLab into the C++ programming language, which allows for lighter packaging and distribution of the toolbox e.g. together with commercial robotic systems. In addition, a C++ implementation of DLR CalDe allows for faster image processing and extensive parallelization, which is useful in the case of large numbers of high-resolution images.

B.3.1 DLR CalDe (DLR *Cal*ibration *De*tection Toolbox)

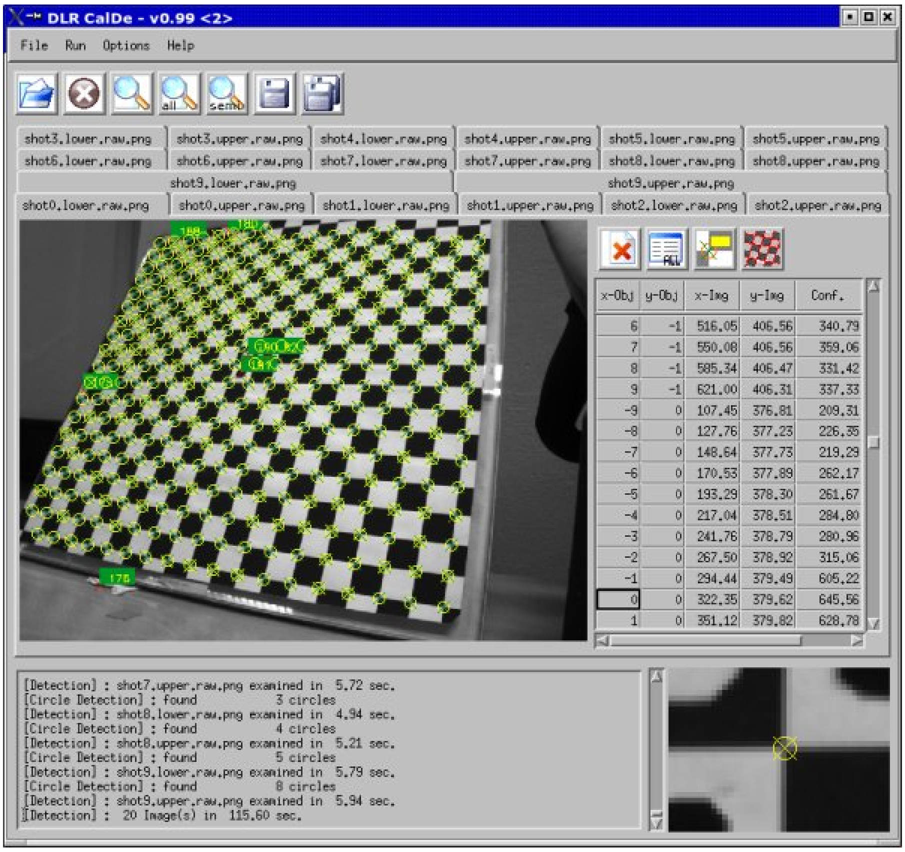
The detection toolbox DLR CalDe serves the need for localizing landmarks/corners on a chessboard-like 2-D calibration panel, viz. with sub-pixel accuracy.

In contrast to the vast majority of similar freely-available applications, here the operation is fully automatic. In addition to that, the calibration pattern no longer has to be fully visible within the images. The implications of this fact are twofold: First, it makes possible to calibrate lens distortion in the peripheral regions of the image. Second, it facilitates the calibration of stereo cameras and eye-in-hand or eye-to-hand systems, since partially visible patterns suffice for calibration.

The application also preserves the possibility of manual interaction and adjustment of the selected landmarks.

In the end, DLR CalDe generates files containing the correspondences between the actual 3-D coordinates of the landmarks of the calibration object and their (stereo) image correspondences, *i.e.*, their detected 2-D projections. These are starting point for the camera calibration toolbox DLR CalLab.

Please find the short tutorial of DLR Calde In Section D.1 within Appendix D.



x-Obj	y-Obj	x-Img	y-Img	Conf.
6	-1	516.05	406.56	340.79
7	-1	550.08	406.56	359.06
8	-1	585.34	406.47	331.42
9	-1	621.00	406.31	337.33
-9	0	107.45	376.81	209.31
-8	0	127.76	377.23	226.35
-7	0	148.64	377.73	219.29
-6	0	170.53	377.89	262.17
-5	0	193.29	378.30	261.67
-4	0	217.04	378.51	284.80
-3	0	241.76	378.79	280.96
-2	0	267.50	378.92	315.06
-1	0	294.44	379.49	605.22
0	0	322.35	379.62	645.56
1	0	351.12	379.82	628.78

```
[Detection] : shot7.upper.raw.png examined in  5.72 sec.
[Circle Detection] : found            3 circles
[Detection] : shot8.lower.raw.png examined in  4.94 sec.
[Circle Detection] : found            4 circles
[Detection] : shot8.upper.raw.png examined in  5.21 sec.
[Circle Detection] : found            5 circles
[Detection] : shot9.lower.raw.png examined in  5.79 sec.
[Circle Detection] : found            8 circles
[Detection] : shot9.upper.raw.png examined in  5.94 sec.
[Detection] :  20 Image(s) in  115.60 sec.
```

Figure B.32: Main window of the corners detection program DLR CalDe.

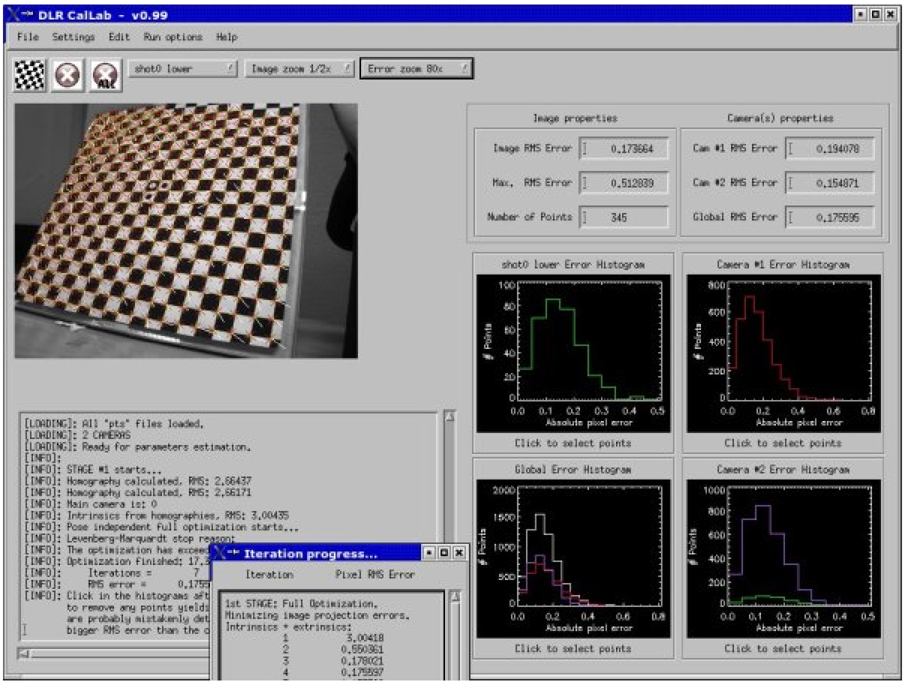


Image RMS Error: 0.173664
Max. RMS Error: 0.512839
Number of Points: 345
Cam #1 RMS Error: 0.194078
Cam #2 RMS Error: 0.154871
Global RMS Error: 0.175595

```
[LOADING]: All "pts" files loaded.
[LOADING]: 2 CAMERAS
[LOADING]: Ready for parameters estimation.
[INFO]:
[INFO]: STAGE #1 starts...
[INFO]: Homography calculated. RMS: 2.66437
[INFO]: Homography calculated. RMS: 2.66171
[INFO]: Main camera is: 0
[INFO]: Intrinsics from homographies. RMS: 3.00435
[INFO]: Pose independent full optimization starts...
[INFO]: Levenberg-Marquardt stop reason:
[INFO]: The optimization has exceed
[INFO]: Optimization finished: 17.3
[INFO]:      Iterations =       7
[INFO]:      RMS error =      0.1755
[INFO]: Click in the histograms aft
        to remove any points yieldi
        are probably mistakenly det
        bigger RMS error than the o
```

```
     Iteration        Pixel RMS Error

1st STAGE: Full Optimization.
Minimizing image projection errors.
Intrinsics + extrinsics:
             1           3.00418
             2           0.550361
             3           0.178021
             4           0.175597
```

Figure B.33: Main window of the parameters estimation program DLR CalLab.

B.3.2 DLR CalLab (DLR ***Cal***ibration ***Lab***oratory)

DLR CalLab estimates both, the intrinsic and extrinsic parameters of either a single camera or a stereo camera (*i.e.*, of a constellation of two or more cameras rigidly attached to each other). It does so on the basis of the previously detected image features (e.g. from DLR CalDe).

Intrinsic parameters describe perspective projection, lens and sensor distortions, as well as the digitization process (refer to Chapter 2). These define the nonlinear transformation between projections in the camera reference frame S_C and themselves when represented in the image memory frame S_M. In the case of a stereo camera calibration, the rigid body transfomation(s) between cameras can be considered as further intrinsic parameters of a more general instance called stereo camera.

Extrinsic parameters describe the rigid body transformations between the main camera reference frame S_C and either the world reference frame S_0 or the **t**ool **c**enter **p**oint (TCP) reference frame S_T. The former transformation changes at different instants (camera stations), whereas the latter (also called hand-eye transformation) remains constant as long as the camera(s) stay rigidly attached to the TCP. The hand-eye calibration is implemented following different methods like 2-D image reprojection minimization (Zhuang *et al.*, 1995; Malm, 2003), closed-form solutions (Zhuang *et al.*, 1994), as well as the novel method presented in (Strobl and Hirzinger, 2006) and Section 3.3.

The program offers extensive interaction possibilities:

- Choice and parametrization of the numerical optimization algorithms.
- Hands-on histograms and images for the selection of mistaken corners to be removed. It is more appropriate to take over this task from DLR CalDe since estimated, reprojected points are useful to promptly verify the detected corners.
- A variety of parameter estimation methods is implemented.
- The lens distortion model can be flexibly selected up to radial distortion in 3rd, 5th, and 7th orders, decentering distortion in 2nd and 4th orders, and thin prism distortion in 2nd and 4th orders (refer to Section 2.2.1).
- It is possible to release the aspect ratio and the absolute scale of the calibration target, see (Strobl and Hirzinger, 2008).
- It is even possible to release the full geometry of the calibration target during intrinsic and extrinsic calibrations, by following the novel methods in (Strobl and Hirzinger, 2011).

Of course, it is also possible to perform automatically the whole calibration process in one-button-mode.

Please find the short tutorial of DLR CalLab In Section D.2 within Appendix D.

B.4 Summary and Discussion

In this chapter I present several implementations of the methods and devices introduced in the last chapters. These implementations constitute the real-world context of the introduced methods and devices. By understanding their context, the reader may now better comprehend the details of their realization, as well as the design guidelines originally introduced in Section 1.2 focusing on the effective implementation of methods and devices in service robotics applications.

Starting out I present the DLR 3D-Modeler as the perception head of the humanoid robot "Justin." In this context, the calibration of the cameras of the DLR 3D-Modeler requires special care both, because of the limited motion range of the pan-tilt unit on which the DLR 3D-Modeler is mounted, and because of the strong accuracy requirements of the applications. In reward, the stereo camera can be also used to support the kinematic calibration of "Justin."

Next the implementation of a light stripe profiler (LSP) acting as an obstacle avoidance sensor mounted on the ExoMars rover has been detailed. Due to the particularities of the platform, the original LSP within the DLR 3D-Modeler had to be extended in aspects concerning both hardware and software. In a further implementation—now in outer space—the visual pose tracking method presented in Chapter 5 has been used to estimate the relative motion of two satellites—one potentially tumbling down to Earth and the other aiming at a controlled docking maneuver unto the latter. The visual pose tracking method presented in Chapter 5 had to be customized to cope with spurious specularities obtained in hardware-in-the-loop experiments.

Special calibration methods for constrained hardware have been also presented targeted at the robotic car RoboMobil featuring 18 cameras, a laser pico projector acting as an inverse camera, and to cameras mounted on 3-D display panels (in this context the patent DE20101004233 was filed).

Medical applications comprise the calibraton of autopointers for augmented reality as well as the insertion of the LSP of the DLR 3D-Modeler for 3-D patient registration; these contributions are realized in the 3-D modeling device VR-Map.

On the software side, the well-known camera calibration toolbox DLR CalDe and DLR CalLab has been introduced (Strobl *et al.*, 2005). The software is ranked in the top three among the freely-available camera calibration toolboxes worldwide. Beyond learning my lessons on algorithmic and computer programming, I learned a lot about maintaining a software package for an active community of users.

In conclusion, from the numerous implementations of the DLR 3D-Modeler and of the methods developed in this thesis, I first and foremost learned that real-world implementations always require a considerable extent of ad hoc customization and developing time.

"If you can't explain it simply, you don't understand it well enough."
—Albert Einstein

C
The DLR 3D-Modeler Documentation

This appendix contains the latest version of the official documentation of the DLR 3D-Modeler (ver. 1.1-pre, 2010-10-29). This document was co-authored by Tim Bodenmüller.

C.1 General System Description

Legal Information

The multisensory DLR 3D-Modeler hardware and software components (Suppa and Hirzinger, 2004; Suppa *et al.*, 2007) have been developed at the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany.

This documentation describes the use of the DLR 3D-Modeler with ego-motion estimation from images, *i.e.*, without using external pose sensing.

C.1.1 Introduction

Generally, 3-D scanning is the task of sampling the surface of an object by manually moving a scanner device along a scan trajectory with respect to (w.r.t.) the surface. A 3-*D scanner system* consists of a *range sensor* that measures a set of distances and a *pose sensor* that measures the pose of the scanner system w.r.t. a global coordinate system (usually one rigidly attached to the object). Here, the range sensor is the *Light Stripe Profiler (LSP)*, see Section C.1.1, and the pose sensor is the *ego-motion estimator*, see Section C.1.1. The DLR 3D-Modeler system further supports the immediate (streaming) 3-D surface reconstruction and visualization of the measured data. The processing concept is summarized in Fig. C.1. In the following, the principles of these modules are summarized.

