

# **Real-Time Range Imaging for Human-Machine Interfaces**

Frank Forster



Lehrstuhl für Mensch-Maschine-Kommunikation

Technische Universität München

# **Real-Time Range Imaging for Human-Machine Interfaces**

Frank Forster

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

**Doktors der Naturwissenschaften**

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. techn. Josef A. Nossek

Prüfer der Dissertation:

1. Univ.-Prof. Dr. rer. nat. Manfred K. Lang
2. Univ.-Prof. Dr. rer. nat. Bernd Radig
3. Univ.-Prof. Dr.-Ing. Gert Hauske

Die Dissertation wurde am 8.11.2004 bei der Technischen Universität München eingereicht und  
durch die Fakultät für Elektrotechnik und Informationstechnik am 20.06.2005 angenommen.



## Vorwort

Die vorliegende Arbeit ist das Ergebnis meiner Forschungstätigkeit als externer wissenschaftlicher Mitarbeiter am Lehrstuhl für Mensch-Maschine-Kommunikation der Technischen Universität München und als Mitarbeiter der Abteilung CT PS 9 der Siemens AG.

Mein ganz besonderer Dank gilt meinem Doktorvater Professor Manfred Lang, der sich freundlicherweise bereit erklärt hat, die wissenschaftliche Betreuung zu übernehmen. Er war jederzeit für wertvolle Diskussionen und fachlichen Rat verfügbar und schuf damit eine wichtige Grundvoraussetzung für den Erfolg dieser Arbeit. Als sehr fruchtbar erwies sich, dass er mir im Rahmen des Themas viel Freiraum bei der Gestaltung der Dissertation ließ. Auch von seiner Unterstützung beim Besuch internationaler Fachtagungen, teilweise durch seine persönliche Anwesenheit, profitierte diese Arbeit.

Desgleichen möchte ich Herrn Professor Radig danken, der diese Dissertation ebenfalls wissenschaftlich betreute. Trotz seiner vielen Verpflichtungen fand er Zeit, mir beratend zur Seite zu stehen, und trug somit viel zum Gelingen dieser Arbeit bei.

Ein herzliches Dankeschön geht an die Mitarbeiter des Fachzentrums CT PS 9 der Siemens AG, insbesondere an Herrn Rummel, Dr. Doemens und Dr. Laloni, die mich, soweit der stressige Berufsalltag es zuließ, bei meinem Promotionsvorhaben stets unterstützten.

Weiter danke ich allen an dieser Arbeit beteiligten Diplomanden, Praktikanten und Werkstudenten für ihr Engagement.

Abschließend danke ich allen Personen aus meinem persönlichen Umfeld, die mich bei dieser Arbeit unterstützt haben, vor allem natürlich meiner Monika, die leider oft hinter der Promotionsarbeit zurückstehen musste. Vielen Dank für das Verständnis und die Unterstützung!

München, im Juli 2004

Frank Forster



## Abstract

This thesis presents a new approach to automatically acquire accurate high-resolution range images in real-time. While this work focuses on scenes relevant for human-machine communication such as human faces or hands, the proposed technique can be used with arbitrary close-range scenes. Moreover, it is well suited for an integrated 2D-3D vision approach as it provides a color image of the scene along with the range data.

The central part of the presented technique is a color coded light approach with a single static projection pattern. Existing methods using this technique are limited to scenes with neutral or uniform reflectance. This work concludes from an abstract scene model that employing local color edge patterns for encoding is a way to overcome this limitation. It furthermore establishes the properties a coded light projection pattern should ideally have so that an algorithm is able to reliably demodulate and decode it from an image. From these theoretical considerations it derives a corresponding new type of projection pattern that also permits a high lateral resolution of the range data. It proves that this pattern type permits detecting if the reflection of a local color edge pattern is disrupted in virtually all practically relevant cases. Next, it introduces an algorithm that exploits the properties of the projection pattern to robustly convert a color image of a scene illuminated by such a pattern into a range image. It finally describes a pseudo-random approach to generate the necessary complex color edge patterns.

The proposed coded light approach works well with most scenes, but has certain intrinsic weaknesses, e.g. at surface singularities. The thesis shows that stereo algorithms are typically well suited for obtaining range values for the parts of the scene where the coded light step fails; also that such algorithms are capable of operating in real-time in this case because these problematic regions tend to make up only a small percentage of the scene. A corresponding stereo algorithm that complements the coded light step is presented, yielding a two-stage ranging technique suited for arbitrary scenes.

It is a precondition for range image acquisition that both camera and projector are calibrated. This work introduces an approach to camera calibration that extends Tsai's well-known monoview calibration method to one based on several views of a planar calibration target. It describes how the task of projector calibration can be solved with this approach. The experimental results given indicate the technique has certain advantages over comparable state-of-the-art calibration methods and permits the accurate calibration of a coded light, respectively stereo system on the basis of a simple planar target.

The thesis further performs a range error analysis based on a parameterized model of a triangulation-system, including the complex case of a convergent geometric set-up. It develops exact as well as approximate formulas for the error in the measured coordinates as function of these parameters.

A prototype system based on the presented ranging approach, integrated using low-cost off-the-shelf components, is evaluated. Experiments show that it is able to acquire range maps of resolution 780 by 580 at up to 25 frames per second on a standard PC with an exemplary measurement accuracy of 0.2 mm standard deviation over a cubical working space of about 0.5 m side length; also that the method is robust against background illumination and works well with scenes that are strongly colored and textured.

Finally, this thesis describes a face recognition system based on embedded hidden Markov models that works with color/gray level and range data; also a database of 2700 color and corresponding range images of a test population of 20 people acquired with a prototype system of the proposed approach. The evaluation of the face recognition system on the database demonstrates that using range data improves the performance of a standard face recognition technique significantly over its color/gray level only version.





## Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	MOTIVATION AND PURPOSE OF THIS WORK .....	1
1.2	ORGANIZATION OF THIS WORK.....	2
<b>2</b>	<b>FUNDAMENTALS OF RANGE IMAGE ACQUISITION.....</b>	<b>3</b>
2.1	PHYSICS OF RADIATION .....	3
2.1.1	<i>Basic Concepts and Quantities of Radiometry.....</i>	<i>4</i>
2.1.2	<i>The Propagation of Radiation.....</i>	<i>6</i>
2.2	GEOMETRIC CAMERA MODELS.....	9
2.2.1	<i>The Pinhole Camera Model.....</i>	<i>9</i>
2.2.2	<i>The Lens Camera Model.....</i>	<i>10</i>
2.3	THE CAMERA SENSOR MODEL.....	12
2.3.1	<i>Formation and Description of Digital Images.....</i>	<i>12</i>
2.3.2	<i>The Concept of Color, Color Vision and Color Images.....</i>	<i>14</i>
2.4	SUMMARY .....	16
<b>3</b>	<b>STATE OF THE ART IN RANGE IMAGING .....</b>	<b>17</b>
3.1	BASIC RANGING TERMS.....	17
3.2	ASPECTS OF RANGE ACQUISITION SYSTEMS .....	19
3.3	RANGE IMAGING METHODS .....	19
3.3.1	<i>Time-of-Flight Ranging.....</i>	<i>20</i>
3.3.2	<i>Unidirectional Interferometry.....</i>	<i>22</i>
3.3.3	<i>Moiré Interferometry.....</i>	<i>23</i>
3.3.4	<i>Depth-From-Focus and Depth-From-Defocus .....</i>	<i>24</i>
3.3.5	<i>Photometric Stereo.....</i>	<i>25</i>
3.3.6	<i>Shape-from-Shading.....</i>	<i>26</i>
3.3.7	<i>Shape from Motion.....</i>	<i>28</i>
3.3.8	<i>Triangulation .....</i>	<i>30</i>
3.3.9	<i>Stereo Vision .....</i>	<i>31</i>
3.3.9.1	<i>The Principle of Stereo Vision.....</i>	<i>31</i>
3.3.9.2	<i>The Constraints of Stereo Vision.....</i>	<i>33</i>
3.3.9.3	<i>Area-Based Stereo Algorithms .....</i>	<i>34</i>
3.3.9.4	<i>Feature-Based Stereo Algorithms .....</i>	<i>35</i>
3.3.9.5	<i>Cooperative Stereo Algorithms.....</i>	<i>36</i>
3.3.9.6	<i>Global Optimization Stereo Algorithms .....</i>	<i>37</i>
3.3.9.7	<i>Variants of Stereo Vision.....</i>	<i>38</i>
3.3.9.8	<i>Conclusions On Stereo Vision .....</i>	<i>38</i>
3.3.10	<i>Structured Light .....</i>	<i>39</i>
3.3.10.1	<i>The Principle of the Structured Light Approach.....</i>	<i>39</i>
3.3.10.2	<i>The Constraints of Structured Light.....</i>	<i>42</i>
3.3.10.3	<i>Structured Light without Encoding.....</i>	<i>43</i>
3.3.10.4	<i>Coded Light Based on Point-Wise Encoding.....</i>	<i>47</i>
3.3.10.5	<i>Coded Light Based on Temporal Encoding .....</i>	<i>48</i>
3.3.10.6	<i>Coded Light Based on Spatial Gray Level Encoding .....</i>	<i>50</i>
3.3.10.7	<i>Coded Light Based on Spatial Color Encoding .....</i>	<i>52</i>
3.3.10.8	<i>Conclusions on Structured Light.....</i>	<i>54</i>
3.4	SUMMARY .....	54

<b>4</b>	<b>A NEW APPROACH TO RANGE IMAGE ACQUISITION .....</b>	<b>55</b>
4.1	PROBLEM STATEMENT .....	55
4.2	PRINCIPLE AND KEY IDEAS OF THE NEW APPROACH.....	56
4.3	THE CODED LIGHT SUBSYSTEM .....	60
4.3.1	<i>Calibration</i> .....	60
4.3.1.1	Tsai's Camera Calibration Technique.....	60
4.3.1.2	An Improved Version of Tsai's Camera Calibration Technique.....	63
4.3.1.3	Projector Calibration .....	67
4.3.1.4	Experimental Results.....	68
4.3.2	<i>The Choice of Colors</i> .....	75
4.3.3	<i>Encoding – The Projection Pattern</i> .....	75
4.3.3.1	Required Pattern Resolution.....	75
4.3.3.2	Towards Optimal Patterns for Spatially Encoded Light.....	76
4.3.3.3	A New Kind of Projection Pattern .....	77
4.3.3.4	Pattern Generation.....	86
4.3.4	<i>Data Processing</i> .....	89
4.3.4.1	Demodulation and Decoding.....	89
4.3.4.2	Triangulation .....	95
4.3.4.3	Interpolation .....	97
4.4	THE STEREO SUBSYSTEM .....	97
4.4.1	<i>Rectification</i> .....	99
4.4.2	<i>Mutual Update</i> .....	99
4.4.3	<i>Stereo Step</i> .....	100
4.4.3.1	Area-Based Stereo.....	101
4.4.3.2	Feature-Based Stereo.....	102
4.4.4	<i>Dense Correspondence Step</i> .....	102
4.5	OPTIONAL SYSTEM COMPONENTS .....	104
4.5.1	<i>Scene Reflectance Compensation</i> .....	104
4.5.2	<i>Autonomous Threshold Optimization</i> .....	108
4.5.3	<i>Scene Color Estimation</i> .....	109
<b>5</b>	<b>EVALUATION OF A PROTOTYPE SYSTEM.....</b>	<b>111</b>
5.1	A PARAMETERIZED MODEL OF A TRIANGULATION SYSTEM .....	113
5.2	THEORETICAL ANALYSIS OF THE MEASUREMENT ERROR WITH TRIANGULATION .....	115
5.2.1	<i>Definition of Basic Terms</i> .....	115
5.2.2	<i>Causes of Inaccuracy</i> .....	117
5.2.3	<i>Exact Formulas for the Measurement Error</i> .....	118
5.2.4	<i>Simple Estimates for the Measurement Error of a Structured Light System</i> .....	121
5.2.5	<i>Stochastic Analysis of the Measurement Error</i> .....	122
5.2.6	<i>Conclusions Regarding the Accuracy – Set-Up Relationship</i> .....	124
5.3	EXPERIMENTAL EVALUATION OF THE RANGE ERROR .....	125
5.3.1	<i>The Precision or Repeat Accuracy</i> .....	126
5.3.2	<i>Experimental Determination of the Localization Error</i> .....	125
5.3.3	<i>Measured Shape of a Planar Object</i> .....	127
5.3.4	<i>Reconstruction of the Calibration Target</i> .....	129
5.4	EXPERIMENTAL EVALUATION OF THE FRAME RATE .....	131
5.5	EXPERIMENTAL QUALITATIVE EVALUATION: EXEMPLARY SCENES .....	132

<b>6</b>	<b>FACE RECOGNITION: AN EXEMPLARY APPLICATION.....</b>	<b>139</b>
6.1	MOTIVATION FOR 3D FACE RECOGNITION .....	139
6.2	THE METHOD USED FOR FACE RECOGNITION .....	140
6.3	THE FACE DATABASE .....	142
6.4	EVALUATION RESULTS .....	143
<b>7</b>	<b>CONCLUSIONS .....</b>	<b>147</b>
7.1	CONCLUSIONS .....	147
7.2	FUTURE RESEARCH .....	149
7.3	SUMMARY OF CONTRIBUTIONS.....	150
<b>8</b>	<b>REFERENCES.....</b>	<b>153</b>



## List of Most Important Symbols and Abbreviations

$\Delta x_l$	Error in the left image coordinate, respectively localization error (in the context of the structured light approach)
$\Delta x_r$	Error in the right image coordinate, respectively projection error (in the context of the structured light approach)
$\Delta x$	Shorthand for $\Delta x_l - \Delta x_r$
$\delta$	Measurement error or uncertainty as n-dimensional vector
$\delta_{abs}$	Absolute error in a coordinate measurement
$\delta_{rel}$	Relative error in a coordinate measurement
$\delta x, \delta y, \delta z$	Error in a measured x, y or z coordinate, respectively
$\delta(x)$	Dirac impulse
$(\theta, \phi)$	Spherical coordinates, where $\theta$ represents the polar, $\phi$ the azimuth angle
$(\theta_i, \phi_i)$	Spherical coordinates of the illumination direction
$(\theta_r, \phi_r)$	Spherical coordinates of the viewing direction
$\kappa$	Radial distortion coefficient
$\lambda$	Wavelength
$\sigma$	Standard deviation of a random variable, respectively of a sample
$\sigma(k)$	Codeword reading rule or function
$\Phi$	Radiant flux
$\Omega$	Solid angle or space angle
$\mathfrak{R}^2, \mathfrak{R}^3$	Two and three-dimensional Euclidean space
$\mathbf{A}=(a_{ij})$	Camera-projector (color-)coupling matrix of dimensions 3 by 3
$b$	Image plane distance; in the context of stereo vision or structured light also length of the baseline
$C$	Image center (optical); in the context of an encoded pattern also symbol for the code of the pattern
$C'$	Code of an edge pattern
$(c_x, c_y)$	Position of the image center in image coordinates
$c$	Codeword
$c_\sigma(i_p, j_p)$	Codeword for slide coordinates $(i_p, j_p)$ under codeword reading function $\sigma(k)$
$c'$	Derived codeword
$d$	Slide margin (in the context of the coded light approach), respectively disparity (in the context of stereo vision)
$dx, dy$	Width and height of a sensor element of a camera
$E$	Irradiance or irradiation

$E_p$	Irradiance or irradiation caused by a projection device (in the context of the structured light approach)
$E_0$	Background illumination (in the context of the structured light approach)
$F$	F-number or focal ratio of a lens
$f$	Focal length
$f_k$	Effective focal length (i.e. the image plane distance)
$f_r$	Bi-directional Reflectance Distribution Function (BRDF)
$g$	Object distance
$h$	Hamming distance of two codewords, respectively minimal distance of a code
$I(x, y)$	Image function
$I_l(x, y)$	lth component function of a vector-valued image function
$I_p(x, y)$	Slide function
$I_p'(x, y)$	Slide function describing the edge pattern associated with the slide $I_p(x, y)$
$I_p(i, j, k)$	Projection pattern, i.e. potentially time-varying slide
$(i, j)$ or $(i_i, j_i)$	Discrete image coordinates
$(i_p, j_p)$	Discrete slide coordinates
$k$	Constant numeric (proportionality) factor
$L$	Radiance
$M$	Radiant exitance or emittance
$N_x, N_y$	Image dimensions
$m, n$	Image dimensions
$m_p, n_p$	Slide dimensions
$n$	Index of refraction
$O$	Optical center of a pinhole camera
$Q$	Set of quantization levels
$Q_p$	Code Alphabet
$Q_r$	Radiant energy
$(p, q)$	Shorthand notation for the surface normal $(-p, -q, 1)^T$
$q$	Code symbol
$q_i$	ith symbol of a codeword
$q_p$	Size of code alphabet, respectively number of distinct graylevels, colors or pattern primitives of a projection pattern
$\mathbf{R}=(r_{ij})$	Rotation matrix of the $\mathfrak{R}^3$
$R_x, R_y, R_z$	Euler angles
$R(p, q)$	Reflectance map
$r$	Radius (occasionally refers to a constant as well)

$r_{ij}$	Element of a rotation matrix
$(R, G, B)$	Tristimulus vector of a color
$r(\lambda), g(\lambda), b(\lambda)$	tristimulus or standard observer curves; respectively, $r(\lambda)$ also describes the wavelength-dependent reflectivity or spectral reflectance function of a scene patch
$S$	Scene point or surface patch
$S'$	Image point at which the scene point $S$ is imaged
$s$	Codeword length
$s(\lambda)$	Spectral response or sensitivity curve of a sensor
$t$	Number of distinct slides part of a projection pattern; also used as variable of time
$\mathbf{t}=(t_x, t_y, t_z)$	Translation vector of the $\mathfrak{R}^3$
$v, w$	Window size where $v$ is the height, $w$ the width of the window
$XY$	2D Cartesian, right handed coordinate system with an $x$ and $y$ axis
$XY_i$	Image coordinate system
$XYZ$	3D Cartesian, right handed coordinate system with an $x, y$ and $z$ axis
$XYZ_c$	Camera coordinate system
$XYZ_w$	World coordinate system
$(x, y)$ or $(x_i, y_i)$	Continuous image coordinates
$(x_f, y_f)$	Frame coordinates
$(x_d, y_d)$	Distorted continuous image coordinates
$(x_p, y_p)$	Continuous slide coordinates
$(x_s, y_s)$	Sensor coordinates
$(x_u, y_u)$	Undistorted (i.e. ideal) continuous image coordinates
$(x_c, y_c, z_c)$	Coordinates within the camera coordinate system $XYZ_c$
$(x_w, y_w, z_w)$	Coordinates within the world coordinate system $XYZ_w$
$z(x, y)$	Depth map
$z_{\min}, z_{\max}$	Minimal (standoff) and maximal possible range value





# 1 Introduction

## 1.1 Motivation and Purpose of this Work

Machine vision is a central component of human-machine communication. Traditionally, it is based on color or gray level images. Such images – more generally referred to as *intensity images* – are strongly influenced by the type of illumination, the illumination direction and the viewing angle. This dependency, well known from photography, poses a fundamental problem to any intensity-based vision system. It particularly affects visual human-computer interfaces such as gesture or face recognition systems as they typically operate in real-world environments – e.g. cars or factories – where the above factors cannot be controlled. As a result, such interfaces are often too unreliable for every-day use.

Visual interfaces based on *range images*, i.e. the three-dimensional surface data of their environment, avoid the above problematic altogether. They have the additional advantage that many tasks are much easier to solve given such spatial data than given intensity information only. At the same time, they face a new problem, namely that of obtaining the range images in the first place. Many researchers consider the effort required to solve this task, known as (*non-tactile automated computer*) *acquisition of range images* or *ranging* for short, to outweigh the advantages of range data. In this context, Chellappa et al. [1995] remark with respect to face recognition: “Although range information is richer than the 2D intensity array, we feel that cost considerations will make range image based techniques less attractive for field use.”

As Chellappa et al. point out, the challenge lies in obtaining range images with an effort and of a quality that is acceptable in practice. This task is relevant far beyond the scope of human-machine communication: range data is needed for many other applications such as robot navigation, industrial surface inspection, volume measurement, or the creation of 3D models of real-world objects. This wide scale of uses explains why the acquisition of range images has been a primary objective of computer vision and related fields for many decades.

Nevertheless, ranging is still an unsolved problem but for certain special cases. More accurately, it is an open question how machines can acquire accurate high-resolution range images of arbitrary, potentially moving scenes robustly, in real-time and with reasonable effort. This thesis attempts to answer this question. To that end, it develops a ranging method of its own. Its central part is a new approach to color coded light with a single static projection pattern that overcomes certain limitations of comparable methods. This coded light approach works well with most scenes, but has certain intrinsic weaknesses, e.g. at singularities of the scene surface. For this reason, it is complemented by a subsequent stereo vision step, yielding a new two-stage ranging technique that meets all of the above criteria. The proposed method actually allows an integrated 2D-3D vision approach as it obtains a color image of the scene besides the range data.

The evaluation of a prototype implementation of the proposed technique demonstrates that it solves the outlined problem for scenes up to a few meters away from the acquisition system, i.e. for the typical working space of a human-computer interface.

The thesis further demonstrates for an exemplary visual human-machine interface, a 3D face recognition system, that employing range data can make such an interface more reliable than one based on intensity data only. To that end, it shortly describes a face verification system based on a standard face recognition technique that exploits both range and color images. It then presents the results of an evaluation of the system on a database of 2700 range and color images. These show that the use of range data allows reducing the false acceptance and false rejection rates of the system significantly compared to a strictly intensity-based approach.

## 1.2 Organization of this Work

This work is organized as follows:

The second chapter presents the physical and mathematical concepts needed for a discussion of the non-tactile automated computer acquisition of range data, starting out with the essential facts from the physics of radiation and focusing on the topic of reflection. It then explains what an intensity image is and how it is formed, based on classical models such as the pinhole or lens camera model.

Given this fundament, chapter three presents the state of the art with respect to range imaging. It first uses the concepts introduced in chapter two to define essential terms such as range image or range image acquisition system. It establishes an abstract model of a range acquisition system and a common set of aspects of such systems as basis for a generic discussion and comparison of ranging techniques. A review of the most important range imaging methods then forms the core of the chapter. As mentioned above, 3D perception has received vast attention in the past; there is consequently a great amount of previous work. For this reason, the chapter focuses on the approaches that have the potential of solving the problem given the constraints laid down: arbitrary scenes, real-time ability, high accuracy, robustness and moderate resource requirements.

The fourth chapter introduces a new approach to range image acquisition. First, it precisely defines the problem to be solved and shows by reference to the state of the art that it is still unsolved. Next, it outlines the principles and key ideas of a solution. Its detailed description is structured into three parts: the coded light step, the stereo vision step and finally a set of optional features such as a scene color estimation component that might or might not be needed or applicable, depending on the application in mind.

Chapter five evaluates a prototype system based on the proposed ranging approach. In this context, it treats the topic of accuracy in detail in the course of a theoretical error analysis. To that end, it identifies the factors that cause the measurement error in the first place. The chapter then introduces a parameterized model of a triangulation system. It explores how the measurement error caused by these factors depends on the choice of set-up parameters, considering the general case of a triangulation system as well as the special case of a structured light system. The chapter concludes with describing experiments conducted to assess the prototype's accuracy, its frame rate and its ability to acquire range data of arbitrary scenes.

Chapter six describes an exemplary application of the proposed range acquisition system to the task of face recognition. It briefly discusses the technique employed for face verification with range and color images. It then describes a database of range and color images recorded for the evaluation of the recognition algorithm, followed by a report on the results of the evaluation of the algorithm with the database.

The final chapter lists the inferences made in the course of the work and summarizes the contributions of new knowledge made. It further discusses how future work could extend on the results of this thesis.

## 2 Fundamentals of Range Image Acquisition

This chapter presents the fundamentals of range image acquisition. On their lowest conceptual layer, all non-tactile ranging approaches are based on receiving and processing radiation. For this reason, we need to understand what radiation is, how it propagates through space and how we can quantify it, shortly we have to obtain at least working knowledge of the physics of radiation (2.1). The next two sections deal with (digital) cameras seeing that some of the most important ranging approaches derive their data from intensity images taken with such cameras. Moreover, we will treat range image acquisition systems as a special type of camera that acquires spatial rather than brightness information of surfaces. Section 2.2 introduces the most important geometric camera models, section 2.3 the model of a sensor for digitizing images. In the latter context, we also discuss color sensors, and as prerequisite for that the concept of color. The chapter concludes with a summary of its main results (2.4).

### 2.1 Physics of Radiation

Radiation is defined as *energy propagating through space*. It is not a rare phenomenon but an essential part of our everyday life that we perceive for example as light or sound. Radiation typically spreads in the form of three-dimensional waves or moving particles. Electromagnetic radiation may even be regarded as either form as it behaves partly like waves and partly like particles. However, the case of particle radiation is negligible for the purpose of this work, and we will use the terms radiation and (three-dimensional) wave interchangeably throughout unless explicitly stated otherwise.

As a first step in discussing radiation, we have to find a way to describe and measure it. The corresponding branch of physics is called radiometry. The next section outlines its basic concepts and quantities (2.1.1).

Image acquisition systems, whether for range or intensity data, collect radiation sent out from objects to form an image. Even though all objects of our world constantly emit radiation of their own, e.g. thermal radiation, we ignore this kind of emission for the scenes considered in this work. Namely, we always assume the presence of at least one separate source that emits radiation several orders of magnitude more powerful than the scene's own active emission. This implies the waves the acquisition system registers are not actively emitted by the scene itself, but by a different source and only reflected by the scene surface as visualized in figure 1. Consequently, it is essential to understand how radiation propagates through space and in particular which laws and models describe its reflection by a surface (2.1.2).

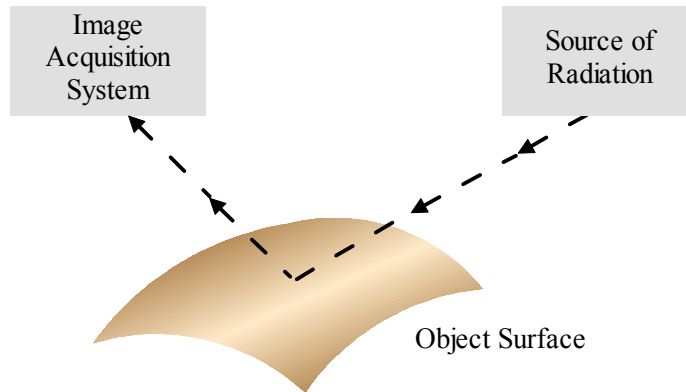


Figure 1: General case of range or intensity image formation. A source radiates on a scene surface that reflects a fraction of this radiation into the direction of the acquisition system.

### 2.1.1 Basic Concepts and Quantities of Radiometry.

As stated above, radiation is a form of propagating energy. Consequently, the most basic quantity in radiometry is the *radiant energy*  $Q_r$  emitted by a source, measured in Joule. The definition of the radiant power emitted by a source, the *radiant flux*  $\Phi$ , follows as radiant energy per time unit dt:

$$\Phi = \frac{dQ_r}{dt} [W] \quad (1)$$

It is typically specified in Watt. The radiant flux  $\Phi$  emitted by a surface per unit square is called its *radiant exitance* or *radiant emittance*  $M$ . The radiant flux  $\Phi$  per unit square impinging on a surface is called *irradiance* or *irradiation*  $E$ . Radiant exitance and irradiance have a different interpretation, but the same definition as they both represent flux per surface unit  $dA$ , measured in Watt per  $m^2$ :

$$M = \frac{d\Phi}{dA} [W m^{-2}] \quad E = \frac{d\Phi}{dA} [W m^{-2}] \quad (2)$$

A source can radiate into the full sphere of directions. Things remain simple if it does so radially and uniformly in all directions, in which case we call it *isotropic*. However, the radiant flux of real world sources typically varies strongly with the direction. To deal with such an-isotropic sources, we need a way to represent directions in 3D space. To this end, we employ *spherical coordinates*, i.e. we specify a direction by its *polar angle*  $\theta$  (co-latitude) and its *azimuth angle*  $\varphi$  (longitude) within some well-defined coordinate system. When dealing with planar surfaces such as infinitesimal surface patches, we use in the following implicitly a local coordinate system whose polar axis corresponds to a surface normal and whose azimuth axis lies on the surface plane.

We further need to extend the notion of an angle to three dimensions via the concept of a *solid angle*  $\Omega$  (also called *space angle*). Its unit is the *steradian*, the straightforward extension of radians to the three-dimensional sphere. The International System of Units (SI) defines a steradian as the solid angle that has its vertex in the center of a sphere of radius  $r$  and cuts off an area of  $r^2$  of the surface of the sphere [NIST 2002]. Accordingly, the full sphere subtends  $4\pi \approx 12.56$  steradian, and for a given surface  $S$  its size  $\Omega_S$  in steradian is the area of its projection on the surface of a sphere of radius  $r$ , normalized by  $r^2$ . We obtain the latter analytically by integrating over the differential patches of the surface. Each of these infinitesimally small planar patches  $dA$  subtends the solid angle  $d\Omega$ , implying the total surface  $S$  subtends the integral over all its patches (where  $\alpha$  is the angle between the surface normal of  $dA$  and the line connecting  $dA$  to the sphere's origin):

$$d\Omega = \frac{dA \cos\alpha}{r^2} = \frac{dA'}{r^2} [sr] \quad \Omega_S = \int_S \frac{\cos\alpha}{r^2} dA = \iint_S \sin\theta d\theta d\varphi [sr] \quad (3)$$

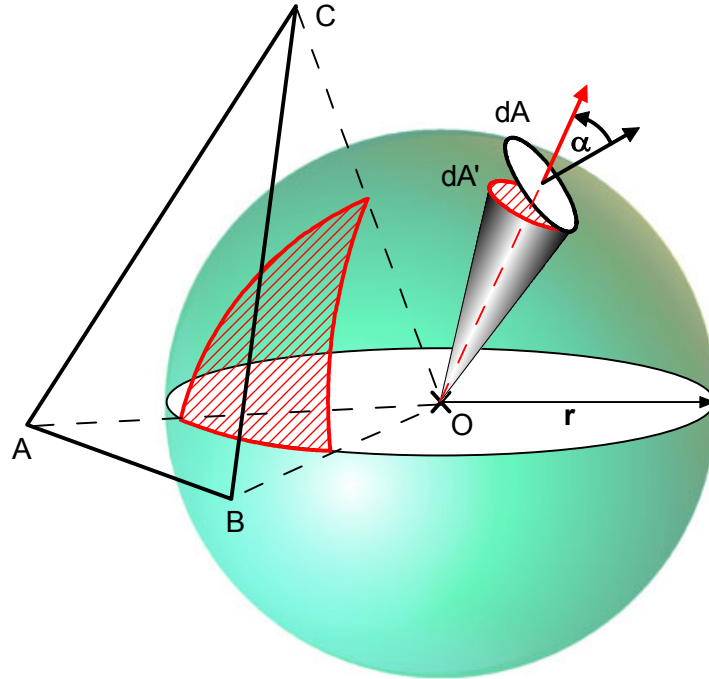


Figure 2: The solid angle of a surface is the area of its projection on the surface of a sphere of radius  $r$  and origin  $O$ , e.g. the hatched area for the surface  $ABC$ , normalized by  $r^2$ . For an infinitesimal surface  $dA$  at distance  $r$ , this area corresponds to  $dA' = dA \cdot \cos(\alpha)$ , its solid angle to  $dA'/r^2$ .

Figure 2 illustrates these definitions. The concept of a solid angle allows defining the *radiance*  $L$  of a surface, its radiant flux per unit steradian and unit area:

$$L = \frac{d^2\Phi}{d\Omega dA'} = \frac{d^2\Phi}{d\Omega dA \cos\alpha} \quad [W sr^{-1} m^{-2}] \quad (4)$$

In this definition, only the effective (foreshortened) surface is considered. Its area is computed as product of the surface area and the cosine of the angle subtended by its normal and the direction of the solid angle. Using these definitions, we describe an isotropic source via  $L(\theta, \varphi)$ , its radiance in the direction  $(\theta, \varphi)$ . To obtain its radiant emittance  $M_\Omega$  into a given solid angle  $\Omega$ , we integrate  $L(\theta, \varphi)$  over this angle. The total exitance  $M_{tot}$  follows from integrating over the whole hemisphere.

$$M_\Omega = \int_\Omega L(\theta, \varphi) \cos\theta d\Omega \quad M_{tot} = \int_0^{2\pi} \int_0^{\pi/2} L(\theta, \varphi) \cos\theta \sin\theta d\theta d\varphi \quad (5)$$

Again the foreshortening accounts for the cosine term, while the  $d\Omega$ , respectively  $\sin\theta d\theta d\varphi$  term represents the infinitesimal solid angle. Analogously to  $L(\theta, \varphi)$ , we define  $E(\theta, \varphi)$  as the irradiance per unit solid angle coming from the direction  $(\theta, \varphi)$ .

We often have to consider the distribution of radiation over the spectrum. For this reason, we introduce the *spectral density distribution*  $C(\lambda)$  of a radiometric quantity. Let  $C(\lambda, \lambda + d\lambda)$  be value of a quantity  $C$  considering only waves with a wavelength from  $\lambda$  to  $\lambda + d\lambda$ . We then define its spectral distribution  $C(\lambda)$  as the following limit (which is assumed to exist):

$$C(\lambda) = \lim_{d\lambda \rightarrow 0} \frac{C(\lambda, \lambda + d\lambda)}{d\lambda} \quad (6)$$

### 2.1.2 The Propagation of Radiation

The propagation of radiation in 3D space is a complex subject, mostly due to the wave nature of radiation. Fortunately we can mostly ignore this wave nature and limit ourselves to *geometric or ray optics*, which deal with the rectilinear propagation of infinitesimal narrow bundles of radiation. This widely used approach facilitates the discussion of the propagation of radiation greatly.