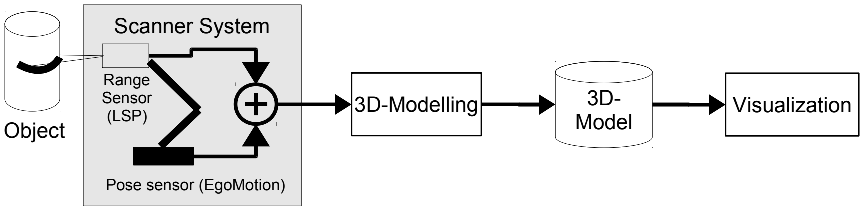


Figure C.1: 3-D modeling using the DLR 3D-Modeler.

Range Measurement: The Light Stripe Profiler (LSP)

The *Light Stripe Profiler (LSP)* (Strobl *et al.*, 2004) is a 1-D range sensor (*i.e.*, it delivers range data for a 1-D stripe of directions) that uses one camera and a laser-line module. The laser beam illuminates a stripe on the surface while the camera records the diffuse reflection. An advantage of our implementation of this sensor is that we are not using any optical filter[1] for direct line segmentation on the images. By doing this, concurrent applications can simultaneously use unfiltered camera images. Possible applications are here stereo reconstruction, texturing of the resulting 3-D model, image-based pose estimation (*ego-motion tracker*), and the use of the camera live stream for user-friendly visualization during scanning, e.g. augmented reality.

Pose Measurement: The Ego-Motion Tracker (EMT)

The *Ego-Motion Tracker* (Strobl *et al.*, 2009a; Mair *et al.*, 2009, 2010b; Strobl *et al.*, 2011) is an accurate, real-time localization system based on a single calibrated camera. Its efficiency and accuracy are based on monocular feature tracking and sub-pixel accurate stereo triangulation respectively. By using motion tracking, no external referencing system is necessary anymore. Furthermore, it is a purely passive method as it does not require any physical action on the environment. Efficient pose estimation is achieved by using V-GPS (Burschka and Hager, 2003) in a novel, robust manner (Mair *et al.*, 2009). Intelligent feature management robustifies the pose estimation additionally. The algorithm is designed for close range applications like hand-held 3-D scanning, but it also allows for mobile robots to estimate their own motion in real-time, without any knowledge about their environment except for its rigidity.

3-D Modeling and Visualization

In order to provide a suitable visual feedback to the user it is necessary to allow for in-the-loop integration and processing of range measurements. Rendering of the raw measurement data is the most simple way of providing visual feedback. However, this typically results in poor visualization quality as no reasonable model shading is possible and it is difficult for the user to judge the quality of

[1] Optical filters may be mounted in front of the camera lens in order to filter out light at frequencies different from the characteristic frequency of the filter, *i.e.*, only laser light passes through.

the eventual 3-D model. Hence, it would be beneficial to generate the desired 3-D model in-the-loop and to be able to visualize it in real time.

The DLR 3D-Modeler software suite integrates in-the-loop generation and visualization of 3-D surface models (Bodenmüller and Hirzinger, 2004). Surface reconstruction incrementally generates a dense and homogeneous triangular mesh from measurement data by extending and refining the surface model with every newly inserted point. This process implicitly filters out outliers and rejects under-sampled regions until enough sample points are available. The method is not scanner-specific but it is suitable for generic types of 3-D scanners. The user can instantly begin scanning objects and does not need to parametrize the workspace beforehand. Further, the generated model is available at any time and no additional post-processing is required for visualization.

C.1.2 Hardware Components

The multisensory DLR 3D-Modeler system (in its ego-motion setup) comprises the DLR 3D-Modeler unit with attached carry handle, the sensor PC, and some cabling, as shown in Fig. C.2. In detail:

- 1x DLR 3D-Modeler main unit with handle and stand
- 1x 1394b Firewire cable
- 1x Sensor PC with mouse and keyboard
- 1x 27" Monitor

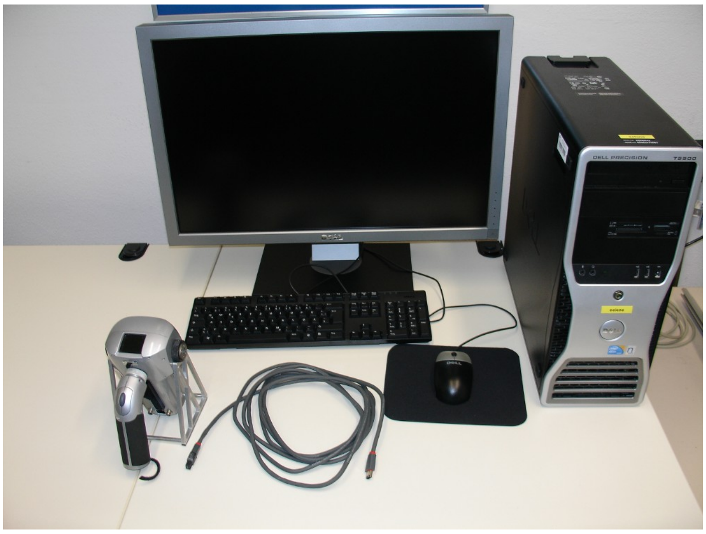


Figure C.2: Components of the multisensory DLR 3D-Modeler (ego-motion setup).

C.1.3 The DLR 3D-Modeler Overview

The *multisensory DLR 3D-Modeler main unit* (Fig. C.3) is a sensor system for manual and automatic digitization of object surfaces. Its sensing components are a pair of FireWire cameras, two laser-line modules, and a DLR laser-range scanner (LRS). The latter is an independent laser-range sensor in its own and is described by Hacker *et al.* in Ref. (Hacker *et al.*, 1997). The integrated AVT Marlin cameras[2] feature an image resolution of 780×580 pixel and a maximum full-frame rate of 50 Hz. The base distance of the cameras is 50 mm, which represents a trade-off between perspective projection dissimilarity and range precision, assuming a general working range between 100 and 2000 mm for stereo triangulation. The laser-line modules have an opening angle of 60° and wavelength of 635 nm. The DLR 3D-Modeler main unit can be used either robot-mounted, hand-guided in combination with an external pose measurement system (e.g. optical tracking system), or without any external sensors in the current ego-motion setup. The three mechanical couplers/adapters are identical and they can be used for mounting a handle, tracking markers or robot flanges.

Three buttons are integrated in the handle. They are used to control and configure the sensor modules of the DLR 3D-Modeler. The configuration is visualized on a TFT display on top of the DLR 3D-Modeler main unit. Data and power transmission are via Firewire bus.

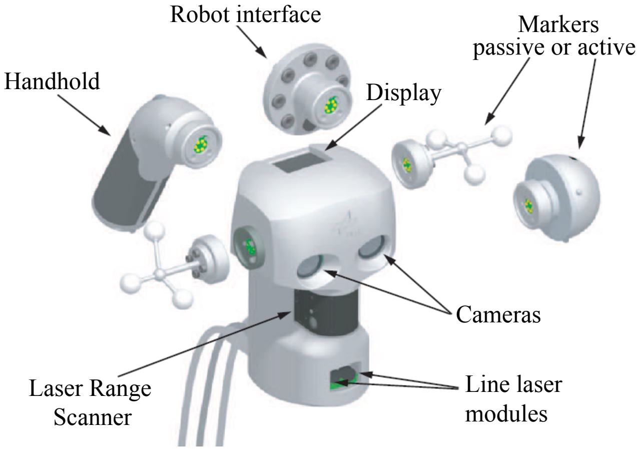


Figure C.3: The multisensory DLR 3D-Modeler and its different potential adapters.

The 3-D modeling software suite is structured in several processing stages, which are implemented as individual programs. Fig. C.4 shows the relations between these components. A hardware abstraction layer is implemented in the program `SensorServer`. This program directly communicates with the DLR 3D-Modeler. On top of `SensorServer` the individual pose and range measurement modules (*i.e.*, LSP and EMT) are arranged, see (Bodenmüller *et al.*,

[2] Allied Vision Technologies (`http://www.alliedvisiontec.com`, 2008).

2007). Data from both sensor modules are received by the `Visu3D` software, which in turn generates and displays the resulting 3-D model. Further, the software supports the simultaneous display of live camera streams and textured 3-D models (augmented reality).

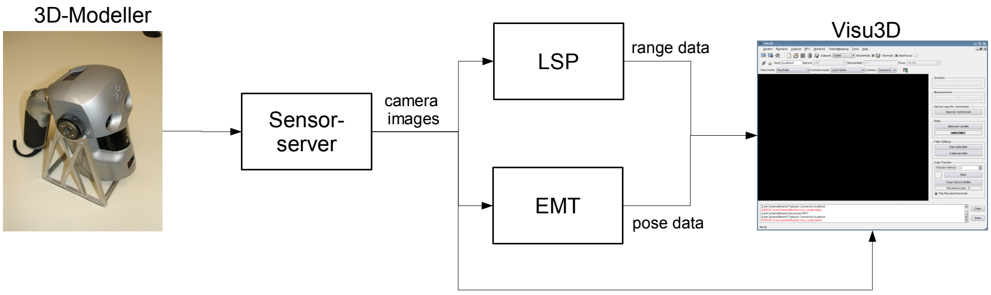
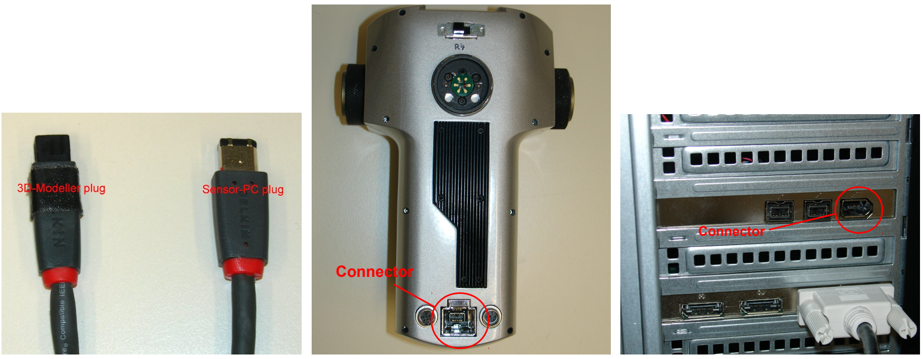


Figure C.4: Overview of the DLR 3D-Modeler's software architecture.

C.2 System Installation

C.2.1 Wiring

Fig. C.5 shows the required physical connections between the DLR 3D-Modeler's components. The Firewire cable connects the DLR 3D-Modeler with the Sensor-PC. The 1394a plug has to be connect to the PC, the 1394b plug must be connected to the DLR 3D-Modeler main unit (see Fig. C.5).



(a) Plugs (b) Modeler connector (c) PC connector

Figure C.5: Connecting the DLR 3D-Modeler to the Sensor-PC.

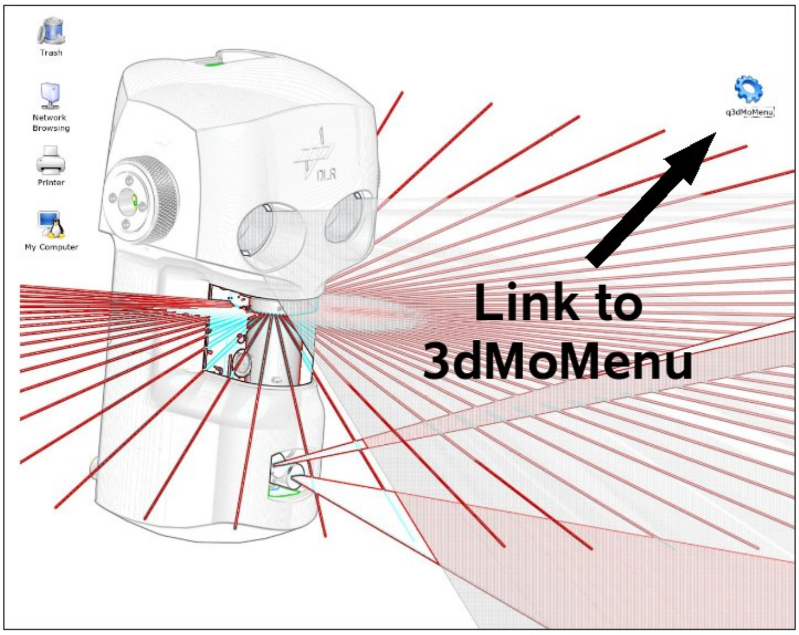


Figure C.6: Initial desktop of the PC and link to *Q3dMo-Menu*.

C.2.2 Starting up the System

The DLR 3D-Modeler system has to be started in the following order:

1. Hardware setup:

   (a) Turn on PC (in EXTERN mode if outside DLR-RM's local area network) and login.

   (b) Turn on power switch of the DLR 3D-Modeler main unit; wait until the DLR 3D-Modeler display shows its menu.

2. Software setup (via *Q3dMo-Menu*):

   (a) Start *Q3dMo-Menu* (see Fig. C.6).

   (b) Select the DLR 3D-Modeler main unit used (Q3dMo-Menu: *device* → *3dMo-R4*).

   (c) Select the desired pose sensor (Q3dMo-Menu: *pose* → *VSLAM*).

   (d) Start the `SensorServer` module (Q3dMo-Menu: *SensorServer* button).

   (e) Configure the stereo camera system (see Section C.2.3).

   (f) Start and configure `Visu3D` (see Section C.5).

   (g) Start and configure the *LSP* module (see Section C.2.4).

   (h) Start the *EMT* module (*ego-motion*, see Section C.2.5).

A more detailed description of the *Q3dMo-Menu* program is given in Section C.3.

C.2.3 Configuration of the Stereo Camera

The parameters of the stereo camera system (*shutter time*, *gain* and *white balance*) have to be set by the user because environmental lighting conditions differ depending on both the system's location and potentially the time of the day as well. The `coriander` software can be used for this purpose. The necessary steps are as follows:

1. Start stereo viewer module (Q3dMo-Menu: *extras → start stereo viewer*)

2. Start `coriander` twice (Q3dMo-Menu: *extras → start coriander*) and select a different camera on each `coriander` window.

3. Set *shutter time* and *gain* (`coriander`: *Controls* label) so that the stereo viewer's images are not overexposed (see Fig. C.7). Furthermore, the *shutter times* of both cameras must be identical. Typical values are in the range of 300 to 800. The *gain* is set by `coriander` to 1 but should be increased to values up to 150 if necessary: Higher *gain* values allow for lower *shutter times* in the case that *shutter time* is higher than 400 (a value that would be desirable).

4. Set *white balance*. First, the DLR 3D-Modeler main unit's cameras should solely aim at a rather white/neutral scene. After that, the *white balance* mode (`coriander`: *Controls* label) has to be set to *auto* for a few seconds (the color weighting of the cameras now changes automatically). Then, the *white balance* mode has to be set back to *man*.