When radiation impinges on the surface of a material object, it is absorbed, transmitted or reflected. Usually all three effects occur concomitantly. The absorbed part is transformed into other forms of energy, primarily thermal energy. This leads to an increased temperature radiation of the object, yet as we ignore this type of radiation the absorbed fraction is effectively lost for our purposes.

The transmitted fraction propagates through the object with a velocity that depends on the properties of the medium, e.g. on its electric and magnetic properties in case of electromagnetic radiation. This speed can vary drastically with different media: sound waves travel through steel with 5100 m/s as compared to 342 m/s through air (at sea-level, 18° C and a frequency of 440kHz [Gerthsen 1960]). If the speed of a wave changes when it enters a new medium, its propagation direction changes at the interface between the media as well. This fact is described by *Snell's law*:

$$v_2 \sin \theta_1 = v_1 \sin \theta_2 \quad \text{or} \quad n_1 \sin \theta_1 = n_2 \sin \theta_2 \quad (7)$$

In the left form of Snell's law,  $v_1$  and  $v_2$  are the respective velocities of the wave in the two media while  $\theta_1$  and  $\theta_2$  represent the respective angles between the interface normal and the wavefront normal. For electromagnetic waves, the right form is more common. It uses the *index of refraction*  $n$  of the media, the ratio of speed of an electromagnetic wave in vacuum to the one in the medium. This index is not a constant, but a function of the frequency since the propagation velocity of radiation for a given medium often varies with the frequency.

Finally, the fraction of the radiation not absorbed and not transmitted is sent out again from the object surface, i.e. reflected. We formally describe the reflectance of a surface via its point-wise *Bidirectional Reflectance Distribution Function* (BRDF)  $f_r$ . The National Institute of Standards [NIST 2002] defines the BRDF  $f_r$  of a surface point (or rather infinitesimal patch)  $dA$  as

$$f_r(\theta_i, \varphi_i, \theta_r, \varphi_r) = \frac{dL(\theta_r, \varphi_r)}{dE(\theta_i, \varphi_i)} \left[ \frac{1}{sr} \right] \quad (8)$$

i.e. as ratio of  $dL(\theta_r, \varphi_r)$ , the radiance of  $dA$  in the direction  $(\theta_r, \varphi_r)$ , to  $dE(\theta_i, \varphi_i)$ , the irradiance incident on  $dA$  from  $(\theta_i, \varphi_i)$ . This definition is illustrated in figure 3. Put another, less formal way, the BRDF describes how bright a surface appears viewed from one direction when illuminated from a certain other direction. It is often possible to reduce the four parameters of the BRDF to three as most surfaces can be rotated about their normal without altering the radiance, i.e. only the difference  $\varphi_r - \varphi_i$  is of relevance, not the separate angles [Horn 1986]. The only exceptions are surfaces with oriented microstructure such as e.g. the iridescent feathers of certain birds [Horn 1986] or some types of wheel rims [Klette et al. 1995]. If the BRDF of a surface is known as well as the light sources illuminating it, the surface's radiance can be computed by integrating over all possible directions of incidence:

$$L(\theta_r, \varphi_r) = \int_0^{\pi/2} \int_0^{2\pi} f_r(\theta_i, \varphi_i, \theta_r, \varphi_r) \underbrace{E(\theta_i, \varphi_i) \cos \theta_i \sin \theta_i d\varphi_i d\theta_i}_{dE(\theta_i, \varphi_i)} \quad (9)$$

The BRDF for a given material is either measured or derived from a reflection model. The former approach is obviously very cumbersome owing to the BRDF's many degrees of freedom. The latter method works out only for certain ideal surfaces such as the ones described next.

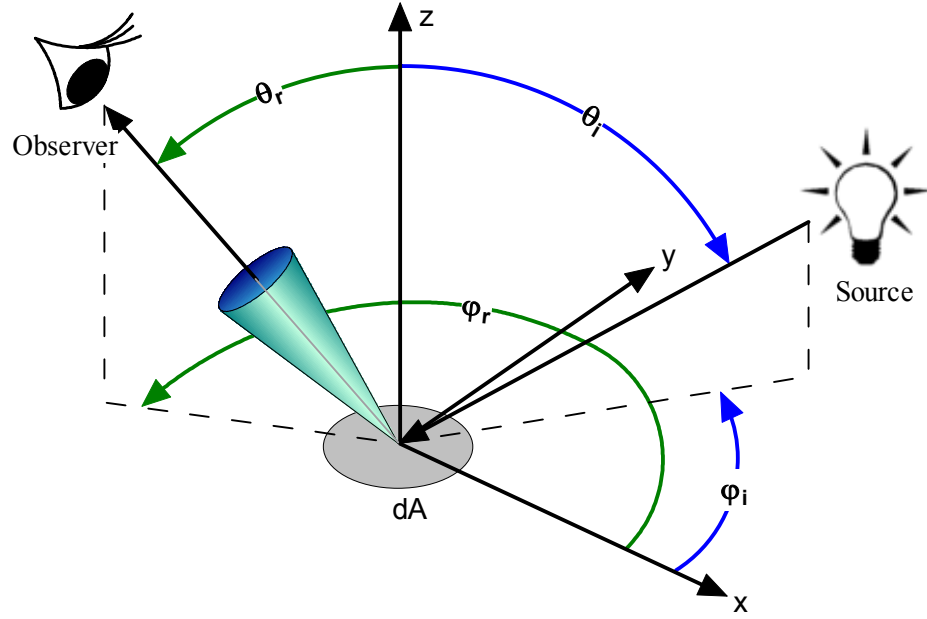


Figure 3: The BRDF describes the ratio of the radiance of a surface  $dA$  viewed from the direction  $(\theta_r, \varphi_r)$  to the irradiance incident from the direction  $(\theta_i, \varphi_i)$ .

The simplest case of reflection occurs at an ideally planar interface between two media. In that case, the basic *law of reflection* applies. It states that the angle of incidence  $\theta_i$  equals the angle of reflection  $\theta_r$  of the reflected beam of radiation, i.e.  $\theta_i = \theta_r$ . This type of reflection is called *specular* or *regular reflection*. The best example of a surface that exhibits almost exclusively specular reflection (for visible light) is an ordinary mirror. We can derive the BRDF of an ideally specular surface easily because its radiance is zero but for the direction  $(\theta_i, \varphi_i - \pi)$ . Consequently, the BRDF  $f_r$  is proportional to the product of the two Dirac impulses  $\delta(\theta_r - \theta_i)$  and  $\delta(\varphi_r - \varphi_i - \pi)$ . We determine the proportionality factor  $k$  (here as in the rest of this work  $k$  denotes a proportionality factor) by exploiting that with an ideal mirror no energy is lost, i.e. ignoring other light sources  $M_{\text{tot}} = dE(\theta_i, \varphi_i)$ , and by using the sifting property of the impulses:

$$\begin{aligned}
 M_{\text{tot}} &= \int_0^{2\pi} \int_0^{\pi/2} L(\theta_r, \varphi_r) \cos\theta_r \sin\theta_r \, d\theta_r \, d\varphi_r \\
 &= \int_0^{2\pi} \int_0^{\pi/2} dE(\theta_i, \varphi_i) f_r(\theta_i, \varphi_i, \theta_r, \varphi_r) \cos\theta_r \sin\theta_r \, d\theta_r \, d\varphi_r \\
 &= \int_0^{2\pi} \int_0^{\pi/2} dE(\theta_i, \varphi_i) k \delta(\theta_i - \theta_r) \delta(\varphi_r - \varphi_i - \pi) \cos\theta_r \sin\theta_r \, d\theta_r \, d\varphi_r \\
 &= dE(\theta_i, \varphi_i) k \cos\theta_i \sin\theta_i
 \end{aligned} \tag{10}$$

By resolving the last line with regard to  $k$ , the BRDF of an ideally specular surface follows as:

$$f_r(\theta_i, \varphi_i, \theta_r, \varphi_r) = \frac{\delta(\theta_r - \theta_i) \delta(\varphi_r - \varphi_i - \pi)}{\cos\theta_i \sin\theta_i} \quad \text{for } 0 < \theta_i < \frac{\pi}{2} \tag{11}$$

Planar interfaces in the mathematical sense do not physically exist; planar is better read as "of negligible roughness relative to the incident signal's wavelength". In industrial environments, polished and smooth surfaces that act as mirror for most wavelengths are predominant. In natural settings, rough surfaces are much more common. They exhibit mostly diffuse reflection described next.

If a surface is rough relative to the wavelength, the reflection of incoming waves is scattered in many directions; as a result, the integrity of an incident wave front is lost. This type of reflection is called *diffuse or matte reflection*. An example for diffuse reflection is the reflection of light on this page of paper. A perfectly diffuse surface appears equally bright from all viewing directions, i.e. its radiance is constant over all viewing directions:  $L(\theta_r, \varphi_r) = k$ . If it furthermore reflects all impinging radiation completely, it is called Lambertian reflector. The proportionality factor  $k$  for a Lambertian reflector again follows from equating  $M_{\text{tot}}$  with  $E_{\text{tot}}$ . To that end, we first compute  $M_{\text{tot}}$  as:

$$M_{\text{tot}} = \iint_S L(\theta_r, \varphi_r) \cos\theta_r \sin\theta_r d\varphi_r d\theta_r = \int_0^{\pi/2} \int_0^{2\pi} k \cos\theta_r \sin\theta_r d\varphi_r d\theta_r = \pi k \int_0^{\pi/2} \sin 2\theta_r d\theta_r = \pi k \quad (12)$$

The radiance and BRDF of a Lambertian reflector is obtained by equating  $E_{\text{tot}}$  and  $M_{\text{tot}}$ :

$$L = \frac{E_{\text{tot}}}{\pi} \Rightarrow f_r(\theta_i, \varphi_i, \theta_r, \varphi_r) = \frac{dL(\theta_r, \varphi_r)}{dE(\theta_i, \varphi_i)} = \frac{1}{\pi} \quad (13)$$

If a point source of irradiance  $E$  illuminates a matte surface from the direction  $(\theta_i, \varphi_i)$ , the surface's radiance is proportional to  $\cos(\theta_i)$  as  $E_{\text{tot}} = E \cdot \cos(\theta_i)$ . This result is known as *cosine* or *Lambert's law of reflection from matte surfaces*.

Of course, reflection is in general more complex than in the above ideal cases; real world surfaces do not reflect all incoming radiation; also their reflectance properties depend on the wavelength considered. So more realistic reflection models such as the *Dichromatic Reflection Model* ([Shafer 1985], [Klunker et al. 1990]) incorporate the fraction of the incoming radiation reflected by a surface and the wavelength as additional parameters. The DRM describes the reflection from a point on a dielectric non-uniform material as mixture of radiation reflected at the material surface, the *surface reflection component*, and of the radiation reflected from the material body, the *body reflection component*. Each component is separated into its spectral and its geometric reflection properties, i.e. is modeled as the product of a spectral power distribution and a geometric scale factor.

According to the model, the surface reflection component has about the same spectral power distribution as the incident radiation. It is perceived as highlight or gloss. The DRM does not specify a term for the geometric scale factor; any of the numerous ones proposed in the literature for non-ideal specular reflectance can be used (e.g. [Phong 1975], [Horn 1977]; see [Nayar et al. 1991] for an overview). Details regarding the surface and the incoming radiation being unknown, all of them describe the reflection rather similarly: the reflection is maximal for the surface normal halfway between source and viewer, the so-called *halfway vector*, since the viewing angle is  $(\theta_i, \varphi_i - \pi)$  for a surface with this normal. It then drops off sharply with increasing angle between the halfway vector and the normal of the considered scene patch, e.g. with some power of its cosine ([Phong 1975], [Horn 1977]) or the exponential of its negative amount [Torrance and Sparrow 1967]. The body reflection component represents the radiation that has penetrated the surface and entered the material body, where it is scattered until it arrives again at the surface and exits the material. It provides the characteristic object color, as the radiation traveling through the body is increasingly absorbed at wavelengths characteristic for the material. Its geometric scale factor is usually modeled as approximation of a perfectly diffuse reflector.

Even the complex DRM does not include phenomena such as fluorescent surfaces. Also real world surfaces are usually inhomogeneous, there is typically more than a single source of radiation, and some or all of the sources are uncontrolled. But for a single convex surface also mutual illumination has to be taken into account. Finally, the topic of this work is acquiring range images of unknown scenes whose surface orientation, let alone reflection properties, are correspondingly unknown as well. So a more sophisticated model would be of little use as all its parameters would be unknown anyway. We consequently proceed with a generically applicable qualitative understanding of reflection and the factors that influence it rather than with an exact quantitative model.



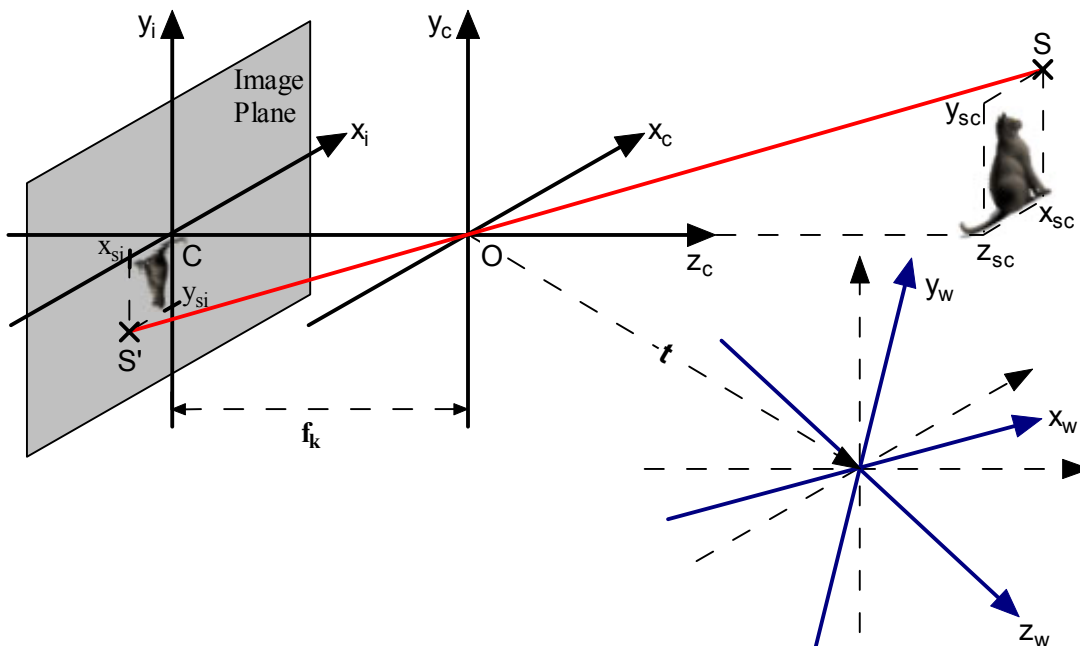


Figure 4: A pinhole camera is defined by its optical center  $O$  and its image plane. It produces an image that is a perspective projection of the scene on the image plane with projection center  $O$ . The  $z_c$ -axis of the camera coordinate system  $XYZ_c$  coincides with the optical axis, while the world coordinate system  $XYZ_w$  is defined independently of the camera.

## 2.2 Geometric Camera Models

We use two models to describe the geometric aspects of a camera: The simple, but in many cases adequate pinhole camera model (2.2.1) and the more complex lens camera model (2.2.2).

### 2.2.1 The Pinhole Camera Model

A basic and widely used geometric camera model is that of a *pinhole camera* as depicted in figure 4. A pinhole camera is defined by its *optical center*  $O$  and its *image plane*, also called *retinal plane*. Its *optical axis* follows as the straight line through the optical center that is perpendicular to the image plane. The intersection of the optical axis with the image plane is called (*optical*) *image center*  $C$ , the distance from  $O$  to  $C$  is termed (*effective*) *focal length*  $f_k$  or *image plane distance*  $b$ . The *image coordinate system*  $XY_i$  with origin  $C$  is a two-dimensional Cartesian coordinate system of the image plane. The *camera coordinate system*  $XYZ_c$  is a right-handed, three-dimensional Cartesian coordinate system of the space  $\mathcal{R}^3$ . Its origin is the optical center  $O$ ; its  $z_c$ -axis coincides with the camera's optical axis and its  $x_c$ - and  $y_c$ -axis are parallel to the  $x_i$ -axis and  $y_i$ -axis of the image plane coordinate system. A scene point  $S$  with camera coordinates  $(x_{sc}, y_{sc}, z_{sc})$  is projected at a point  $S'$  of the image plane with camera coordinates  $(x_{si}, y_{si}, -f_k < 0)$  and image coordinates  $(x_{si}, y_{si})$ . The relationship between camera and image coordinates is given by the following equation, which is sometimes called *perspective imaging equation*:

$$x_{si} = -\frac{x_{sc} f_k}{z_{sc}} \quad y_{si} = -\frac{y_{sc} f_k}{z_{sc}} \quad (14)$$

With other words, a pinhole camera defines a *perspective projection* on the image plane with projection center  $O$ . Which implies that – without additional knowledge – it is inherently impossible to reconstruct the exact 3D form of an object from images taken with a single pinhole camera: Two objects of the same shape (e.g. two spheres), one twice the size and at twice the  $z_c$ -distance to the camera compared to the other, give rise to identical images.

Real-world cameras as described above acquire horizontally and vertically mirrored images. For convenience, we work in the following mostly with cameras with imaginary retinal planes located in front of the optical center. They acquire images that are un-mirrored, but otherwise identical to the ones of real-world cameras (i.e. described by equation 14 but for the negative signs).

If the variation in depth over the scene is small in relation to the average distance scene-camera, we may use *parallel* or *orthographic* projection (with some scale factor  $k$ ) as close approximation to the actual perspective projection. The corresponding *parallel imaging equation* is much simpler:

$$x_{si} = k x_{sc} \quad y_{si} = k y_{sc} \quad (15)$$

We occasionally employ an additional, arbitrarily chosen and camera independent coordinate system of the  $\mathfrak{R}^3$ , the *world coordinate system*  $XYZ_w$ . As usual, we convert coordinates between two coordinate systems via a rotation, i.e. multiplication of a given vector with a rotation matrix  $\mathbf{R} = (r_{ij})$ , and a subsequent translation, i.e. vector addition of the rotated vector with a translation vector  $\mathbf{t} = (t_x, t_y, t_z)$ . Consequently, a point with world coordinates  $(x_{sw}, y_{sw}, z_{sw})$  is imaged at

$$x_{si} = f_k \frac{r_{11}x_{sw} + r_{12}y_{sw} + r_{13}z_{sw} + t_x}{r_{31}x_{sw} + r_{32}y_{sw} + r_{33}z_{sw} + t_z} \quad y_{si} = f_k \frac{r_{21}x_{sw} + r_{22}y_{sw} + r_{23}z_{sw} + t_y}{r_{31}x_{sw} + r_{32}y_{sw} + r_{33}z_{sw} + t_z} \quad (16)$$

In practice, pinhole cameras are simple to build and have for that reason been used since the 14<sup>th</sup> century [Bergmann and Schäfer 1987]. Even though the pinhole cannot be made arbitrarily small to avoid diffraction of the incoming light, they give sharp images without significant aberrations. They are nevertheless rarely used because the necessarily small pinhole generally subtends only a very small solid angle of a given surface and accordingly receives very little of its radiant emission. To obtain a measurable quantity of light, this has to be compensated with a long exposure time, which limits the use of pinhole cameras to scenes that are static during this time span. For this reason, representative modern cameras use lenses described next rather than pinholes.

### 2.2.2 The Lens Camera Model

A lens camera is a special type of a pinhole camera, with which the conceptually infinitesimal pinhole is replaced with a finite-sized lens. In principle, the perspective image formation equation 14 also applies to lens cameras, the most relevant difference being that lens cameras produce well-focused images only of objects at a certain *object distance*  $g$ . The relation between  $g$ , the *image plane distance*  $b$  and the *focal length*  $f$  of the lens is given by the well-known (*thin-*) *lens equation*:

$$1/f = 1/g + 1/b \quad (17)$$

The *aperture stop* of a lens camera is an adjustable device that limits the lens aperture and thus the diameter of the bundle of incoming rays as shown in figure 5. The (effective size of the) aperture is commonly specified relatively via the *f-number* (also *focal ratio*, *relative aperture* or *speed*), defined as the ratio of the focal length to the diameter of lens aperture and denoted by the symbol  $F$ .

Figure 5 also illustrates that the image of a point  $S_2$  not at the object distance is a disc of radius  $r$  (whose outline is called *blur circle*) rather than a single point. The *depth of field* of a camera is the range of object-side distances for which the radius of the resulting blur circle is acceptably small (not to be confused with the corresponding image space range, the *depth of focus*). Clearly what is acceptable depends on the situation, but conventionally a blur radius of half the width of a sensor element is chosen as maximal permissible one. Given this radius  $r_{\max}$ , the depth of field follows via the lens equation: it is the range of object distances  $g_2$  for which the image-plane distance between the top- and the bottom-most ray emerging from a point  $S_2$  does not exceed the permissible diameter  $2r_{\max}$ . It suffices to consider points on the optical axis as elementary calculations show that the sum of the slopes of the two rays and consequently the diameter of the blur circle depends on the object distance only, not on the object height.

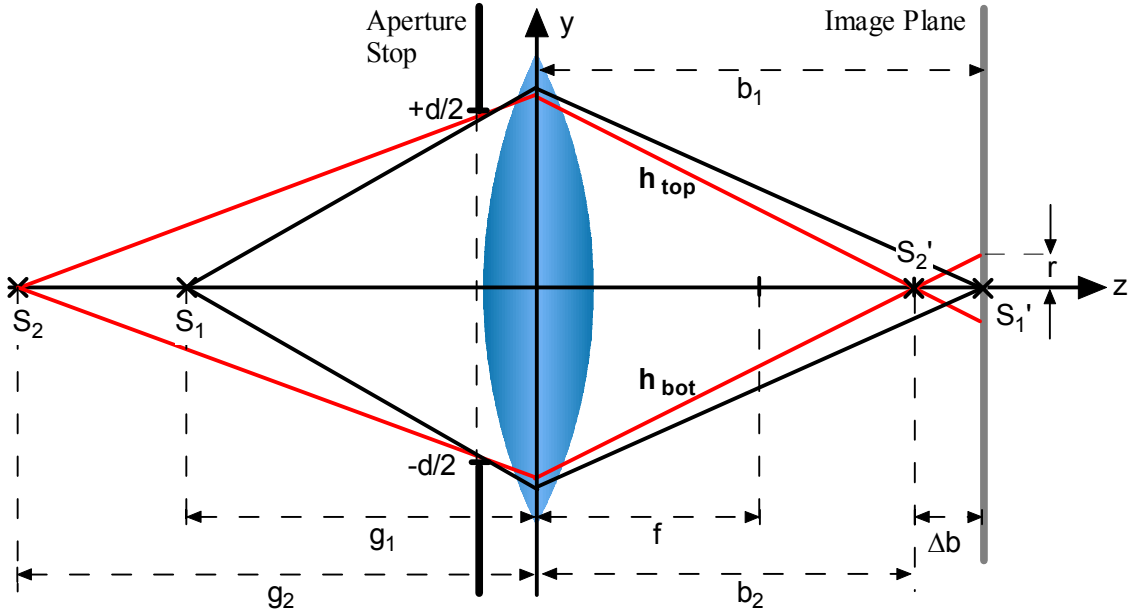


Figure 5: Lens with focal length  $f$  and effective diameter  $d$ . When focused at distance  $g_1$ , the image of a point  $S_1$  located at  $g_1$  is an infinitesimal point while that of a point  $S_2$  outside of the focused plane is a disc of radius  $r > 0$ .

Using the terms introduced in figure 5, we obtain as formula for the blur radius  $r$ :

$$r = \frac{d \Delta b}{2b_2} = \frac{d}{2} \left( \frac{b}{b_2} - 1 \right) = \frac{d}{2} \left( \left( \frac{1}{f} - \frac{1}{g_2} \right) \left( \frac{f g_1}{g_1 - f} \right) - 1 \right) = \frac{d f (g_2 - g_1)}{2g_2 (g_1 - f)} \quad (18)$$

Assuming  $g_1 \approx g_2$  and a large object distance ( $g_1 \gg f$ ) as the standard case, respectively a small object distance as for microscopy ( $g \approx f$ ), the formula for the depth of field follows as:

$$DoF_{standard} = 2r_{max} F \left( \frac{g_1}{f} \right)^2 \quad DoF_{microscopy} = \frac{2r_{max} F g_1}{b_1} \quad (19)$$

Real-world cameras take images that differ from the ones ideal lens cameras would acquire. The deviations are called *image aberrations* and can be attributed to various causes, a major one being that the lens equation is only valid for rays close and approximately parallel to the optical axis. Bergmann and Schäfer [1987] differentiate between *image degrading aberrations* such as the coma and *image deforming aberrations* as e.g. chromatic aberrations and image distortion (figure 6). The former are less relevant for range measurements as they mostly affect the image quality – they typically cause blurring – and can be minimized with suitable optics. We can further ignore chromatic distortion as its effect is mostly negligible for the purpose of this work.

(Symmetric) *Radial distortion* occurs with almost all lenses. It cannot be disregarded as it changes the retinal position at which a world point is imaged significantly. More precisely, radial distortion causes the imaging scale to change with the distance to the optical image center. It is caused by an asymmetric position of the aperture stop within a lens (system) [Bergmann and Schäfer 1987], [Luhman 2000]. Consequently, the type of change depends mostly on the location of the aperture stop: If it is located in front of the lens, the imaging scale grows with the image center distance. This type of radial distortion is referred to as *pincushion distortion*. Analogously, an aperture stop behind the lens causes the scale to drop off with the image center distance. Figure 6 shows the effect of the latter type of radial distortion, termed *barrel distortion*, on an image of a checkerboard.

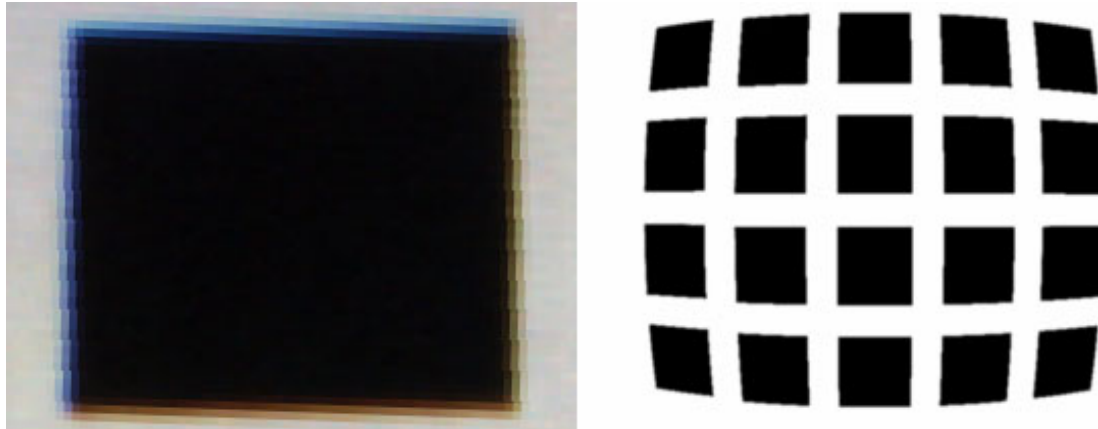


Figure 6: Effects of chromatic aberration (resulting in blue, respectively red square fringes) and radial barrel-distortion on the color-image of a regular black-and-white checkerboard.

[Slama et al. 80] describe the effect of radial distortion via the following infinite series:

$$x_d = x_u - x_d(\kappa_1 r^2 + \kappa_2 r^4 + \dots), \quad y_d = y_u - y_d(\kappa_1 r^2 + \kappa_2 r^4 + \dots), \quad r = (x_d^2 + y_d^2) \quad (20)$$

In this equation as in the following,  $(x_u, y_u)$  represent the unobservable undistorted coordinates at which a point would be imaged if the lens were distortion-free and  $(x_d, y_d)$  the actually observed distorted image coordinates. For most purposes, including ours, the first term of the series describes the effect of radial distortion with sufficient accuracy. We associate for that reason barrel distortion with a positive, pincushion distortion with a negative distortion coefficient  $\kappa$ .

## 2.3 The Camera Sensor Model

### 2.3.1 Formation and Description of Digital Images

The fundament of almost all electronic image sensors is the *photoelectric effect*. It causes electrons to be knocked out of certain surfaces when photons strike them. The number of freed electrons is proportional to the irradiance incident on the image plane (for a fixed spectral distribution). Consequently, the photoelectric effect allows measuring irradiance by determining well-known electric quantities such as the electric charge. To obtain an image, a finite portion of the image plane of a camera is covered with suitable uniform sensor elements of size  $dx$  by  $dy$ . These elements are exposed to the incident light during the exposure time. For each element, the freed electrons are collected; the resulting charge is quantified and A/D-converted to a digital value, called *pixel value*  $I$ . Each (unit-less) pixel value  $I$  is proportional to the number of freed electrons and consequently to the irradiance  $E$  of its sensor element, as expressed by the following equation:

$$I = k \int E(\lambda) s(\lambda) d\lambda \quad (21)$$

In this equation,  $k$  is again a proportionality factor that encapsulates aspects such as the sensor size or the exposure time and  $s(\lambda)$  represents the sensor's spectral response or sensitivity curve. The latter describes the sensor-specific, wavelength-dependent relationship between incoming photons and freed electrons, respective generated charge.

We now show with the help of figure 7 that the irradiance is in turn proportional to the scene brightness, or more precisely, to the radiance of the imaged scene point into the solid angle subtended by the lens. We first note that the solid angles  $\Omega_I$  and  $\Omega_S$  subtended by  $dI$ , respectively  $dS$ , relative to the camera coordinate origin are equal as one is the perspective projection of the other. Equating the two yields the following expression for the ratio of  $dS$  to  $dI$ :

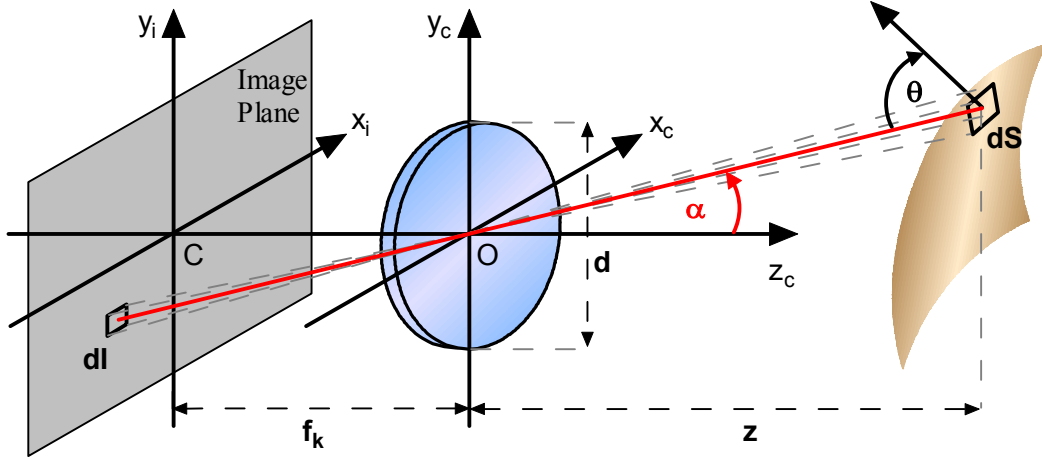


Figure 7: The irradiance at the image plane segment  $dI$  is proportional to radiance of the imaged surface patch  $dS$ . Redrawn from [Ballard and Brown 82] with adaptations.

$$\Omega_I = \frac{dI \cos \alpha}{(f_k / \cos \alpha)^2}; \quad \Omega_S = \frac{dS \cos \theta}{(z / \cos \alpha)^2}; \quad \Omega_I = \Omega_S \Rightarrow \frac{dS}{dI} = \left( \frac{z}{f_k} \right)^2 \frac{\cos \alpha}{\cos \theta} \quad (22)$$

The radiant flux  $\Phi_{lens}$  emanating from  $dS$  and gathered by the lens is equal to the radiance of  $dS$ , multiplied by its foreshortened surface and the solid angle subtended by the lens of diameter  $d$ :

$$\Phi_{lens} = L \cdot dS \cos \theta \cdot \frac{\pi(d/2)^2}{(z/\cos \alpha)^2} \cdot \cos \alpha = L \cdot dS \cos \theta \cdot \frac{\pi}{4} \left( \frac{d}{z} \right)^2 (\cos \alpha)^3 \quad (23)$$

If we ignore losses in the lens, we can equate the radiant flux  $\Phi_{lens}$  collected by the lens with the radiant flux arriving at  $dI$ . By substituting the term for the ratio of  $dS$  to  $dI$  into the definition of the irradiance, we obtain the following expression for the irradiance incident on the sensor element:

$$E = \frac{d\Phi_{lens}}{dI} = L \cdot \frac{dS}{dI} \cos \theta \cdot \frac{\pi}{4} \left( \frac{d}{z} \right)^2 (\cos \alpha)^3 = L \cdot \frac{\pi}{4} \left( \frac{d}{f_k} \right)^2 (\cos \alpha)^4 = L \cdot \frac{\pi}{4} \cdot \left( \frac{1}{F} \right)^2 (\cos \alpha)^4 \quad (24)$$

This implies that each pixel value is proportional to the radiance  $L$  of the imaged surface patch. Or rather, since the sensor elements are not infinitesimally small, but occupy a finite area, that each pixel value is proportional to the average radiance of the imaged surface patch. The smaller the size of this patch, the higher is the lateral resolution of the sensor. Since the patch size depends on the distance of the camera to the scene, we usually specify this lateral resolution as angular size, namely the arc tangent of the ratio of the sensor element size to the effective focal length.