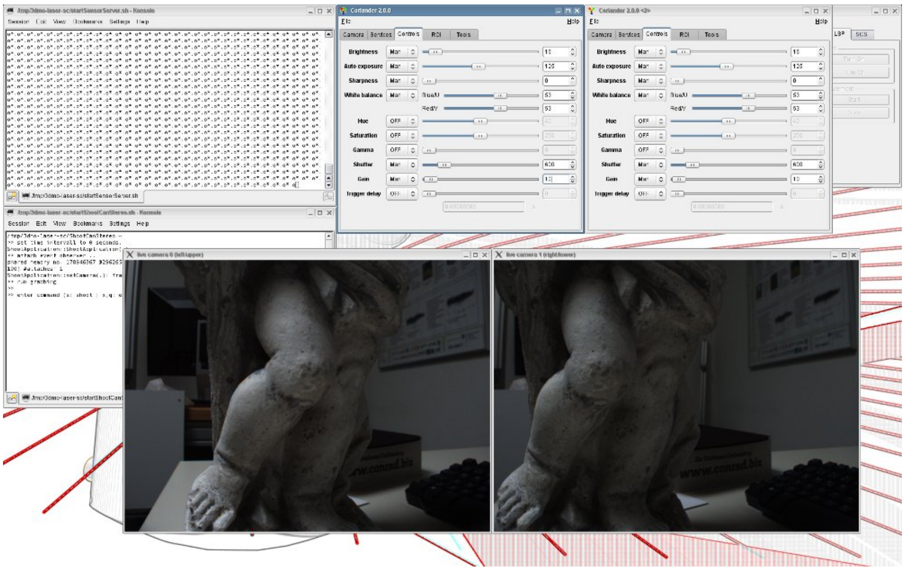
5. Close the *stereo viewer* module (type '`x`' or '`q`' in its console window, and then `<ENTER>`).

6. Close—or minimize—the `coriander` windows.

C.2.4 Start and Configuration of the LSP Module

The *Laser Stripe Profiler* (*LSP*) uses a color look-up table (LUT) of the scene in order to distinguish between (scene) background and laser-line reflection. Even though a typical LUT is already provided, the LUT should be adapted by the user to include the color values of the scene that will be digitized. A strongly inadequate LUT may cause either wrong segmentation results of the laser-line or no segmentation at all.

In order to learn the LUT anew, the LSP must be first started (click on 'RangeSensor LSP' button in *Q3dMo-Menu*). After that (and, if required, at any other moment during operation), 'LUT measurement' must be selected either using the display, wheel and buttons on the DLR 3D-Modeler main unit (see Section C.4), or using the 'Sensor special command' button in `Visu3D` (see Section C.5). Now, the DLR 3D-Modeler main unit has to be directed to the scene or object to be modeled. By pressing (and maintain pressed) the unit's *fire button*, or alternatively clicking on `Visu3D`'s 'Start LUT' button, the system acquires images of the scene *without* laser projections. These images are used



```
/tmp/3dmo-laser-sc/ShootCamStereo ~
>> set time intervall to 0 seconds.
ShootApplication::ShootApplication(
>> attach event observer ..
shared memory no. 178946367 9296265
100) #attaches 1
ShootApplication::setCamera(.): fra
>> run grabbing
>>
>> enter command (s: shoot | x,q: e
```

Brightness	Man		15
Auto exposure	Man		125
Sharpness	Man		0
White balance	Man	Blue/U	53
		Red/V	53
Hue	OFF		
Saturation	OFF		
Gamma	OFF		
Shutter	Man		600
Gain	Man		10
Trigger delay	OFF		

/tmp/3dmo-laser-sc/startSensorServer.sh - Konsole
Session Edit View Bookmarks Settings Help
/tmp/3dmo-laser-sc/startSensorServer.sh
/tmp/3dmo-laser-sc/startShootCamStereo.sh - Konsole
/tmp/3dmo-laser-sc/startShootCam
Coriander 2.0.0
Coriander 2.0.0 <2>
File
Help
Camera
Services
Controls
ROI
Tools
live camera 0 (left/upper)
live camera 1 (right/lower)

Figure C.7: Screen of the PC during configuration of the stereo camera system: For each camera a *coriander* tool is started (upper, right-hand side windows). The *SensorServer* (upper, left-hand side console) and the *stereo viewer* (lower, left-hand side console and two lower image windows) are also running.

for generating the new LUT. The user should rapidly scan the scene. When finished after some seconds, the user can finish the LUT learning phase by releasing the fire button or pressing 'Stop LUT' in `Visu3D`.

This LUT learning phase should only be performed once unless either the scene or its ilumination change.

C.2.5 Start of the EMT Module

To start the ego-motion estimation module (*EMT*) click on the 'PoseSensor VSLAM' button in *Q3dMo-Menu*. A camera window appears and the *EMT* module automatically starts delivering pose estimations to the `SensorServer`.

The *EMT* module's operation is as follows: First, scene features at close range (10 to 50 cm) are searched for that are convenient for eventual feature tracking. Only after successful (stereo) initialization of these features, the *EMT* module delivers accurate pose estimations by tracking these features using the image stream of one camera only. If the features are lost by saccadic hand movements or by the absence of other scene features at close range, pose estimation is interrupted. Every time that this happens, the object reference frame where 3-D results are being represented by `Visu3D` is lost. If further data has to be acquired the already modeled scene has to be deleted, or alternatively a separated, new 3-D scene can be opened.

Software extensions that are convenient for extended scanning of a scene, like automatic relocalization on former scenes or loop closing, have been already implemented but are not delivered in this software package yet.

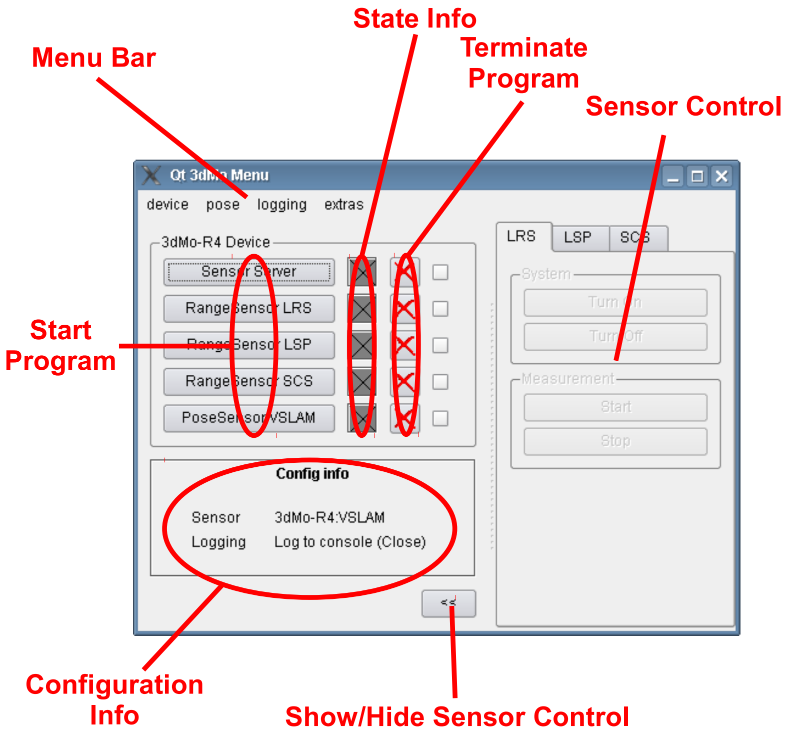
C.2.6 Shutting down the System

The DLR 3D-Modeler system should be turned off in the following order:

1. Disconnect and close `Visu3D`.
2. Close all modules (Q3dMo-Menu: *extras* → *terminate all*).
3. Turn off the DLR 3D-Modeler main unit.
4. Shut down PC (by briefly pressing the power button).

C.3 The Q3dMo-Menu

Q3DMo-Menu is a a graphical user interface (GUI) program for starting and surveilance of the software components of the DLR 3D-Modeler. The menu is shown in Fig. C.8 with labeled components. In this section the components functions are explained.



Config info

Sensor 3dMo-R4:VSLAM
Logging Log to console (Close)

Figure C.8: The components of the *Q3dMo-Menu*.

C.3.1 Starting and Stopping Modules

The main area of the *Q3DMo-Menu* shows a start button, a state info and a terminate button for every sensor module (program) that is available. This is for a selected combination of a DLR 3D-Modeler device and a pose sensor. The DLR 3D-Modeler device can be selected by the **device menu** in the menu bar. Analogously, the pose sensor is selected in its **pose menu**. A module is started by pressing its respective labeled button and terminated by pressing the red cross button beside it. The state info field (colored square) shows the current state of the respective module, which can have the following colors:

☒ program has not been started yet

🟥 sensor is turned off (state OFF)

🟩 sensor is ready (state READY)

🟨 sensor is measuring (state RUNNING)

⬛ sensor is busy (state UNDEFINED)

Logging Options

The **logging menu** controls the programs output, e.g. error messages. The following options can be used:

- **No logging output** - no output
- **Log messages to console** - program is executed in a separate console
- **Log messages to files** - program messages are written in a file

The option *Close console* defines whether a console ought to be closed after program termination (in effect only if *Log messages to console* is active).

If *Log messages to files* is selected, the user can assign the destination directory via *Log. file directory*. Each program writes to a separate file named `[program name].log` (e.g. `SensorServer.log`).

Configuration Info Area

The *Configuration Info Area* shows the currently selected DLR 3D-Modeler device and the pose sensor. Further, the logging options are displayed.

Extras

The **extras** menu contains additional useful operations:

(re-)start all: Starts all modules, if not already started.

(re-)start selected: Starts all modules checked in the main window.

terminate all: Terminates all running programs.

kill all: Stops all programs immediately.

start coriander: Starts an instance of the camera configuration tool `coriander`.

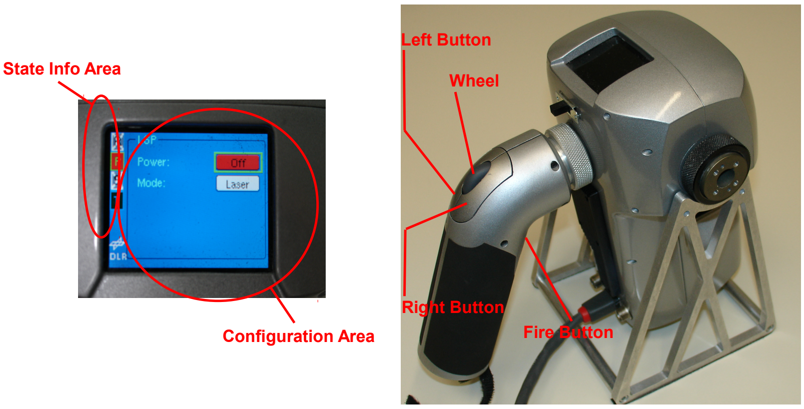
start stereo viewer: Starts an instance of a viewer that displays the live stream of the stereo camera.

start visu3d: Starts an instance of the `Visu3D` program, see Section C.5.

C.3.2 Range Sensor Control

The *Range Sensor Control* area provides direct control of the started range sensors (e.g. of the *LSP*). Every range sensor has three states (and one error state):

- **OFF**: The program is running but no resources are allocated – the system is idle.
- **READY**: The program is running and resources are allocated – the system is ready for immediate measurement.
- **RUNNING**: The program is measuring data.



(a) Menu

(b) Buttons

Figure C.9: The DLR 3D-Modeler display areas and buttons on the DLR 3D-Modeler handle.

C.4 The DLR 3D-Modeler Display and Buttons

The DLR 3D-Modeler can be operated using the handle and display of its main unit after software initialization (refer to chapter C.2). However, data transfer to `Visu3D` still has to be released manually. The displayed menu, the buttons and the wheel are shown in Fig. C.9.

C.4.1 The Display Menu

The display menu has two areas: the state info area (left-hand side column) and the configuration area (right-hand side window). The state info area shows the state of the available range sensors as colored squares. These colors correspond to the state encoding of the *Q3dMo-Menu* (see Section C.3). At the moment only the state square of the *LSP* (labeled with **P**) and the state of the `SensorServer` (labeled with **T**) are relevant. Similar to the *Range Sensor Control* area at the *Q3dMo-Menu*, the configuration area shows sensor-specific settings: For the *LSP*, the sensor can be turned *on* and *off* and the function of the fire button can be toggled between measurement (**laser**) and learning the color look-up table (**LUT**) as explained in Section C.2.4.

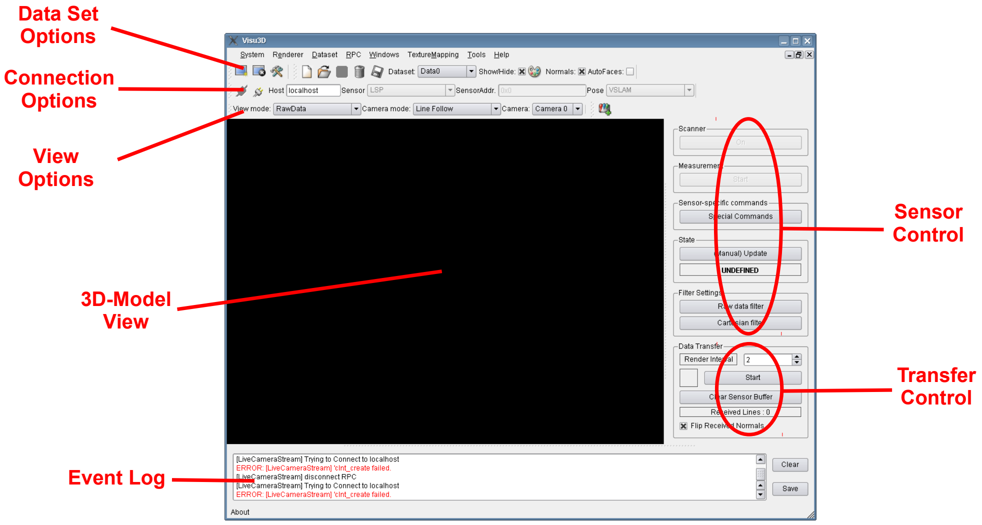
C.4.2 Usage of the Buttons and the Central Wheel

To navigate the menu use the left and right buttons as well as the central wheel. The left button has the cancel/back function, the right button has the okay/next function. Fields in focus can be browsed using the wheel. The fire button toggles (start/stop) the measurement of the selected range sensor (in this software version only the *LSP* (**P**) is available).

In the state info area (left-hand side column) different sensors are browsed by turning the wheel. The right button will select the sensor, activating its configuration area on the right-hand side. Another right button press applies possible changes and jumps to the next entry. A left button press returns to the state info area.