The amount of charge generated within a sensor element is physically limited with real-world sensors; so the range of pixel values is restricted to a certain interval. The accuracy of the A/D-conversion is finite as well. It accordingly suffices to use  $q \in \mathbf{N}$  quantization steps, where  $q$  is typically a fairly low number such as 256; that is, we specify pixel values as integers in the range  $Q = [0, q - 1]$ . With  $N_x = n$  sensor elements along the  $x_i$ -axis and  $N_y = m$  along the  $y_i$ -axis, we represent a (grayscale) image as function  $I(i, j)$  that maps the 2D integer set  $[1, N_x] \times [1, N_y]$  on the 1D integer interval  $Q$ , or alternatively as matrix  $I \in Q^{m \times n}$ . In situations where we can disregard their finite and discrete character, we also model images as continuous real-valued signals  $I(x, y)$ , i.e. as mapping of the  $\mathcal{R}^2$  on  $\mathcal{R}$ . In the following,  $(i, j)$  refers to discrete,  $(x, y)$  to ideal continuous image coordinates. Another consequence of the finiteness of real world cameras is that their field of view is limited, too. We specify it via the horizontal (vertical, diagonal, etc.) half angle of view  $\beta_{hor}$  (see figure 8) calculated as  $\arctan N_x \cdot dx / 2f$  or as the visible area for a given object distance.

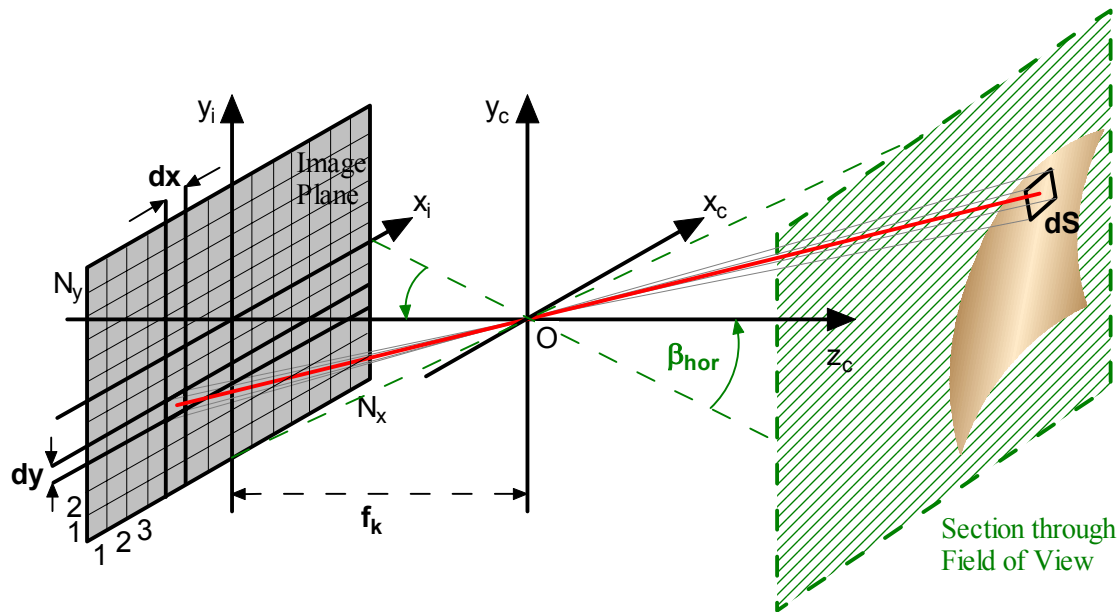


Figure 8: With digital cameras, a finite part of the image plane is covered with  $N_x \times N_y$  sensor elements of size  $dx$  by  $dy$ , resulting in a finite lateral resolution and field, respectively angle of view.

Filters allow limiting the sensitivity of a sensor to a certain spectral interval of interest. In particular, they permit acquiring multi-spectral images by applying  $r$  different filters, e.g. one after another or by splitting the incoming radiation into  $r$  several beams and applying a distinct filter to each beam. Analogously to the definition of a grayscale image, we describe a multi-spectral image as  $r$ -dimensional signal, i.e. as vector-valued function of the image coordinates on the codomain  $Q^r$ :

$$I(i, j) = (I_1(i, j), \dots, I_r(i, j)), \quad I_l(i, j) = k \int_{-\infty}^{+\infty} E_{i,j}(\lambda) s_l(\lambda) d\lambda \quad (25)$$

where  $s_l$  is the (effective) spectral response of the  $l$ -th sensor, i.e. the product of the sensor's original spectral sensitivity and the filter's spectral transmission (ranging unitless from 0 to 1). An important example of multi-spectral imaging is human color vision. It is of special interest as sensors that emulate it are widely available and form the basis of the ranging method proposed in this work.

### 2.3.2 The Concept of Color, Color Vision and Color Images

Light is defined as the type of electromagnetic radiation capable of producing visual sensation in most humans. Its spectrum ranges from ca. (the exact interval differs from human to human) 390 to 740 nm. Color is the aspect of visual perception that allows an observer to distinguish between two fields of light of the same size, shape, structure, duration and luminance [Wyszecki and Stiles 1982]. Humans perceive color because their retina contains three different types of neurochemical receptors called cones, each of them sensitive to a particular spectral band: One peaking in the long- (red), one in the mid- (green), and one in the short (blue) visible range. How exactly our vision system translates the cone signals into color sensations is not completely understood. What is known – that color perception is non-linear, dependent on e.g. the surroundings and state of adaptation of the viewer [Pratt 1991] – indicates that the underlying processes are very complex.

In any case, for our purposes it suffices to measure color in some well-defined way. Doing so implies relating psychological phenomena to physical phenomena [Grum and Bartelson 1980] and is for that reason quite different from other measurement processes. The most common approach to color measurement is to let humans compare and match colors, especially as Grassmann found out

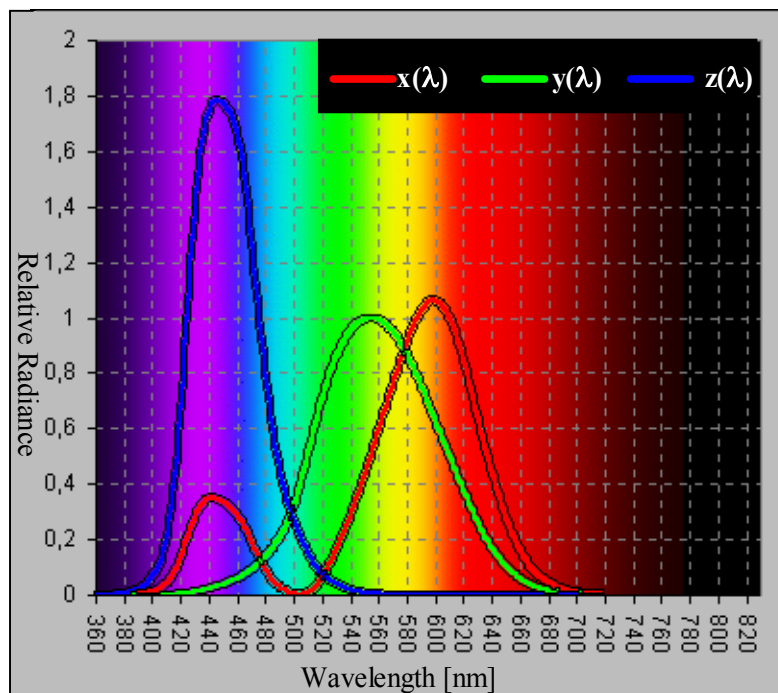


Figure 9: The CIE 1931 photopic standard observer 2° field of view standard observer curves for the imaginary primaries.

that in principle all colors can be matched by superimposing three so-called *primary colors* [Grum and Bartelson 1980]. This rule is also called the *tri-chromatic theory of color*. The only restriction on the choice of the primaries is that none of them may be a mixture of the two others. We are – for a given set of three primaries – for that reason able to specify a color by stating the intensities of the primaries necessary to match it. In this context, a negative intensity of a primary means the corresponding amount of the primary has to be added to the original color to achieve a match.

The Commission Internationale de l'Eclairage (CIE) conducted extensive color matching experiments using three monochromatic sources of spectral centroids 700.0 nm, 546.1 nm and 435.8 nm as primaries. For each monochromatic color in the visible spectrum, a large number of observers determined the radiances of the primaries necessary to match the color. The results of these experiments are called the *tristimulus* or *standard observer curves*  $r(\lambda)$ ,  $g(\lambda)$  and  $b(\lambda)$  for a 2° field of view (as color perception changes with the field of view) and the aforementioned primaries. According to the tri-chromatic theory of color, the tristimulus values RGB of an arbitrary color follow from integrating over the product of the spectral irradiation distribution  $E(\lambda)$  of its source and the respective spectral tristimulus curve:

$$R = k \int_{-\infty}^{+\infty} E(\lambda) r(\lambda) d\lambda \quad G = k \int_{-\infty}^{+\infty} E(\lambda) g(\lambda) d\lambda \quad B = k \int_{-\infty}^{+\infty} E(\lambda) b(\lambda) d\lambda \quad (26)$$

However, it is not possible to directly match all monochromatic colors with the primaries the CIE originally used, i.e. some monochromatic colors result in partially negative tristimulus values. For this and some other reasons, the CIE introduced three imaginary primaries for which the corresponding tristimulus values are strictly positive for the monochromatic colors of the visible spectrum – imaginary because these primaries do not physically exist. Their calculated standard observer curves  $x(\lambda)$ ,  $y(\lambda)$  and  $z(\lambda)$  are shown in figure 9. As before, we use these curves to obtain the matching dimensionless tristimulus values X, Y and Z of a given spectral irradiation distribution  $E(\lambda)$  by integrating the product of the distribution with the curve of interest over the spectrum:

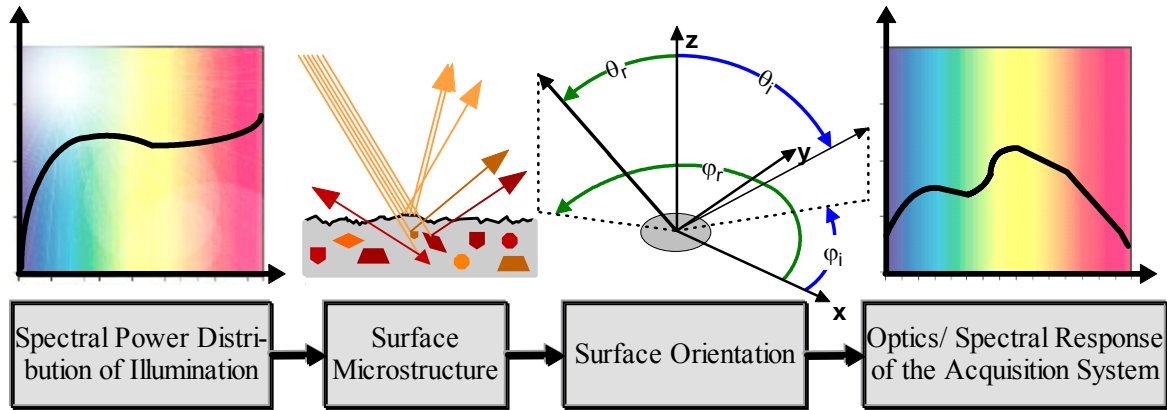


Figure 10: The main factors determining the intensity image of a scene.

$$X = k \int_{-\infty}^{+\infty} E(\lambda) x(\lambda) d\lambda \quad Y = k \int_{-\infty}^{+\infty} E(\lambda) y(\lambda) d\lambda \quad Z = k \int_{-\infty}^{+\infty} E(\lambda) z(\lambda) d\lambda \quad (27)$$

A *color space* is a way of organizing colors, i.e. the result of colorimetry is specified as a vector of a color space. So far we introduced two color spaces, namely the RGB and the XYZ color space. As color cameras emulate human color vision, most of them use three filter types modeled after the cone types. Hence they acquire their data principally within the RGB color space, or, more precisely, within their own, device dependent color space that is more or less similar to the RGB color space defined by the CIE. In view of that, we describe the response of a color camera as:

$$R_c = k \int_{-\infty}^{+\infty} E(\lambda) r_c(\lambda) d\lambda \quad G_c = k \int_{-\infty}^{+\infty} E(\lambda) g_c(\lambda) d\lambda \quad B_c = k \int_{-\infty}^{+\infty} E(\lambda) b_c(\lambda) d\lambda \quad (28)$$

where the subscript *c* indicates that the spectral responses, let alone the proportionality factors, and thus the total response vector is a camera-dependent approximation to the actual RGB tristimulus vector. We correspondingly specify the resulting multi-spectral image type as  $I(x, y) = (R_c(x, y), G_c(x, y), B_c(x, y))$ , or shorthand  $(R(x, y), G(x, y), B(x, y))$  if it is obvious from the context that we intend to refer to the latter expression and not to the CIE tristimulus values.

## 2.4 Summary

This chapter introduced, among other things, the main factors that determine the intensity image of a scene. Figure 10 summarizes them as shown in figure 10:

- **The spectral power distribution** of the radiation arriving at the scene.
- **The surface microstructure** of the scene, including its spatial surface properties such as its roughness and its micro-orientation as well as its absorption characteristics (e.g. its color).
- **The orientation** of the imaged surface patch, or, put another way, the scene geometry along with its spatial position and orientation relative to acquisition system and incoming radiation.
- **The optical and sensorial properties** of the image acquisition system such as its focal length, the size of its sensor elements or its spectral response.

These results explain why 3D vision systems have the potential of being more reliable than comparable systems based on intensity images: While the latter are involuntarily affected by all listed factors, the former are in principle independent of them but for the one they are actually interested in, namely the scene geometry, position and orientation.



### 3 State of the Art in Range Imaging

This chapter presents the state of the art with respect to (non-tactile) range imaging. In its first section, it uses the concepts introduced in the previous chapter to give terms such as range image or range image acquisition system a precise definition (3.1); in that context it also develops an abstract model of a range image acquisition system. A prerequisite for discussing the various approaches to range imaging is a set of common aspects; section 3.2 establishes it. The central section of this chapter gives a review of the most important ranging methods (3.3). The chapter concludes with a summary of its main results (3.4).

#### 3.1 Basic Ranging Terms

Following a proposal by Sanz [1989], we define a *range image* as set of several elements that represent a well-defined distance measurement between a common reference point and a point on an object's surface. Its synonyms are range map, range data, surface data, 2½D image, 3D data, 3D image, topographic map, surface distance matrix, to name only a few. Some of these names have to be used with caution as they carry a different meaning in other contexts, respectively with some authors, but all of them have been used in the literature to refer to range images.

We define a *range (image) acquisition system* simply by its ability to obtain range images, i.e. as any combination of hard- and software capable of acquiring range images of its visible spatial environment at a given time [Sanz 1989]. Some of the numerous synonyms for such a system are 3D imaging system, 3D digitizer, rangefinder, range imaging system, range sensor, 3D-sensor, or 3D-surface acquisition system. This work exclusively considers non-tactile range sensors; it models them as special digital pinhole cameras that acquire a certain distance value of surface points instead of their brightness (figure 11). This approach allows reusing the terms introduced in the context of pinhole cameras such as angular lateral resolution or effective focal length. Analogously to irradiance sensors, real-world range sensors have a nonzero sensor element size. As the former measure the average brightness, the latter acquire accordingly the average distance of an imaged surface patch rather than the well-defined distance of a surface point. To avoid mathematical subtleties regarding the definition of an average distance value, we always assume the distance to vary only negligibly over an imaged surface patch and to take on a finite value at all times.

We define a *depth map*  $z(x, y)$  as special type of range image that maps the image plane of a pinhole camera to the  $z_c$  camera coordinate – the depth – of the imaged surface point. Clearly a depth map can take on values greater than zero only and we are able to reserve the value zero for points for which no depth value could be obtained. Range and depth data are geometrically equivalent – given range data, the corresponding depth map can be computed and vice versa; so unless stated otherwise, we do not distinguish between range and depth data, respectively sensors, in this work.

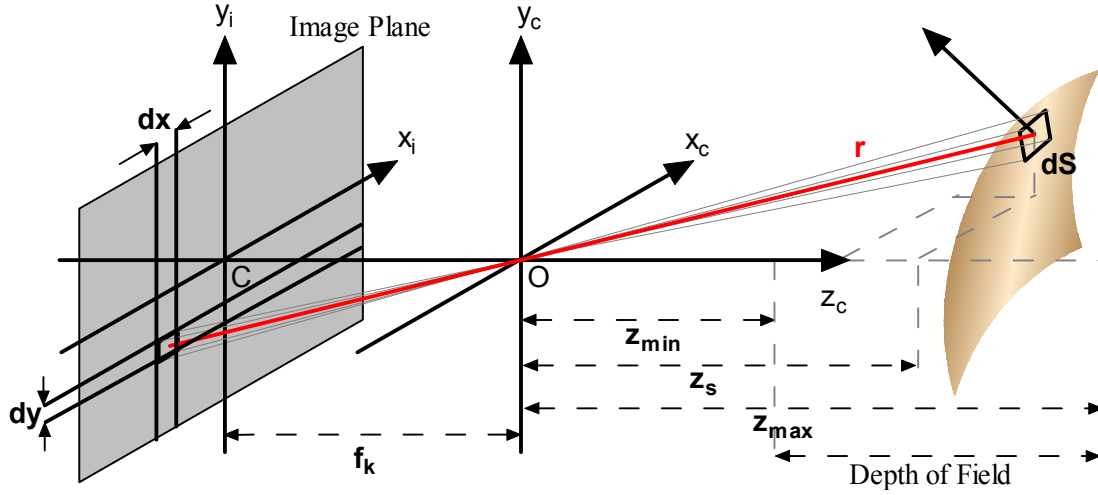


Figure 11: Abstract model of a range sensor. The standoff  $z_{\min}$  is the minimal,  $z_{\max}$  the maximal possible depth value and the depth of field is the distance between  $z_{\min}$  and  $z_{\max}$ .

The above pinhole camera interpretation implies that imaged depth data undergoes a perspective projection. Its effect is best exemplified with some ideal body such as a sphere of radius  $r$  with camera coordinate center  $(0, 0, b)^T$ . Its depth map, obtained by substituting the sphere equation into equation 14, is quite complex because of perspective foreshortening:

$$z(x, y) = f \frac{bf - \sqrt{r^2(x^2 + y^2 + f^2) - b^2(x^2 + y^2)}}{x^2 + y^2 + f^2} \quad \text{for } b^2(x^2 + y^2) \leq r^2(x^2 + y^2 + f^2) \quad (29)$$

As pointed out before, we may approximate perspective with parallel projection if the depth varies only negligibly over a scene compared to the distance scene-camera. We will assume just that on some occasions as orthographic projection allows a much simpler mathematical treatment of certain problems, in some cases making them tractable in the first place. E.g. the above depth map simplifies under parallel projection, i.e. by substituting into equation 15 instead of 14, to:

$$z_{\text{par}}(x, y) = b - \sqrt{r^2 - x^2 - y^2} \quad \text{for } x^2 + y^2 \leq r^2 \quad (30)$$

In this context, it is important to note that it is principally impossible to derive exact range maps from intensity images acquired under parallel projection as they do not contain any depth information by definition: translating a scene along the  $z_c$ -axis does not affect its orthographic image at all. Consequently, at most relative depth data, that is a depth map that contains an unknown constant depth offset and an equally unknown scale factor, can be extracted from an image taken under parallel projection (given no further information).

The above definition of a depth map as function allows drawing on mathematical concepts to characterize scenes. We call a scene continuous (differentiable etc.) if its depth map as mapping of the  $\mathcal{R}^2$  on  $\mathcal{R}$  is a continuous (differentiable etc.) function, i.e. if  $z(x, y) \in C^0$ .

There are several further important terms relating to range sensors: For a given configuration, the *standoff*  $z_{\min}$  of a range image acquisition system is the minimal possible depth of a scene point  $P$  it permits. We define the *depth of field* as the difference of the maximal acceptable depth  $z_{\max}$  and the standoff, i.e. as  $z_{\max} - z_{\min}$ . The *working space* for a given configuration is the volume defined by the depth of field and the field of view, i.e. it corresponds approximately to the product of the two. The *depth range* of a range acquisition system is the distance between the minimal  $z_{\min}$  and the maximal  $z_{\max}$  over all configurations. Figure 11 illustrates some of these definitions.

### 3.2 Aspects of Range Acquisition Systems

A prerequisite for discussing the pros and cons of various ranging approaches is establishing a common set of aspects. Of course, we will not consider each aspect for every implementation reported in literature; we are more interested in general insights that relate to the underlying ranging principle. In our discussion, we focus on the following aspects:

- **(Relative Spatial) Resolution:** We define the (relative spatial) resolution of a range imaging system as the number of its (potentially imaginary) sensor elements, i.e. as  $N_x \times N_y$ .
- **Data Rate:** The data rate follows as the product of the resolution with the number of images the system acquires per second, i.e. as pixels per second. Its inverse is called *pixel-dwell-time*.
- **Geometric Parameters:** What are the depth range, the possible values of the standoff, the volume of its working space, etc.?
- **Accuracy:** Accuracy and related aspects such as depth resolution are complex concepts discussed in detail in chapter 5. Also accuracy figures stated in the literature have to be taken with care: they are often on selected well-suited objects in controlled environments only, respectively it is unclear how the authors define accuracy. So the subsequent general discussion uses a preliminary, broader understanding of accuracy and associates it with the order of magnitude (e.g. nano- or centimeters) of the approach-characteristic deviation of a given depth measurement from the correct value, the ground truth, to be expected over a typical working space.
- **Robustness:** We define robustness as lack of restrictions regarding the ambient conditions (illumination, heat, etc.) under which a rangefinder can operate. A perfectly robust system is one that functions under all conditions; a less robust system would be one that requires that all sources radiating on the scene to be controlled and which can therefore only be used indoors.
- **Scene Constraints:** The scene constraints of a range acquisition system are the requirements a scene has to meet for the system to be able to image it. Due to their radiometric nature, all approaches discussed here implicitly require the scene surface to reflect at least some radiation into their direction. However, the systems differ significantly regarding any additional scene constraints. Some require a diffuse scene reflection, a single continuous surface, or impose the restrictive constraint of a static scene, while others cope with almost any kind of reflection and moving objects.
- **Safety:** Does the system represent a potential danger to humans, especially to their eyes?
- **Hardware Requirements:** What are the typical resource requirements in terms of hardware? Does the approach require highly specialized hardware or can it be implemented using standard components such as a digital video camera and a personal computer?

### 3.3 Range Imaging Methods

This chapter discusses the state of the art in range imaging. Ranging is by no means a novel concept; bats [Griffin 1958] and porpoises [Kellogg 1961] have used (ultrasonic) rangefinders successfully for millions of years. In 1903, Hülsmeyer [1904] demonstrated and later patented his Telemobiloskop, the first serviceable range acquisition system that fits our definition and that we would now classify as radar. Since then, many other ranging approaches have been proposed. Keeping in line with the topic of this work, we focus on approaches with the potential of acquiring accurate and dense depth data of unknown dynamic scenes in unconstrained environments rapidly, reliably and with reasonable effort in terms of resources. That is, we will not consider techniques such as shape from texture ([Stevens 1979], [Witkin 1981], [Blostein and Ahuja 1989]) that are principally unsuited for this task as they are limited to very special types of scenes (e.g. ones with a regular texture) or due to other principal fundamental downsides. As shown in figure 12, we employ two key aspects to classify the relevant ranging methods into broad categories:

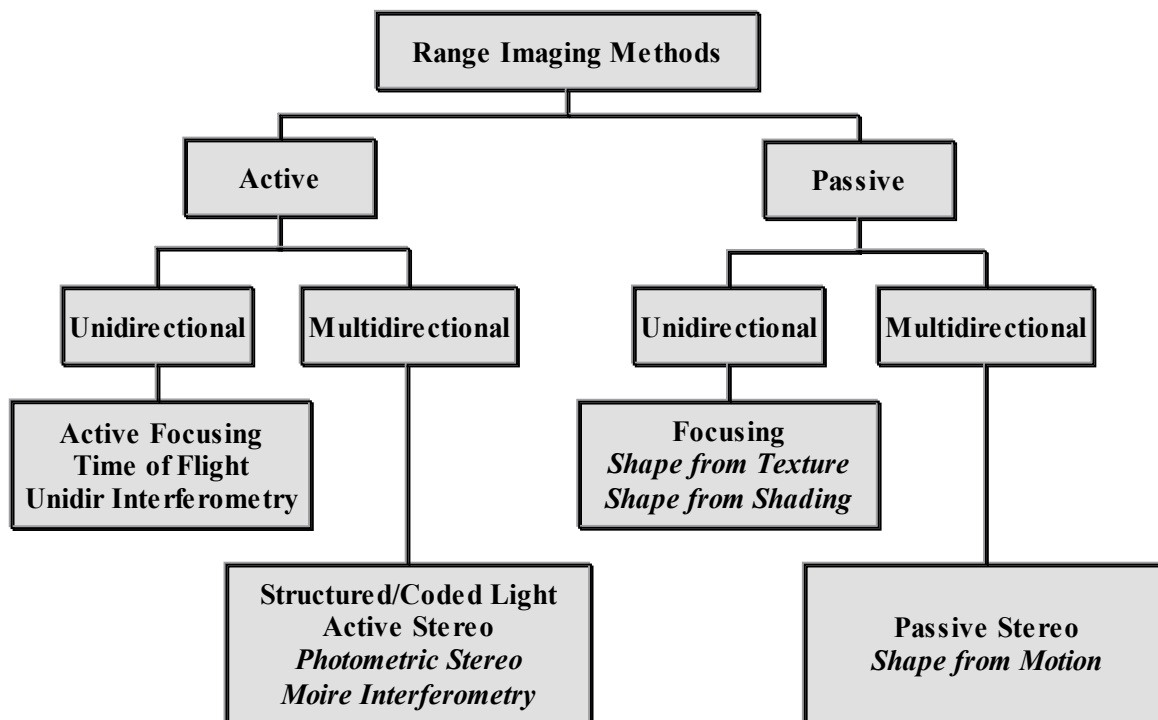


Figure 12: Overview of range imaging methods. *Italic print* implies a method gives shape only.

- **Active** ranging methods are methods that require a special source of radiation to be part of the range acquisition system. The opposite are **passive** methods that operate with the ambient radiation present, i.e. which can do without a source of radiation of their own.
- **Multidirectional** methods have to observe and/or radiate on a scene point from at least two points in space that are distinct relative to the scene point. They consequently suffer from the problem of occlusion or missing data: not all scene points might be visible from both points at the same time. **Unidirectional** or collinear approaches require only a single direction of view and/or illumination. Occlusion does for that reason not occur with them.

### 3.3.1 Time-of-Flight Ranging

The relationship between a signal's average propagation speed  $v$  and the distance  $\Delta s$  it travels during a time interval  $\Delta t$  is given by  $\Delta s = v \cdot \Delta t$ . *Time-of-Flight* (ToF) sensors exploit this equation by emitting a signal that travels at a known speed and by measuring the time that elapses until its echo is received. With  $v$  and  $\Delta t$  known, the distance  $\Delta s$  the signal traveled can be computed. The desired range value, which corresponds to the distance to the reflecting surface patch, follows as half this distance, i.e. as  $\frac{1}{2} \Delta s$ . As with most ToF systems transmitter and receiver are mounted coaxially, we classify ToF as an active, unidirectional approach to range imaging. The signals types commonly used for it are *ultra-sonic waves*, *radio-/microwaves* and *visible or near-visible light*. The former two types have important applications (e.g. ultra-sonic waves: low-cost rangefinders for consumer products such as cameras, radio/microwaves: synthetic aperture radar, GPS), but are not well suited for range measurements with high angular resolution due to diffraction limitations. They are for that reason not discussed here.

There are several different classes of ToF sensors (e.g. [Jähne 2002]). All of them are characterized by the fact that they modulate their carrier signal in some way and that they do not require coherent signals.

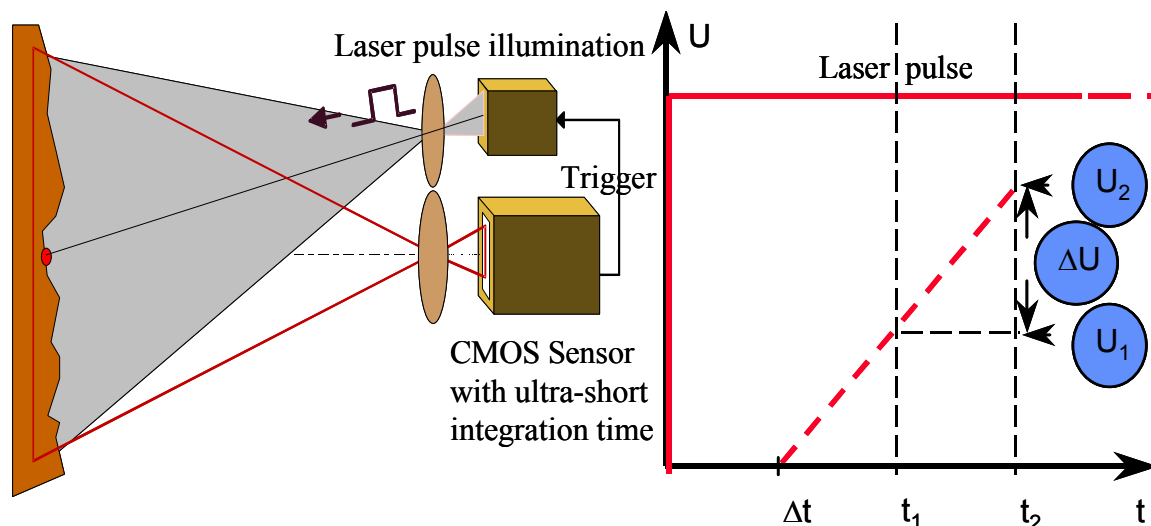


Figure 13: Principle of a Laser ToF rangefinder by Mengel et al. [2001].

The conceptually most straightforward ToF sensors use *pulse-modulation*, i.e. emit short signal pulses and determine the delay between sending a pulse and receiving its echo. Due to its short duration, such a pulse can have a high energy while still being eye-safe. This allows reducing the influence of ambient illumination, respectively achieving a high signal-to-noise ratio. However, generating such pulses requires an expensive light source capable of sharp rise and fall times. Also, due to the high propagation speed of light, the delay has to be quantified very accurately, e.g. for a ranging accuracy of  $\pm 1$  mm to within  $\pm 33$  ps. This is a very challenging task further complicated by the fact that the originally sharp pulses are often deformed by the reflection [Shirai 1987]. Consequently, many researches explore alternatives to directly measuring the delay, a representative example being the one by Mengel et al. [2001]: the latter suggest emitting a short, rectangular shaped laser pulse that is synchronized with the integration window of an image sensor (figure 13). After a travel time  $\Delta t$ , the reflected pulse generates the sensor signal  $U(t)$ . As discussed in section 2.3.1, its strength is proportional to the exposure time. Two distinct shutter times  $t_1$  and  $t_2$  give rise to the responses  $U_1$  and  $U_2$ . Then the two points  $(t_1, U_1)$  and  $(t_2, U_2)$  uniquely determine the linear sensor signal and the abscissa of its zero-crossing represents the pulse travel time  $\Delta t$ . This process is repeated with ultra-short exposure (integration) time, i.e. a very high frequency, to average out noise.

Another class of ToF sensors employs *continuous-wave (CW) modulation*, i.e. emits a continuous, periodically modulated signal. The most common types of modulation are *frequency-* (FMCW), and *amplitude-modulation* (AMCW). The travel time and thus the depth is then determined indirectly via the phase shift  $\Delta\phi$  between the in- and the out-going signal, i.e.

$$z(\Delta\phi) = \Delta\phi \cdot \lambda / 4\pi \quad (31)$$

where  $\lambda$  is the modulation wavelength. Accordingly, the accuracy of a sensor depends mainly on the accuracy of the phase difference measurement and the modulation frequency. So the accuracy of CW-modulation techniques is typically higher than the one of methods involving direct time measurement, respectively less effort is needed to achieve a comparable accuracy. The disadvantage of a modulation approach is that the phase difference can only be established within a range of  $\pm\pi$ . This implies the distance of the reflecting patch can only be determined up to half a modulation wave length, or, with other words, that the depth of field of CW sensors is limited.

To overcome this shortcoming, *pseudo-noise-modulation* (PNM, e.g. [Klein 1993]) has been developed; in its case the amplitude is randomly modulated. This permits uniquely (or at least with a large ambiguity interval) determining the phase-shift due to the travel-time; PNM systems therefore combine the large depth of field of pulse modulation with the high accuracy of CW-modulation.

The following aspects are shared by all ToF sensors: With them, a detectable portion of the signal has to be reflected from the surface back to the receiver. In general, this requires a sufficiently diffuse reflection of the signal – to receive the signal's specular reflection the surface normal would have to be approximately parallel to the incident direction of the signal. Their main advantage over triangulation methods introduced below is their unidirectional nature; they do not suffer from the missing-data problem. Also their accuracy depends only on the accuracy of the time, respectively phase measurement, and is consequently in principle independent of the distance to the object. The latter argument is somewhat imprecise, as of course the strength of the back-scattered signal and consequently also the accuracy decrease with growing object-distance. Nevertheless, the loss of accuracy with increasing distance is not nearly as pronounced as with triangulation systems.

The main disadvantage of ToF sensors – from the viewpoint of this work – is their need for custom-made and consequently often expensive hardware. Many ToF rangefinders have for that reason only a single or at most a few sensor elements and scan their field of view, e.g. via an rotating mirror. Their resolution and field of view is therefore often adjustable; their data rate can nevertheless be quite high (up to  $10^7$  pixels per second and more with some advanced systems) as it is mostly a matter of resources one is willing to spend. An inherent disadvantage of such mechanically moving parts is that they typically lose accuracy over time due to mechanic wear. Lately, quite a few distinct low-cost ToF rangefinders with larger arrays of up to 100 000 pixels and frame-rates up to 50 fps have been proposed, e.g. [Schwarte 1995], [Lange 2000], [Mengel 2001], [Canesta 2002]. Such systems benefit from recent advances in CMOS technology that permit integrating more and more additional functionality on an inexpensive image sensor, respectively allow producing affordable sensor with very short shutter times. As of today, all of these devices still have certain major drawbacks (e.g. susceptibility to background illumination, problems with moving objects or noise in general) and are limited to centimeter accuracy at best, under noisy real-world conditions in combination with high data rates typically to even worse accuracy.

### 3.3.2 Unidirectional Interferometry

*Unidirectional interferometry* (e.g. [Jähne 2002]) is the study of interference patterns created by the interaction of several sets of waves for the purpose of length or range measurement. It is often regarded as special case of CW-modulation; respectively, CW-modulation can be interpreted as interferometric approach. In line with most of the literature, we nevertheless treat it as separate ranging approach and distinguish between the two related techniques by considering only these methods as interferometric that require coherent radiation and exploit the modulation inherent to the radiation.

The standard signal type for interferometry is a laser, which implies the “built-in” modulation wavelength is in the range of 400 – 700 nm. As the phase difference can again only be determined up to  $\pm\pi$ , the effective depth of field of a straightforward interferometric system is extremely small. *Multi-interferometry* with multiple reference signals of typically closely spaced wavelengths (e.g. [Dändliker et al. 1995]) can be used to widen this interval, but usually only to the milli- or centimeter range. Another way to increase the depth of field is *white light interferometry* (e.g. [Notni et al. 1997]), also called *low-coherence interferometry*. It exploits that notable interference of light waves occurs only with coherent light. White light has typically a coherence length of a few wavelengths only. Corresponding systems send white light through two optical paths, one for measurement, and one as reference. Only if the optical path difference of the two paths is smaller than the coherence length, interference occurs; it becomes maximal for a minimal path difference. So the sought depth value is found by shifting the scene along the measurement path, respectively varying the length of the reference path and determining the position of maximal interference.

What has been said in the previous section about ToF rangefinders, especially about their hardware requirements, also applies to interferometric ones; the main difference is that the latter tend to achieve a much higher accuracy – typically fractions of the wavelength of light – over a much smaller effective depth of field.

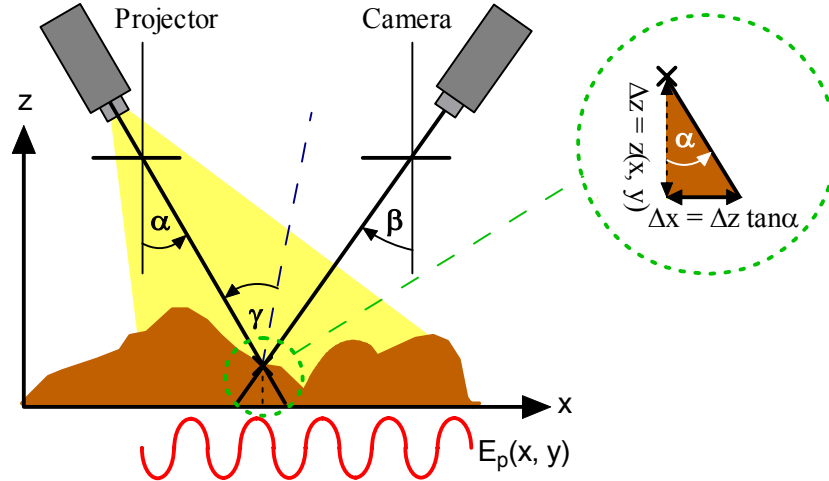


Figure 14: The principle of Moiré interferometry.

### 3.3.3 Moiré Interferometry

*Moiré interferometry* or *Moiré range imaging* ([Takasaki 1970], [Sanz 1989], [Wechsler 1990], [Klette et al. 1996]) is another example of interferometry. With the most common set-up for this active, multi-directional ranging method, a scene is illuminated by a projector and imaged with a black-and-white camera. Two periodically spaced gratings – one placed in front of the projector, the other in front of the camera lens – create interfering light signals. Assuming orthographic projection, the range data is derived from the resulting interference pattern as follows: Let the intensity modulation of the first grating be described by the following up-shifted sine wave of period  $d$

$$T(x, y) = 1/2 + 1/2 \sin(2\pi x/d) \quad (32)$$

If the projector emits a constant illumination, the resulting modulated irradiance  $E_p(x, y)$  at height 0 has the form  $k \cdot T(x, y)$  as shown in figure 14. If the surface is sufficiently matte, the reflection of the illumination is proportional to the cosine between the illumination direction and the surface normal. We can equate the change in illumination due to the nonzero height  $\Delta z$  of a given surface patch with that caused by a horizontal offset of  $\Delta z \cdot \tan(\alpha)$  at the zero height level. Consequently, we may describe the reflected illumination  $E_r(x, y)$  as (assuming a negligibly varying albedo of the scene):

$$E_r(x, y) = \cos \gamma \cdot k \cdot (1 + \sin(2\pi(x + \tan \alpha \cdot z(x, y))/d)) \quad (33)$$

If the cosine term changes negligibly compared to the illumination, we can include it into the constant factor  $k$ . Placing a second identical grid in front of the camera yields the following image:

$$\begin{aligned} I(x, y) &= k \left( 1 + \sin\left(\frac{2\pi}{d}(x + \tan \alpha z(x, y))\right) \right) \left( 1 + \sin\left(\frac{2\pi}{d}(x - \tan \beta z(x, y))\right) \right) \\ &= k + k \sin\left(\frac{2\pi}{d}(x + \tan \alpha z(x, y))\right) + k \sin\left(\frac{2\pi}{d}(x - \tan \beta z(x, y))\right) - \\ &\quad \frac{k}{2} \cos\left(\frac{2\pi}{d}(2x + z(x, y)(\tan \alpha - \tan \beta))\right) + \frac{k}{2} \cos\left(\frac{2\pi}{d} z(x, y)(\tan \alpha + \tan \beta)\right) \end{aligned} \quad (34)$$

The second term of the above product is due to the effect of the second grid on the image formation; the sum results from applying the identity  $2 \sin(x) \cdot \sin(y) = \cos(x + y) - \cos(x - y)$ . The last term of the sum depends exclusively on the depth of the scene patch. If the first three terms are above the resolution limit of the camera, we can identify each iso-brightness contour with a contour of constant depth. Adjacent fringes of minimal and maximally brightness then differ in depth by

$$|\Delta z| = d / (\tan \alpha + \tan \beta) \quad (35)$$

This permits extracting relative depth information from Moiré images starting out from an arbitrarily chosen contour. This process is called *phase unwrapping* and requires a continuous scene surface. It is not possible to tell the sign of the depth change between two adjacent iso-depth contours, i.e. to know which one is closer to the camera. A wrong decision with respect to the sign (or some other mistake) during unwrapping propagates into all subsequently unwrapped phases.

The main appeal of Moiré ranging is that it allows a human observer to directly “see” range values: with Moiré interferometry an image exhibits a striped pattern where contours of minimal or maximal brightness represent surface curves of equal depth. Its relative spatial resolution depends in one dimension on the number of visible contours (i.e. on the period  $d$  and the scene) and on the resolution of the camera in the other. Moiré interferometry is principally suited for real-time ranging of moving objects because only a single image has to be acquired and evaluated; its data-rate can therefore be very high. As ambient illumination can affect the measurement drastically, it has to be controlled. All in all, even though Moiré interferometry gives highly accurate range data up to the micron range, it is rarely used as generic, automated ranging technique, primarily because of its ambiguous output and its special hardware requirements in form of the gratings.

### 3.3.4 Depth-From-Focus and Depth-From-Defocus

*Depth-from-Focus* (DfF) is a passive, unidirectional ranging method (e.g. [Horn 1968], [Wechsler 1990], [Jähne 2002]). As discussed in section 2.2.2, only a single object-side plane of constant depth is in focus when acquiring an image with a lens camera. Points outside of this plane are imaged as blur circles. DfF methods acquire a series of images for distinct image plane distances, focal length settings or camera-scene distances. Determining for each point of the scene the image of best focus yields the point’s depth as the corresponding object distance. Of course, the scene has to have an appropriate visible structure whose blurring can be exploited. If it does not, a suitable pattern can be projected on it; depth-from-focus then becomes an active, but still unidirectional ranging approach. If camera and projector share the optical path, and if the depth of field is very small as with microscopy, only points in the focal plane are well-illuminated. So segmenting them becomes trivial. This is the principle of *confocal microscopy*, which also solves the problem of acquiring sharp intensity images of non-flat objects with a microscope.

The variation *Depth-from-Defocus* DfD (e.g. [Rioux and Blais 1986], [Ens and Lawrence 1993]) exploits the degree of blurring (e.g. the blur circle radii) for ranging. This can in principle be done given a single image, but for reasonably accurate results at least two images taken with different settings are needed whose relative blurring can be compared. Nayar et al. [1996] describe a corresponding active approach that achieves 30 fps for a resolution of 256 by 240 pixels (which could be increased easily) and millimeter accuracy in for a (ranging) depth of field of 300 mm. Applying this approach to dynamic scenes again requires at least two lens-camera combinations that share the same optical path. Nayar et al. [1996] also need to project their illumination through this path.

DfF is able to produce dense and highly accurate depth maps (up to microns) if the depth of field of a lens is very small. However, it requires a large number of image acquisitions. It is thus primarily suited – and employed – for obtaining highly accurate range images of static, microscopic scenes. DfD is well suited for real-time ranging with high-resolutions (implying a high data rate). Its accuracy, though, tends to be significantly worse than that of DfF. As stated above, with moving scenes it requires at least two cameras with a common optical path, i.e. a fairly complicated optical set-up. A fundamental disadvantage of both variations is a limited working space (unless the then necessarily static scene or the camera are moved mechanically): According to equation 18, the principle cannot be applied to object distances much larger than the longest focal length of the system, because then the effect of defocusing quickly becomes almost imperceptible and the accuracy of DfF and DfD degrades correspondingly.



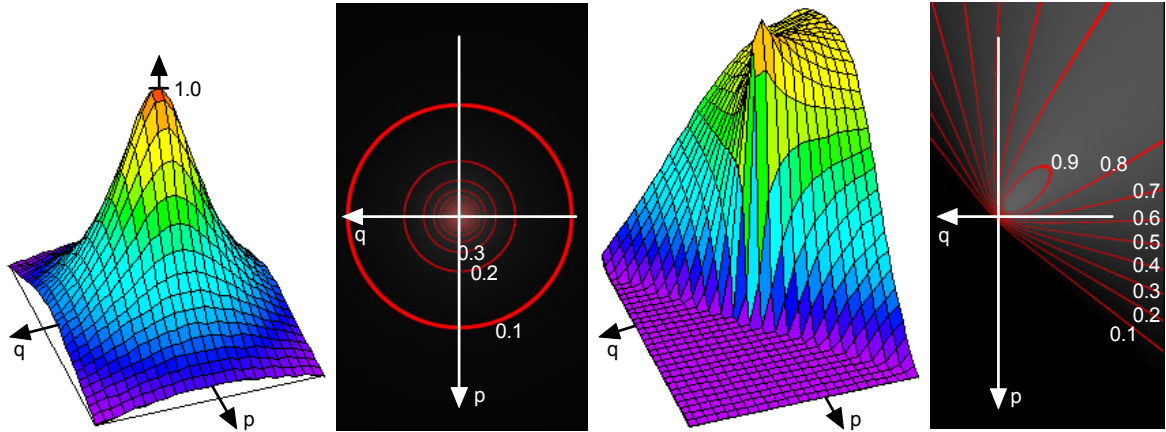


Figure 15: Reflectance maps of a Lambertian surface illuminated from  $(0, 0, 1)^T$  and from  $(1, 1, 1)^T$  drawn as function plots and as iso-brightness contours.

### 3.3.5 Photometric Stereo

As discussed in section 2.1, there is often a strong dependency between the orientation of a surface and its apparent brightness. With both a remote viewer and a remote illumination, that is in the case of parallel projection, the viewing and illumination direction are approximately constant over a scene. If a scene has furthermore a fairly homogenous surface, the orientation is the sole factor determining how bright each surface patch appears in the image. In the following, we investigate if it is possible to exploit this brightness-orientation dependency for ranging. To this end, we first need to introduce the concepts of the *gradient space* and of the *reflectance map*.

Given a (continuous) depth map and a scene point  $S$  with coordinates  $(x, y, z(x, y))^T$ , the straight lines  $(x, y, z(x, y))^T + \lambda(1, 0, \partial z(x, y)/\partial x)^T$  and  $(x, y, z(x, y))^T + \mu(0, 1, \partial z(x, y)/\partial y)^T$  lie in the tangent plane of  $S$ , provided the total derivative of the depth map exists for  $S$ . This follows directly from the definition of the (partial) derivative. The cross product of the directional vectors of the two straight lines is consequently normal to the tangent plane. We can therefore represent the surface orientation of a patch imaged at  $(x, y)$  as this cross-product:

$$\mathbf{n}(x, y) = \begin{pmatrix} -\partial z(x, y)/\partial x \\ -\partial z(x, y)/\partial y \\ 1 \end{pmatrix} = \begin{pmatrix} -p \\ -q \\ 1 \end{pmatrix} \quad (36)$$

We define the two-dimensional gradient space as the set of all real-valued pairs  $(p, q)$ , where  $(p, q)$  is the shorthand notation for the surface normal  $(-p, -q, 1)^T$ . This allows expressing the condition “surface orientation determines brightness” as

$$I(x, y) = R(-\partial z(x, y)/\partial x, -\partial z(x, y)/\partial y) = R(p(x, y), q(x, y)) = R(p, q) \quad (37)$$

where  $R$  is a mapping of the set of orientations into the set of brightness values. We call such a function  $R(p, q)$  *reflectance map* [Horn 1977]. Its range, the set of brightness values, is typically normalized in some way, usually to the dimension-less interval  $[0.0, 1.0]$ , where 1 represents the maximal brightness. The simplest example of a reflectance map is a Lambertian surface illuminated by a point source from  $(-p_s, -q_s, 1)^T$ . As shown in section 2.1.2, such a surface’s radiance is proportional to the cosine between the illumination direction and the surface normal. With  $k$  as constant normalization factor, its reflectance map therefore follows as:

$$R(p, q) = k \frac{1 + p_s p + q_s q}{(1 + p^2 + q^2) (1 + p_s^2 + q_s^2)} \quad (38)$$

This equation is clearly not invertible; for all irradiance values but the maximum 1.0, the set of solution forms a conic section in the gradient space. Figure 15 shows Lambertian reflectance maps for two illumination directions with selected iso-brightness contours. A way to come to a unique solution for the remaining irradiances is to move the light source to several locations, i.e. to use several distinct reflectance maps. Each one gives a conic section in the gradient space. Unless they coincide, two conic sections intersect at two points only [Horn 1986]. By assuming a smooth surface, it is usually possible to rule out the incorrect orientation. Alternatively, a third illumination direction directly yields the orientation uniquely, and one more robust against noise on top. In the special case of a matte surface, it also allows coping with locally varying albedo [Horn 1986].

This active, multidirectional approach called *photometric stereo* [Horn 1986], [Klette et al. 1996] can be applied not only to Lambertian, but to all scenes whose reflectance map contains suitable iso-brightness contours. Its relative spatial resolution is typically that of the camera used. Based on two or three images as above, it is only suited for static scenes. This constraint could be disposed of by using a different spectral interval for each illumination and a multi-spectral camera, at the cost of increasing the otherwise modest hardware requirements of photometric stereo. In any case, the reflectance map of the scene has to be known a priori. It is in some cases possible to obtain it experimentally, e.g. by analyzing an object of an identical material and with known surface orientation. However, – with the exception of some special cases – photometric stereo cannot deal with a locally varying albedo, let alone with non-constant reflectance properties as they occur with most scenes due to specular reflections or distinct surface materials. It also obtains the surface orientation only. Under the assumption of a smooth surface, the orientation information can in principle be converted to shape data. Klette et al. [1996] discuss several solutions to this problem. With all of them, the conversion is quite time-consuming, resulting in a fairly low data rate. Of course, due to the parallel projection, the output can at best be a depth map containing an unknown constant depth offset and an equally unknown scale factor. Even if we adapt photometric stereo to perspective projection, we cannot get rid of the latter. For this reason, we refrain from quantifying the accuracy of photometric stereo; it is certainly not the method of choice to obtain accurate depth maps in real-world environments, especially as it furthermore requires a controlled illumination.

### 3.3.6 Shape-from-Shading

Photometric stereo requires at least two images of a scene, each taken with a distinct illumination direction. Yet humans are often able to conclude the shape of objects from a single photograph. The corresponding active or passive unidirectional ranging technique *Shape-from-Shading* (SfS, e.g. [Horn 1977], see [Zhang et al. 1999] for a survey and comparison of six algorithms) tries to reconstruct the shape of an object given a single image. As before, we assume constant viewing and illumination directions, that the surface orientation determines the brightness and that the reflectance map is known. We have already seen that the reconstruction problem is ill-posed without further constraints; for a Lambertian reflectance map, we do not obtain a single solution for a given shading, but a conic section of the gradient space, for a linear one a straight line etc. We therefore assume the scene surface to be smooth. In that case, there are three different types of approaches to SfS:

*Propagation Approaches* represent the earliest SfS methods [Horn 1977]. They propagate the shape information along characteristic curves starting out from surface points with known orientation. We can comprehend this principle best by considering a linear reflectance map  $R(p, q) = h(ap + bq)$ , where  $h$  is known and invertible. From a given starting point  $S$  located at  $(x_0, y_0, z_0)^T$ , we take a small step into the direction  $\theta = \arctan(b/a)$ . We then compute the directional derivative  $m$  of the depth map  $z(x, y)$  at the point  $S$  in that direction as

$$m = p \cos \vartheta + q \sin \vartheta = \frac{pa}{\sqrt{a^2 + b^2}} + \frac{qb}{\sqrt{a^2 + b^2}} = \frac{pa + qb}{\sqrt{a^2 + b^2}} = \frac{h^{-1}(I(x, y))}{\sqrt{a^2 + b^2}} \quad (39)$$

where the last term contains only known quantities. Elementary calculus tells us we can reconstruct a smooth function up to a constant given its derivative. Consequently, we are able to recover the 1D depth map  $z(\xi)$  relative to the starting point  $z_0$  in the direction  $\theta$  since we know its derivative:

$$z(\xi) = z_0 + \frac{1}{\sqrt{a^2 + b^2}} \int_0^\xi h^{-1}(I(x_0 + \xi \cos \theta, y_0 + \xi \sin \theta)) d\xi \quad (40)$$

Such a 1D depth map is called characteristic curve. Given an initial curve of known depth values that is nowhere parallel to the direction of the characteristic curves, we use its points as starting points for the latter and obtain the shape of the whole scene by combining all characteristic curves.

It is possible to generalize the propagation approach from linear to arbitrary reflectance maps. In that case, we also need to know the orientation along the initial curve, i.e. we need  $p$  and  $q$  besides  $(x_0, y_0, z_0)^T$ , the reflectance map  $R(p, q)$  and the image  $I(x, y)$ . Given that, we compute  $I_x$ , the partial derivative of equation 37 with respect to  $x$ , by applying the chain rule for the partial derivative:

$$I_x = \frac{\partial I(x, y)}{\partial x} = \frac{\partial R(p, q)}{\partial x} = \frac{\partial R(\partial z / \partial x, \partial z / \partial y)}{\partial x} = \frac{\partial R(p, q)}{\partial p} \cdot \frac{\partial^2 z}{\partial x^2} + \frac{\partial R(p, q)}{\partial q} \cdot \frac{\partial^2 z}{\partial x \partial y} = rR_p + sR_q \quad (41)$$

Analogously, we obtain  $I_y = sR_p + tR_q$ , where  $t$  is the second partial derivative of  $z$  with respect to  $y$ . We now have two equations, plus the image and the reflectance map gradient; we cannot generally solve for the three unknowns  $r$ ,  $s$  and  $t$  with them, but we are again able to solve for a special direction  $\xi$ . Let's consider an infinitesimal step  $(dx, dy)^T$  in the image plane in the direction of the gradient of the reflectance map, i.e.  $(dx, dy)^T = (R_p, R_q)^T \cdot d\xi$ . The change in  $p$  caused by this step is

$$dp = \frac{\partial p}{\partial x} dx + \frac{\partial p}{\partial y} dy = r dx + s dy = rR_p d\xi + sR_q d\xi = (rR_p + sR_q) d\xi = I_x d\xi \quad (42)$$

In the same manner, we obtain  $I_y d\xi$  for the change in  $q$ . This implies a small step in the direction of the reflectance map gradient corresponds to a change in orientation proportional to the image gradient. This allows reconstructing the surface starting out from the initial curve and moving along these now arbitrarily shaped characteristic curves along the point-wise reflectance map gradient. Horn [1986] summarizes this fact with five ordinary first-order differential equations:

$$\frac{\partial x}{\partial \xi} = \frac{\partial R}{\partial p}, \quad \frac{\partial y}{\partial \xi} = \frac{\partial R}{\partial q}, \quad \frac{\partial z}{\partial \xi} = p \frac{\partial R}{\partial p} + q \frac{\partial R}{\partial q}, \quad \frac{\partial p}{\partial \xi} = \frac{\partial I}{\partial x}, \quad \frac{\partial q}{\partial \xi} = \frac{\partial I}{\partial y} \quad (43)$$

The disadvantage of propagation methods is that they propagate and thus accumulate errors; for that reason they have to be considered as unstable if the intensity data is noisy [Bakshi 1994]. In the latter, very common case, they cannot be relied upon to produce usable shape data.

*Local Methods* ([Pentland 1984], [Lee and Rosenfeld 1985]) infer the orientation of surface patches by analyzing the perceived intensity and its first two derivatives over small neighborhoods. They are motivated in part by the theory that biological visual systems carry out such a local analysis of images. They assume local neighborhoods to have certain elementary geometries, typically that of a sphere, and Lambertian reflectance. Given these strong constraints, they operate without further knowledge, specifically without a reflectance map. They also cope well with non-linear transformations of the image irradiance as they are often introduced in the digital imaging chain, as long the transformations are smooth and monotonic. Finally, they usually forgo iterative computations and are consequently potentially faster than the other approaches. Their main disadvantage is that they add a further, strong constraint (for which surfaces do local patches approximate a sphere?) to the already heavy restrictions of shape-from-shading and further need to rely on noisy 2<sup>nd</sup> derivatives.

*Global Minimization Methods* ([Horn 1986], [Klette et al. 1996], [Zheng and Capella 1991]) determine the shape of a scene by computing the shape that minimizes an error term also called energy or cost function. The latter is usually a weighted sum of a data or shading-consistency component and a surface smoothness component such as (with  $p$  and  $q$  shorthand for  $p(x, y)$  and  $q(x, y)$ ):

$$e = \iint (I(x, y) - R(p, q))^2 dx dy + \lambda \iint ((\partial p / \partial x)^2 + (\partial p / \partial y)^2 + (\partial q / \partial x)^2 + (\partial q / \partial y)^2) dx dy \quad (44)$$

This approach of stabilizing an ill-posed problem by explicitly introducing consistency with a direct model or a desired property, traditionally smoothness, as an additional quantitative objective for (usually least-square) optimization is referred to as *regularization*. Of course, choosing appropriate functionals and suitable weights for the regularizing term is both very important and tricky. Several additional or alternative expressions of the energy function have been proposed such as an integrability component enforcing  $z_{xy} = z_{yx}$  ([Zheng and Capella 1991]) or the unit normal constraint ([Horn and Brooks 1989]). In any case, the shape minimizing the weighted sum of the functionals is calculated. This approach to SfS thus turns out to be an optimization problem and is solved using one of the classic general-purpose, typically iterative optimization techniques. A problem of the latter is that they – besides being computationally expensive – usually need good initial values; otherwise they tend to get stuck in some local minimum or to not converge at all.

What has been said regarding the aspects, especially the constraints, of photometric stereo also applies to its sibling SfS. Moreover, for SfS mutual illumination and noise present an even greater problem. So Forsyth and Zisserman [1991] conclude that it is impossible to “obtain veridical dense depth or normal maps from a shading analysis”. Furthermore, all but local approaches require at least some reasonable initial values, which are generally not available. Horn [1986] discusses ways to obtain them; all of them have their major drawbacks and limitations. All in all, our conclusion concerning photometric stereo applies even more pronouncedly to SfS: it is an interesting area of research, but currently by no means a technique for obtaining accurate range data in practice.

### 3.3.7 Shape from Motion

If objects move relative to a camera or vice versa, it is under certain circumstances possible to derive their shape from the resulting image sequence or, more formally, time-varying image  $I(x, y, t)$ . The corresponding active, multidirectional ranging approach is called *shape-from-motion* or *dynamic stereo* ([Wong and Pugh 1987], [Klette 1996], [Negahdaripour 1998], [Jähne 2002]).

The simplest case is that of a moving camera and a static scene [Roach and Aggarwal 1980]. Let's assume it is possible to identify  $n > 0$  scene points in each of the  $m > 1$  images corresponding to  $m$  different points of view. This yields  $3n - 1 + 6(m - 1)$  unknowns;  $3n$  because each of the  $n$  scene point has 3 coordinates,  $3n - 1$  because one  $z$  coordinate can be fixed arbitrarily as in any case only dimensionless shape data can be obtained. Each of the  $m$  views results in 6 unknowns (3 rotation and 3 translation parameters, see section 2.2). By choosing the first viewpoint as reference point, its rotation and translation parameters can be set to an arbitrary value, so all in all the camera parameters give  $6(m - 1)$  unknowns. Each image provides  $2n$  equations, i.e. the total number of equations is  $2nm$ . In principle, already 5 points in two images (20 equations) suffice to determine the camera motion as well as the scaled coordinates of the scene points (20 unknowns), i.e. the shape. Of course, additional points will make the results more robust against noise or collinear equations.

Clearly the fundamental problem is to track scene points over the distinct images. This task is called *correspondence problem of shape-from-motion*. It is notably more difficult than the one of static stereo vision discussed below as the epipolar constraint does not apply and the search space for correspondence is consequently two-dimensional. A popular approach to solve it is based on two concepts, the *motion field* and the *optical flow*.

The motion field is a time-varying vector field formed by assigning a velocity or local displacement vector to each or selected image points in an image sequence. Such a vector connects the different images of a given scene point  $S$ . If  $S$  is imaged at the pixel  $(x, y)$  at the time  $t_i$  and at  $(x', y')$  at the time  $t_{i+1}$ , then the displacement vector at the time  $t_i$  is given by  $(x' - x, y' - y)$ . With other words, the motion field is the formal solution of the correspondence problem.

What we see in the images, however, is the optical flow, the apparent motion of brightness patterns in time-varying images [Horn 1986]. Objects that move in the front of a camera can give rise to optical flow, but a varying illumination, moving objects outside of the scene that cast a shadow or other effects may do so as well. However, it is all that is available to solve the correspondence problem for unstructured scenes. The basic assumption of shape from motion is that the optical flow is caused by the object movement and approximately identical to the motion field. That being said, it becomes immediately obvious that shape-from-motion is inherently unreliable and inaccurate. We therefore discuss a representative approach to determine the optical flow only briefly.

Let's consider the intensity of the image point  $(x, y)$  at time  $t$ . With  $u(x, y)$  and  $v(x, y)$  as the  $x$ , respectively  $y$  component of the optical flow/motion field, it is by definition equal to the intensity at the point  $(x + udt, y + vdt)$  at the time  $t + dt$ . Expanding the resulting equation into a Taylor series and ignoring the higher, non-linear terms yields the following constraint on the optical flow:

$$I(x, y, t) = I(x + udt, y + vdt, t + dt) \approx I(x, y, t) + dx \frac{\partial I}{\partial x} + dy \frac{\partial I}{\partial y} + dt \frac{\partial I}{\partial t} \Rightarrow u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \quad (45)$$

where we obtain the latter expression by dividing the former equation by  $dt$  and exploiting the identity  $u = dx/dt$ , respectively  $v = dy/dt$  for an infinitesimal time span  $dt$ . We have thus established a linear dependency of  $u$  and  $v$ , i.e. not a unique solution for the optical flow, but one with only one degree of freedom. However, that is all we can obtain locally, a fact known as the *aperture problem of the optical flow*. As with shape from shading, we introduce a constraint to arrive at a solution, namely we assume the motion field and thus the optical flow to vary smoothly almost everywhere. Just as in the case of the global optimization approaches to shape from shading, we determine the optical flow  $(u(x, y), v(x, y))$  as the function minimizing the integral

$$\iint \left( u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \right)^2 dx dy + \lambda \iint \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 dx dy \quad (46)$$

where the first term enforces the constraint on the optical flow derived above and the second represents a smoothness component (of course there are again many other ways to formulate the energy function). With other words, we again choose regularization to stabilize an otherwise under-constrained problem. This way we arrive at a solution, but one accompanied by the abovementioned problems and disadvantages of regularization. However, other than with shape from shading, there is a more or less realistic chance of coping with some degree of discontinuity of the optical flow, for instance at object borders, by using *discontinuity-preserving regularization*. There are two important approaches to this task (see e.g. [Jähne 2002]); the first is splitting up the integration area into several separate integration areas, each of which is smooth. Of course, determining such areas is a hen-egg problem that is typically solved via a complex iterative procedure. The second way to tackle this problem, called *controlled smoothness*, is to modify the smoothness term according to criteria such as local signs of discontinuity. For example, if there is an edge indicator such as a zero crossing of the Laplace-filtered image, the algorithm might attenuate the smoothness component proportional to the degree to which the depth discontinuity cues are present or even ignore it altogether.

We again conclude that shape-from-motion is a promising research subject, but at the current state of the art not suited for reliably obtaining accurate range images in real-time. We refrain for this reason from giving figures regarding aspects such as the achievable accuracy or data rate.

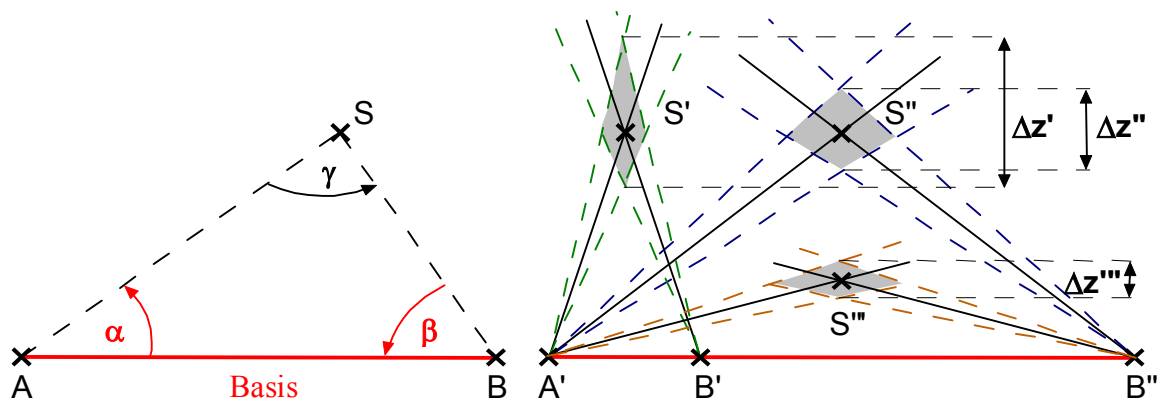


Figure 16: The principle of triangulation: Given a triangle  $ABS$ , we can deduce the spatial position of the point  $S$  from the length of the basis  $AB$  and the two adjoining inner angles  $\alpha$  and  $\beta$  (left). All other settings being equal, the depth error resulting from a given fixed error in the angle values decreases with growing baseline and increases with the object distance (right).

### 3.3.8 Triangulation

A triangle is defined by the length of one of its sides and two of its inner angles. We can exploit this fact for measuring distances or spatial coordinates of a point of interest  $S$ . To that end, we observe  $S$  from two known distinct positions  $A$  and  $B$ . These two points define a line (segment)  $AB$ , the so-called *baseline*, *basis* or *separation*. We then measure the angles enclosed by the line of view from each position to  $S$  and the basis as illustrated in the left part of figure 16. As the points  $ABS$  form a triangle, and as we know the length of the side  $AB$  and the two adjoining angles, the spatial position of  $S$  is determined uniquely. It can be computed easily, e.g. by intersecting the two lines of view. This simple measurement principle is called *triangulation*. It has been employed to measure distances e.g. for astronomy or geodesy for a long time. We are able to trace its use back to at least 1533 when Regnier Gemma Frisius proposed in his *Libellus de Locorum* to apply triangulation to accurately locate places.

We discuss the accuracy of ranging by triangulation in detail in chapter 5; for now, we observe with the help of figure 16 that the primary source of inaccuracy is an incorrect measurement of the two angles, given it should be possible to determine the baseline fairly accurately off-line. The depth error resulting from an incorrect angle depends on two factors, the object distance and the length of the basis: Figure 16 illustrates that, all other parameters remaining unchanged, the depth error decreases with growing separation and increases along with the object distance. So it seems that with respect to range accuracy, a triangulation system should be positioned as close as possible and should have a baseline as large as possible. One has to be careful with this reasoning as it ignores certain aspects. However, we show in chapter 5 that it holds true for most cases of practical interest and we will consequently employ it as helpful rule of thumb in the following.

The inherent downside of any triangulation technique is the *problem of occlusion* or *missing data*; a system can obtain range data only for scene points visible from both vertices  $A$  and  $B$ . Therefore triangulation systems are intrinsically better suited for scenes with gradual changes in depth and less so for scenes with significant and frequent depth gaps. Clearly the occlusion problem aggravates as the slope of the two rays of views become more dissimilar, that is with increasing baseline and as well with decreasing object distance. Consequently, we cannot reconcile accuracy and minimal occlusion because they are inversely related; there is no set-up that optimizes both aspects. It depends on the task at hand to which criterion we attach greater importance.

In sum, we cannot give a fixed figure on the accuracy of a given triangulation-based system; it depends strongly on the chosen components, the set-up and the object distance. Also, accuracy cannot be considered by itself: a wide baseline makes a triangulation system very accurate, but at the same time occlusion effects render it unusable but for flat objects.