C.5 3-D Modeling using the 3-D Software Visu3D

Measurements can be acquired, visualized and stored using the 3-D software `Visu3D`. The software also enables control of the used range sensors. The data is maintained in one or more data sets, which can be visualized simultaneously. After data acquisition, the data sets can be post-processed, e.g. texturized. Fig. C.10 shows the main window of `Visu3D` with some labeled components. In the following the configuration of the software, the control of range sensors, data acquisition as well as texturing of 3-D models is explained.



```
[LiveCameraStream] Trying to Connect to localhost
ERROR: [LiveCameraStream] 'clnt_create failed.
[LiveCameraStream] disconnect RPC
[LiveCameraStream] Trying to Connect to localhost
ERROR: [LiveCameraStream] 'clnt_create failed.
```

Figure C.10: The components of *Visu3D*.

C.5.1 Introduction

The `Visu3D` main window consists of four areas (see Fig. C.10): the **3-D viewer** that displays the generated model, the **sensor and data transfer control**, an **event logger** window, and the **tool- and menu bars**. The sensor control provides direct access to the selected range sensor (e.g. the *LSP*). The commands are similar to the sensor control of the *Q3dMo-Menu* (see Section C.3). The upper area of the program window consists of a menu bar with all options and three toolbars for data set, sensor connection, and 3-D viewer options.

The 3-D viewer area consists of one or more viewers (or render windows). Each viewer can hold different types of data sets. These determine the kind of data that is displayed and also the type of processing to be performed on these data. At the moment the following data set types are supported:

- **Viewer**: The gathered data is directly shown, in raw. Newly inserted data is not further processed and it is shown as a point set.
- **TriangleMesh**: All inserted data is processed by the surface reconstruction module to generate a triangle mesh in real time.
- The types **Ext. Viewer**, **Viewer+Reduction** and **Normalizer** are only for internal use and should not be used.

C.5.2 Connecting to the DLR 3D-Modeler

In order to receive data from the DLR 3D-Modeler system, `Visu3D` has to be connected to the data streams of both *LSP* and *EMT*, in other words it has to be connected to their respective sensor modules. This is done by the following procedure:

1. Choose *LSP* as sensor.

2. Choose *VSLAM* as pose.

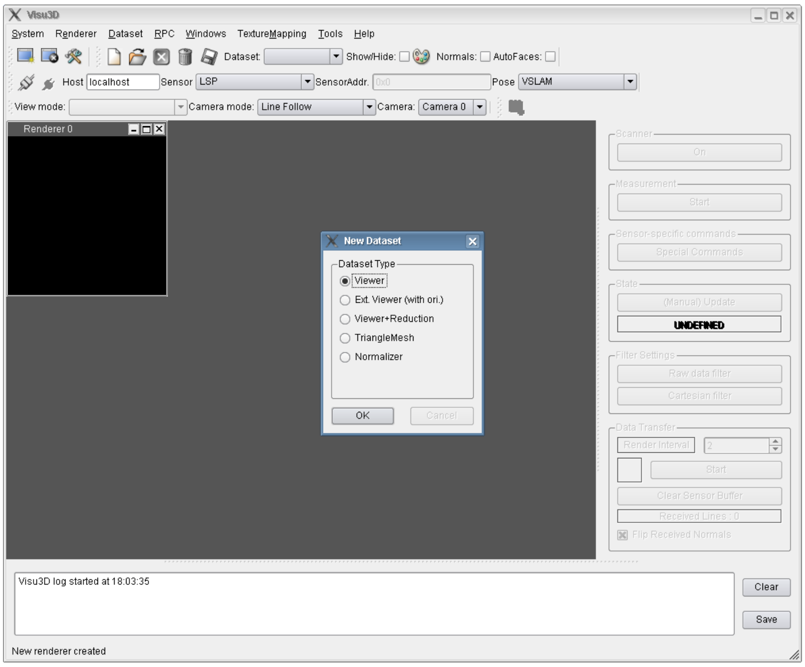
3. Enter `localhost` as *host*.

4. Press the *connect* button.

5. If no data set has been created yet, the data set creation dialog will be displayed as in Fig. C.11. Select either *Viewer* to show the raw 3-D points or *TriangleMesh* for activating surface reconstruction.

6. If the range sensor has not been already started (*i.e.*, showing the state *READY*) by e.g. the *Q3dMo-Menu* or the DLR 3D-Modeler main unit, the user can also use the sensor control section in `Visu3D` to start the range sensor.

7. Start the *EMT* using the *Q3dMo-Menu* (if not already performed before).



Host localhost
Sensor LSP
SensorAddr. 0x0
Pose VSLAM
Camera mode: Line Follow
Camera: Camera 0

New Dataset
Dataset Type
● Viewer
○ Ext. Viewer (with ori.)
○ Viewer+Reduction
○ TriangleMesh
○ Normalizer

UNDEFINED
Render Interval 2
Received Lines : 0
☒ Flip Received Normals

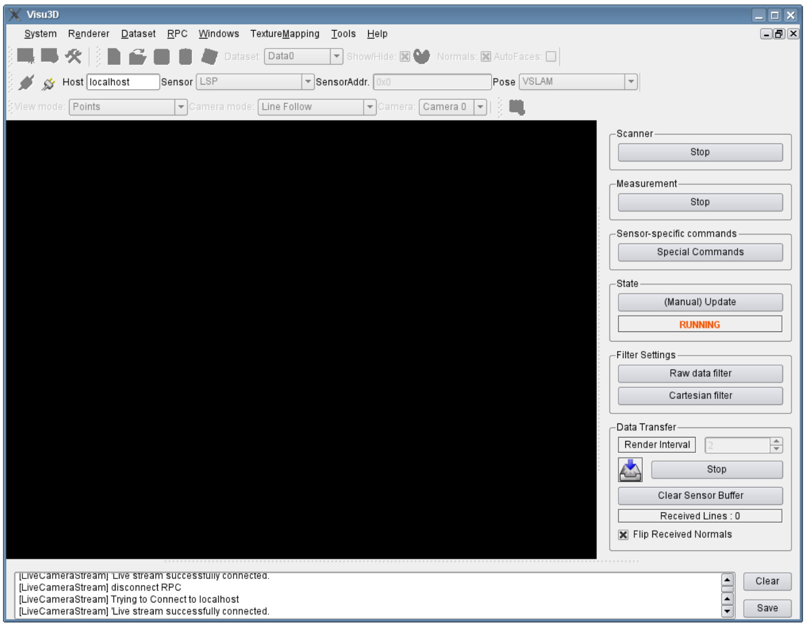
Visu3D log started at 18:03:35

New renderer created

Figure C.11: Selection of the data set type. *In this version: use only Viewer or TriangleMesh!*

C.5.3 Receiving Data

`Visu3D` is set into data reception mode by pressing the *start/stop transfer button* in the transfer control area. The reception mode is indicated by a symbol beside that button. When the program is in this mode, mouse interaction with the model in the 3-D viewer is locked. `Visu3D` immediatley processes all data by the DLR 3D-Modeler system. To stop this data reception mode press the *start/stop transfer button* again.



Dataset: Data0
Host localhost
Sensor LSP
SensorAddr. 0x0
Pose VSLAM
View mode: Points
Camera mode: Line Follow
Camera: Camera 0
Render Interval 2
Received Lines : 0
☒ Flip Received Normals

```
[LiveCameraStream] 'Live stream successfully connected.
[LiveCameraStream] disconnect RPC
[LiveCameraStream] Trying to Connect to localhost
[LiveCameraStream] 'Live stream successfully connected.
```

Figure C.12: The *Visu3D* in transfer mode.

C.5.4 Working with the 3-D Viewer Window

When `Visu3D` is **not** in data reception mode the user can interact with the acquired 3-D model using mouse events on the 3-D viewer window:

- Mouse motion with pressed left button rotates the model.
- Mouse motion with pressed right button modifies the representation scale (in other words, the range from the 3-D viewer's point of view to the 3-D model).
- Mouse motion with pressed central button (wheel) shifts the position of the 3-D model w.r.t. the 3-D viewer's point of view, but without modifying its range.

The user can also choose between different perspective representation modes for the 3-D viewer using *camera-mode* at the the third toolbar of the top menu bar:

- **Line follow** representation follows range data as it arrives from the sensor, rescaling the 3-D modeling at that.
- **Live pose** represents the acquired 3-D model in its correct perspective as perceived by the selected camera (0 or 1), using pose estimation from *EMT*.
- **Live pose and image** is similar to the last option but it additionally displays actual 2-D images gathered by one of the DLR 3D-Modeler's cameras at the back of the virtual, 3-D model – *i.e.*, the 3-D model is overlayed on the actual, 2-D image stream. This corresponds to so-called augmented reality, see Fig. C.13.

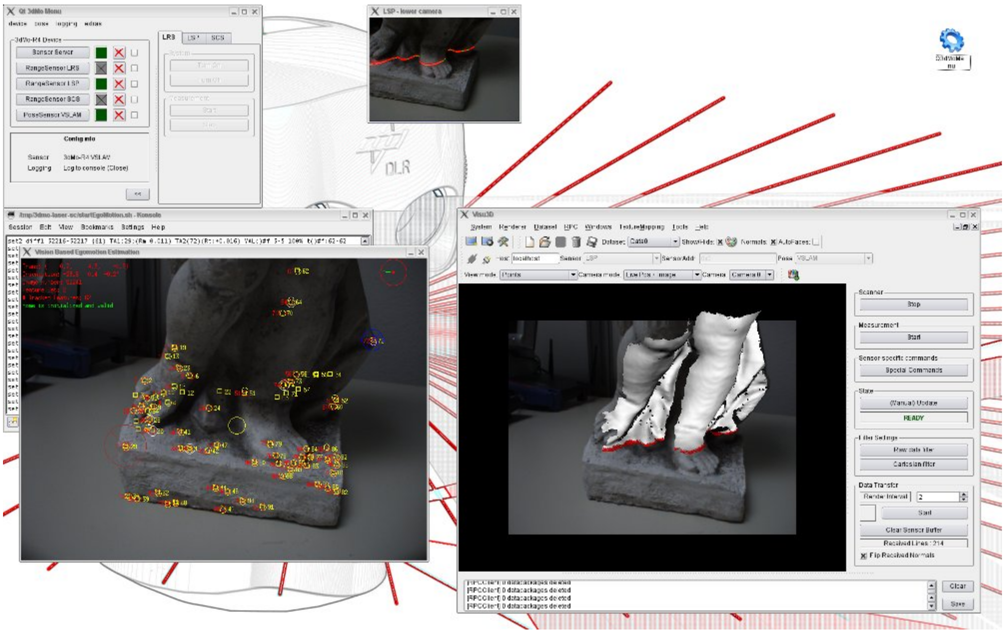


Figure C.13: *Visu3D* transfer mode with live image background.

Further viewer options can be activated by keyboard shortcuts, see Fig. C.14.

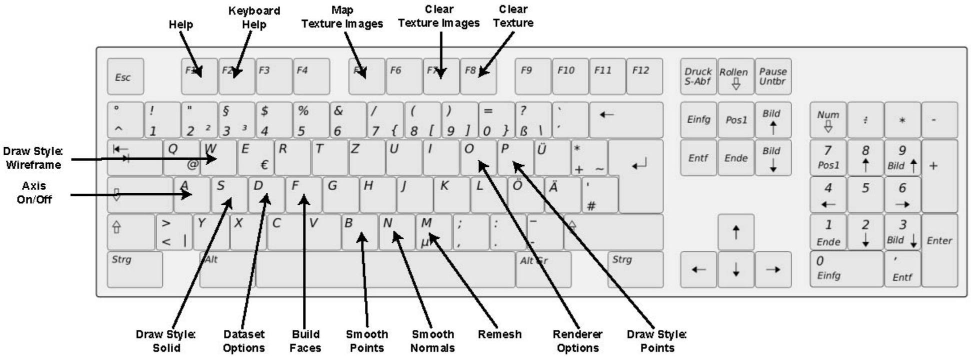
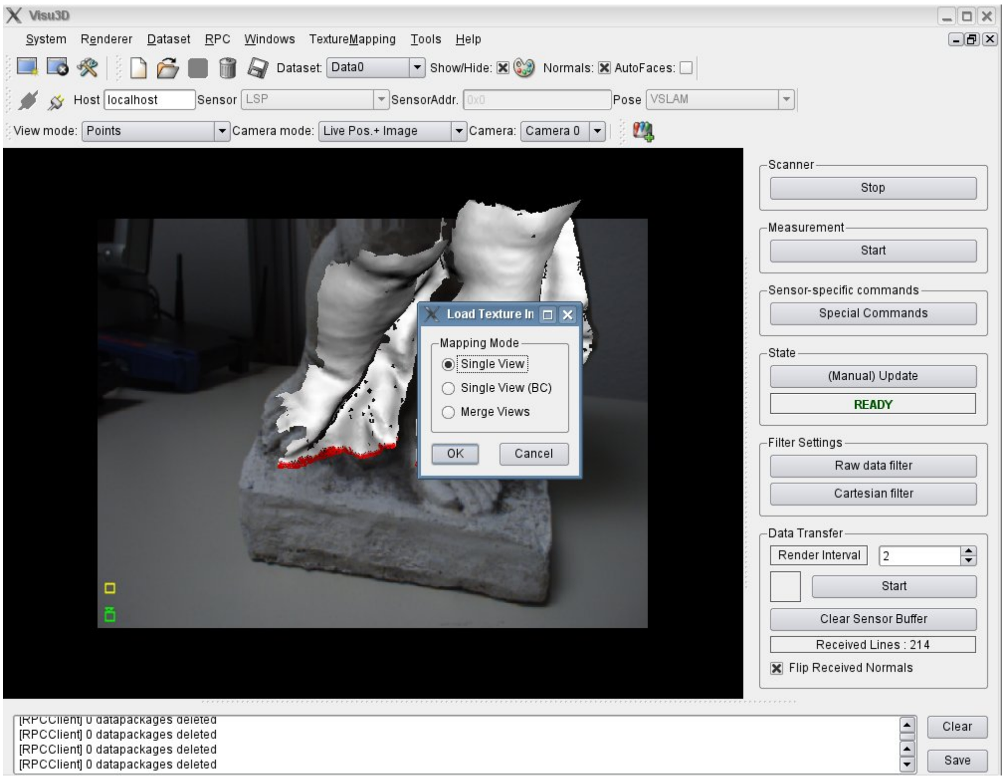


Figure C.14: Viewer commands on the keyboard.