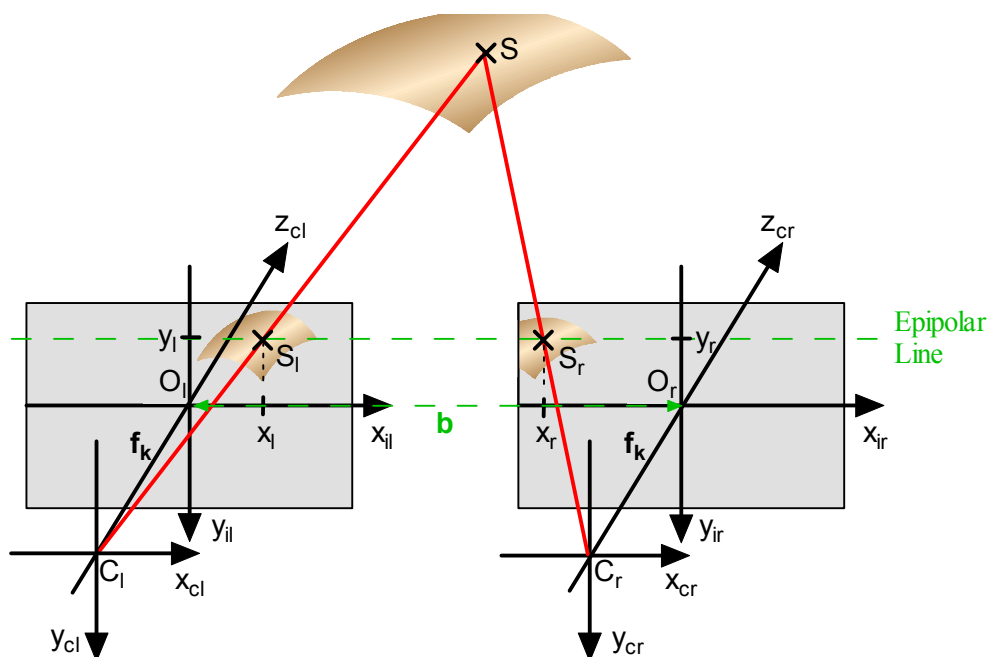


Figure 17: With stereo vision, images of a scene are taken from two or more distinct viewpoints; the spatial position of scene points such as  $S$  is then determined by locating the two image points  $S_l$  and  $S_r$  where  $S$  is imaged and by intersecting the two resulting lines of view.

### 3.3.9 Stereo Vision

Precisely how humans manage to perceive objects in depth is still an open question. What is known is that we derive the three-dimensional structure of our world at least to some extent by comparing the images acquired with the left eye to the ones acquired with the right eye [Horn 1986]. The generalization of this process, the acquisition of two (or more) images of a scene from different points of view and their transformation into a three-dimensional scene model is called (static) *stereo vision*. It has piqued the interest of researchers from early on, not only because it provides us with some insight into the way we “see in three-dimensions”, but also because it is a useful passive multidirectional technique to acquire range data (e.g. [Marr and Poggio 1976], [Marr and Poggio 1979], [Grimson 85], [Ayache 1991], [Cochran and Medioni 1992], [Faugeras 1993], [Zitnick and Kanade 2000], [Hirschmüller 2001], [Porr et al. 2002], [Lin 2002], see [Scharstein and Szeliski 2002] for an up-to-date comparison and evaluation of 22 stereo algorithms).

In this section, we discuss stereo vision from the latter perspective. We first outline its principle. As stereo vision is per se an ill-posed problem, algorithms have to impose constraints on the set of solutions to turn it into a well-defined one. We discuss the most common constraints before we move on to a review of existing stereo algorithms organized by their algorithmic principles. Next, we briefly consider some variants of stereo vision. We conclude the chapter with an evaluation of the principal advantages and shortcomings of stereo vision as ranging technique.

#### 3.3.9.1 The Principle of Stereo Vision

The first step of all stereo vision algorithms is to acquire the images to be compared. Stereo systems obtain them by using several distinct cameras, a single moveable camera such as one mounted on a mobile robot or by employing mirrors. In the following, we assume two distinct cameras (binocular stereo) as the most intuitive and most frequently used set-up, but everything being said principally applies to the other set-ups as well, including ones with more cameras. We further assume to know the internal optical and geometrical parameters of the camera(s) and their spatial position and orientation. Finally, we do not permit any movement between the acquisitions of the two images. To this end, either the two cameras have to be well synchronized, or the scene has to be static.

After the image acquisition, stereo algorithms register or match both images: they identify pixel pairs, one pixel from the first, the other one from the second image, that correspond to the same scene point. We call two such pixels a *conjugate pair* and refer to the task of matching as the *correspondence problem (of static stereo vision)*. Stereo algorithms typically approach it by designating one image as *reference image* and by solving it for its pixels or certain of its features. The correspondence problem represents the core of stereo vision – if it is solved, the task of stereo vision is essentially solved and vice versa. It is, however, fundamentally difficult to solve, and in the general case machines can still not solve it infallibly on their own. This is despite the fact that it is essentially a one-dimensional problem: Given an image point such as  $S_r$ , its conjugate pixel  $S_l$  in the other image is known to lie on the projection of the line of view of  $S_r$  onto the image plane of  $S_l$  (figure 17) for simple geometric reasons. This circumstance is often referred to as the *epipolar geometry* of stereo vision because each scene point together with the two optical camera centers defines an epipolar plane in space. Accordingly, we call the intersection of an epipolar plane with an image plane *epipolar line*; we are therefore able to rephrase that  $S_l$  is known to lie on the epipolar line defined by  $S_r$ , a fact known as the *epipolar constraint* of stereo vision.

Which other constraints reduce the complexity of the correspondence problem and how algorithms solve it follows below in detail. For now, let's assume the system identified a conjugate pair. It then computes the coordinates of the pair's common scene point by intersecting the two lines of views, i.e. via ray-ray triangulation. This is most easily done if the system set-up adheres to the *standard geometry* shown in figure 17. In its case, the two cameras have the same effective focal length, orientation and position, the only difference being that the second camera is translated along the x-axis common to both camera coordinate systems; the *baseline*  $b$ , the straight line connecting the two optical centers, is then a segment of this x-axis. With the standard geometry, the cameras acquire *rectified* image pairs, i.e. images with which epipolar lines coincide with image rows (or columns, but we focus on the former case in the following). The standard geometry is the set-up of choice if a scene is far away (relative to the baseline length). For close-range stereo, the cameras are usually rotated so that their optical axes converge and come closest within the center of the scene. This yields a more apt working space at the cost of non-rectified images. It is always possible to simulate the standard geometry for set-ups of practical interest by computationally transforming images onto a suitable common virtual image plane. With other words, we may assume it without loss of generality. In its case, a conjugate pair provides three simple equations containing three unknowns, namely the coordinates  $(x_s, y_s, z_s)$  of the imaged scene point  $S$  within the so-called cyclopean coordinate system, a coordinate system of a virtual camera exactly in-between and aligned with the two real ones:

$$\frac{x_l}{f} = \frac{x_s + b/2}{z_s}, \quad \frac{x_r}{f} = \frac{x_s - b/2}{z_s}, \quad \frac{y_l}{f} = \frac{y_r}{f} = \frac{y_s}{z_s} \quad (47)$$

Solving these equations for the three unknown coordinates  $x_s$ ,  $y_s$  and  $z_s$  yields:

$$x_s = b \frac{x_l + x_r}{2(x_l - x_r)}, \quad y_s = b \frac{y_l}{x_l - x_r}, \quad z_s = b \frac{f}{x_l - x_r} \quad (48)$$

With the standard geometry, the depth of a scene point is inversely proportional to the image plane distance  $x_l - x_r$  of its conjugate pair, the so-called *disparity* or *parallax*. We can easily convert the disparity into a depth value. Repeating this calculation for all matched items of the reference image transforms it into a – more or less dense – depth map.

In a next step, most algorithms employ interpolation to obtain disparity estimates for the pixels for which they could not establish it directly. Almost any of the classical interpolation methods can be used as long as special care is taken at depth discontinuities; ideally, an approach should segment the depth map into regions of continuous disparity and perform a separate interpolation for each region. Unfortunately, there is no reliable, generic way to determine such regions.



### 3.3.9.2 The Constraints of Stereo Vision

Stereo vision is – from a mathematical point of view – an ill-posed problem; algorithms have to impose constraints on the set of solutions to turn it into a well-defined one with a unique and stable solution. They derive these constraints – often implicitly – from a model of the world and thus the scene. For example, Marr and Poggio [1976] conclude from the cohesiveness of matter that “only a small fraction of the area of an image is composed of boundaries that are discontinuous in depth”. Constraints also contribute to an efficient solution of the correspondence problem, e.g. by narrowing down the search space. Its constraints are consequently a central aspect of a stereo algorithm; the following list discusses the most common constraints besides the epipolar one:

- **Uniqueness:** A given image item corresponds to at most one item in the other image. Lin [2002] differentiates between one-way (each reference item is assigned at most one match), asymmetric two-way (one-way uniqueness plus distinct reference items never share the same match) and symmetric two-way uniqueness (both images serve as reference images, both disparity maps respect asymmetric two-way uniqueness). The classic contraindicating example for the uniqueness constraint given by Marr and Poggio [1976] is that of a pixel receiving light both from a fish and the bowl it swims in, i.e. that of (semi-) transparent objects.
- **Photometric Compatibility or Color Constancy:** Two pixels form a conjugate pair only if they are photometrically similar, that is if their intensity or color values are sufficiently close according to some metric. As discussed in section 2.1, this assumption applies only to certain surfaces, i.e. it does not apply in general and can be completely off with specular reflection.
- **Geometric Similarity:** Two conjugate features such as two line segments need to be geometrically similar: that is, their length, their orientation etc. need to be reasonably close. This constraint does not hold in all cases: E.g. a reflection edge of the scene might very well show up as a single edge segment in the one image, but be split up into several separate segments in the other image because of occlusion or imaging noise.
- **Continuity or (Disparity) Smoothness:** The disparity varies smoothly almost everywhere. Due to the discrete nature of images the meaning of smoothness (or continuity, the two are used interchangeably in the literature in this context) is not well defined; the two are best read as “varying fairly slowly, with a low frequency compared to the sampling frequency”.
- **Figural Continuity:** The disparity varies smoothly along image features such as edges.
- **Disparity Limit:** Only matches resulting in a disparity within a given interval are permissible. This constraint is frequently used with close-range stereo as in its case only a certain range of depth/disparity values can realistically occur, e.g. because of known depth of field limitations.
- **Disparity Gradient Limit:** The disparity gradient describes the rate of change of disparity from the cyclopean perspective. Given a conjugate pair  $(x_l, y_l) - (x_r, y_r)$ , its cyclopean coordinates are defined as  $((x_l + x_r)/2, (y_l + y_r)/2)$ . We define the disparity gradient  $\Gamma$  of two pairs  $(x_{l1}, y_{l1}), (x_{r1}, y_{r1})$  and  $(x_{l2}, y_{l2}), (x_{r2}, y_{r2})$  as their ratio of disparity difference to cyclopean distance:
 
$$\Gamma = 2 \left| (x_{l1} - x_{r1}) - (x_{l2} - x_{r2}) \right| / \sqrt{\left( (x_{l1} - x_{l2}) - (x_{r1} - x_{r2}) \right)^2 + 2(y_{l1} - y_{l2})^2}$$
 After conducting a number of psychophysical experiments, Pollard et al. [1985] conjectured that the human vision system imposes an upper limit on it. Such a limit effectively restricts the maximal tilt of a scene surface with respect to the viewer. For a given meaningful limit, it is easy to construct a scene that violates this constraint, e.g. by tilting a planar object adequately.
- **Ordering (Preservation):** The spatial order of image items along epipolar lines is the same for both images: If the item A is to the left (to the right) of the item B in one image, it is also to the left (to the right) in the other image. This constraint tends to be violated e.g. with small objects residing in front of other objects. It always holds with a single continuous surface as scene.

### 3.3.9.3 Area-Based Stereo Algorithms

The most direct approach to the correspondence problem – predicated upon the photometric compatibility constraint – is to correlate intensity or color values. In principle, an algorithm could attempt to match each pixel with its photometrically most similar counterpart. However, the discrete and noisy character of real images renders such an approach unreliable. With continuity, adjacent pixels have by definition similar disparity. Clearly in that case groups of adjacent pixels tend to be much more distinctive than single pixels. The widely used area-based stereo algorithms presume distinctive enough to allow determining correspondence reliably: they compare and match local brightness patterns, usually regularly sized blocks (windows) of pixels.

How should the window size be chosen? As a coarse rule, the smaller the window size, the less computational effort an algorithm has to spend for matching. So a small window seems to be the best choice if efficiency is of relevance. However, such windows often do not contain enough information to determine correspondence dependably; large windows are more likely to do so. At the same time for all but fronto-parallel surfaces the disparity changes over a given window, typically smoothly, but at object borders also abruptly. Area-based approaches tend to smooth this disparity variation and to respond to discontinuities with surface fattening or shrinkage: surfaces of high brightness variance extend across occluding boundaries into adjacent surfaces of lesser variation. Finally, even given otherwise perfect conditions, area-based approaches can end up with a false match because with a non-constant disparity the brightness distribution of one window is a spatially condensed/expanded/or more complexly transformed version of the other, respectively partially not related at all. All the latter effects become more frequent and severe with increasing block dimensions. Most importantly, the latest effect implies that a large window size is not even a means to achieve a crude but reliable solution of the correspondence problem.

All in all, there is no universal answer to the above question; the sizes recommended in the literature vary accordingly strongly. A resort is to use windows of variable dimensions that are adapted according to local intensity variations alone ([Levine et. al. 1973]) or in combination with local disparity estimates ([Kanade and Okumoti 1996]). However, this technique is computationally expensive, which wipes out the very benefit of an area-based approach, its efficiency. So Fusiello et al. [1997] chose the via media of employing several windows of distinct preset sizes. In any case, such modifications alleviate, but cannot solve the problems inherent to a window-based approach; a truly useful adaptive approach would need to know the disparity values, i.e. the solution to the problem it intends to solve.

Irregardless of the chosen window size, all area-based algorithms quantify the similarity of window pairs by computing some kind of statistical correlation coefficient such as the sum of squared intensity (or color) differences (Kanade et Okutomi [1996]), the sum of absolute differences (e.g. Hirschmüller [2001]) or the normalized cross correlation [Cochran and Medioni 1992]. Porr et al. [2002] operate in the frequency domain: they use Gabor filters, localized frequency filters, to correlate windows via the phase difference of their filter responses. The motivation for the latter approach is its parallel efficiency and its relevance to the study of human stereo vision given that such phase-based stereo computations are believed to take place in the visual cortex of mammals.

To become more robust, some algorithms transform the blocks before computing correlation coefficients; respectively, such transforms are part of the coefficient as with the normalized cross correlation. Most such transforms are parametric, that is based on intensity distribution related statistical parameters, for instance the sample mean or the variance of a window. Less frequently used are non-parametric local transforms (Bhat and Nayar [1998] or Zabih and Woodfill [1994]) that do not rely on such distribution parameters. Examples are rank filters that sort the local intensity values and map a window on the ordinal rank of its central pixel or on a matrix whose entries represent the ordinal rank of the corresponding pixels. Such transforms tend to cope better with non-linear monotone intensity transformations between the two images, with outliers and with depth discontinuities, at a higher computational cost.

To find a match for a given reference block, area-based algorithms calculate the correlation coefficient for a set of candidates narrowed down by constraints such as the epipolar or the disparity limit constraint. They then match the reference block with the counterpart of maximal correlation. If several maxima occur, they typically cull which of them receives the most support from the local neighborhood according to the hypothesized scene model, e.g. which results in the smoothest disparity change. Given a match, the resulting disparity is either assigned to all pixels of the reference block or only to its central pixel. Most implementations ignore that some blocks might not have a match because of occlusion; only few are reported to try to identify such blocks and leave them unmatched. Exemplary methods for the latter approach are bi-directional matching, i.e. using both images as reference image and keeping only matches consistent between the two disparity maps, or discarding blocks for which even the best match has a low correlation coefficient.

Some area-based approaches operate in a coarse-to-fine manner, i.e. apply the steps described above iteratively on an image pyramid created by (typically Gaussian) smoothing and subsequent sub-sampling of the original images at decreasing rate (e.g. [Koschan et al. 1996]). There are several distinct strategies of dealing with the results of previous steps; matches are simply kept, used for guiding/constraining the next finer stage or only consulted if several equally likely match candidates occur at a finer stage. Reported advantages of this hierarchical approach are an increased robustness and efficiency – despite the overhead for creating the image pyramid.

In summary, area-based methods suffer from a number of serious problems and are for that reason inferior to other, more sophisticated stereo algorithms (e.g. [Scharstein and Szeliski 2002]). Their functioning depends on the scene: They tend to produce rather poor results with scenes of significantly non-Lambertian reflection, of strong disparity variation, with frequent depth gaps or with no or repetitive texture. However, they do work well with mostly fronto-parallel scenes of distinctive texture. Most importantly, they are very efficient, up to real-time capability, and result in dense depth maps; area-based methods are for that reasons primarily employed for applications such as teleconferencing where the combination of the latter two criteria outweighs accuracy aspects.

#### **3.3.9.4 Feature-Based Stereo Algorithms**

Feature- or token-based methods ([Marr and Poggio 1979], [Grimson 85]) do not rely on intensity distributions per se, but on image features such as edge pixels, corners, edge segments, contours, regions or objects. Matching thus takes place on a semantically higher, more abstract level, motivated in part by the theory that human stereopsis operates this way. Also from a strictly result-oriented standpoint, features exhibit certain advantages over uninterpreted intensity data: they occur less frequently and their appearance is less affected by effects such as a view-dependent reflection, changes in perspective and noise in general. Moreover, many attributes such as the edge polarity (low-high vs. high-low transition) can be determined robustly even given noisy conditions.

Naturally, feature-based methods start by extracting the features from the images. This is one of the basic tasks of image processing and related areas; for all listed features a number of detection methods exist, most of which can and have been used for stereo vision. A classic example is edge pixel localization by identifying zero-crossings of the Laplacian of a Gaussian. Next, a similarity measure is computed, with edge pixels e.g. by quantifying the difference of aspects such as the direction or scalar norm of the gradient, the sign of the intensity change, etc. Feature-based methods typically do not assign a match by simply looking at single similarity scores; they rather strive for a globally consistent match according to their scene model, e.g. by enforcing disparity smoothness or two-way uniqueness. Respectively, this consistency is inherent to certain features; e.g. with edge segments, inter-row consistency (figural continuity) of the matching follows automatically.

Feature-based approaches often operate in a coarse-to-fine manner as well. To that end, the image is smoothed in various degrees, starting with a large filter width that typically results in a few edges only and proceeding iteratively to finer scales (e.g. [Grimson 85]). The strategies of combining the results from distinct stages are the same as with area-based approaches; so are the pros and cons.

For the reasons discussed above, feature-based algorithms tend to be more reliable than area-based methods. Given that most features such as edges can be located with sub-pixel precision, they typically also produce more accurate results. Of course, feature-based algorithms have their downsides as well, the primary being that the scene has to exhibit the sought-for features in the first place. Considering edges, many scenes will not exhibit reflectivity or shadow edges. Only abrupt changes in surface orientation give rise to orientation discontinuity edges: corresponding feature-based stereo algorithms are for that reason better suited for polyhedral objects than for spherical ones. Also, edges need to have a certain orientation: edges parallel to the epipolar lines are rather useless for determining correspondence. Furthermore, specular and occlusion edges are highly viewpoint dependent and thus not only useless, but also misleading. Finally, a major reason for the effectiveness of feature-based approaches – the relative scarceness of features – also implies they produce only sparse depth maps of a-priori unknown resolution since they obtain disparity values exclusively for detected and matched features.

Researches have for these reasons tried to combine feature- with area-based approaches: they perform the matching using both photometric information and features (e.g. [Cochran and Medioni 1992]), giving precedence to the latter and using them as anchors that guide the area-based correlation of feature-less regions of the image.

### 3.3.9.5 Cooperative Stereo Algorithms

A central weakness of area-based stereo approaches is their local nature: they typically do not consider that a match at one point might support or clash with another match at a distant image point because of global constraints such as (two-way) uniqueness, continuity or the disparity gradient limit. Feature-based algorithms consider such global aspects, but produce only sparse depth maps. Cooperative or the related relaxation stereo methods (e.g. [Marr and Poggio 1976], [Jiang and Bunke 1997], [Zitnick and Kanade 2000]) combine both aspects, i.e. provide dense output and exploit the global interdependency of matches following from the world model. They do so using a massively parallel interactive process motivated by and modeled after certain biological nervous systems that master complex tasks in real-time, believed to include the human vision system.

This cooperative process operates in the three-dimensional discrete disparity space where each element  $(x, y, d)$  projects to the pixel  $(x, y)$  in the left and the pixel  $(x + d, y)$  in the right image. A match value function  $L$  assigns each element of the disparity space a real number that indicates the probability that its two associated pixels form a conjugate pair.  $L$  is initialized by evaluating one of the previously discussed correlation functions, e.g. by assigning  $(x, y, d)$  the inverse of the squared intensity difference of the left image point  $(x, y)$  and the right image point  $(x + d, y)$ . The cooperative process then iteratively updates the match value function: During each iteration, matches diffuse support for (increase the match value of) those conjugate pairs consistent with them according to the world model and inhibit (decrease the match value of) those that are not. E.g. for symmetric uniqueness, the match  $(x_1, y_1, d)$  inhibits all other matches within an inhibition area  $\Psi$ , that is all disparity space elements of the form  $(x_1, y_1, d')$  and  $(x', y_1, d + x_1 - x')$ . And for continuity, the match supports nearby matches  $(x_1', y_1', d')$  within its excitation area  $\Phi$ , e.g. within a fixed-sized box in the disparity space. Zitnick and Kanade [2000] propose the following update of the match function  $L$ :

$$L_{n+1}(r, c, d) = L_0(r, c, d) \cdot \left( \frac{S_n(r, c, d)}{\sum_{(r', c', d') \in \Psi(r, c, d)} S_n(r', c', d')} \right)^\alpha, \quad S_n(r, c, d) = \sum_{(r', c', d') \in \Phi} L_n(r + r', c + c', d + d') \quad (49)$$

where the consistent use of the photometric similarity term  $L_0$  ensures only reasonably similar pixel form conjugate pairs. This process iterates until convergence of the match value function. Then each pixel in the left image  $(x, y)$  is assigned the disparity for which  $(x, y, d)$  is maximal. Respectively, the pixel is classified as occluded if this maximum is below a certain threshold in its case.

With a fixed local support area, cooperative algorithms tend to blur edges; a support area limited to smooth surfaces would be preferable, but determining such regions is a hen-egg problem. Due to their three-dimensional, iterative approach, cooperative algorithms are computationally more expensive than e.g. area-based approaches; they are not suited for real-time applications on today's hardware. Another disadvantage is that the initial matches established during the first iteration have to be reasonable; otherwise the algorithm might converge not at all, or in the worst case to an incorrect solution. In general, however, modern cooperative stereo algorithms tend to produce dense depth maps of a quality superior to that of area based approaches [Scharstein and Szeliski 2002]).

### 3.3.9.6 Global Optimization Stereo Algorithms

Global optimization stereo algorithms interpret the correspondence problem in the broader context of the classical task of non-linear optimization. Generally spoken, they try to find the disparity function that – as a whole – balances the data momentarily provided by the camera best with the static scene model. Computationally this is done by determining the function that minimizes a cost or energy function made up of several competing components. Which ones depends on the respective world model, but almost all cost functions contain a data or photometric consistency term (e.g. the squared intensity difference of a conjugate pair) and most a continuity or smoothness term (e.g. the squared Laplacian of the resulting disparity map). The following distinct approaches to formulate and solve stereo correspondence as optimization problem have been proposed in the literature:

*Dynamic Programming* (DP, e.g. [Jiang and Bunke 1997]) represents a non-iterative optimization scheme of discrete, combinatorial nature. It rephrases the correspondence problem as the task of finding the set of ordered pairs  $(x_l, x_r)$  that minimizes the cost function, where each  $x_l$  occurs exactly once (asymmetric uniqueness, there are also formulations requiring symmetric uniqueness) and where  $x_l > x_l'$  implies  $x_r \geq x_r'$  (ordering constraint). DP is a non-iterative technique and thus significantly faster than the other global optimization methods discussed below. It is limited to optimization within a single image row. For this reason, some implementations add a post-processing stage of propagating results between distinct rows to refine the results and identify errors which e.g. inevitably occur if the scene violates the ordering constraint. According to the evaluation by Scharstein and Szeliski [2002], even such enhanced DP approaches are reported to be notably less accurate than other optimization methods.

*Graph-Based or Maximum-Flow* methods are a 2D optimization scheme of discrete, combinatorial nature. They reformulate the discontinuity preserving minimization of the stereo energy function as the task of finding the minimum cost-cut or maximum flow through a network graph [Boykov et al. 1998]. Their main advantage over other 2D optimization methods is their performance; Boykov et al. [2001] develop an efficient approximation algorithm applicable to stereo that provably gives results within a constant factor of the global minimum of the cost function. However, according to a recent evaluation, current implementations still miss real-time capacity by a wide margin ([Scharstein and Szeliski 2002]). With respect to accuracy, they are among the best stereo algorithms.

*Regularization* methods (e.g. [March 1988], [Roberts and Deriche 1996]) have already been discussed in the context of shape-from-shading and the optical flow; they are formulated in a similar manner for stereo correspondence using the smoothness constraint for (typically edge-preserving) regularization. The primary weakness of regularization as a way of solving stereo vision is – besides aspects discussed before such as the need for a good initial guess – the problem of occlusion [Lin 2002]. The fact that some pixels do not have a match at all cannot be integrated easily in the generic framework of regularization.

*Layered Stereo* algorithms (e.g. [Baker et al. 1998], [Lin 2002]) explicitly segment the scene into several distinct layers, each of which is modeled as a continuous surface. Most importantly, the layer model also includes a special tier for pixels that do not have a match because of occlusion. In this manner, layered stereo algorithms directly address the problem of discontinuities at object borders and the issue of occlusion. A welcome side effect of layered stereo is that it also provides a

segmentation of the scene into distinct continuous surfaces, respectively occluded areas. So the task at hand becomes twofold: Find an optimal decomposition of the scene into separate layers and at the same time an optimally smooth and consistent scene model. A way to algorithmically solve this is e.g. to minimize a now more complex energy function that also incorporates the task of segmentation. Layered stereo algorithms tend to give about the best results of all stereo algorithms; however, they are also among the most time-consuming (up to several hours for a single stereo pair) because of their more complex problem formulation.

### 3.3.9.7 Variants of Stereo Vision

With *axial stereo*, one camera is positioned on the optical axis of the other, only closer to the scene. In practice, a small offset is used, since the frontal camera would otherwise obscure the view of the posterior. Clearly such a set-up minimizes, if not overcomes the problem of occlusion. However, e.g. Nguyen and Huang [1992] report a significant loss of accuracy over comparable lateral stereo systems. Axial stereo is for this reason barely used in practice and not further discussed here.

Current passive stereo vision systems cannot cope with scenes that do not have any features, or exhibit regularly repeating patterns. Therefore *active methods* have been proposed. They illuminate the scene with a suitable pattern, e.g. a black-and-white one of sinusoidally varying intensity [Kang et al. 1995] or a colored one such as a rainbow spectrum [Koschan et al. 1996]. Such methods solve the problem of featureless, unstructured surfaces at least for scenes close to the acquisition system; of course the other aspects of stereo vision are not affected by actively illuminating the scene.

### 3.3.9.8 Conclusions On Stereo Vision

Today's stereo vision algorithms are able to produce depth maps of high resolution; most compute a depth value for every pixel of the reference image. Their relative spatial resolution depends consequently primarily on the camera used. Their data rate varies strongly with the considered approach and its implementation; according to a recent benchmark test ([Scharstein and Szeliski 2002]) the computation time for a single disparity map (without rectification) varies from as low as a tenth of a second (for area-based stereo) up to many minutes (global optimization) for a single frame of 400 by 400 pixels, with a roughly inversely proportional relationship between accuracy and efficiency.

The difficulty of putting a figure on the accuracy of a triangulation system has been discussed before. Naturally this also applies to stereo vision systems; with them also another aspect becomes relevant: with increasing baseline, i.e. more divergent angles of view/perspectives, the correspondence problem becomes more difficult. Most stereo systems are for that reason limited to a small separation; wide baseline stereo is often considered as a related, but nevertheless distinct approach of its own that is typically based on three or more cameras. Accordingly, the accuracy of standard binocular stereos tends to be on the low end of that of triangulation systems for simple set-up reasons. The accuracy of stereo further depends on the algorithm considered as well as to a significant extent on the scene. If the scene complies with the world model of the algorithm (typically if it is opaque, smooth and approximately Lambertian) and has many distinct optical features, the accuracy of a first-rate stereo algorithm will be high; conversely, almost all stereo algorithms are less accurate if the scene is optically unstructured and fail with scenes that collide with their world model. Finally, their accuracy (particularly in the least-square sense) suffers from the fact that even the best stereo algorithms occasionally produce false matches where the stated depth value differs arbitrarily from the ground truth. Scharstein and Szeliski [2002] show the rate of false matches to be in the range of a few percent with state-of-the-art algorithms and realistically complex scenes.

With respect to all other aspects, stereo systems compare favorably; their hardware requirements are modest, they have a wide dynamic range primarily restricted by the permissible baseline of a given set-up, their robustness is excellent and passive systems pose no danger or nuisance whatsoever to humans. In sum, their only major limitation is their current inability to reliably obtain accurate high-resolution range data of realistic, potentially optically unstructured scenes in real-time.

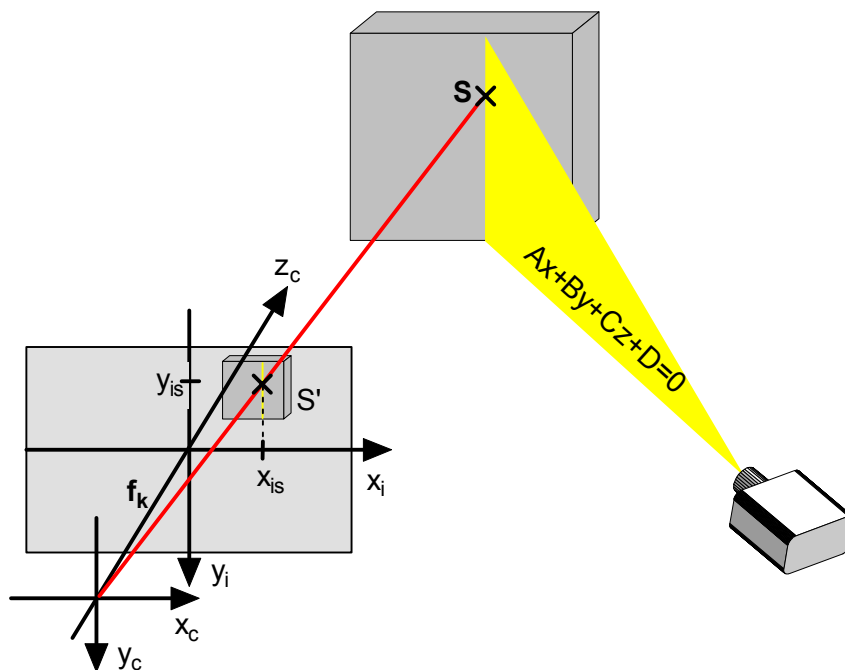


Figure 18: The principle of the Structured Light Approach.

### 3.3.10 Structured Light

This section analyses the state of the art regarding the *Structured Light Approach* SLA ([Shiray 1972], see [Battle et. al 1997] for a survey). We start with its principle, followed by a review of several ways of implementing it and conclusions.

#### 3.3.10.1 The Principle of the Structured Light Approach

The SLA is an active, multidirectional ranging technique. We treat it as modification of stereo vision where a directional illumination device replaces one of the cameras of stereo. This device, which is abstractly modeled as digital pinhole projector, illuminates the scene with a light pattern, while the remaining camera acquires one or several images of the scene (called *pattern images*) as shown in figure 18. A point on the projection slide results in a projected ray, a straight line in a projected plane in 3D space. So given a suitable dot or line pattern, the projector can be said to illuminate the scene with *light rays*, respectively *light planes*. In the following, we focus on the more important case of light plane projection, yet everything said will mutatis mutandis apply to a light ray projection as well. Barring degenerate configurations, the intersection of a plane with a straight line is a well-defined single point in 3D space. Accordingly, if it is known which plane illuminates an imaged scene point, the point's 3D coordinates are given by the intersection of this plane with the line of view of the camera, i.e. via plane-ray triangulation. With SL, the correspondence problem of stereo vision is consequently replaced by the simpler task of locating and identifying light planes in the pattern images, a task known as *identification* or *indexing problem*.

There is no agreed way of formally describing SL systems. For this reason, we propose the following formalism: We treat a slide as discrete image  $I_p(i, j)$  of resolution  $n_p$  by  $m_p$  and of codomain  $Q_p$ , where  $Q_p$  is a set of  $q_p$  elements representing distinct gray levels or colors. As a projection pattern may comprise  $t$  distinct slides, we represent it as time-varying slide  $I_p(i, j, k)$ , where  $1 \leq k \leq t$ ,  $t \geq 1$ . In the following, we interpret slide and pattern coordinates as elements of the vector spaces  $\mathfrak{R}^2$ , respectively  $\mathfrak{R}^3$  (over the scalar field  $\mathfrak{R}$ ). Doing so gives access to the well-defined algebraic structure of these vector spaces and allows using canonical concepts such as the L-infinity-norm with coordinates. We call vectors of the  $\mathfrak{R}^2$  ( $\mathfrak{R}^3$ ) *valid* slide (pattern) coordinates if they are elements of the domain of  $I_p(i, j)$ , respectively  $I_p(i, j, k)$ , and *invalid* if they are not.