C.5.5 Texturing

Texturing 3-D models can be performed after geometrical 3-D model (range data) acquisition. It is mandatory that data transfer is suspended (see *Scanner Control–Data Transfer*) for this. In order to texture models the following steps are indicated (see Fig. C.15):

1. *Tools–Texture Mapping–Grab Image* grabs a single image for texturing.
2. The amount of overlayed yellow boxes represents the number of acquired images. The user can readily acquire several images following *Tools–Texture Mapping–Grab Image*.



Dataset: Data0 Show/Hide: ☒ Normals: ☒ AutoFaces: □

Host localhost Sensor LSP SensorAddr. 0x0 Pose VSLAM

View mode: Points Camera mode: Live Pos.+ Image Camera: Camera 0

Mapping Mode
● Single View
○ Single View (BC)
○ Merge Views

READY

Render Interval 2

Received Lines : 214

☒ Flip Received Normals

```
[RPCClient] 0 datapackages deleted
[RPCClient] 0 datapackages deleted
[RPCClient] 0 datapackages deleted
[RPCClient] 0 datapackages deleted
```

Figure C.15: *Visu3D* texturing.

3. Applying *Tools–Texture Mapping–Map Texture Images* places textures on the 3-D model surfaces. Note that the model has to consist of triangle faces (*TriangleMesh* data set).

4. The textured model is then visualized. If several images were acquired, choose the option *Merge views* for sensibly merging textures before mapping, see Fig. C.15.

"Every science begins as philosophy and ends as art."
—Will Durant, The Story of Philosophy, 1926

D

DLR CalDe and DLR CalLab Short Tutorial

This appendix contains the latest version of the short tutorials on the usage of the camera calibration toolbox DLR CalDe and DLR CalLab (Strobl *et al.*, 2005), including the short tutorial of Wolfgang Sepp on DLR CalDe, and my own short tutorial on DLR CalLab.

D.1 Short Tutorial on DLR CalDe

1. Preparation:

   a) Create a calibration pattern using "Options→Create calibration pattern" or use one of the supplied patterns. The size of the pattern should be chosen according to the lens aperture as well as to the range for sharp imaging. A bigger calibration pattern usually fits the demands. The pattern should be sticked onto a flat surface, which may be difficult in the case of large objects.

   b) Shoot 3 to 10 images of the calibration pattern from vantage points featuring varying orientations and distances, using either monocular or synchronized stereo cameras. Note that perpendicular images to the pattern are discouraged, see (Strobl *et al.*, 2009b). An optimal calibration can be only achieved if the calibration pattern fills the whole image.

   c) Save the images as Portable Network Graphics (PNG) files named "`*.(left|right|upper|lower).*png`". Additionally, save the corresponding **t**ool **c**enter **p**oint (TCP)-to-robot base homogeneous transformation matrix to text files "`*.coords`" only if you also require the extrinsic hand-eye calibration.

2. Load data.

    a) Load the configuration file corresponding to the particular calibration pattern in “Options→Settings” and edit the measured size of the checkboard rectangles (remember that off-the-shelf printers do not necessarily preserve pattern dimensions, refer to (Strobl and Hirzinger, 2008, 2011)). Note that the circles on the calibration pattern indicate its $\boldsymbol{x}/\boldsymbol{y}$-axis directions as well as the origin of the plate. The $\boldsymbol{z}$-axis is perpendicular to the calibration plane and points inside the object.

    b) Load the PNG images.

3. Detect corner points in the images.

    a) Run the automatic corner point detection for the first image/tab.

    b) If the circles of the calibration pattern have not been recognized, then either enter an appropriate binarization threshold in “Options→Settings” and return to step 3.a), or run semi-automatic detection by mouse-clicking in the circles of the calibration pattern.

    c) If too few corners of the calibration pattern have been detected in spite of correctly detected central points, then the confidence threshold in “Options→Settings”is too high; please lower it and return to 3.a). The images probably present low contrast.

    d) Run the corner point detection for all images/tabs with the above configuration.

    e) Check the result for example by using the “repaint grid”-icon at the top of the table. The detected corners are then linked by a line to their horizontal and vertical neighbors. Wrongly identified points can be now easily detected. Either:

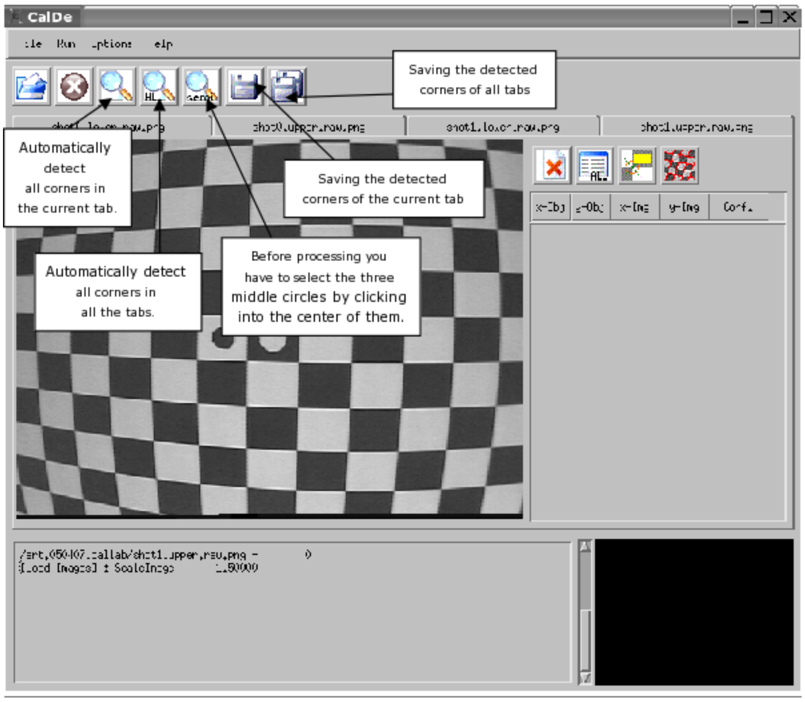
        i. select these points in the table on the right-hand side and click on the “delete-point”-icon to remove these points, or

        ii. adjust the parameters as in 3.b) or 3.c) and repeat full-automatic detection.

    Manual corner point localization is the last resource in the case of really compromised images. For this purpose, select a corner point in the table on the right-hand side and click near to the corresponding corner in the image. These adjustments are actually very rarely needed.

4. Save data.

    Save the detected corner points for all images/tabs by clicking on the “save-points-of-all-tabs”-icon.



Saving the detected corners of all tabs

Automatically detect all corners in the current tab.

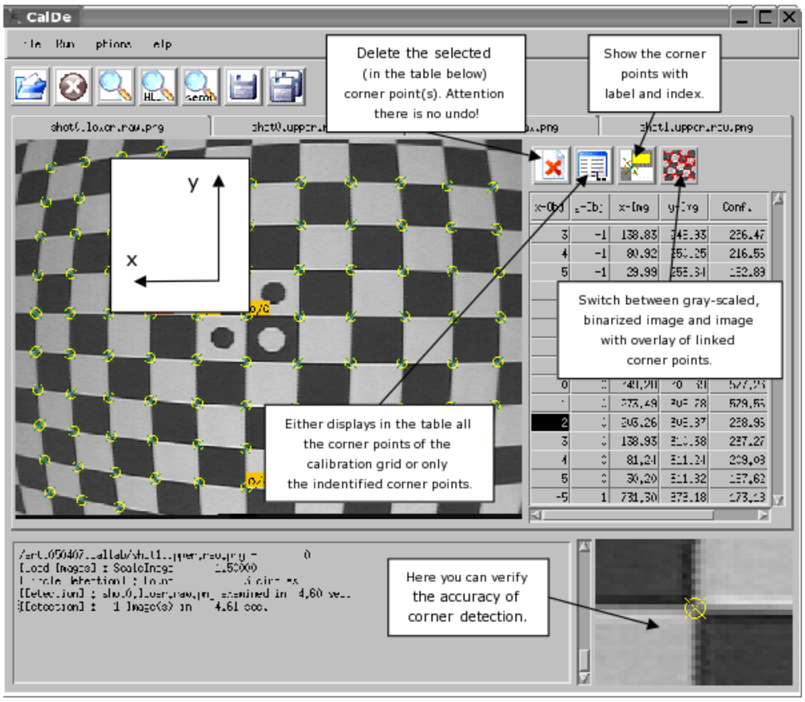
Saving the detected corners of the current tab

Automatically detect all corners in all the tabs.

Before processing you have to select the three middle circles by clicking into the center of them.

```
/ert,050407_callab/shot1_upper,raw.png -          0
[Load Images] : ScaleImage        1.50000
```

Figure D.1: Upper interface buttons on the GUI of DLR CalDe.



Delete the selected (in the table below) corner point(s). Attention there is no undo!

Show the corner points with label and index.

x-Obj	z-Obj	x-Img	y-Img	Conf.
3	-1	138.83	345.33	226.47
4	-1	80.92	350.25	216.55
5	-1	29.99	355.34	152.89
0	0	441.20	30 81	577.25
1	0	373.49	305 78	579.55
2	0	206.26	305.37	228.95
3	0	138.93	310.38	227.27
4	0	81.24	311.24	209.03
5	0	30.20	311.32	157.62
-5	1	731.30	373.18	173.13

Switch between gray-scaled, binarized image and image with overlay of linked corner points.

Either displays in the table all the corner points of the calibration grid or only the indentified corner points.

```
/ert_050407_callab/shot1_upper,raw.png -          0
[Load Images] : ScaleImage        1.50000
[Circle Detection] : Found                3 circles
[Detection] : shot0_lower,raw,png examined in  4,60 sec.
[Detection] :   1 Image(s) in    4.61 sec.
```

Here you can verify the accuracy of corner detection.

Figure D.2: Further indications on the GUI of DLR CalDe.

D.2 Short Tutorial on DLR CalLab

1. Preparation.

   Run DLR CalDe or any other similar program to detect and identify features of a calibration object.

2. Load data.

   Load the PNG image files as in DLR CalDe. The application is flexible to the names and numbers used. At least three images (or stereo image pairs) are required for initialization. You can select several files at the same time by pressing the SHIFT key and selecting the first and the last files, or by pressing the CTRL key and selecting each file. Load the corresponding points files for the images as from DLR CalDe.

3. First calibration stage.

   The general settings should not be modified in a standard calibration case. If the command output says that you are ready for the estimation of the parameters, do proceed with the first calibration stage. Intrinsic camera parameters (including the camera-#1 to camera-#n transformation(s) in the case of a stereo camera rig) are to be estimated. At the beginning, an initial estimation is performed based on homographies, refer to (Zhang, 2000; Sturm and Maybank, 1999). Then, this last estimation is numerically optimized. After that, the reprojection errors are overlaid on the images. You can browse through images searching for mistakenly detected corners. Here it is useful to hide the actual image (click on "Switch monochrome image"), and it is also useful to augment arrows corresponding to the reprojection residuals (click on "Error zoom"). In addition, you can use the histograms for rapidly finding mistakenly detected corners. Repeat this first calibration stage if you did remove corner points.

4. Second calibration stage.

   The second calibration stage provides the TCP-to-camera transformation (or hand-eye transformation). This transformation is not always required by the user and can only be estimated if TCP-to-robot-base transformations for every calibration image have been collected. These transformations have been already used in DLR CalDe, hence embedded into the points files. DLR CalLab implements different algorithms for estimating this transformation ("Settings→General settings"):

   a) The method of Strobl and Hirzinger in (Strobl and Hirzinger, 2006) minimizes errors in the erroneous world-to-TCP transformation. The residual errors in translation can be located either at the top of the manipulator, at its bottom, or on both ends. The algorithm uses a linear least squares solution for initialization.

   b) You can also minimize features reprojection errors. This is only a sound solution in the case of highly noisy cameras e.g. with very low image resolution.

5. Third calibration stage.

   The third calibration stage only serves to verify correct estimation in prior stages. It furthermore produces accuracy estimations for the positioning device (e.g. a robotic manipulator like the Kuka KR 16, an infrared tracking system like the ARTtrack2, etc.); these devices produce the abovementioned world-to-TCP estimations.

6. Save data.

   "File→Save" saves the calibration results in the desired output format. For information on these formats refer to "Help→Documentation."

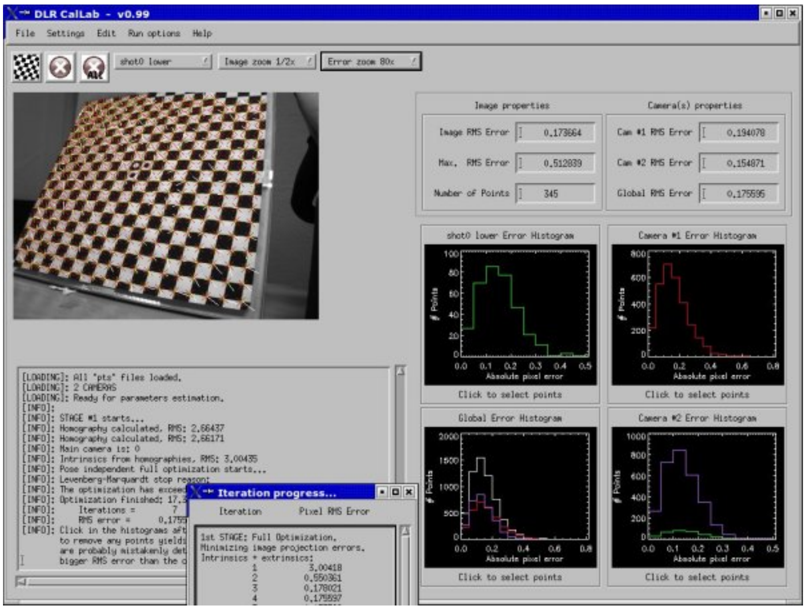


Image properties		Camera(s) properties	
Image RMS Error	0.173664	Cam #1 RMS Error	0.194078
Max. RMS Error	0.512839	Cam #2 RMS Error	0.154871
Number of Points	345	Global RMS Error	0.175595

```
[LOADING]: All "pts" files loaded.
[LOADING]: 2 CAMERAS
[LOADING]: Ready for parameters estimation.
[INFO]:
[INFO]: STAGE #1 starts...
[INFO]: Homography calculated. RMS: 2.66437
[INFO]: Homography calculated. RMS: 2.66171
[INFO]: Main camera is: 0
[INFO]: Intrinsics from homographies. RMS: 3.00435
[INFO]: Pose independent full optimization starts...
[INFO]: Levenberg-Marquardt stop reason:
[INFO]: The optimization has exceed
[INFO]: Optimization finished: 17.3
[INFO]:     Iterations =       7
[INFO]:     RMS error =     0.1759
[INFO]: Click in the histograms aft
        to remove any points yieldi
        are probably mistakenly det
        bigger RMS error than the o
```

```
    Iteration        Pixel RMS Error

1st STAGE: Full Optimization.
Minimizing image projection errors.
Intrinsics + extrinsics:
           1            3.00418
           2            0.550361
           3            0.178021
           4            0.175597
```

Figure D.3: The DLR CalLab GUI after successful calibration.