A pattern slide may contain arbitrarily oriented lines. Parallel lines avoid intersections, simplify the formal description of a pattern and are used exclusively throughout the literature. For this reason, we assume a pattern of  $n_p$  parallel vertical straight lines – also called stripes or columns to emphasize the discrete nature of the slide – of slide equations  $i_p = k$  ( $1 \leq k \leq n_p$ ). The choice of vertical lines does not imply a loss of generality as we make no assumptions about the geometric set-up of the SL system (including the projector's orientation) but the one that the camera's optical center does not lie in any of the projected planes. A point and a straight line define a plane unless they are collinear. The projector's optical center has a distance of  $f_{kp} > 0$  from the slide. We may for that reason equate light planes with pattern lines, in our case additionally with certain horizontal slide positions. That is, we can restate the identification problem as that of mapping image positions  $(i_i, j_i)$  on the slide coordinate  $i_p$  of the plane illuminating the scene patch imaged at  $(i_i, j_i)$ .

Solving the identification problem is quite simple if each slide contains a single straight line only. If many planes are projected simultaneously, keeping them apart becomes significantly more difficult. In general, this is only possible if each projected plane is visually unique over time, i.e. with an *encoded pattern*. The following definitions give this term a well-defined meaning:

- A *codeword-reading-rule* or *codeword function*  $\sigma$  is a mapping of the integers  $\{1, \dots, s\}$ ,  $s > 0$ , on a subset of the  $\mathcal{R}^3$ , namely the Cartesian product  $S = \{-n_p + 1, \dots, -1, 0, 1, \dots, n_p - 1\} \times \{-m_p + 1, \dots, -1, 0, 1, \dots, m_p - 1\} \times \{1, \dots, t\}$ . The first two elements of a vector from  $S$  are called its *relative spatial slide coordinates*; the third is referred to as its *temporal pattern coordinate*.
- The range of a given codeword function is a set of  $s$  vectors of the  $\mathcal{R}^3$ . Projecting each vector on its relative spatial slide-coordinates, applying the L-infinity norm to the  $s$  vectors of the  $\mathcal{R}^2$  and determining the maximum  $d$  of the resulting  $s$  numbers is a well-defined operation. The integer value  $d$ ,  $0 \leq d \leq \min(n_p - 1, m_p - 1)$ , is called *slide margin*. A position on the slide  $(i_p, j_p)$  is called *within the margin* if  $i_p \leq d$  or  $j_p \leq d$  or  $n_p - d < i_p$  or  $m_p - d < j_p$ ; otherwise it is called *central*. For valid central slide coordinates  $(i_p, j_p)$ , the component-wise  $\mathcal{R}^3$  addition  $(i_p, j_p, 0) + \sigma(k)$ ,  $1 \leq k \leq s$ , yields again valid pattern coordinates.
- For a valid central slide position  $(u, v)$ , a codeword function  $\sigma$  defines a projection pattern  $I_{p(u,v)}$  of its own (via  $I_{p(u,v)}(i, j, t) = I_p(i, j, t)$  if  $(i, j, t) = (u, v, 0) + \sigma(k)$  for a certain integer  $k$ , where  $1 \leq k \leq s$ , undefined otherwise). It is termed *subpattern* of  $(u, v)$ .
- The *codeword* of a valid central slide position  $(i_p, j_p)$  under  $\sigma$ , denoted by  $c_\sigma(i_p, j_p) \in Q_p^s$ , is defined as the sequence  $q_1 \dots q_s$ , where  $q_i = I_p((i_p, j_p, 0) + \sigma(i))$ . Invalid positions or ones within the margin have by definition no codewords associated with them. The set of all codewords of a pattern is called its (*block*) *code*  $C$ ; it is a subset of the code space  $Q_p^s$ . In this context, the mapping of slide positions to codewords is called *encoding schema*  $c_\sigma$ , the set  $Q_p$  *code alphabet*, its elements (*code*) *symbols* and the integer  $s$  *codeword length*.
- A pattern  $I_p(i, j, t)$  is called *encoded* (under  $\sigma$ ) if it comprises less slides than projected planes ( $t < n_p$ ), and if the encoding schema is injective with respect to the horizontal position. Formally, this is expressed as:  $i_p, i_p' \in \{d + 1, \dots, n_p - d\} \wedge j_p, j_p' \in \{d + 1, \dots, m_p - d\} \wedge c_\sigma(i_p, j_p) = c_\sigma(i_p', j_p') \Rightarrow i_p = i_p'$ . This allows associating a horizontal slide position and thus a light plane with a given codeword or subpattern (but not necessarily vice versa). It is important to note the difference between codeword and subpattern: with an encoded pattern both identify a plane, yet the former is a mathematical abstraction, the latter a physically existing signal block.
- An encoded pattern is called *redundantly encoded* or *h-error-detecting* ( $h \geq 1$ ) if two codewords that refer to distinct light planes have a Hamming distance (the count of positions in which the two words have distinct symbols) that is greater than  $h$ .

Variants of the SLA that employ encoded patterns are accordingly called *Coded Light Approach* (CLA). In the following, we discuss several aspects that allow broadly categorizing the many different ways to exploit the SL/CL principle for ranging.



*Projection Model (Perspective or Parallel Projection):* Most SL approaches consider perspective projection, but some are based on the simpler parallel projection model. To approximate the latter, camera and projector have to be far away from the scene, somewhere between 20 [Asada et al. 1988] to 30 [Wang et al. 1987] times its size. For larger scenes, this is quite unpractical. As discussed before, only the shape, that is the surface data scaled by an only approximately known factor and containing an unknown depth-offset, can be obtained with orthographic projection.

*Type of Illumination (Color or Gray Levels):* A core aspect of a SL system is the type of its illumination. We distinguish between color (monochromatic or not) and gray level patterns (binary, that is black and white, or made up of more than these two gray-shades).

*Dimensionality of Encoding (1D or 2D):* If the codeword function of an encoded pattern is injective, we call its encoding *two-dimensional*. In this case, a codeword can be uniquely associated with a slide position  $(i_p, j_p)$ , i.e. with a light ray. As mentioned above, encoding of the horizontal position (*1D encoding*) suffices for triangulation. With 2D encoding, computing 3D coordinates amounts to intersecting an illuminating ray and a line of view (ray-ray triangulation), or rather determining their closest approximation. The magnitude of the latter allows making out calibration problems or misidentifications, respectively represents a good indicator of ranging accuracy. Under the same token, 2D encoding allows checking whether an identified ray originates from the expected epipolar line and discarding the id if it does not. Finally, it is convenient for projector calibration, e.g. for establishing the distortion of its lens, and simplifies considering the projector's lens distortion during the triangulation step. However, straight 2D encoding roughly squares the number of necessary codewords over 1D encoding. The resulting decrease in relative spatial resolution or reliability tends to more than offset the listed benefits.

*Encoding Technique (None, Point-wise, Temporal or Spatial):* We call an encoding with the trivial codeword function  $\sigma: \{1\} \rightarrow \{(0, 0, 1)\}$  *point-wise* encoding. In its case, each codeword corresponds to a distinct color (or gray level) and vice versa. Clearly such a pattern has no margin (i.e.  $d = 0$ ) and requires employing at least as many different colors as light planes ( $q_p \geq n_p$ ). In general, however, only very few distinct colors, if any, can be distinguished dependably in the pattern image. For this reason, point-wise encoding is often considered too unreliable to encode a large number of light planes. Given only  $q_p \ll n_p$  reliably discernable colors/gray levels, there are two foremost encoding techniques that overcome the limitations of a point-wise approach.

*Temporal or time-space encoding* uses  $t > 1$  distinct slides, i.e. a time-varying pattern. Typically, a codeword function of a temporally encoded pattern has the form  $\sigma(k) = (0, 0, k)$  for  $1 \leq k \leq t$ . So the codeword length equals the number of slides ( $s = t$ ). Identification is done by establishing for each image pixel the sequence of projected color/gray levels. Temporal encoding has one decisive disadvantage: To cope with moving scenes, to speed up the data acquisition and to make do with a simple projector, systems should be able to compute a depth map from a single snapshot of a scene. These capable of doing so are called *one-shot systems*. Formally, they are characterized by  $t = 1$ . Clearly it is by definition impossible to build a one-shot system on the basis of temporal encoding.

*Spatial encoding* opens up a simple way of building one-shot systems. It increases the codeword length by employing spatially extended codewords. Its patterns are consequently characterized by a nonzero margin and subpattern size. Their formal description is complicated by the fact that in their case often several adjacent physical slide elements are grouped into conceptual units. The values such units take on are termed *pattern primitives*. Other than physical slide elements, primitives can have distinct shapes. Nevertheless, as long as the units are of uniform rectangular size, the above formalism still holds with the primitive abstraction. In this case, we simply treat a slide as being made up of primitives; its resolution of  $n_p$  by  $m_p$  then refers to primitive units. As the resulting slide has the familiar lattice structure, and as its domain – the set of primitives – can be identified with the abstract set  $Q_p$ , we are able to apply the above definitions without changes to spatial encoding. We only have to bear in mind that even with a slide margin of 0 on the pattern primitive layer such patterns may have a margin of  $d > 0$  at the physical slide level. Unless stated otherwise, we de-

scribe spatially encoded patterns in the following on the pattern primitive layer. Spatial encoding increases the number of potential codewords almost arbitrarily over point-wise while retaining its one-shot nature, yet at a price to be paid during decoding (with CL, solving the identification problem is also called decoding): The appearance and the spatial relationship of primitives as (if at all) visible in the pattern image can differ substantially from what has been originally projected, e.g. because of the scene texture or foreshortening. This makes the task of recognizing primitives, let alone subpatterns, in the pattern image potentially very demanding. As the patterns are composed of shapes of principally non-negligible size rather than dimensionless color or gray level points, only certain boundaries (or other conceptually infinitesimal features) of primitives should be used for triangulation for accuracy reasons. Which implies it is not possible to obtain a depth value for each image pixel; the non-interpolated relative spatial ranging resolution is subject to the number of suitable boundaries visible in the pattern image, which in turn depends on the pattern design and on the scene. As we cannot factor in the latter into a general discussion, we take the horizontal slide resolution  $n_p$  as measure for the resolution of the depth map, given it represents an upper bound on the number of observable vertical boundaries. It is important to note that due to the discrete nature of real-world cameras and projectors this restriction applies to some extent to other SL/CL techniques as well. Finally, as the projected subpattern are not even approximately spatially dimensionless, depth values can only be obtained for surfaces large enough to reflect them more or less integrally. Of course, the impact of this restriction – which roughly corresponds to the continuity constraint of stereo vision – depends on the (actual physical) size of the subpatterns.

*Type of Encoding (Unique, Periodic, Sparse):* The more distant (according to a suitable metric of the space  $Q_p^s$  such as the Hamming distance) codewords are, the more robust the recognition process becomes. Yet in general the more redundant a code is, the less words it has, implying a trade-off between robustness and a large number of codewords/a high relative spatial resolution (for a fixed space  $Q_p^s$ ). A way to reconcile these two conflicting objectives is *periodic “encoding”*, where codewords are repeated with a horizontal period  $p$ . Formally, this is expressed as  $I_p(i, j, t) = I_p(i + p, j, t)$ . Periodic encoding permits a high resolution even with a robust code of few words. It does, however, create a ranging ambiguity – as several distinct light planes are encoded with the same codeword – that cannot be resolved unless certain scene constraints apply. With a low pattern frequency, a known limited depth of field (the disparity limit of stereo) might allow identifying the correct plane from the set of all planes associated with a given codeword (if all others exceed the disparity limit). In general, absolute range values can only be obtained for a globally continuous scene surface in combination with at least one unique codeword: all remaining ambiguities are then resolved by exploiting spatial adjacency relations relative to the latter (an id propagation similar to the phase unwrapping of Moiré interferometry). Interspersing at least one unique codeword into an otherwise non-encoded or periodically encoded pattern is called *sparse encoding*.

Many SL/CL patterns consist of a single slide and are a function of the horizontal slide coordinate only, i.e.  $I_p(i, j, k) = I(i, 0, 0)$ . With other words, the color/ gray level does not change over a vertical line/stripe. We describe such patterns as one-dimensional function  $I_p(i)$  of the horizontal projection slide position, respectively  $I_p(x)$  if the slide resolution is conceptually infinitesimal.

### 3.3.10.2 The Constraints of Structured Light

Just as stereo vision algorithms, most SL systems (implicitly) impose constraints on the scene. The following list enumerates the most important ones besides those already known from stereo vision.

- **Reflectivity Smoothness:** This is the analogue to the disparity smoothness constraint: The reflectivity of the scene changes smoothly (with a low frequency compared to the frequency of the illuminating pattern) almost everywhere.
- **Neutral Reflectivity:** The reflectivity of the scene is neutral with respect to the wavelength.
- **Neutral Ambient Light:** The ambient light, the illumination arriving at the scene that is not emitted by the projector, is approximately color-less.

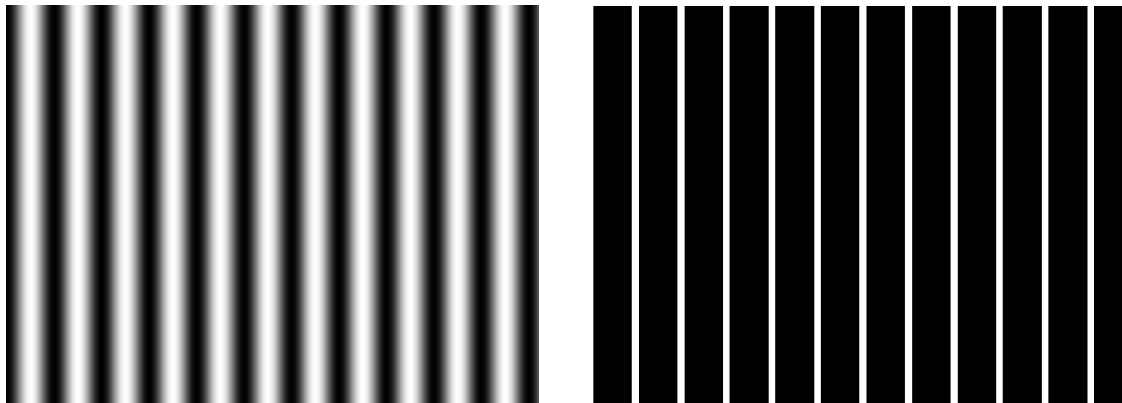


Figure 19: Sinusoidal illumination (left) and line grid pattern (right).

- **Negligible Ambient Light:** The ambient light is negligible compared to the projected pattern.
- **Global Continuity:** The scene is globally continuous in the sense that between two imaged scene points an image path exists along which the scene depth changes continuously.
- **Local Planarity:** The surface orientation changes only slowly almost everywhere, allowing small surface segments to be reasonably well approximated by a plane in 3D space.
- **No Movement:** The scene is static, i.e. does not contain moving objects.
- **Limited Movement:** Objects in the scene move with image plane velocities below a set limit.

### 3.3.10.3 Structured Light without Encoding

This section discusses approaches to SL that do not encode their pattern. The earliest such systems illuminate the scene with a single (typically laser) beam or stripe ([Schmaltz 1932], [Shiray 1972], [Agin and Binford 1973]). These light structures result in distinctive blobs or stripes in the pattern image, which can usually be detected and identified easily. Of course, only a few range values can be extracted from a single pattern image, making the approaches ineffective. To obtain a dense range image, the beams or stripes have to be swept across the necessarily static scene. This tends to be a slow process that typically requires expensive mechanically moving parts such as a deflection mechanism of one or two degrees of freedom, in which case the results suffer from mechanical imprecision and wear. Despite these drawbacks such systems are still widely used today (e.g. [Schmallfuss et al. 2002]), primarily because they are easy to implement, reliable, with suitable mechanics also very accurate and have a comparatively large depth of field. They are particularly well suited for scenes moving with precisely known constant speed such as objects on a conveyor belt, since this allows them to acquire dense depth maps without having to actively scan the scene.

To speed up the acquisition and to avoid complex mechanics, projecting not just one, but many distinct light planes at once with a simple projector has been proposed. We distinguish between two corresponding un-encoded SL categories whose designations have mostly historical reasons:

- **Fringe pattern approaches** interpret the illumination as continuous signal, exploit its intensity values for the range calculation and obtain depth values for all image pixels. They either project a sinusoidal illumination of the form  $I_p(x) = I_{p0} + A \cos(2\pi x/d + \varphi)$  or a grid of equidistant parallel vertical lines (a line (grid) pattern). Figure 19 shows examples for both types. Fringe pattern approaches are, by and large, a topic of optical research.
- **Grid coding approaches** understand the pattern as being composed of discrete shapes, locate these shapes in the image and compute depth values only for pixels part of a shape. They use the signal values only for localization, not for range calculation. They either project a line grid pattern or one of equidistant vertical and horizontal lines, i.e. a square grid pattern (figure 20).

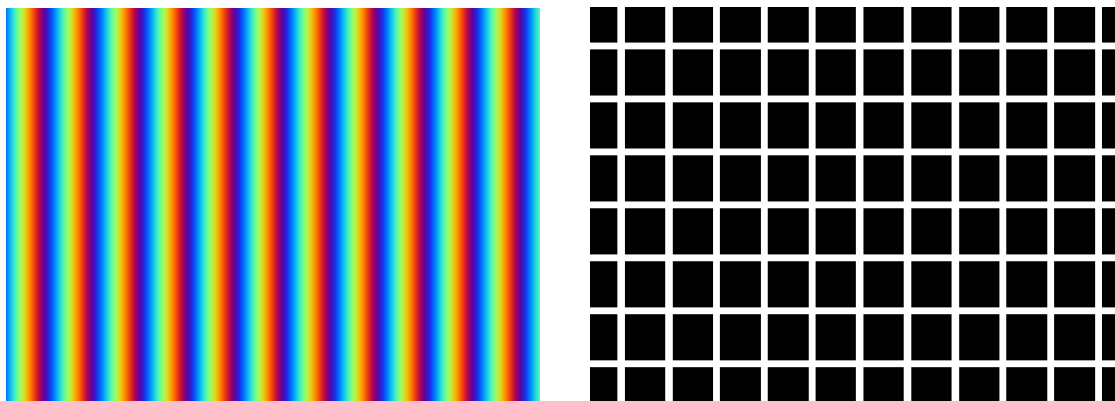


Figure 20: Color pattern by Wust and Capson [1991] whose R, G and B components represent sinusoidal illuminations phase-shifted relative to each other (left). A square grid pattern (right).

The basic idea of fringe pattern techniques is best understood by assuming parallel projection and a sinusoidal illumination impinging under an angle  $\theta$  onto a reference plane parallel to the camera's image plane. Then a shift in height relative to the reference plane results in a phase shift of the observed illumination, but affects neither its frequency nor its amplitude. Determining this phase shift for a scene point, a process called phase demodulation, yields its depth offset  $z$  relative to the reference plane, wrapped into a depth interval  $\Delta z$  corresponding to a  $2\pi$  phase interval.

*Phase shifting* or *stepping*, also called *phase measurement interferometry* ([Creath 1988], [Halioua and Liu 1989], [Larkin and Oreb 1992]), is a popular way to demodulate the phase. In its case, the illumination is shifted, e.g. mechanically, by at least three discrete phase steps such as  $\varphi_1 = \pi/4$ ,  $\varphi_2 = 3\pi/4$ ,  $\varphi_3 = 5\pi/4$ . This defines for each pixel a series of image intensities, e.g. with three steps the intensities  $I_1$ ,  $I_2$  and  $I_3$ . With the aforementioned system geometry, these can then be related via

$$\arctan((I_2 - I_3)/(I_2 - I_1)) = \arctan(\tan 2\pi(x + z \tan \theta)/d) = 2\pi x/d + 2\pi z \tan \theta/d \in ]-\pi, +\pi[ \quad (50)$$

where merely elementary trigonometric transformations have been applied. As the 3D space  $x_c$  position of an imaged scene point is by definition known with parallel projection ( $x_i = kx_c$ ), the depth is the only unknown of equation 50; resolving it for  $z$  determines the wrapped depth. Since solely the illumination is shifted, yet the imaged scene point remains fixed over a series, unwanted influences on the perceived brightness such as the local reflectivity or the ambient illumination are mostly cancelled out. There is a wide range of phase stepping algorithms that differ mainly in the number of phase steps, the formula for computing the phase/depth from the intensities and the assumed geometric configuration. Their common disadvantage – and the reason they are not covered in more detail here – is the need to acquire at least three images, typically even more to limit the influence of noise. To overcome this drawback, Wust and Capson [1991] combine three phase shifted illuminations into a single color pattern whose  $R(x)$ ,  $G(x)$  and  $B(x)$  components represent three shifted sinusoids (figure 20), i.e. propose a one-shot phase shifting system. However, phase stepping relies on shifting the very same signal: the observed illumination should differ in phase only. With the combined pattern, this will typically not be the case, e.g. the red and the blue signal will differ in their dc components and amplitudes due to the reflectivity and sensor response being quite distinct for the two spectral bands. The approach thus requires a scene-dependent color calibration to at least approximate the above prerequisite; of course it is even then limited to scenes of constant color. Huang et al. [1999] propose a similar pattern and end up with the same limitations. They do not perform a color calibration, but rather paint their scenes white and state that "meaningful results can be obtained only on surfaces of neutral color content". One-shot phase shifting is principally problematic as the literature reports that with traditional phase shifting, i.e. given more favorable conditions, shifts are repeated up to 30 times (e.g. [Teschner 1996]) to get the better of noise. So it does not come as a surprise that Wust and Capson report a comparably low accuracy.

Other approaches to phase demodulation operate in the frequency domain, namely the closely related techniques of *Fourier filtering* [Hobson et al. 1997] and the *Fourier transform profilometry* [Takeda and Mutoh 1983]. Their main advantage over phase-shift methods is their one-shot nature. Their main problem is separating the sought phase from the unwanted amplitude modulation introduced by a varying surface reflectivity. They solve it in the following way: Provided a surface's slope is limited and its reflectivity and height distribution have a much lower frequency than the pattern's carrier frequency  $f_0 = 1/d$ , the spectra corresponding to the integral multiples of  $f_0$  are well separated. It is then possible to isolate single spectra in the 2D FT image via band-pass filtering and to frequency-shift each of them to be centered at zero frequency. Transforming this modified spectrum back into the spatial domain yields, after some rather simple transformations, the phase distribution of the image. Clearly doing so represents an elegant one-shot technique to demodulate the phase. Its central weakness is the above reflectivity smoothness and local planarity assumption: high frequency depth changes result in wide bandwidth image signals, in which case band-pass filtering becomes difficult and unavoidably passes noise as well. High-frequency variations in the reflectivity cause similar problems.

With perspective projection, a height change modulates the observed illumination frequency in a way indirectly proportional to the distance scene-projector: the further away the scene, the lower this frequency and vice versa. Hung [1993] proposes to determine the local projection frequency (and thus depth) by computing the first derivative of the instantaneous phase. The approach is limited to special geometric set-ups that guarantee a strong perspective effect of the projection while avoiding it in the pattern image. It is rather unclear how it deals with locally varying reflectivity, especially as it relies on a noise-amplifying derivative. Only simulated results are given.

The main advantage of fringe methods is their accuracy, given they determine phase shifts at up to 1/1000 of the fringe period [Halioua and Liu 1989]. On top of that, they obtain the depth for every image pixel, i.e. they tend to achieve a high resolution of the depth map. At the same time they are very sensitive to noise, as noise in the observed intensity directly propagates into the computed range value. Moreover, they are typically not one-shot approaches, with all associated disadvantages. The few one-shot variations require special types of surfaces (reflectivity smoothness, etc.) and are according to Creath [1988] significantly less accurate. A severe limitation is that all fringe methods obtain the wrapped phase/depth only. To obtain absolute range data, the mod  $2\pi$  ambiguities have to be resolved via phase unwrapping. Usually this is done by comparing the phase of adjacent pixels and adding (subtracting)  $2\pi$  at jumps from  $2\pi$  to 0 (0 to  $2\pi$ ). Which requires that the phase does not change by more than  $\pi$  over adjoining pixels, i.e. a globally continuous surface of limited slope. Furthermore, at least one point with known absolute phase/height is needed as starting point for the unwrapping. Clearly it is not always possible or convenient to insert such a reference point into a scene. With orthographic projection, an arbitrary point can be chosen as reference, yielding a height profile relative to this unknown starting position that is scaled by an equally unknown factor. With perspective projection and in the absence of a reference point, so-called fringe-tracking methods [Pearson 1996] insert a large meta-fringe into the pattern that allows correctly indexing the fringes, effectively turning the approach into a sparsely encoded one. Or the issue is solved indirectly by combining fringe approaches with ranging methods that are less accurate, but yield absolute range values; the former then serve the purpose of refining the results of the latter.

*Grid coding* was originally proposed by Will and Pennigton [1971] to segment an image of a polyhedron into its planar components of distinct orientation. The 2D Fourier transform of an image of a planar area illuminated by a line pattern is a crossed set of harmonically related delta functions. Separating the resulting spectral clusters and transforming them singly back into the spatial domain yields the sought decomposition (but for a certain remaining ambiguity).

Determining the 3D form of a scene is difficult with grid coding due to the un-encoded pattern; it is in general impossible to identify which imaged line or quadrilateral matches up with which projected line, respectively square. Yet each such mapping (that respects the epipolar constraint) corresponds to a different surface. An algorithm needs significant a-priori knowledge to decide on the

right out of the many possible mappings, respectively 3D shapes. This becomes the more difficult, the more lines are projected, as then the more plausible shapes that differ only slightly are there to choose between. This implies a tradeoff between a high lateral resolution of the depth map and the probability of generating the correct one. Hu and Stockman [1989] project a square grid as in figure 20 and try to derive the mapping of recognized to projected squares from a set of general geometrical and topological constraints. Despite using a coarse grid (20 x 20 lines), they are in most cases still not able to settle on a single solution, but rather output several potential depth maps.

As it is virtually impossible to obtain high-resolution depth maps of unknown scenes in the latter fashion, all following grid coding systems assume parallel projection. Then the illumination can be interpreted as set of parallel, equidistant planes in space, i.e. the  $i$ th vertical plane can be written as  $A_v x + B_v y + z + D_v + id = 0$  and the  $j$ th horizontal as  $A_h x + B_h y + z + D_h + jd = 0$ . Will and Pennigton [1971] use a square grid pattern and determine for each imaged quadrilateral the transformation matrix that restores it back to the projected square; this matrix allows concluding the surface normal. Asada et al. [1985, 1986, 1988] project a line grid pattern and calculate the surface orientation for each point on an imaged line from the line's image plane slope and distance to the next imaged line. Wang et al. [1987] as all following authors choose a square grid pattern and derive the orientation for each grid intersection point from the image plane slopes of the horizontal and the vertical line. Shrikhande and Stockmann [1989] discuss two similar approaches, which both exploit the size and orientation of the quadrilateral sides as seen in the image. All these related approaches that analyze the deformation of imaged grid lines require the surface to be locally planar and exclusively determine its orientation; only Wang et al. actually try to determine the surface shape by integrating from orientation to shape, with all the obvious disadvantages of such a method (inaccuracies add up, drastic errors occur if perceived lines do not break across jump boundaries, etc.).

Wang [1991] discusses how to obtain intrinsic surface properties, i.e. ones independent of the coordinate system, the viewer and the chosen surface representation such as the principal curvature. He computes these properties for a given square grid illuminated scene from the second derivative of the imaged grid line curves, which makes his technique very sensitive to noise. Moreover, his approach works only for surfaces that locally approximate certain elementary surface types, namely planes, spheres, cones and cylinders; which surface patch is of which type of shape has to be known in advance. Wang and Cheng [1992] show how the latter information can be derived from the intensity data (primarily by exploiting iso-brightness contours) given a controlled illumination.

Proesmans et al. [1996a, b] select an arbitrary square grid intersection point as origin (for which they set  $i = j = 0$ ). They "identify" the remaining light planes recursively via their position relative to this origin (assuming global continuity). Then for each grid point connected to the origin the illuminating plane is known, as are the point's  $x$  and  $y$  coordinates due to the orthographic projection. Solving the plane equation for the remaining unknown  $z$  yields the depth value. Accordingly, even two depth values can be computed for each intersection point (one from the horizontal, one from the vertical plane), which are averaged. Only these averaged values are kept. Consequently, their range output resembles a 3D grid or mesh rather than a dense depth map. Unidentifiable intersection points lead to holes in the depth map. In the worst case, misidentifications occur, which are especially problematic as they propagate into all subsequently identified grid points. Their algorithm also computes a grayscale image of the scene by "reading between the grid lines" [Proesmans et al. 1998] and non-linearly diffusing the intensity data over the areas occluded by the grid.

The disadvantage of grid coding as a ranging technique – besides the downsides of parallel projection and the fact that most approaches do not even obtain shape data – is that errors with respect to localizing and/or identifying the grid lines can go unnoticed or even propagate as the regular patterns have by definition no built-in error detection capabilities. Distinguishing between projected lines and texture is especially problematic if the latter is of high frequency. This implies the approaches inherently assume reflectivity smoothness besides global continuity, just as one-shot fringe pattern methods. Compared to the latter, they are somewhat more robust, yet achieve a lower relative spatial resolution as range values are obtained only for imaged grid lines.

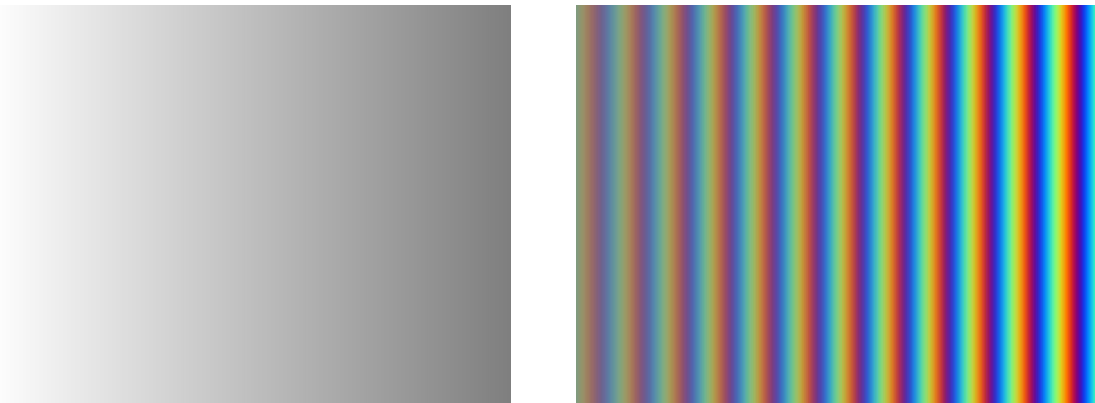


Figure 21: Illumination created by a 2D linear wedge filter (left) and RGB phase shift pattern with monotonously increasing amplitude as proposed by Schubert [1996].

#### 3.3.10.4 Coded Light Based on Point-Wise Encoding

The most intuitive way of encoding is to exclusively use the data transmitted by a single projector pixel, respectively received by a single image pixel, i.e. point-wise encoding via the intensity or color of the illumination. As discussed above, corresponding CL approaches typically employ the trivial codeword function  $\sigma: \{1\} \rightarrow \{(0, 0, 1)\}$ . The earliest such method due to Carrihill and Hummel [1985] utilizes the intensity ratio between two gray-level views of a scene to that end. One image is taken with constant illumination  $I_p(x, y, 0) = k$ , the other with the illumination passing through a 2D linear wedge filter. The transmittance of the latter decreases horizontally from 100% to 50% and is vertically constant, i.e.  $I_p(x, y, 1) = k - (kx)/(2n_p)$ , resulting in the pattern of figure 21. For each image pixel, the ratio of the two intensity values is formed. In the absence of noise, this ratio identifies a plane of light, provided the reflected intensity is proportional to the projected.

Alternative approaches attempt to exploit that the projected light and that reflected by the scene differ only in intensity, but not in the spectral composition if the employed colors are monochromatic (barring the unlikely cases of fluorescent or phosphorescent surfaces). In this case, the wavelength appears to be a reliable means to per-pixel encode a large number of light planes. A practical way to create such a monochromatic color pattern is to diffract collimated white light with a grating, effectively generating a rainbow. The approach is accordingly called *rainbow approach* ([Tajima 1987, 1990], [Häusler and Ritter 1993], [Geng 1995, 1996], [Smutny and Pajdla 1996]).

Schubert [1996] combines the ideas of the phase-shift and the rainbow approach: He modifies the RGB phase shift pattern to one of monotonously increasing amplitude; consequently the amplitude uniquely encodes the illuminating plane. In this manner, Schubert gets rid of the periodic encoding of the traditional phase shift, respectively the need to find out the exact wavelength of the reflected light. However, without monochromatic light the reflected amplitude depends strongly on the scene color, limiting the approach to surfaces of neutral color. Also, as even more information is packed into the color signal, the technique is even more susceptible to noise than a stand-alone phase shift or rainbow approach.

The approaches of this section have several advantages: Their patterns encode in principle a continuum of light planes, which allows generating depth maps of arbitrarily high relative spatial resolution. The codeword function is trivial, so decoding is simple and efficient as it involves single image pixels only; it can easily be done in real-time. Dynamic scenes pose no problem but for the intensity ratio sensor. At the same time, any real-world implementation will unavoidably suffer from a number of major practical problems: The foremost is a strong susceptibility to noise such as imaging noise or mutual illumination, given it is necessary to distinguish between subtle nuances in the projected intensity, respectively wavelength. So Tajima and Iwakawa [1990] propose to average 10 color images to generate a single, less noisy pattern image and to further average the measured depth over 5 by 5 windows.

With respect to the rainbow approach, standard RGB color cameras are generally not well suited for recognizing the exact wavelength of monochromatic light. Häusler and Ritter employ a sophisticated optical set-up to eliminate the green range from the spectrum as for this band the mapping of monochromatic colors to the camera's RGB system is the worst. Geng [1995] discusses that it is in most cases advisable to obtain two more images, one without and one with all-white illumination to cancel the effects of the background illumination and the intrinsic scene color. All approaches cannot cope with non-neutral ambient illumination. In short, to obtain usable range images with the methods of this section, measures have to be taken that completely offset their theoretical advantages, with the exception of a few selected applications involving cooperative scenes and well-controlled laboratory environments.

### 3.3.10.5 Coded Light Based on Temporal Encoding

Altschuler et al. [1979] and many others describe the temporal encoding of light planes on the basis of only two intensity levels, i.e.  $Q_p = \{\text{black, white}\}$ . As discussed before, with temporal encoding the codeword function has typically the form  $\sigma(k) = (0, 0, k)$ ; the resulting set of potential codewords is the set of binary sequences of length  $s = t$  such as 01001110. Schemata for encoding the planes are simple binary codes [Altschuler et al. 1979], a Hamming code [Minou et al. 1981] or the Gray code proposed by Potsdamer and Altschuler [1982] and implemented by Inokuchi et al. [1984] and Wahl [1984]; the latter is by far the most common choice as in its case codewords for adjacent planes differ in exactly one bit; the most likely errors, the ones occurring at the single resulting intensity transition between adjacent planes, are thus guaranteed to cause only minor ranging errors. Also, the frequency of its finest pattern is only half of that of straightforward binary coding. Regardless of the schema, the encoding translates directly to a projection pattern: a plane in the  $n$ th pattern is white if its  $n$ th code letter is 1, otherwise black. Implementing such a system is rather simple: Typically first two images are taken under all-white and without illumination. From these two images a space-variant threshold is derived, respectively pixels that do not exhibit a significant intensity change (e.g. because of occlusion) are marked as invalid. Then the  $t$  pattern images are acquired, and its threshold is applied to each pixel of each image. This defines for each valid camera pixel a sequence of black and white, respectively 0s and 1s, that uniquely identifies the light plane illuminating the imaged scene point. Alternatively, Sato et al. [1986, 87] propose to project each pattern once as positive and once as negative and to determine the sign of the intensity difference between the resulting positive and negative images for every pixel. The threshold is then not only space-variant, but also adapted for each pattern. They report a more robust decoding at the price of a twofold number of slides and pattern images. As mentioned before, the discrete character of real-world cameras and projectors implies that projection slides and pattern images contain stripes rather than infinitesimal lines; only boundaries and to some extent stripe centers should be associated with planes and used for triangulation. So to obtain a precise depth value for every valid pixel, temporal encoding is often refined with a phase shift step, or a related technique such as a sub-pixel [Hattori and Sato 1996] or line shift [Guehring 2001] of the finest Graycode pattern.

Several modifications have been put forward to speed up the above principle by using more than two gray levels. Horn and Kiryati [1998] present a theoretical framework for the design of patterns with  $q_p \geq 2$  gray shades. It allows deriving the pattern/code resulting in the smallest mean squared expected identification error given one wants to project  $n_p$  distinct stripes, use  $t$  distinct slides and has to cope with a known fixed level of zero-mean Gaussian noise. Respectively, determining this pattern for distinct values of  $t$  yields the minimal value of  $t$  needed to stay below a given error threshold. As finding such a code is an optimization problem of too large a dimensionality, they propose two sub-optimal solutions, namely filling the  $t$ -dimensional code space with space-filling Hilbert or Peano curves of finite order and placing  $N$  points on the curves at equal distances. Conditional on the input parameters, their patterns utilize three or more distinct gray levels or converge to the binary Graycode, if  $2^t > n_p$  or the noise level is too high. Caspi et al. [1998] extend this principle to colored patterns as color potentially allows using even more distinct levels per slide.



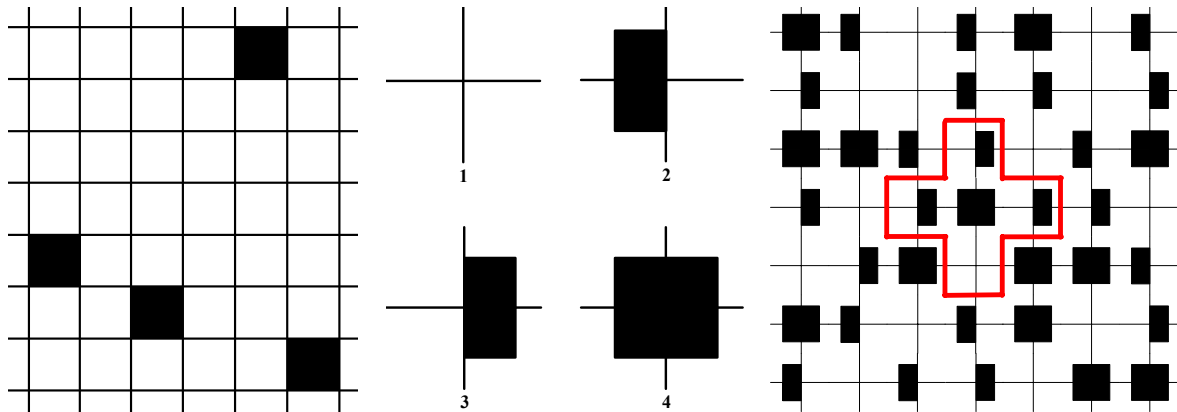


Figure 22: Pattern proposed by Le Moigne and Waxman [1984] (left). Griffin and Yee [1991, 1994] propose the four primitives shown in the middle. A clipping from a resulting pattern is shown to the right, with an exemplary subpattern/four-neighborhood highlighted.

Hall-Holt and Rusinkiewicz [2001] propose a time-varying encoding that allows limited movement. They project a sequence of slides where the color of each of their 111 stripes varies over the slides between black and white, just as with classical binary temporal encoding. The key difference is that Hall-Holt and Rusinkiewicz exclusively focus on the boundaries between stripes, which can take on the following four values:  $Q_p = \{\text{black-to-black (invisible), black-to-white, white-to-black and white-to-white (invisible)}\}$ . Tracking the status of a boundary over four frames yields a code-word that identifies it uniquely. Yet this tracking is quite complicated as boundaries are sometimes invisible. To deal with this problem, the code design allows at most one invisible between two visible boundaries and ensures each boundary is visible at least every other frame. The tracking algorithm then hypothesizes all potential locations of invisible boundaries and simply matches each visible boundary of a frame to the closest hypothetical or visible one of the previous image. Clearly this approach permits only a limited movement where the position of a boundary varies by less than half a stripe width over two frames (if it moves faster, the algorithm does not notice it, but produces erroneous data) and a mostly continuous surface. It furthermore makes a reflectivity smoothness assumption, as it otherwise could not extract the boundaries from a single pattern image.

Binary temporal encoding copes very well with locally varying surface reflectance properties and background illumination. Especially when combined with phase shifting, it represents an exceptionally reliable and accurate ranging approach that is widely used in practice. It is, however, time-consuming compared to single-shot approaches and requires static scenes and an expensive projector capable of switching between several slides without introducing a positioning error. The multi-gray level or color variations reduce the number of patterns if circumstances in conjunction with the robustness requirements permit it. However, they rely on adapting the pattern to the scene, which in itself entails additional processing time. Consequently, their practical benefit is somewhat limited, and for most applications the more straightforward and robust binary approaches will be given preference. The tracking method by Hall-Holt and Rusinkiewicz trades the advantages of traditional methods with the ability to deal with restricted movement; currently the latter limit is rather severe. Clearly the idea's relevance depends on the availability and cost of multimedia projectors capable of very fast slide switching.

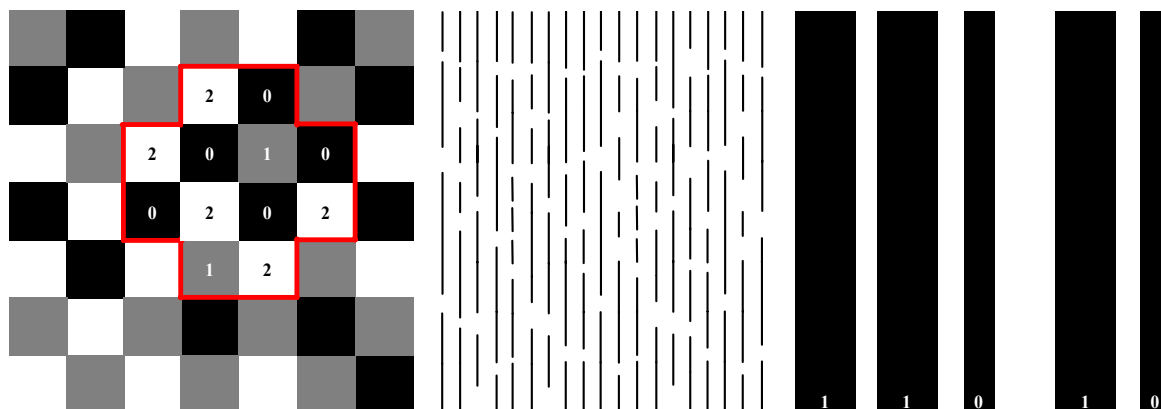


Figure 23: Pattern by Ito and Ishi [1995] with black, gray and white squares as pattern primitives and with subpatterns of 12 primitives (left). Slit pattern with random cuts by Maruyama and Abe [1993] (middle). Beumier and Acheroy [1999] employ thin and thick stripes as primitives (right).

### 3.3.10.6 Coded Light Based on Spatial Gray Level Encoding

This section discusses one-shot ( $t = 1$ ) CL approaches based on spatially encoded gray level patterns. Recognizing projected grayscales in an image is difficult as a camera captures only the reflection of the illumination, whose brightness depends on the unknown scene reflectivity. Evidently, gray levels are the easier to discern, the fewer of them are used; so almost all corresponding approaches described in the literature employ only the two levels black and white. Which, however, creates a new problem: typically adjacent pattern primitives of the same gray level cannot be discerned – e.g. how to tell two contiguous black squares apart from one in the presence of foreshortening? This implies the succession of black and white within a binary pattern is predetermined and cannot carry any information: Neighbors of a black pattern element have to be white and vice versa. So binary approaches have to use differently shaped primitives or sparse encoding.

Le Moigne and Waxman [1984] project a pattern formed by two primitives, black squares and white ones with a black margin, or, from another point of view, a square grid of a resolution of ca. 10 by 10 lines (figure 22). Since they use a geometric set-up corresponding to the standard geometry of stereo, projected horizontal lines remain horizontal in the image and can be detected easily. They are used for albedo normalization and thus help to segment the vertical lines, which alone carry the range information. The black squares serve as means of sparse encoding. Ids are propagated starting out from these unique marks (there is at most one square per epipolar line) along the grid. Once identified, vertical lines are used for triangulation, yielding a very sparse range map.

Morita et al. [1988] suggest using a *M-array* or *Pseudo-Random Array* (PRA) as pattern. In the context of CL, a PRA is defined as  $Q_p$ -ary array  $P = (p_{ij})$  of size  $m_p$  by  $n_p$  where each  $v$  by  $w$  window occurs at most once (as usual,  $v$  refers to the height,  $w$  to the width of the window). The concept is borrowed from fields such as cryptography and data communications (e.g. [MacWilliams and Sloane 1976]), where it refers to a typically binary array where each possible  $v$  by  $w$  window but the all-zero one occurs exactly once; it is related to the idea of De Bruijn and pseudo-random sequences discussed below and also to *perfect maps*, arrays where each such window including the all-zero one occurs once [Paterson 1994]. Its definition is extended to error detecting PRAs by requiring the windows to have a minimal distance  $h > 1$  [Morano et al. 1998]. In the terminology of this work, a PRA represents a 2D encoding with a codeword function that scans a local  $v$  by  $w$  window, i.e.  $\sigma(k) = (-w/2 + (k - 1)/v, -v/2 + (k - 1) \bmod v)$ , where  $1 \leq k \leq s = v \cdot w$ . To implement their PRA-based approach, Morita et al. use a pattern of 32 by 27 black and white dots and a window size of 3 by 4. They first illuminate with a pattern of only white dots, then with one where some dots are blanked out, and locate the blanked dots by comparing the two resulting images, even though this implies their approach is not truly a one-shot one. From type and location of the dots, they construct a binary array that contains the imaged adjacency relations and match its win-

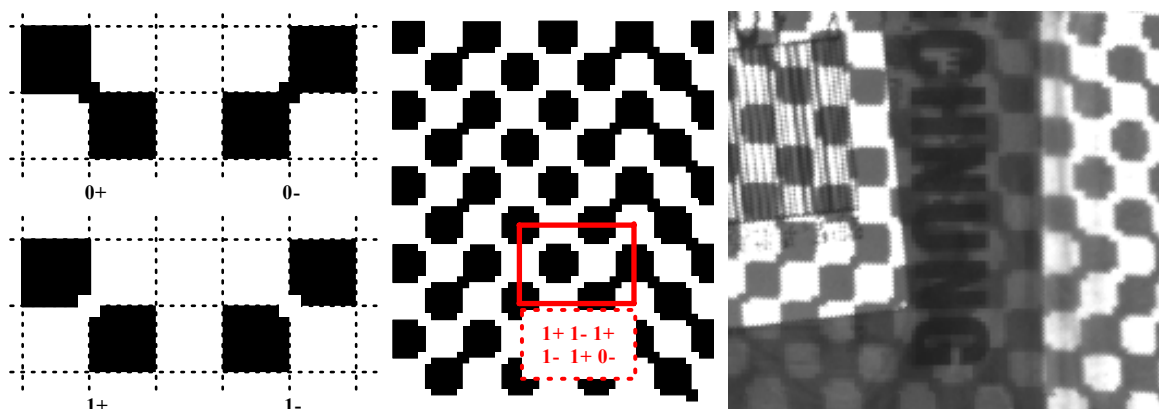


Figure 24: Vuylsteke and Oosterlinck [1990] use two distinct pattern primitives (0 and 1) which occur in a positive and a negative form (left) to form a PRA with unique 3 by 2 windows (see excerpt in the middle). A clipping from a resulting pattern image is shown to the right.

dows with that of the PRA. Unmatched points are filled with a set of heuristics, even though the authors report that doing so can lead to erroneous results.

Griffin and Yee [1991, 1992, 1994] employ a 2D-encoded pattern of size 32 by 32, based on the four primitives shown in figure 22. Each primitive and its four-neighborhood constitute a subpattern, i.e. the approach relies on base-4 codewords of length 5. The authors connect adjoining primitives with straight lines to simplify establishing their adjacency relations and propose template matching [1991] or an adaptive threshold [1994] to locate the pattern primitives in the pattern image. The most interesting aspect of their work is a deterministic encoding schema (for non-error-detecting 4-neighborhood codewords) that is optimal in the sense that every possible codeword occurs exactly once in the pattern [1992].

In a similar approach, Ito and Ishi [1995] utilize a 2D-encoded pattern of resolution 34 by 34 with squares of three distinct gray levels as primitives. Twelve adjacent squares form a cross-like subpattern as shown in figure 23, implying base-3 codewords of length 12. Despite the large subpattern size, respectively codeword length, their encoding is not error-detecting. It is not even an encoding according to the definition of this work as some arbitrary slide positions share the same codeword. Figure 23 shows a clipping from their type of pattern. Clearly it is generally difficult to discern three distinct gray shades reliably in a pattern image. For this reason, they propose to acquire two additional images if there is background illumination or if the scene reflectance is not constant, which of course contradicts the principle of a one-shot approach.

Vuylsteke and Oosterlinck [1990] propose a binary pattern in form of a chessboard as shown in figure 24. Its squares serve to locate the actual pattern primitives, which are smaller squares superimposed on the regular checkerboard. They use two pattern primitives, which each occur in a positive and a negative form. Positive and negative primitives alternate horizontally and vertically, i.e. the four neighborhood of a positive primitive contains only negative ones and vice versa. Their pattern represents in principle a 64 by 2 binary PRA with 3 by 2 windows (i.e. 1D encoding) that is repeated 32 times vertically, yielding a total resolution of 64 by 64. Repeating the PRA introduces the problem of determining whether a given row represents the upper or the lower row of the array. The authors solve it by exploiting the alternate appearance of their primitive forms: codewords are formed by reading positive ones first. E.g. the codeword resulting from the highlighted subpattern in figure 24 reads: 1+ 1+ 1+ 1- 1- 0-, i.e. 111110. A further advantage of their pattern design is that the positions of the primitives are associated with the intersection of horizontal and vertical checkerboard square edges of alternating polarity. According to the authors, the latter can be detected easily and accurately by correlating the image with a reference pattern, making the approach fairly robust with respect to the scene texture. Its main shortcoming is the large size of its subpatterns and its low relative spatial resolution, both due to the necessarily large checkerboard squares.

Maruyama and Abe [1993] project a 1D-encoded pattern of ca. vertical 50 lines or slits with random cuts (figure 23), i.e.  $Q_p = \{\text{cut, non-cut}\}$ ). They use a geometric set-up corresponding to the standard geometry of stereo vision where horizontal slide lines correspond to image rows. Codewords are read by determining the succession of cuts and non-cuts along an image row; the authors do not specify a codeword length. Positive identifications are propagated vertically along the slits. In a similar approach, Beumier and Acheroy [1999] project a white slide with an unspecified number of thin and thick black stripes (figure 23), which replace the cuts and non-cuts of Maruyama and Abe. This idea seems to have been proposed earlier by Yonezawa and Tamamura [1978] in a Japanese-only paper. The disadvantage of the latter approach is that to recognize the width of a stripe in the pattern image the scene not only has to be continuous, but also needs to be locally planar; only then the foreshortening effect is locally constant and the relative width of stripes is preserved from the projection to the pattern image. Also employing stripes of distinct width results inevitably in large subpatterns and sparser than necessary range data, since at least the thick stripes have to be significantly larger than the minimal detectable size.

The main point in favor of the methods of this section is their (in all cases at least conceptual) one-shot nature, their efficiency and their ability to deal with colored scenes; their key problem is recognizing the pattern primitives in the pattern image. None of the proposed primitives can be relied upon to be discernable from the scene texture even with standard scenes; e.g. the scene displayed in figure 24 (taken from a real-world application, namely measuring the volume of parcels) seems likely to confuse all approaches of this section. So the methods either resort to additional images or im- or explicitly impose a reflectivity smoothness assumption, besides the continuity constraint inherent to spatial encoding. Also it is difficult to create a code of sufficiently many reasonably short words given the low transmission capacity of the typically binary patterns. This limits the approaches, especially those with 2D encoding, to a coarse relative spatial resolution; the highest stated in the literature is a low resolution of 64 by 64. The latter aspect also prevents the use of error-detecting codes that would make the systems more robust.

### 3.3.10.7 Coded Light Based on Spatial Color Encoding

This section discusses one-shot methods based on spatially encoded color patterns. In principle, the latter offer a tripled transmission capacity compared to gray level ones (if one interprets R, G and B as separate transmission channels). It is paid for with a strong dependency on the scene color: only if the latter is neutral, the reflected color resembles the projected one. With a strongly colored scene, the two can differ drastically as demonstrated by figure 25. The use of monochromatic colors alleviates this problem to some extent, but there is currently no practical way to generate arbitrary projection patterns composed of monochromatic colors (for some close-range applications LEDs might provide sufficient light).

The first color-coded light system by Boyer and Kak [1986] utilizes a pattern composed of 96 colored vertical stripes. The horizontal concatenations of stripes  $1 \dots s$ , of  $s + 1 \dots 2s$  etc. each represent a unique signature. The pattern is consequently not encoded in the sense of this work (at least not when considering the word length  $s$ ) as only selected – as opposed to every possible –  $s$ -primitive-sequences are guaranteed to be unique. This creates the problem of how to find the beginning of such a sequence. For their experiments, Boyer and Kak choose a sequence length  $s$  of 4 and employ stripes of the three colors red, green and blue. To recognize these colors in the pattern image, they locate the signal peaks of the R, G and B color signals. Given the recovered color sequence along an image row, they compare all received sequences of length  $s$  to the codewords and note the matches. For each projected codeword, 0, 1 or  $m$  matches exist. Each match is then interpreted as crystal and enlarged as long as the projected and received colors match. Next, a heuristically motivated, iterative process chooses the longest crystal and eliminates or trims all remaining crystals that clash or overlap with it. The authors remark that this process can be "fooled by particular occlusive effects". As their approach is computationally inexpensive, it is well suited for real-time processing.

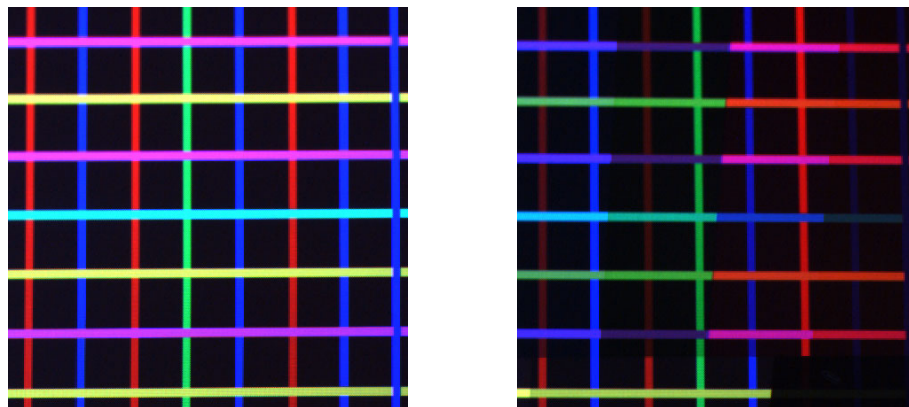


Figure 25: Pattern images obtained for the projection pattern proposed by Salvi [1998] for a neutral (left) and a colorful (right) scene. Comparing the identical clippings shows that the latter scene modulates the pattern reflection to the extent of the projected colors being no longer recognizable.

Hügli and Maitre [1988] improve upon the approach by Boyer and Kak by changing the pattern to a truly encoded one, i.e. one where each sequence of  $s$  stripes forms a codeword. The resulting projection patterns, respectively code symbol sequences are also called *pseudo-random*, *pseudo-noise* or *m-sequences* and play an important role in areas such as radar, cryptography and Monte Carlo simulation [Goresky and Klapper 2004]. Such pseudo-random sequences are closely related to *De Bruijn-sequences* [De Bruijn 1946]; a De Bruijn-sequence of order  $s$  over the alphabet  $Q_p$  is a circular sequence of length  $q_p^s$  where each possible substring of length  $s$  occurs exactly once; a pseudo-noise sequence is a De Bruijn-sequence without the all-zero subword [MacWilliams and Sloane 1976]; respectively, in the context of coded light, this definition is typically understood to refer to a sequence of undefined length where each substring of length  $s$  occurs at most once.

Monks [1994] builds on the version of Hügli and Maitre. He chooses the pattern colors red, green, blue, yellow, magenta and cyan ( $q_p = 6$ ). With a codeword length of 3, he achieves a resolution of 120 distinct light planes. The new aspect of his work is an enhanced decoding schema based on a-cyclic directed graphs. It is based on the ordering constraint and thus of limited general applicability; it seems to work well with human mouths, which is the only type of scene Monks considers.

Paul and Stahnke [2000] use a similar pattern of 108 stripes; the main difference is that they adapt their four projection pattern colors to the scene. For this reason, they are able to cope with uniformly colored scenes despite directly exploiting – as all other approaches of this section – the reflected color signals. However, such a global adaptation of the projected colors still fails with scenes whose color varies significantly. Also a scene-dependent color adjustment is often unpractical and calls for a flexible multi-media projector.

Griffin et al. [1992] describe a PRA of size 11 by 29 that uses red, green and blue circles as primitives. Apart from this aspect, it is identical to their grayscale PRA described above. Davies and Nixon [1994, 1998] use this pattern where they change the colors to yellow, magenta and cyan. Their innovation is to apply a special Hough transform to locate the circles in the pattern image and to directly derive the local surface orientation from the elliptic shape of the imaged circles.

Salvi et al. [1998] again adopt the approach by Griffin et al. [1992] to generate their pattern. They project two separate PRAs of size 27 by 1 and window dimensions of 3 by 1 at the same time. One of the arrays is made up of horizontal, the other of vertical stripes. They choose red, green and blue for horizontal and yellow, magenta and cyan for vertical stripes. As there is a large intervening space between the stripes, the two patterns can be superimposed as shown in figure 25. This way they achieve 2D encoding with doubling rather than squaring the number of codewords, however at the cost of a coarse resolution of 29 by 29 due to the big interspaces needed to avoid interferences of the two superimposed PRAs.

Morano et al. [1998] also propose employing PRAs to implement a coded light system. They compute PRAs of size 45 by 45 for distinct choices of the remaining parameters, e.g. for  $h = 1$  up to 4,  $q_p = 3$  up to 9 and windows sizes of 3 by 3 up to 4 by 5. For their real-world experiments, they employ circles of the three colors red, green, and blue ( $q_p = 3$ ). Even though they put forward using error-detecting codes, their small code alphabet prevents them from actually employing one for their experiments.

Compared to gray level techniques, this section's color-based one-shot methods share the continuity assumption. The larger number of codewords possible with colored patterns allows them to achieve a higher resolution, respectively in some cases to employ error-detecting codes. However, they are restricted to neutrally colored scenes, with the exception of Paul and Stahnke [2000] who cope with uniformly colored scenes. This fundamental limitation, caused by directly relying on the absolute color signals, is very pronounced. Boyer and Kak [1986] require the color content of scenes to be "predominantly neutral". Griffin et al. [1992] state that their approach "works best with environments that are color neutral". Davies and Nixon [1998] paint their test scene white. Monks [1994] recommends white make-up. According to Salvi et al. [1998], their approach is limited to "pale and neutral scenes" and permits no ambient light. Morano et al. [1998] remark that color identification may be compromised if a scene has "complex reflectivity properties". They use for that reason only three distinct colors and seem to employ objects of rather neutral color for their experiments.

### 3.3.10.8 Conclusions on Structured Light

When realized with a simple slide projector, SL has modest resource requirements. With the exception of laser-based systems, it is also a safe technique, though one limited to relatively small (up to a few meters) working spaces as the projection strength fades away relative to the background illumination proportional to the squared distance projector-scene. Its variations differ widely regarding all other aspects such as accuracy or data rate. The most important conclusions are: In the form of temporal encoding, SL is by far the simplest and reliable (and according to Nayar et al. [1996] most widely used) method to acquire accurate depth maps of high resolution of static scenes. This task can be considered practically solved but for special scenes such as strongly specular ones. Only one-shot approaches are currently suited for acquiring range images of unknown, arbitrarily fast moving scenes in uncontrolled environments. Most of them operate in real-time as they solve the identification problem very efficiently, if they solve it. However, un-encoded one-shot approaches acquire at most shape data. The ones based on black-and-white spatial encoding suffer from a reflectivity smoothness assumption, to a lesser extent from a continuity assumption and, most importantly, from a low relative spatial resolution. Color pattern approaches tend to have a higher resolution, but are limited to neutral or at most uniformly colored surfaces. So the task of acquiring accurate high-resolution range maps of arbitrary dynamic scenes with the SLA/CLA is, by and large, unsolved.

## 3.4 Summary

This chapter shows that there is a wide spectrum of approaches to range image acquisition; for most of them furthermore a wide number of variations exist. Despite this wealth of techniques, there is no single best approach to ranging; each of them has its intrinsic advantages and fundamental limitations. No method solves the problem of range image acquisition in a manner that covers the needs of 95 % of the applications such as e.g. today's CCD cameras do in the case of color or grayscale image acquisition. For this reason, a productive discussion of which approach should be chosen requires to first narrow down certain conditions and requirements – for which distance, accuracy, resolution etc. Accordingly, this is what we do in the beginning of the next chapter. On this basis we then utilize the concepts and insights of this chapter to develop a new approach to range imaging.

## 4 A New Approach to Range Image Acquisition

This chapter develops a new approach to range image acquisition. First, it precisely defines the problem to be solved (4.1). Next, it outlines the principles and key ideas of the proposed solution (4.2). It structures the latter into three parts: the core principle (4.3), its extension (4.4) that yields better results, but requires more resources and finally a set of optional features (4.5) that might or might not be needed or pertinent, depending on the application in mind.

### 4.1 Problem Statement

The introduction described the main topic of this thesis as that of acquiring the high-resolution 3D structure of an arbitrary scene accurately, robustly and in real-time with reasonable effort. The concepts of the previous two chapters allow re-stating the problem in a more precise form: The objective is to acquire range images with the following properties and given the following conditions:

- **(Relative Spatial) Resolution:** The relative spatial resolution of the depth maps should be at least comparable to today's standard video cameras, i.e. be in the area of CCIR (ca. 780 x 580 pixels) or RS-170 (ca. 750 x 480 pixels) systems.
- **Data Rate:** The approach should work in real-time. As commonly accepted, we understand this as being able to provide range maps of the above resolution at the CCIR video frame rate of 25 images per second with a time lag corresponding to at most one frame, i.e. to less than 40 ms.
- **Geometric Parameters:** As we are mostly interested in human-machine interfaces, the working space should be sufficiently large to acquire images of faces or gestures, i.e. approximate a cube of 0.5 to 1 meter side length.
- **Accuracy:** Accuracy is a complex aspect as it is influenced by many factors – the components used, the working space, the scene's reflection properties etc. For this reason, we specify the required accuracy rather vaguely: the quality of the range data should allow solving tasks such as 3D face or 3D gesture recognition. We conclude from the typical dimensions of the features of face and hand that the standard deviation between measured and correct range value (the ground truth) should be notably less than a millimeter to this end. A more precise definition is not helpful in the context of a general problem statement, given that it would require a set of very specific follow-up definitions regarding the above factors.
- **Robustness:** The technique should be able to obtain range data in an uncontrolled real-world environment, especially in the presence of uncontrolled background illumination of realistic levels and types; it should e.g. function outdoors.
- **Safety:** The system should pose no danger whatsoever to humans.

- **Scene Constraints:** The approach should impose, if any, only few and minor scene constraints. Excluding dynamic scenes with objects in motion, strongly textured scenes, strongly colored scenes, or ones with frequent surface discontinuities (depth gaps) is not acceptable.
- **Hardware Requirements:** The system should be exclusively based on today's low-cost off-the-shelf hardware, and as little of it as possible.

Comparing these requirements with the capabilities of the approaches discussed in the previous chapter leads to the following general conclusions: Contemporary shape-from-shading or -motion techniques miss the outlined criteria by far. Moiré interferometry and photometric stereo do not give actual 3D data (just dimensionless shape data) and are too sensitive to uncontrolled background illumination. Depth-from-defocus is better suited for the task. Applying it to dynamic scenes, however, requires significant hardware resources (several lenses that share the same optical path), especially given the targeted sub-millimeter accuracy over the targeted large working space. Current time-of-flight and interferometric systems require complex special-purpose hardware, have only a single or at most a few sensor elements and scan the scene rather than acquiring a single snapshot of it. They are for that reason not suited for acquiring high-resolution depth maps of strongly dynamic scenes. Stereo systems are not able to reliably obtain accurate high-resolution range data of unknown complex, potentially optically unstructured scenes in real-time. Finally, reasonably robust one-shot structured light systems that acquire 3D data, not just shape information, either do not achieve an adequate resolution (with spatial encoding based on gray levels) or are limited to neutral or uniformly colored scenes (with spatial encoding based on color patterns).

We conclude that we have not found an approach in the open literature that is able to solve the stated problem, despite the fact that it would have numerous uses. As pointed out in the introduction, this is undisputable considering there are many applications that require the exact 3D structure of objects such as shape inspection of industrial goods, gauging the size of objects in general or the creation of 3D models for virtual reality.

## 4.2 Principle and Key Ideas of the New Approach

This section outlines the principle and key ideas of the solution to the problem stated above. Which of the techniques discussed in the previous chapter (ignoring the ones already principally ruled out) could be improved or extended to that end?

- Developing a high-resolution time-of-flight system for dynamic scenes appears to be a promising idea. These methods clearly have the potential of meeting the set objectives. However, building a corresponding range image sensor with hundred thousands of sensor elements calls for major hardware development, which is by far beyond the scope of this work and conflicts with the outlined hardware requirements.
- Stereo vision is capable of meeting most, but not all of the set requirements. Passive stereo has major problems with optically unstructured surfaces. Active stereo solves this problem, yet is limited to a working space of a few meters (which is, however, in line with the requirements) and its active illumination might appear intrusive to humans. The main problem – and one that appears to be fundamentally difficult to solve – is implementing a stereo vision system with high resolution and accuracy as real-time system on today's off-the-shelf hardware.
- Single-shot CL is quite similar to active stereo, but has certain key advantages compared to it: CL requires only a single camera. It is intrinsically faster because CL algorithms are able to solve the identification problem by analyzing small local neighborhoods in a single image rather than having to consider at least a good part of an epipolar line to work out the much more difficult correspondence problem. In addition, it is straightforward to make CL systems reliable via error-detecting encoding whereas all stereo systems occasionally produce mismatches where the range data is flat-out wrong. It shares the disadvantage of an active illumi-



nation. Further relevant issues are: a low resolution (with gray level encoding) and that the seemingly necessary spatial encoding fails with surfaces too small to reflect the subpatterns integrally, with certain types of texture or in the case of color encoding also with surfaces of non-neutral, respectively varying reflectivity. However, these problems seem minor compared to the efficiency issue of accurate hi-res stereo.

In light of the above conclusions, we put forward the following two-stage ranging method:

*Coded Light Stage* (see also [Forster et al. 2001a], [Forster et al. 2001b], [Forster et al. 2002] and [Forster et al. 2003]): A new approach to coded light based on spatial encoding represents the initial stage of the proposed ranging technique. Its purpose is to compute a first depth map of the scene, which can then be improved by the latter stages, if necessary. As it uses spatial encoding, the first stage is able to compute this map from a single snapshot of a scene; it consequently copes with dynamic scenes and is in principle capable of a 3D frame rate equal to that of the employed color camera. Moreover, a simple slide projector suffices to generate the necessary un-varying illumination. The higher transmission capacity of a color pattern permits combining small subpatterns – that is ones which take up only a few pixels in the pattern image when projected on suitable surfaces – with a code space large enough for high resolutions and error detecting codes. This would be difficult, if not impossible, with a gray level pattern. Of course, the disadvantage of a color pattern seems to be that decoding requires recognizing the projected colors in the pattern image, yet the reflected colors cannot be relied upon for that task: Perceived red might be due to projected red, but also to projected white and a red surface or projected green in combination with a red surface plus strong white ambient light. Existing color-coded systems exploit the perceived color anyway and simply avoid this problem by assuming a neutral scene reflectivity or by adapting the projected colors with uniformly colored scenes.