Bibliography

Y. I. Abdel Aziz and H. M. Karara. Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry. In *Proceedings of the Symposium on Close-Range Photogrammetry, American Society of Photogrammetry*, pages 1–18, Falls Church, VA, USA, 1971. 54

M. Agrawal, K. Konolige, and M. R. Blas. CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 102–115, Marseille, France, October 2008. 168

A. Alahi, R. Ortiz, , and P. Vandergheynst. FREAK: Fast Retina Keypoint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, June 2012. Open Source Award Winner. 168

A. Albarelli, E. Rodolá, and A. Torsello. Robust Camera Calibration using Inaccurate Targets. In *Proceedings of the British Machine Vision Conference (BMVC)*, Aberystwyth, UK, September 2010. 78, 79, 90, 101, 102, 103, 105

P. F. Alcantarilla, A. Bartoli, and A. J. Davison. KAZE Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, October 2012. 168

M. Andreessen. Why Software is Eating the World. *The Wall Street Journal*, August 20 2011. 4

N. Andreff, R. Horaud, and B. Espiau. Robot Hand–Eye Calibration Using Structure–from–Motion. *International Journal of Robotics Research*, 20(3): 228–248, March 2001. 61, 64

ARTtrack2. Advanced Realtime Tracking GmbH.
URL `http://www.ar-tracking.com`. 13, 15, 20, 41, 45, 49, 64, 67, 72, 107, 152, 221, 245, 247, 279

AscTec AutoPilot. Ascending Technologies GmbH.
URL `http://www.asctec.de`. 20, 47, 129

J. P. Barreto. A Unifying Geometric Representation for Central Projection Systems. *Computer Vision and Image Understanding (CVIU)*, 103, 2006. 31

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008. 168, 194

E. Bayro–Corrochano, K. Daniilidis, and G. Sommer. Motor–Algebra for 3D Kinematics: The Case of Hand–Eye Calibration. *Journal for Mathematical Imaging and Vision*, 13(2):79–100, October 2000. 61, 64

P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 163, 251

S. Birchfield. KLT: Kanade-Lucas-Tomasi Feature Tracker. Dept. of Electrical and Computer Engineering, Clemson University, Clemson, SC, USA. URL `http://www.ces.clemson.edu/~stb/klt/`. 177

F. Blais. Review of 20 Years of Range Sensor Development. *Journal of Electronic Imaging*, 13(1):231–240, January 2004. 8

T. Bodenmüller. *Streaming Surface Reconstruction from Real Time 3D Measurements.* PhD thesis, Institute for Real-Time Computer Systems, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, Munich, Germany, October 2009. 5, 45, 159, 199, 209

T. Bodenmüller and G. Hirzinger. Online Surface Reconstruction from Unorganized 3-D Points for the DLR Hand-Guided Scanner System. In *Proceedings of the 2nd Symposium on 3D Data Processing, Visualization, and Transmission*, Thessaloniki, Greece, 2004. 259

T. Bodenmüller, W. Sepp, M. Suppa, and G. Hirzinger. Tackling Multisensory 3D Data Acquisition and Fusion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2180–2185, San Diego, CA, USA, October 2007. 5, 49, 161, 208, 260

M. Born, E. Wolf, and A. B. Bhatia. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light.* Cambridge University Press, ISBN: 0521642221, seventh, illustrated, revised edition, 1999. 27

C. Borst *et al.* Rollin' Justin - Mobile Platform with Variable Base. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009. Video contribution, best video award. 18, 102, 139, 159, 220

J.-Y. Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker. Description of the Algorithm. Technical report, Microprocessor Research Labs, Intel Corporation, Santa Clara, CA, USA, 2000. 178, 184

J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. Computer Vision Research Group, California Institute of Technology (Caltech), Pasadena, CA, USA, 2002. URL `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`. 109, 113

J. M. Brady. Seeds of Perception. In *Proceedings of the Alvey Vision Conference*, pages 259–265, Cambridge, UK, September 1987. 167

S. S. Brandt and J. Kannala. A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, August 2006. 107

C. Brauer-Burchardt and K. Voss. A New Algorithm to Correct Fish-Eye and Strong Wide Angle Lens Distrotion from Single Images. In *Proceedings of the IEEE International Conference on Image Processing*, Thessaloniki, Greece, 2001. 31

J. Brembeck, L. M. Ho, A. Schaub, C. Satzger, and G. Hirzinger. ROMO - the Robotic Electric Vehicle. In *22nd International Symposium on Dynamics of Vehicle on Roads and Tracks*. IAVSD, 2011. 239

D. C. Brown. Decentering Distortion of Lenses. *Photogrammetric Engineering and Remote Sensing*, 32(3):444–462, May 1966. 28, 32, 34

D. C. Brown. Close-Range Camera Calibration. *Photogrammetric Engineering and Remote Sensing*, 37(8):855–866, 1971. 32, 54, 59

D. Burschka and G. D. Hager. V-GPS – Image-Based Control for 3D Guidance Systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1789–1795, Las Vegas, NV, USA, October 2003. 175, 188, 208, 258

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 778–792, Crete, Greece, 2010. 168

G. Carrera, A. Angeli, and A. J. Davison. SLAM-Based Automatic Extrinsic Calibration of a Multi-Camera Rig. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2652–2659, Shanghai, China, May 2011. 241

C. H. Chen and A. C. Kak. Modelling and Calibration of a Structured Light Scanner for 3D Robot Vision. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 807–815, Raleigh, NC, USA, 1987. 121

F. Chen, G. M. Brown, and M. Song. Overview of Three-Dimensional Shape Measurement Using Optical Methods. *Optical Engineering*, 39(1):10–22, January 2000. 8, 48

H. H. Chen. A Screw Motion Approach to Uniqueness Analysis of Head–Eye Geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 145–151, Maui, Hawaii, June 1991. 61, 63, 64

Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate. *ACM Trans. Math. Softw.*, 35(3):22:1–22:14, Oct. 2008. ISSN 0098-3500. 198

Y. Cheng, M. W. Maimone, and L. Matthies. Visual Odometry on the Mars Exploration Rovers. *IEEE Robotics and Automation Magazine*, 13(2):54 –62, June 2006. 169, 189, 208

M. Chli and A. J. Davison. Active Matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 72–85, Marseille, France, October 2008. 166, 174, 179, 181, 182

M. Chli and A. J. Davison. Active Matching for Visual Tracking. *Robotics and Autonomous Systems*, 57(12):1173–1187, 2009a. 168, 181, 182, 183, 184

M. Chli and A. J. Davison. Automatically and Efficiently Inferring the Hierarchical Structure of Visual Maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 387–394, Kobe, Japan, May 2009b. 184

J. Chou and M. Kamel. Finding the Position and Orientation of a Sensor on a Robot Manipulator Using Quaternions. *International Journal of Robotics Research*, 10(3):240–254, June 1991. 61, 64

S. Christy and R. Horaud. Euclidean Shape and Motion from Multiple Perspective Views by Affine Iterations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1098–1104, 1996. ISSN 0162-8828. 108

J. Civera, D. R. Bueno, A. J. Davison, and J. M. M. Montiel. Camera Self-Calibration for Sequential Bayesian Structure from Motion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3411–3416, Kobe, Japan, May 2009. Best vision paper finalist. 53, 59

D. Claus and A. W. Fitzgibbon. A Rational Function Lens Distortion Model for General Cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 213–219, San Diego, CA, USA, June 2005. 31

B. Clipp, J. Lim, J.-M. Frahm, and M. Pollefeys. Parallel, Real-Time Visual SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3961–3968, Taipei, Taiwan, October 2010. 174, 199

A. Comport, M. Meilland, and P. Rives. An Asymmetric Real-Time Dense Visual Localisation and Mapping System. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1st International Workshop on Live Dense Reconstruction from Moving Cameras*, Barcelona, Spain, November 2011. 169

A. E. Conrady. Decentered Lens Systems. *Monthly Notices Royal Astronomical Society*, 79:384–390, 1919. 32

A. E. Conrady. *Applied Optics and Optical Design, Part 1*. Dover Publications, 1958. 28, 29, 30, 32

A. E. Conrady. *Applied Optics and Optical Design, Part 2*. Dover Publications, 1960. 29, 32

B. Coudrin, M. Devy, J.-J. Orteu, and L. Brèthes. An Innovative Hand-Held Vision-Based Digitizing System for 3D Modelling. *Optics and Lasers in Engineering*, 49(910):1168 – 1176, 2011. ISSN 0143-8166. 163

B. Curless. *New Methods for Surface Reconstruction from Range Images*. PhD thesis, Stanford University, June 1997. Technical Report CSL-TR-97-733. 8, 147

B. Curless and M. Levoy. Better Optical Triangulation through Spacetime Analysis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 987–994, Cambridge, MA, June 1995. 121

N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, San Diego, CA, USA, June 2005. 168

K. Daniilidis. Hand–Eye Calibration Using Dual Quaternions. *International Journal of Robotics Research*, 18(3):286–298, June 1999. 61, 64, 66, 69

DAVID-Laserscanner. DAVID Vision Systems GmbH.
URL `http://www.david-laserscanner.com`. 165

A. J. Davison. *Mobile Robot Navigation Using Active Vision*. PhD thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, UK, October 1999. 108, 171

A. J. Davison. Active Search for Real-Time Vision. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 66–73, Nice, France, October 2005. 162, 181

A. J. Davison and D. W. Murray. Simultaneous Localization and Map-Building Using Active Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, July 2002. ISSN 0162-8828. 168

F. W. DePiero and M. M. Trivedi. 3-D Computer Vision Using Structured Light: Design, Calibration, and Implementation Issues. In M. V. Zelkowitz, editor, *Advances in Computers*, volume 43, pages 243–278. Elsevier, 1996. 121

F. Devernay and O. D. Faugeras. Automatic Calibration and Removal of Distortion From Scenes of Structured Environments. In *In SPIE, volume 2567*, pages 62–72, 1995. 30

F. Devernay and O. D. Faugeras. Straight Lines Have to be Straight: Automatic Calibration and Removal of Distortion from Scenes of Structured Enviroments. *Machine Vision and Applications*, 13(1):14–24, August 2001. 30, 31, 59, 114

DLR Lightweight Robot III. Deutsches Zentrum für Luft- und Raumfahrt and KUKA Roboter GmbH. URL `http://www.robotic.dlr.de/LBR`. 13, 15, 20, 49, 60, 159

F. Dornaika and R. Horaud. Simultaneous Robot–World and Hand–Eye Calibration. *IEEE Transactions on Robotics and Automation*, 14(4):617–622, August 1998. 62, 63, 65, 69

E. Eade and T. Drummond. Monocular SLAM as a Graph of Coalesced Observations. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007. 172

H. Ebner. Self Calibrating Block Adjustment. In *XIII. Congress of the International Society for Photogrammetry*, Helsinki, Finland, 1976. Invited paper. 31

M. T. El Melegy and A. A. Farag. Nonmetric Lens Distortion Calibration: Closed-form Solutions, Robust Estimation and Model Selection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 1, pages 554–559, Nice, France, 2003. ISBN 0-7695-1950-4. 30, 59

J. F. Seara, K. H. Strobl, E. Martin, and G. Schmidt. Task-oriented and Situation-Dependent Gaze Control for Vision Guided Autonomous Walking. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 1–23, Munich and Karlsruhe, Germany, October 2003. 108, 301

W. Faig. Calibration of Close-Range Photogrammetry Systems: Mathematical Formulation. *Photogrammetric Engineering and Remote Sensing*, 41:1479–1486, 1975. 54

FaroArm Gold. FARO Technologies Inc. URL `http://www.faro.com/`. 13, 15, 20, 41, 45, 49, 60, 64, 67, 152

I. Fassi and G. Legnani. Hand to Sensor Calibration: A Geometrical Interpretation of the Matrix Equation AX = XB. *Journal of Robotic Systems*, 22(9): 497–506, 2005. 61, 64

O. D. Faugeras. *Three-Dimensional Computer Vision*. Artificial Intelligence. MIT Press, Cambridge, MA, USA, 1993. ISBN 0-262-06158-9. 77

O. D. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images*. The MIT Press, Cambridge, MA, USA, 2004. 53, 57

O. D. Faugeras and G. Toscani. Camera Calibration for 3D Computer Vision. In *Proceedings of the International Workshop on Machine Vision and Machine Intelligence*, pages 240–247, Tokyo, Japan, February 1987. 26, 31, 54, 77, 109

O. D. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. The MIT Press, ISBN: 0262062208, 2001. 25, 108

A. W. Fitzgibbon. Simultaneous Linear Estimation of Multiple View Geometry and Lens Distortion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, Kauai, Hawaii, USA, December 2001. 31

M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka. Optimization Based IMU Camera Calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3297–3304, San Francisco, CA, USA, Sept. 2011. 129, 162, 181

J. G. Fryer, T. A. Clarke, and J. Chen. Lens Distortion for Simple 'C' Mount Lenses. *International Archives of Photogrammetry and Remote Sensing*, 30: 97–101, 1994. 59

J. G. Fryer; Duane C. Brown. Lens Distortion for Close-Range Photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 52:51–58, Jan. 1986. ISSN 0099-1112. 28, 59

K. A. Gavaghan, M. Peterhans, T. Oliveira-Santos, and S. Weber. A Portable Image Overlay Projection Device for Computer-Aided Open Liver Surgery. *IEEE Transactions on Biomedical Engineering*, 58(6), June 2011. 246

J. A. Grunert. Das Pothenotische Problem in erweiterter Gestalt; nebst Bemerkungen über seine Anwendungen in der Geodäsie. In *Grunerts Archiv für Mathematik und Physik*, volume 1, pp. 238–248. 1841. 170, 183, 195

J. Guivant and E. Nebot. Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001. 172

F. Hacker, J. Dietrich, and G. Hirzinger. A Laser-Triangulation Based Miniaturized 2-D Range-Scanner as Integral Part of a Multisensory Robot-Gripper. In *EOS Topical Metting on Optoelectronic Distance/Displacement Measurements and Applications*, Nantes, France, 1997. 19, 43, 44, 45, 126, 260

HandyScan 3D. Creaform Inc. URL `http://www.creaform3d.com`. 38, 165

R. M. Haralick. Propagating Covariance in Computer Vision. In *Theoretical Foundations of Computer Vision*, pages 95–114, 1998. 140, 152

R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem. *International Journal of Computer Vision*, 13(3):331–356, Dec. 1994. ISSN 0920-5691. 170

C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proceedings of the Alvey Vision Conference*, pages 147–151, Manchester, UK, August 1988. 168

R. I. Hartley. In Defense of the Eight-Point Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997. ISSN 0162-8828. 66, 218

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition, 2004. 25, 108

P. Hébert. A Self-Referenced Hand-Held Range Sensor. In *3-D Digital Imaging and Modeling 3DIM*, pages 5–12, Quebec City, Que., Canada, May 2001. 165

E. Hecht. *Optics.* Addison Wesley, third edition, 1998. 27, 29

J. Heikkilä. Geometric Camera Calibration Using Circular Control Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10): 1066–1077, 2000. 30

J. Heikkilä and O. Silvén. A Four-step Camera Calibration Procedure with Implicit Image Correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1112, San Juan, Puerto Rico, 1997. 30

P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D Mapping: Using Kinect-Style Depth Cameras for Dense 3D Modeling of Indoor Environments. *International Journal of Robotics Research*, 31(5):647–663, 2012. 163, 169

U. Hillenbrand. Pose Clustering from Stereo Data. In L. Iocchi and D. G. Sorrenti, editors, *International Conference on Computer Vision Theory and Applications (VISAPP), International Workshop on Robotic Perception – RoboPerc*, pages 23–32, Funchal, Madeira, Portugal, 2008. 159

A. Hilton and J. Illingworth. Geometric Fusion for a Hand-Held 3D Sensor. *Machine Vision and Applications*, 12(1):44–51, 2000. 10

H. Hirschmüller. Stereo Processing by Semi-Global Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, February 2008. 19, 21, 102, 103, 135, 138, 160, 224

D. Hockney. *Secret Knowledge: Rediscovering the Lost Techniques of the Old Masters.* Studio. ISBN: 0142005126, expanded edition, 2006. 22

R. Horaud and F. Dornaika. Hand–Eye Calibration. *International Journal of Robotics Research*, 14(3):195–210, June 1995. 12, 61, 64, 65, 69, 72, 76

M. Irani and P. Anandan. All About Direct Methods. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), Workshop on Vision Algorithms*, pages 267–277, Corfu, Greece, September 1999. 167

F. A. Jenkins and H. E. White. *Fundamentals of Optics.* McGraw-Hill, fourth edition, 1976. 29

O. Jokinen. Self-Calibration of a Light Striping System by Matching Multiple 3-D Profile Maps. In *Proceedings Second International Conference on 3-D Digital Imaging and Modeling 3DIM'99*, pages 180–190, Ottawa, Canada, October 1999. IEEE Computer Society Press. 121, 122

S. J. Julier and J. K. Uhlmann. A Counter Example to the Theory of Simultaneous Localization and Map Building. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4238–4243, Seoul, Korea, May 2001. 171, 191

M. Kaess and F. Dellaert. Covariance Recovery from a Square Root Information Matrix for Data Association. *Robotics and Autonomous Systems*, 57(12): 1198–1210, 2009. 183, 187

G. Kamberova. Understanding the Systematic and Random Errors in Video Sensor Data. Technical report, GRASP Lab., Department of Computer and Information Science, University of Pennsylvania, 1997. 142

KAMRA Inlays$^{\text{TM}}$. AcuFocus, Inc. URL `http://kamrainlay.de/`. 28

S. B. Kang. Radial Distortion Snakes. *IEICE Transactions on Information & Systems*, E84-D(12):1603–1611, 2001. 59

G. Karypis and V. Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*, 20 (1):359–392, 1999. 198

D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette. Visual Servoing in Robotics Scheme Using a Camera/Laser-Stripe Sensor. *IEEE Transactions on Robotics and Automation*, 12(5):743–750, October 1996. 121

R. Khoury. An Enhanced Positioning Algorithm for a Self-Referencing Hand-Held 3D Sensor. In *3rd Canadian Conference on Computer and Robot Vision CRV*, pages 44–50, Quebec City, Que., Canada, June 2006. 165

S. Kielhöfer. Fehleranalyse und Modellierung eines 3D-Laserscansystems. Master's thesis, Lehrstuhl für Feingerätebau und Mikrotechnik, Fakultät für Maschinenwesen, Technische Universität München, 2003. 19, 43, 45, 126, 155, 156, 157

E. Kilpelä. Compensation of Systematic Errors of Image and Model Coordinates. *International Archives of Photogrammetry*, XXIII(B9):407–427, 1980. 31

M. Kimura, M. Mochimaru, and T. Kanade. Projector Calibration using Arbitrary Planes and Calibrated Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, Minnesota, USA, June 2007. 246

Kinect$^{\text{TM}}$. Microsoft Corporation and PrimeSense.
URL `http://www.microsoft.com/en-us/kinectforwindows/`. 159

R. Kingslake. *Optics in Photography*. SPIE, Bellingham, WA, USA, 1992. 27, 32

G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan, November 2007. 170, 171, 173, 191

G. Konecny and G. Lehmann. *Photogrammetrie*. Walter De Gruyter Inc., Berlin, New York, 4 edition, 1985. 31

R. Konietschke. *Planning of Workplaces with Multiple Kinematically Redundant Robots*. PhD thesis, Institute for Real-Time Computer Systems, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, Munich, Germany, 2008. 244

K. Konolige. Sparse Sparse Bundle Adjustment. In *Proceedings of the British Machine Vision Conference (BMVC)*, Aberystwyth, Wales, 8 2010. 198

K. Konolige and M. Agrawal. FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, October 2008. 170, 173, 199

K. Konolige, M. Agrawal, and J. Solà. Large Scale Visual Odometry for Rough Terrain. In *Proceedings of the International Symposium on Research in Robotics (ISRR)*, Hiroshima, Japan, November 2007. 169, 170

S. Kriegel, C. Rink, T. Bodenmüller, A. Narr, M. Suppa, and G. Hirzinger. Next-Best-Scan Planning for Autonomous 3D Modeling. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2850–2856, Villamoura, Portugal, October 2012. 45, 159

Kuka KR 16. KUKA Roboter GmbH. URL `http://www.kuka-robotics.com/`. xxi, 13, 15, 20, 41, 45, 49, 60, 84, 88, 107, 131, 152, 159, 208, 228, 279

F. Lange, K. H. Strobl, J. Langwald, S. Jörg, G. Hirzinger, B. Gruber, J. Klein, and J. Werner. Kameragestützte Montage von Rädern an kontinuierlich bewegte Fahrzeuge. In *VDI-Berichte 2012, Robotik 2008*, pages 155–158, Munich, Germany, June 2008. In German. 250, 301

J.-M. Lavest, M. Viala, and M. Dhome. Do We Really Need an Accurate Calibration Pattern to Achieve a Reliable Camera Calibration? In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 1, pages 158–174, Freiburg, Germany, June 1998. 78, 79, 90, 91

J. J. Leonard and H. J. S. Feder. A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization. In *International Symposium of Robotics Research*, pages 316–321, Snowbird, UT, USA, October 2000. 172

S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 168

D. Liebowitz and A. Zisserman. Combining Scene and Auto-calibration Constraints. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 293–300, Kerkyra, Greece, Sep 1999. 53

J. Lim, J.-M. Frahm, and M. Pollefeys. Online Environment Mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3489–3496, Colorado Springs, CO, USA, June 2011. 174, 199

O. Lorch, J. F. Seara, K. H. Strobl, U. D. Hanebeck, and G. Schmidt. Perception Errors in Vision Guided Walking: Analysis, Modeling, and Filtering. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2048–2053, Washington DC, USA, May 2002. 113, 301

M. I. A. Lourakis. Sparse Non-Linear Least Squares Optimization for Geometric Vision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 2, pages 43–56, Crete, Greece, 2010. 198

M. I. A. Lourakis. levmar: Levenberg-Marquardt Nonlinear Least Squares Algorithms in C/C++, Jul. 2004.
URL `http://www.ics.forth.gr/~lourakis/levmar/`. 193

D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, Corfu, Greece, September 1999. 168

F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333–349, 1997. 173

Y.-C. Lu and J. C. Chou. Eight–Space Quaternion Approach for Robotic Hand–Eye Calibration. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 3316–3321, Vancouver, BC, Canada, October 1995. 61, 64

T. Luhmann, S. Robson, S. Kyle, and I. Harley. *Close-Range Photogrammetry.* Whittles Publishing, 2006. 28

L. Ma, Y. Chen, and K. L. Moore. A New Analytical Radial Distortion Model for Camera Calibration. In *Comput. Res. Repos., Article No.* 0307046, 2003. 32

L. Ma, Y. Chen, and K. L. Moore. Flexible Camera Calibration Using a New Analytical Radial Undistortion Formula with Application to Mobile Robot Localization. In *IEEE International Symposium on Intelligent Control*, pages 799–804, 2003. 32

D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003. available: `http://www.inference.phy.cam.ac.uk/mackay/itila/`. 64, 72

A. A. Magill. Variation in Distortion with Magnification. *Journal of the Optical Society of America*, 45(3):148–149, 1955. 28

E. Mair, K. H. Strobl, M. Suppa, and D. Burschka. Efficient Camera-Based Pose Estimation for Real-Time Applications. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2696–2703, St. Louis, MO, USA, October 2009. 184, 258, 301

E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 183–196, Crete, Greece, 2010a. 168

E. Mair, K. H. Strobl, T. Bodenmüller, M. Suppa, and D. Burschka. Real-time Image-based Localization for Hand-held 3D-modeling. *KI – Künstliche Intelligenz*, 24(3):207–214, May 2010b. 15, 177, 258, 301

J. Mallon and P. F. Whelan. Precise radial un-distortion of images. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 1, pages 18–21, Cambridge, UK, 2004. 30

J. Mallon and P. F. Whelan. Which Pattern? Biasing Aspects of Planar Calibration Patterns and Detection Methods. *Pattern Recognition Letters*, 28 (8):921–930, June 2007. 53, 56, 91, 98

H. Malm. *Studies in Robotic Vision, Optical Illusions and Nonlinear Diffusion Filtering.* PhD thesis, Centre for Mathematical Sciences, Lund Institute of Technology, Lund University, Sweden, 2003. 62, 254

H. Malm and A. Heyden. Stereo Head Calibration from a Planar Object. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 657–662, Kauai, Hawaii, USA, December 2001. 54, 57, 59, 81

D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H.Freeman & Co Ltd., San Francisco, CA, USA, 1982. 168, 177

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *Proceedings of the British Machine Vision Conference (BMVC)*, Cardiff, UK, September 2002. Best paper prize. 168

L. Matthies. Toward Stochastic Modeling of Obstacle Detectability in Passive Stereo Range Imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 765–768, June 1992. 140

L. Matthies and S. A. Shafer. Error Modeling in Stereo Navigation. *IEEE Journal of Robotics and Automation*, 3(3):239–248, June 1987. 53

C. McGlone, E. Mikhail, and J. Bethel (Eds.). *Manual of Photogrammetry.* ASPRS, Bethesda, MD, USA. ISBN: 1-57083-071-1, fifth edition, 2004. 25

A. M. McIvor. Calibration of a Laser Stripe Profiler. In *Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling*, pages 92–98, October 1999. 121

A. M. McIvor. Nonlinear Calibration of a Laser Stripe Profiler. *Optical Engineering*, 41(1):205–212, January 2002. 121

C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. A Constant Time Efficient Stereo SLAM System. In *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK, September 2009. 170

C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. RSLAM: A System for Large-Scale Mapping in Constant-Time using Stereo. *International Journal of Computer Vision*, 94:198–214, 2010. Special issue of BMVC. 174, 199

T. Melen. *Geometrical modelling and calibration of video cameras for underwater navigation.* PhD thesis, Norwegian University of Science and Technology, Trondhein, Norway, 1994. 30

K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10): 1615–1630, Oct. 2005. ISSN 0162-8828. 168

M. Montemerlo and S. Thrun. *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics.* Series: Springer Tracts in Advanced Robotics, Vol. 27. Springer Berlin Heidelberg, 2007. 172

D. C. Moore, A. S. Huang, M. Walter, E. Olson, L. Fletcher, J. Leonard, and S. Teller. Simultaneous Local and Global State Estimation for Robotic Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3683–3688, Kobe, Japan, May 2009. 189

E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real Time Localization and 3D Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 363–370, New York, NY, USA, 2006. 170, 173, 191

E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and Real-Time Structure from Motion Using Local Bundle Adjustment. *Image and Vision Computing*, 27(8):1178–1193, 2009. ISSN 0262-8856. 170

J. Neira and J. D. Tardós. Data Association in Stochastic Mapping Using the Joint Compatibility Test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, December 2001. 168, 182, 187

J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *Computer Journal*, 7:308–313, 1965. 123

R. A. Newcombe and A. J. Davison. Live Dense Reconstruction with a Single Moving Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1498–1505, San Francisco, CA, USA, June 2010. 166

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, Basel, Switzerland, October 2011a. Best paper award. 163

R. A. Newcombe, S. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2320–2327, Barcelona, Spain, November 2011b. Best demo award. 166, 169

D. Nistér. Preemptive ransac for live structure and motion estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 199–206, Nice, France, October 2003. 170

D. Nistér. A Minimal Solution to the Generalised 3-Point Pose Problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 560–567, Washington, DC, USA, 2004a. 170

D. Nistér. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004b. 170

D. Nistér, O. Naroditsky, and J. R. Bergen. Visual Odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–659, Washington, DC, USA, 2004. 169, 170, 208

D. Nistér, O. Naroditsky, and J. Bergen. Visual Odometry for Ground Vehicle Applications. *Journal of Field Robotics*, 23, 2006. 170, 189, 192

F. C. Park and B. J. Martin. Robot Sensor Calibration – Solving AX = XB on the Euclidean Group. *IEEE Transactions on Robotics and Automation*, 10 (5):717–721, October 1994. 61, 64

C. Perwass and G. Sommer. The Inversion Camera Model. In *28. Symposium für Mustererkennung, DAGM 2006, Berlin, 12.-14.09.2006*, number 4174 in LNCS, pages 647–656. Springer-Verlag, Berlin, Heidelberg, 2006. 31

M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual Modeling with a Hand-Held Camera. *International Journal of Computer Vision*, 59(3):207–232, 2004. 166

B. Prescott and G. F. McLean. Line-Based Correction of Radial Lens Distortion. *Graphical Models and Image Processing*, 59(1):39–47, 1997. ISSN 1077-3169. 30, 59

G. Qian and R. Chellappa. Structure from Motion Using Sequential Monte Carlo Methods. *International Journal of Computer Vision*, 59(1):5–31, 2004. 172

N. Qiu and S. D. Ma. The Nonparametric Approach for Camera Calibration. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 224–229, June 1995. 32

I. D. Reid. Projective Calibration of a Laser-Stripe Range Finder. *Image and Vision Computing*, 14(9):659 – 666, 1996. ISSN 0262-8856. 121

F. Remondino and C. Fraser. Digital Camera Calibration Methods: Considerations and Comparisons. In *Proceedings of the ISPRS Commission V Symposium, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS), Vol. XXXVI, part 5*, pages 266–272, Dresden, Germany, September 2006. 53

S. Rémy, M. Dhome, J. M. Lavest, and N. Daucher. Hand–Eye Calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1057–1065, Grenoble, France, September 1997. 62, 63

E. Rosten and T. Drummond. Fusing Points and Lines for High Performance Tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1508–1511, Nice, France, October 2005. 168

G. Roth and A. Whitehead. Using Projective Vision to Find Camera Positions in an Image Sequence. In *Vision Interface VI'2000*, pages 87–94, Montreal, Canada, May 2000. 166

R. Sagawa, M. Takatsuji, T. Echigo, and Y. Yagi. Calibration of Lens Distortion by Structured-Light Scanning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 832–837, Edmonton, Canada, August 2005. 30

J. Salvi, J. Pagès, and J. Batlle. Pattern Codification Strategies in Structured Light Systems. *Pattern Recognition*, 37(4):827 – 849, 2004. ISSN 0031-3203. 8

J. A. Sánchez, E. A. Destefanis, and L. R. Canali. Plane-based Camera Calibration without Direct Optimization Algorithms. In *IV Jornadas Argentinas de Robótica*, Córdoba, Argentina, November 2006. 58, 81

D. Scharstein and R. Szeliski. Middlebury Stereo Vision Page. Department of Computer Science, Middlebury College, Middlebury, VT, USA, 2002. URL `http://vision.middlebury.edu/stereo/`. 21

J. Schmidt, F. Vogt, and H. Niemann. Calibration–Free Hand–Eye Calibration: A Structure-from-Motion Approach. In *Pattern Recognition, 27th DAGM Symposium*, pages 67–74, Wien, 2005. 72

A. Schwier, R. Konietschke, T. Bodenmüller, T. Ende, S. Kielhöfer, and G. Hirzinger. VR-Map: A New Device for Patient Registration and Optimal Robot Positioning. In *9. Jahrestagung der Deutschen Gesellschaft fr Computer- und Roboterassistierte Chirurgie (CURAC)*, Düsseldorf, Germany, November 2010. 245, 251

S. Se and P. Jasiobedzki. Stereo-Vision Based 3D Modeling and Localization for Unmanned Vehicles. *International Journal of Intelligent Control and Systems, Special Issue on Field Robotics and Intelligent Systems*, 13(1):47–58, March 2008. 166

W. Sepp. *Visual Servoing of Textured Free-Form Objects in 6 Degrees of Freedom.* PhD thesis, Institute for Data Processing, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, Munich, Germany, July 2008. 159

J. Shi and C. Tomasi. Good Features to Track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, Jerusalem, Israel, October 1994. 168, 177, 178, 236

Y. C. Shiu and S. Ahmad. Calibration of Wrist–Mounted Robotic Sensors by Solving Homogeneous Tranform Equations of the Form AX = XB. *IEEE Transactions on Robotics and Automation*, 5(1):16–29, February 1989. 61, 62, 64

G. Sibley, C. Mei, I. Reid, and P. Newman. Adaptive Relative Bundle Adjustment. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009. 174, 199

R. Sim and J. Little. Autonomous Vision-Based Exploration and Mapping Using Hybrid Maps and Rao-Blackwellised Particle Filters. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2082–2089, Beijing, China, October 2006. 172

J. Sladczyk. Entwicklung eines Laser-Lichtschnittsystems zur Erfassung von 3D-Geometrien. Master's thesis, Institut für Roboterforschung, Fachbereich Informatik, Technische Universität Dortmund, 2008. 147

S. M. Smith and J. M. Brady. SUSAN-A New Approach to Low Level Image Processing. *International Journal of Computer Vision*, 23:45–78, 1997. ISSN 0920-5691. 168

J. Solà. Multi-Camera VSLAM: from Former Information Losses to Self-Calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, October 2007. in Workshop on visual SLAM. 183

S. Sollinger. Kalibrierung und Implementierung eines Laserprojektors als Autopointer. Master's thesis, Fakultät für Maschinenbau, Hochschule Regensburg, 2012. 248

G. P. Stein. Internal Camera Calibration Using Rotation and Geometric Shapes. In *AITR-1426, Master's Thesis, Massachussets Institute of Technology, Artificial Intelligence Laboratory*, 1993. 30, 33, 59

G. P. Stein. Lens Distortion Calibration Using Point Correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Juan, Puerto Rico, 1997. 30

H. Stewénius, C. Engels, and D. Nistér. Recent Developments on Direct Relative Orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60: 284–294, June 2006. 170

H. Strasdat, J. M. M. Montiel, and A. Davison. Scale Drift-Aware Large Scale Monocular SLAM. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010a. 173, 197, 199

H. Strasdat, J. M. M. Montiel, and A. J. Davison. Real-Time Monocular SLAM: Why Filter? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2657–2664, 2010b. Best vision paper award. 172, 173, 190, 191, 201

H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige. Double Window Optimisation for Constant Time Visual SLAM. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2352–2359, Barcelona, Spain, November 2011. 170, 174, 199

K. H. Strobl and G. Hirzinger. Optimal Hand-Eye Calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4647–4653, Beijing, China, October 2006. 5, 12, 49, 55, 76, 82, 113, 118, 120, 133, 161, 208, 212, 254, 278, 301

K. H. Strobl and G. Hirzinger. More Accurate Camera and Hand-Eye Calibrations with Unknown Grid Pattern Dimensions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1398–1405, Pasadena, CA, USA, May 2008. 12, 31, 54, 78, 79, 90, 95, 105, 107, 113, 133, 212, 254, 276, 301

K. H. Strobl and G. Hirzinger. More Accurate Pinhole Camera Calibration with Imperfect Planar Target. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1st IEEE Workshop on Challenges and Opportunities in Robot Perception*, pages 1068–1075, Barcelona, Spain, November 2011. 12, 54, 79, 105, 107, 113, 133, 212, 254, 276, 301

K. H. Strobl, W. Sepp, E. Wahl, T. Bodenmüller, M. Suppa, J. F. Seara, and G. Hirzinger. The DLR Multisensory Hand-Guided Device: The Laser Stripe Profiler. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1927–1932, New Orleans, LA, USA, April 2004. 13, 14, 19, 36, 122, 125, 133, 135, 141, 160, 205, 209, 213, 258, 301

K. H. Strobl, W. Sepp, S. Fuchs, C. Paredes, M. Smíšek, and K. Arbter. DLR CalDe and DLR CalLab. Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany, 2005.
URL `http://www.robotic.dlr.de/callab/`. 6, 13, 16, 31, 56, 68, 72, 91, 94, 95, 98, 105, 108, 109, 113, 133, 192, 208, 212, 213, 219, 221, 255, 275, 301

K. H. Strobl, E. Mair, T. Bodenmüller, S. Kielhöfer, W. Sepp, M. Suppa, D. Burschka, and G. Hirzinger. The Self-Referenced DLR 3D-Modeler. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 21–28, St. Louis, MO, USA, October 2009a. Best paper finalist. 10, 15, 162, 166, 177, 201, 237, 258, 301

K. H. Strobl, W. Sepp, and G. Hirzinger. On the Issue of Camera Calibration with Narrow Angular Field of View. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 309–315, St. Louis, MO, USA, October 2009b. 12, 18, 120, 224, 275, 301

K. H. Strobl, E. Mair, and G. Hirzinger. Image-Based Pose Estimation for 3-D Modeling in Rapid, Hand-Held Motion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2593–2600, Shanghai, China, May 2011. 15, 162, 166, 168, 183, 201, 205, 206, 258, 301

L. Stroebel. *View Camera Technique.* Focal Press, Boston, MA, USA, seventh edition, 1999. 27

P. F. Sturm and S. J. Maybank. On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 432–437, Fort Collins, USA, June 1999. 53, 54, 55, 56, 57, 59, 78, 79, 80, 90, 91, 107, 113, 133, 278

W. Sun and J. R. Cooperstock. An Empirical Evaluation of Factors Influencing Camera Calibration Accuracy Using Three Publicly Available Techniques. *Machine Vision and Applications*, 17(1):51–67, March 2006. 53, 77, 78

M. Suppa. *Autonomous Robot Work Cell Exploration using Multisensory Eye-in-Hand Systems.* PhD thesis, Institute for Robotics, Fakultät für Maschinenbau, Gottfried Wilhelm Leibniz Universität Hannover, Hanover, Germany, November 2008. 45, 159

M. Suppa and G. Hirzinger. A Novel System Approach to Multisensory Data Acquisition. In *Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS-8)*, pages 996–1012, Amsterdam, Netherlands, March 2004. 126, 128, 257

M. Suppa, P. Wang, K. Gupta, and G. Hirzinger. C-space Exploration Using Noisy Sensor Models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, USA, April 2004.

M. Suppa, S. Kielhöfer, J. Langwald, F. Hacker, K. H. Strobl, and G. Hirzinger. The 3D-Modeller: A Multi-Purpose Vision Platform. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 781–787, Rome, Italy, April 2007. 10, 11, 15, 18, 19, 36, 37, 49, 257, 301

T. Tamaki. Unified Approach to Image Distortion: D-U and U-D Models. *IEICE Transactions on Information & Systems*, E88-D(5):1086–1090, 2005. ISSN 0916-8532. 30

G. Taylor, L. Kleeman, and Å. Wernersson. Robust Colour and Range Sensing for Robotic Applications Using a Stereoscopic Light Stripe Scanner. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 86–91, Lausanne, Switzerland, October 2002. 121

C. Tomasi and T. Kanade. Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991. 177

P. H. S. Torr and A. Zisserman. Feature Based Methods for Structure and Motion Estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), Workshop on Vision Algorithms*, pages 278–294, Corfu, Greece, September 1999. 167

B. Triggs. Autocalibration from Planar Scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 1, pages 89–105, Freiburg, Germany, June 1998. 53, 55

B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle Adjustment – A Modern Synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Kerkyra, Greece, Sep 1999. 173

E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998. ISBN 0132611082. 137

E. Trucco, R. B. Fisher, A. W. Fitzgibbon, and D. K. Naidu. Calibration, Data Consistency and Model Acquisition with a 3-D Laser Striper. *International Journal of Computer Integrated Manufacturing*, 11(4):293–310, 1998. 121, 147

R. Y. Tsai. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 364–374, Miami, Florida, USA, 1986. 77

R. Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987. 26, 30, 33, 53, 54, 55, 77, 109, 114, 132

R. Y. Tsai and R. K. Lenz. A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/eye Calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, June 1989. 61, 62, 63, 64

C.-C. Wang. Extrinsic Calibration of a Vision Sensor Mounted on a Robot. *IEEE Transactions on Robotics and Automation*, 8(2):161–175, April 1992. 61, 62, 63, 64, 67

J. Wang, F. Shi, J. Zhang, and Y. Liu. A New Calibration Model of Camera Lens Distortion. *Pattern Recognition*, 41(2):607–615, 2008. 32

T. Wang, P. F. McLauchlan, P. Palmer, A. Hilton, L. Wang, and L. Wang. Calibration for an Integrated Measurement System of Camera and Laser and its Application. In *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatic*, 2001. 121

G.-Q. Wei and S. De Ma. Implicit and Explicit Camera Calibration: Theory and Experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):469–480, 1994. 30

G.-Q. Wei, K. Arbter, and G. Hirzinger. Active Self–Calibration of Robotic Eyes and Hand–Eye Relationships with Model Identification. *IEEE Transactions on Robotics and Automation*, 14(1):158–166, February 1998. 61, 64

J. Weng, P. Cohen, and M. Herniou. Camera Calibration with Distortion Models and Accuracy Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, 1992. ISSN 0162-8828. 32, 33, 53, 54, 113

B. Williams and I. Reid. On Combining Visual SLAM and Visual Odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, May 2010. 170

R. G. Willson and S. A. Shafer. What is the Center of the Image? *Journal of the Optical Society of America A*, 11(11):16–29, November 1994. 30, 33, 109, 113

W. Wolfe, D. Mathis, C. W. Sklair, and M. Magee. Three Perspective View of Three Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):66–73, 1991. 170

Z. Zhang. On the Epipolar Geometry Between Two Images With Lens Distortion. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume I, pages 407–411, Vienna, Austria, August 1996. 30

Z. Zhang. A Flexible new Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000. 53, 54, 55, 56, 57, 58, 59, 62, 72, 78, 79, 80, 90, 91, 107, 113, 133, 278

H. Zhuang and Z. S. Roth. Comments on 'Calibration of Wrist–Mounted Robotic Sensors by Solving Homogenous Transform Equations of the Form AX = XB'. *IEEE Transactions on Robotics and Automation*, 7(6):877–878, December 1991. 61, 64

H. Zhuang and Y. C. Shiu. A Noise–Tolerant Algorithm for Robotic Hand–Eye Calibration with or without Sensor Orientation Measurement. *IEEE Transactions on Systems, Man and Cybernetics*, 23(4):1168–1175, July 1993. 61, 64

H. Zhuang, Z. S. Roth, and R. Sudhakar. Simultaneous Robot/World and Tool/Flange Calibration by Solving Homogeneous Transformation Equations of the Form AX = YB. *IEEE Transactions on Robotics and Automation*, 10 (4):549–554, August 1994. 62, 63, 254

H. Zhuang, K. Wang, and Z. S. Roth. Simultaneous Calibration of a Robot and a Hand-Mounted Camera. *IEEE Transactions on Robotics and Automation*, 11(5):649–660, October 1995. 62, 254

Annotation

Among the items listed in this bibliography, my publications in the context of the research presented in this thesis are (Lorch *et al.*, 2002), (F. Seara *et al.*, 2003), (Strobl *et al.*, 2004), (Strobl *et al.*, 2005), (Strobl and Hirzinger, 2006), (Suppa *et al.*, 2007), (Lange *et al.*, 2008), (Strobl and Hirzinger, 2008), (Strobl *et al.*, 2009a), (Strobl *et al.*, 2009b), (Mair *et al.*, 2009), (Mair *et al.*, 2010b), (Strobl *et al.*, 2011), and (Strobl and Hirzinger, 2011).