To overcome this shortcoming, we develop a one-shot color-coded light technique that copes with a scene reflectivity that varies both spatially and spectrally. We first note that an intrinsic assumption with spatial encoding – namely that most projected subpatterns are reflected rather integrally – corresponds to the assumption that depth and reflectivity of the scene vary smoothly almost everywhere, i.e. to a continuity and a reflectivity smoothness constraint. This implies we can exploit these two constraints without introducing new restrictions. Then, with reflectivity smoothness, the scene exhibits only occasionally reflectivity edges of its own. With continuity, there are only a few edges due to object boundaries or sharp changes in the scene's surface orientation; it also implies that projected edges appear as edge segments in the pattern image and that spatial adjacency relations of the imaged segments will in most cases remain as projected. We consequently propose to use color edge segments as pattern primitives and local edge segment patterns as subpatterns.

The resulting workflow is as follows: The first step is to *detect and classify the imaged color edges* resulting from projecting a suitable color pattern, the second to *establish their spatial adjacency relationships*. In principle, these two steps reconstruct the subpatterns visible in the pattern image. However, the received color signals will typically be noisy, especially with low cost single-chip RGB cameras and strongly colored scenes exhibiting a low reflectivity for some color bands. Also other sources – for instance certain changes in surface normal, object borders or the scene's texture – will give rise to (unwanted and by assumption infrequent) color edges. The key idea to overcome these problems is to combine error detecting encoding with a detection algorithm optimized for robustness: The former creates a means to distinguish between edges that are projected and those that are not. The latter exploits it as much as possible. Precisely how this is done is a rather complex process; it is described over the next sections in detail. Next, established subpatterns that translate (*decode*) into valid codewords are considered identified. Edge segments that are part of an identified subpattern are associated with known planes in 3D space. For these segments, the depth of the imaged scene patch is calculated via *ray-plane triangulation*. There will necessarily be certain interspaces in-between adjacent edge segments. For these interspaces, that is for all remaining pixels part of an identified subpattern, the depth is *interpolated* in a subsequent step.

*Stereo Stage* (also described in [Forster 2004]): The coded light stage relies on a continuity and a reflectivity smoothness assumption; it potentially breaks down if one of them does not apply, e.g. with surfaces too small to reflect subpatterns integrally or with certain types of texture. In most practically relevant cases, the corresponding problematic areas take up only a small part of a scene. Moreover, they typically exhibit a pronounced optical structure that is almost certainly not of a regular, repetitive nature due to the active illumination. The stereo stage tries to exploit that stereo algorithms are well suited for such optically structured surfaces. To that end, it uses a second camera, i.e. a set-up as with an active stereo system. Initially, each camera-projector combination acts as an independent CL system. This way each system obtains as many range values as possible with a method much faster and more reliable than stereo vision, but one that might not acquire range values for all parts of the scene. Next, each CL system shares its results with the other (*mutual update*). Then a *stereo algorithm* attempts to compute depth values for the imaged parts of the scene where both CL systems failed. As these parts will in most cases occupy a relatively small area of an image only, and as the stereo algorithm can build on the results of the CL step, it can be expected to be sufficiently fast for real-time operation. Finally, the *dense sub-pixel correspondence* is established for pixels located in-between color edges identified in both images. This yields the targeted high (non-interpolated) relative spatial resolution, which the coded light step alone cannot achieve for the reasons explained above.

Optionally, we also solve certain problems not directly related to or unnecessary for range acquisition, but which might be useful or even essential for some applications.

- **Scene Reflectance Compensation:** In some cases, it is possible to acquire an additional image of the scene under white illumination. This image allows determining the reflectance properties of each imaged scene patch and compensating their effect on the reflection of the projected pattern; the resulting modified pattern image appears as if the scene had had a uniform and neutral reflectance. Of course, this makes the further processing easier and more robust as the projected and the imaged color then correspond; the disadvantages of the color compensation step are that it requires a projector capable of switching between two distinct projection patterns and a scene that is approximately static between the acquisitions of the two images.
- **Scene Color Computation:** During the coded light step a color image of the scene is acquired, but one that is rather unsuited for further processing due to the irregular illumination. If the identification problem is solved, the spectral composition of the projected illumination is known for each patch of the scene. This information permits computing the local reflectivity of each patch, i.e. reconstructing the intrinsic color of the scene, should it be of interest.
- **Threshold Optimization:** The choice of algorithm thresholds (including camera settings such as the exposure time) has a major impact on the performance of the proposed approach. Any modification of the set-up or a strong change in the ambient conditions might require an adjustment of the threshold. Ideally, the system should adjust them autonomously. With the CL approach, the total number of pixels part of an identified projected color edge segment represents a numeric quality indicator that allows directly comparing the goodness of different parameter sets. As the system employs error detection techniques, it is practically impossible that a bad choice of parameters introduces misidentifications. An algorithm can use this direct feedback to find the optimal choice of thresholds. Invoking it e.g. every  $n$  frames enables the system to cope with the above-mentioned situations without user interaction.

In sum, the resulting combined approach (figure 26) has the potential of meeting all set objectives. It imposes no scene constraints but a reduced continuity and reflectivity smoothness assumption. Reduced in the sense that it does not require depth and reflectivity to vary smoothly almost everywhere as in the original formulation of the constraints, but rather only expects them to vary smoothly for a large part of the scene. Only few surfaces of practical interest conflict with these assumptions. Moreover, the combined approach minimizes the problem of occlusion: depth values can be computed if a scene spot can be seen by any two of the three system components.

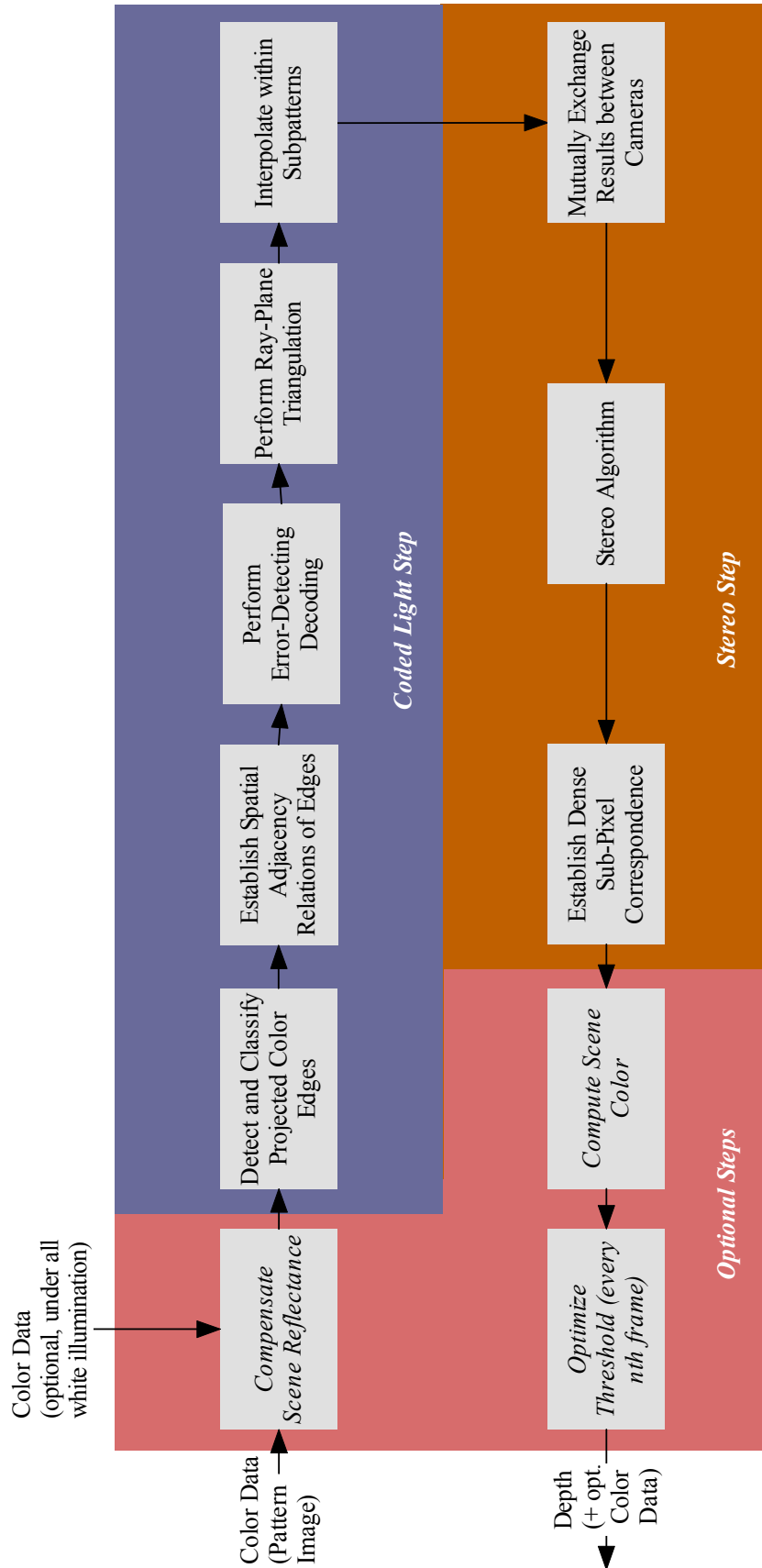


Figure 26: High-level workflow of the proposed approach to range image acquisition.

## 4.3 The Coded Light Subsystem

This section discusses all aspects of the proposed coded light subsystem.

### 4.3.1 Calibration

To obtain range values via ray-plane intersection, the geometric and optical parameters of both camera and projector – according to some suitable geometric camera and projector model – have to be known. Only then can image coordinates be associated with known straight lines (of view), respectively projected light planes with known planes in 3D space. Determining these parameters is a process called *camera*, respectively *projector calibration*. As it is not practical to directly measure the parameters, they are estimated indirectly: For camera calibration, a target that contains marks of known world coordinates is imaged and the image coordinates of the marks are established. For projector calibration, a pattern with marks of known slide coordinates is projected and the world coordinates of the marks are determined. Both approaches yield a set of  $n$  observation vectors, each made up of image with matching world coordinates  $(x_{ij}, y_{ij}, x_{wj}, y_{wj}, z_{wj})_{1 \leq j \leq n}$ . The model parameters that fit best to this data, in our case in the least square sense, are then taken to be the desired camera, respectively projector parameters.

The approach to camera calibration taken in this work builds on and extends a well-established technique introduced by Tsai [1986, 1987]. The next sections present Tsai's method and the one proposed by us, discuss the topic of projector calibration and conclude with experimental results.

#### 4.3.1.1 Tsai's Camera Calibration Technique

This work's calibration technique is based on Tsai's approach as the latter has certain key advantages:

- **Autonomous Operation:** The procedure does not require a good initial guess or any other kind of user intervention, i.e. it operates autonomously. Placing the calibration target in front of the lens and invoking the algorithm is all an operator has to do for camera calibration.
- **Simple Calibration Target:** As a general rule, the systematic error of the calibration mark coordinates should be one order of magnitude smaller than the targeted 3D measurement accuracy [Tsai 1986]. Also the marks should fill the whole field of view. In our case, obtaining a 3D calibration target that spans the targeted large working space and whose systematic error (e.g. due to mechanical instability such as distortion) is and remains over time in the area of 0.01 mm is very difficult. A significant advantage of Tsai's technique is that it works well even with simple planar calibration targets, which can be constructed and shipped much more easily; they also tend to be mechanically more stable than their 3D counterparts. For this reason, we assume and exclusively use a planar calibration target.
- **Accurate Results:** The literature states that the technique produces accurate results.

Tsai's camera model is the pinhole model of 3D-2D perspective projection with 1st order radial lens distortion as introduced in section 2.2. Today's cameras transfer their data digitally to the processing unit; accordingly, effects such as synchronization problems that an analogous transfer of the image data could cause can be ignored. Consequently, the model has in total 10 parameters, namely four *internal parameters*, also called *intrinsic* or *interior parameters*

- **the effective focal length**  $f_k$  of the pinhole camera.
- **the 1st order radial lens distortion coefficient**  $\kappa_1 = \kappa$ .
- **the optical image center**  $C = (c_x, c_y)$ , defined as the intercept of the camera's optical axis with the camera's sensor plane and assumed to coincide with the center of radial lens distortion. It is specified in pixels.

and 6 external parameters, which are

- **the three translation parameters**, i.e. the entries of the three-dimensional translation vector  $\mathbf{t}$  of equation 16 that describes the position of the camera relative to the world-coordinate system.
- **the three rotation parameters**, i.e. the three Euler angles  $R_x$ ,  $R_y$  and  $R_z$  equivalent to the 9 elements of the rotation matrix  $\mathbf{R}$  of equation 16 that describes the orientation of the camera relative to the world-coordinate system.

The core idea of Tsai's approach is to find and exploit linear relationships that involve only a subset of the model parameters rather than directly operating in the full parameter space. The resulting algorithm operates in two stages; the first stage deals with deriving and solving such linear systems, while the second puts it all together and performs a non-linear optimization of the entire model.

*Stage 1a (Resulting parameters  $\mathbf{R}$ ,  $t_x$ ,  $t_y$ ):* So far we more or less ignored that there are two distinct types of image coordinates: *frame coordinates* as e.g. supplied by the frame grabber or used for image processing, which have pixels as units and the top-left pixel as origin, and *sensor coordinates*, which have millimeters as units and whose origin is the optical center of the camera's retinal plane. Of course, the perspective imaging equation (equation 14) applies only to the latter. Originally, the observed positions of calibration marks are given in frame coordinates  $(x_f, y_f)$  only. Accordingly, they need to be transformed into physically meaningful sensor coordinates  $(x_s, y_s)$ . This transformation involves the yet unknown optical image center. Tsai assumes that the arithmetic image center ( $c_x' = N_x/2$ ,  $c_y' = N_y/2$ ) is a reasonably close approximation to the optical one. The conversion of image to sensor coordinates is then done via

$$x_s = (x_f - c_x') \cdot dx \text{ [mm]} \quad y_s = (y_f - c_y') \cdot dy \text{ [mm]} \quad (51)$$

where  $dx$  and  $dy$  are the physical dimensions of the sensor elements in millimeters. As discussed in section 2.2, the above coordinates are the actually observed, distorted sensor coordinates; so in the following we refer to them as  $(x_d, y_d)$  to distinguish them from the (so far unknown and non-observable) ideal undistorted coordinates  $(x_u, y_u)$ . The perspective imaging equation (equation 14) applies only to the latter. From this equation Tsai derives the following constraint:

$$x_u = x_c f_k / z_c \wedge y_u = y_c f_k / z_c \Rightarrow x_u y_c - y_u x_c = 0 \quad (z_c \neq 0) \quad (52)$$

Interpreting this constraint geometrically leads to the conclusion that the ideal image  $(x_u, y_u)$  of a point located at camera-coordinates  $(x_c, y_c, z_c)$  and its  $z_c$ -projection  $(x_c, y_c)$  are radially aligned: their outer vector or cross-product is zero. In contrast to the law of perspective projection, this radial alignment property also applies to the distorted coordinates as the effect of the distortion is – according to the camera model – exclusively radial as well. This follows analytically by combining equations 20 and 52:

$$x_u y_c - y_u x_c = x_d (1 + \kappa r^2 \dots) y_c - y_d (1 + \kappa r^2 \dots) x_c \Rightarrow x_d y_c - y_d x_c = 0 \quad (z_c \neq 0) \quad (53)$$

To exploit the radial alignment property, the  $x$  and  $y$  camera coordinates of the calibration marks are needed. As only their world coordinates  $(x_w, y_w, z_w)$  are known so far, we have to transform the latter into camera coordinates  $(x_c, y_c, z_c)$  via rotation and translation as in equation 16. Equation 53 then reads:

$$x_d (r_{21} x_w + r_{22} y_w + r_{23} z_w + t_y) - y_d (r_{11} x_w + r_{12} y_w + r_{13} z_w + t_x) = 0 \quad (54)$$

As pointed out above, we assume a planar calibration target. We may choose its surface as the  $z_w = 0$  plane of the world coordinate system without loss of generality. Then  $z_w$  becomes zero for all marks and its associated rotation parameters vanish. Separating known from unknown parameters turns equation 54 into the following linear equation of only 5 unknowns:

$$\begin{bmatrix} y_d x_w & y_d y_w & y_d & -x_d x_w & -x_d y_w \end{bmatrix} \begin{bmatrix} t_y^{-1} r_{11} = \tilde{r}_{11} \\ t_y^{-1} r_{12} = \tilde{r}_{12} \\ t_y^{-1} t_x \\ t_y^{-1} r_{21} = \tilde{r}_{21} \\ t_y^{-1} r_{22} = \tilde{r}_{22} \end{bmatrix} = x_d \quad (55)$$

Setting up the above equation for each calibration mark gives an overdetermined system of linear equations that can be solved in the least-square sense for the five unknowns provided that  $n$  is much larger than 5 (barring certain very unlikely degenerate cases). This yields the top-left 2 by 2 sub-matrix of the orthonormal rotation matrix  $\mathbf{R}$  scaled with the unknown factor  $t_y^{-1}$ . Exploiting the orthogonality and normality of a rotation matrix allows determining the squared inverse of the scale factor, i.e.  $t_y^2$ , uniquely (again but for very unlikely degenerate cases) as:

$$t_y^2 = \frac{\tilde{r}_{11}^2 + \tilde{r}_{12}^2 + \tilde{r}_{21}^2 + \tilde{r}_{22}^2 - \sqrt{(\tilde{r}_{11}^2 + \tilde{r}_{12}^2 + \tilde{r}_{21}^2 + \tilde{r}_{22}^2)^2 - 4(\tilde{r}_{11}\tilde{r}_{22} - \tilde{r}_{12}\tilde{r}_{21})^2}}{2(\tilde{r}_{11}\tilde{r}_{22} - \tilde{r}_{12}\tilde{r}_{21})} \quad (56)$$

It remains to derive the sign of  $t_y$ . To that end, we make the following two observations:

- The choice of  $t_y$  uniquely determines the remaining parameters  $r_{11}$ ,  $r_{12}$ ,  $r_{21}$ ,  $r_{22}$  and  $t_x$ . Given these six parameters, the  $x$  and  $y$  camera coordinates of a calibration mark can be computed from its world coordinates. Reversing the sign of  $t_y$  reverses the sign of the other five parameters, i.e. if we obtain the camera coordinates  $(x_c, y_c)$  for given world coordinates  $(x_w, y_w, 0)$  with a positive, we get  $(-x_c, -y_c)$  with a negative sign of  $t_y$ .
- Each calibration mark is in front of the camera, i.e.  $z_c > 0$ , actually  $z_c > f_k$ , and the effective focal length is positive, i.e.  $f_k > 0$ . Then a point with a positive  $x$  (or  $y$ ) coordinate in the camera coordinate system is imaged at a positive image plane  $x$  (or  $y$ ) coordinate and vice versa.

These findings allow hypothesizing a positive  $t_y$ , choosing a calibration mark that is imaged far away from the image center and transforming its world into (incomplete) camera coordinates. If the resulting camera coordinate  $x_c$  has the same sign as  $x_d$  and  $y_c$  and  $y_d$  match as well, the hypothesis is correct. Otherwise, it is wrong and the value of  $t_y$  is set to  $-(t_y^2)^{1/2}$ . With  $t_y$ , the top-left 2 by 2 submatrix of  $\mathbf{R}$  is determined. From the fact that a rotation matrix is orthonormal and in our case right-handed, the remaining entries of  $\mathbf{R}$  follow straightforwardly.

*Stage 1b (Resulting parameters  $f_k$ ,  $t_z$ ):* In a next stage, initial estimates of the effective focal length  $f_k$  and of the remaining unknown translation parameter  $t_z$  are computed. To this end, the results of stage 1 are substituted into equation 16:

$$y_u = f_k \frac{\overbrace{r_{21}x_w + r_{22}y_w + t_y}^{y_c}}{\underbrace{r_{31}x_w + r_{32}y_w + t_z}_{z'_c}} = f_k \frac{y_c}{z'_c + t_z} \Rightarrow y_u z'_c + y_u t_z = f_k y_c \quad (57)$$

The above equation involves the unknown undistorted sensor coordinates. At this point this problem is solved by simply ignoring the radial distortion, i.e. by assuming  $x_d = x_u$ , respectively  $y_d = y_u$ . Separating the known factors from the two remaining unknowns  $f_k$  and  $t_z$  yields for each calibration mark the following linear equation:

$$\begin{bmatrix} y_d & -y_c \end{bmatrix} \begin{bmatrix} t_z \\ f_k \end{bmatrix} = -y_d z'_c \quad (58)$$

The resulting overdetermined linear equation system over all marks can be solved for the two unknowns unless the target plane is parallel to the image plane, in which case  $y_d$  and  $y_e$  are linearly dependent. Accordingly, Tsai recommends an angle between the two planes of at least  $30^\circ$ . It seems that an even larger angle would improve things further, but typically the resulting increased foreshortening amplifies the noise in the observed image coordinates, more than offsetting any gain from a certain tilt on. The solutions of the equation system represent preliminary values only as radial distortion cannot truly be disregarded. Nevertheless they are well suited as initial guess for the non-linear optimization coming next.

*Stage 2 (Resulting parameters are all 10 model parameters):* In the final stage, the radial distortion  $\kappa$ , the exact solutions for the effective focal length  $f_k$ , the translation parameter  $t_z$  and the optical image center  $(c_x, c_y)$  are determined. Substituting what is known so far into equation 16, this time considering radial distortion, yields:

$$x_u = f_k \frac{r_{11}x_w + r_{12}y_w + t_x}{r_{31}x_w + r_{32}y_w + t_z} \Rightarrow x_d (1 + \kappa r^2) = f_k \frac{x_c}{z'_c + t_z} \quad (59)$$

We can formulate an analogous equation for each observed  $y$  sensor coordinate. These two equations are also a function of the image center as it controls the transformation of image to sensor coordinates. All in all, this allows setting up the following non-linear error or cost function of the remaining five unknowns that describes the squared modeling error over all calibration marks:

$$f(c_x, c_y, \kappa, f_k, t_z) = \sum_{j=1}^n \left( f_k \frac{x_{cj} + y_{cj}}{z'_{cj} + t_z} - x_d (1 + \kappa r^2) \right)^2 \quad (60)$$

Minimizing the error function yields the remaining five unknown parameters. Provided the optical is not too far away from the arithmetic image center and the radial distortion is not too strong ( $\kappa \approx 0$ ), a very good initial guess for the parameter set is available. Accordingly, any of the classical non-linear optimization approaches such as steepest descent can be expected to give good results.

Tsai's technique as described in his papers ends with the above step. However, its implementations typically perform a second run of the algorithm where they use the obtained image center to convert image to sensor coordinates in stage 1a and also consider radial distortion in stage 1b. In a last step, they explicitly consider the remaining model parameters (implicit in  $x_{cj}$ ,  $y_{cj}$  and  $z'_{cj}$  of equation 60) of the error function and carry out a non-linear optimization in the full parameter space. Horn [2000] points out that this is rather necessary as the linear equation systems of stage 1 did not minimize the squared modeling error, but their own algebraic error functions. The elements of the 10-dimensional vector of minimal squared error are then taken as the camera's model parameters.

#### 4.3.1.2 An Improved Version of Tsai's Camera Calibration Technique

In general, Tsai's technique yields accurate results. As all calibration methods, it is, however, confronted with two disturbing factors, namely

- a significant dependency between certain model parameters, for instance between the effective focal length and the camera's distance to the scene.
- noisy observation vectors; with low cost cameras as targeted in this work, this applies in particular to the frame/image coordinates. Respectively, a coplanar calibration target is typically built by printing a pattern on a sheet of paper and gluing it on some approximately planar object. In our case, this process resulted in  $z$  world coordinates that depart up to  $\pm 0.05\text{mm}$  from the assumed flatness; even if these deviations are known, Tsai's original approach does not consider such minor, but non-negligible non-coplanarity. Ignoring the actual  $z$  values effectively turns them into high-amplitude, typically non-Gaussian and non-zero-mean noise.

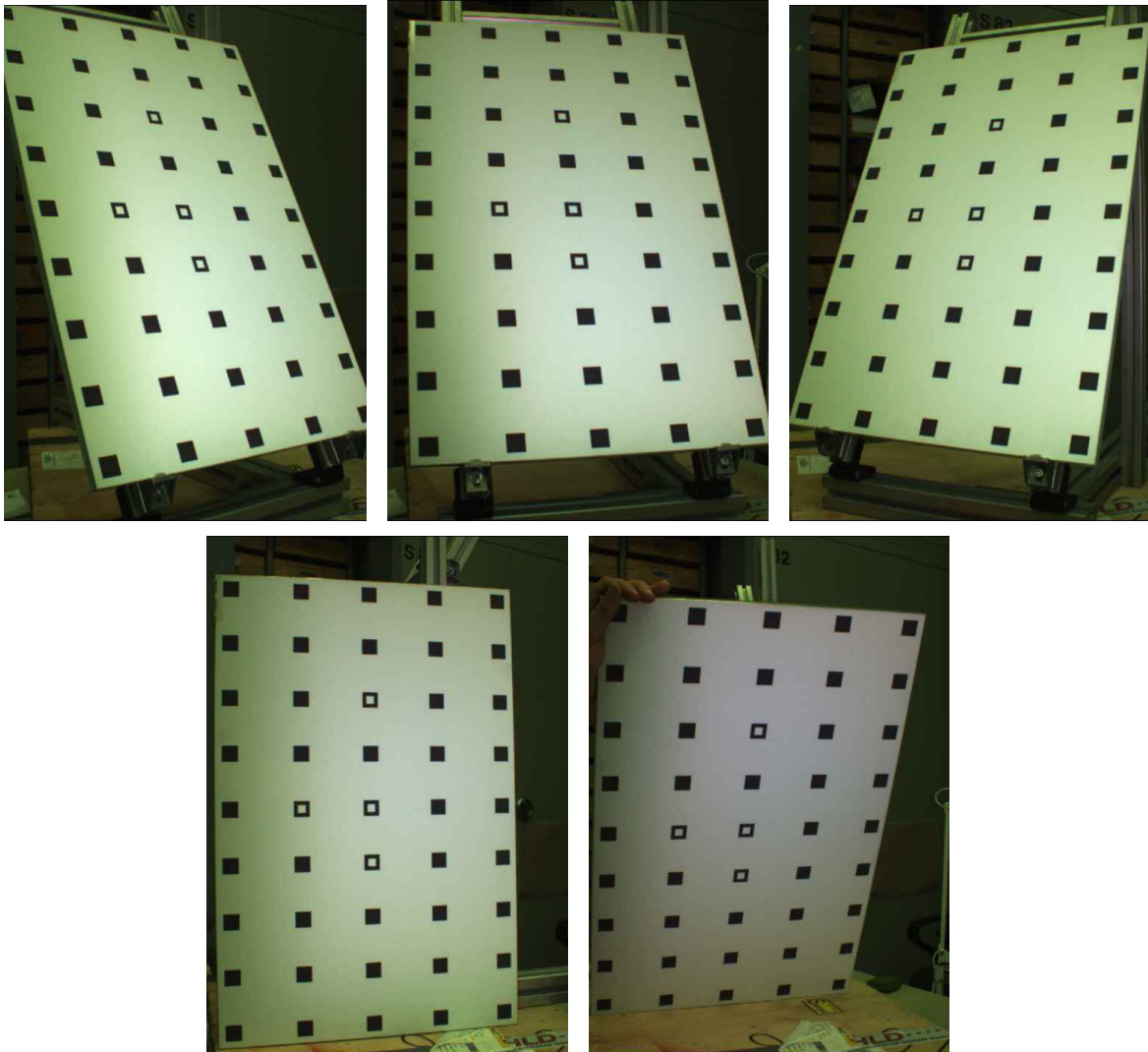


Figure 27: Exemplary views of the calibration target from distinct unknown positions as recommended for the proposed multi-view calibration technique.

Given only the limited information of a single snapshot of a planar calibration target, the interaction of the above factors becomes a serious problem; it causes several slightly distinct parameter sets to describe the observed data equally well, or, with other words, it results in many local minima of the cost function that are almost indistinguishable from the global one. Tsai's algorithm tends for that reason to converge to a local minimum that represents a slightly erroneous parameter set, and to a somewhat different one for each calibration attempt on top. This is barely noticeable when considering points on the calibration plane from the point of view taken for calibration. Only far away in 3D space from this plane or from a distinct perspective the resulting calibration error becomes perceptible. Analyzing the repeat accuracy reveals the problem as well. Both effects are demonstrated experimentally below.

To overcome this effect, we acquire  $m > 1$  images of the planar calibration target from  $m$  distinct viewpoints as shown in figure 27 and establish the set of internal parameters that fits best over all views rather than only one. The  $m - 1$  additional images from unknown viewpoints solely serve the purpose of stabilizing the parameter estimates. Acquiring them is no major effort as it suffices to move the target to a few unknown, but notably distinct positions and orientations, which is equivalent to translating and rotating the camera.



The proposed approach is in principle closely related to the idea of *bundle adjustment* widely used in the field of photogrammetry (see e.g. [Luhmann 2000]). Corresponding photogrammetric calibration methods are typically very accurate, but difficult to use. This is in part due to the fact that they tend to be more complex by design. For example, they often do not distinguish between calibration of the camera(s) and 3D reconstruction of the scene, but combine both steps into a single computation. The advantage of this approach is that it uses the  $m$  points of view not only for calibration, but also for determining the 3D coordinates of certain scene points (i.e. it performs triangulation with up to  $m$  lines of view); consequently, it yields 3D data that is very accurate and that can easily be combined into a complete 3D model of the scene. A key difference between the proposed and photogrammetric approaches is that the latter typically require initial guesses and calibration marks that are not coplanar, for instance a 3D calibration rig such as two orthogonal planes.

There seem to be only two calibration approaches similar to the proposed one – that is operating without initial guesses and relying on observing a simple model plane from several distinct unknown viewpoints – discussed in the literature. The one by Triggs [1998] requires at least five distinct views, the one by Zhang [1998, 2001] can do with two views and appears to be more widely used; it is the calibration method of the popular, freely available Intel Computer Vision (ICV) library [Intel 2001]. It is for that reason used as reference and benchmark in the following.

Zhang's method operates as follows: Initially, it ignores lens distortion and computes for each view the homography between the model plane and its image that gives the least squared error. Even though this homography could be found with linear methods, Zhang determines it via non-linear minimization, probably because the latter approach is more robust against noise ([Faugeras 1993]). From the  $m > 1$  homographies, a closed form solution for the intrinsic camera parameters (excluding distortion related ones) is derived from geometric constraints. Given the most relevant internal parameters, the extrinsic ones of each view follow easily. These results allow setting up an overdetermined linear system to estimate the distortion parameters. Zhang's original method considers two radial distortion coefficients, its ICV version also two tangential ones. Given estimates of all parameters, a non-linear optimization refines the complete model; again the minimization criterion is the squared modeling error.

Clearly deriving initial estimates for all parameters from a linear image formation model is a process that is more vulnerable to parameter interdependencies than a two-stage approach. Zhang's results reflect this: for instance, he reports that the detected outer points lie consistently further away from the image center than the ones predicted by the initial linear model. Consequently, the linear estimation of the radial distortion yields a distortion of the wrong type (pincushion instead of the actual barrel distortion) because it tries to push the modeled outer points further away from the optical image center to bring observation and model closer together.

It seems for this reason promising to combine Tsai's robust two-stage single-view approach with  $m$ -view bundle adjustment. We show in the following that this can be done with relatively little effort. In the following, we describe the proposed changes to Tsai's technique for each stage:

*Stage 1a (Resulting parameters  $c_x$ ,  $c_y$  and  $\mathbf{R}$ ,  $t_x$ ,  $t_y$  for each view):* The first stage again exploits the radial alignment constraint. As explained above, doing so requires knowledge of the optical image center. Tsai's solution is to hypothesize that the optical and the arithmetic image center are about the same. This assumption is a potential weakness of his method: even though it holds true in most cases, it is occasionally quite off. In our experience, this applies in particular to projection devices; for instance, the optical slide plane center of the LCD projector used in this work is located at pixel coordinates (512, 760), which is non-negligibly distinct from the arithmetic center (512, 384) determined by the native projector resolution of 1024 by 768 pixels.

For this reason, we determine the optical image center in stage 1a to prevent latter stages from being guided into the wrong direction. Tsai proves in his paper that the overdetermined equation system 55 has full column rank, i.e. a unique solution for each view. Equation 54 is a function of the optical image center as the image coordinates depend on it. This implies the function  $f(c_x, c_y)$  that maps optical image center coordinates on the squared residual error over all views is well defined:

$$f(c_x, c_y) = \sum_{i=1}^m \sum_{j=1}^{n_j} \left( \begin{array}{l} (x_{f,i,j} - c_x)(r_{21,i}(c_x, c_y)x_{w,i,j} + r_{22,i}(c_x, c_y)y_{w,i,j} + t_{x,i}(c_x, c_y)) - \\ (y_{f,i,j} - c_y)(r_{11,i}(c_x, c_y)x_{w,i,j} + r_{12,i}(c_x, c_y)y_{w,i,j} + t_{x,i}(c_x, c_y)) \end{array} \right)^2 \quad (61)$$

where the rotation and translation parameters are the uniquely defined solution of the overdetermined equation system resulting from the choice of the image center. The non-linear minimization of equation 61 typically yields a better estimate of the optical image center. It is, however, not guaranteed that a wrong choice of the optical center results in a greater squared residual error over all views than the correct one, i.e. that the above minimization actually converges to the intended solution. For this reason, the proposed algorithm is split up into two branches, where the first uses the result of the above optimization as preliminary image center, while the second uses the arithmetic image center as with Tsai's original approach. The branch with the least residual error provides the final output. The only potential disadvantage of this approach is an approximately doubled execution time of algorithm, which is irrelevant in the case of a one-time off-line calibration.

*Stage 1b (Resulting parameters  $t_z$  for each view, plus  $f_k$  common to all views):* The key idea of the modified algorithm is to find a linear relationship that relates a subset of the internal parameters to observations made from many distinct unknown viewpoints. To this end, we observe that we are able to modify equation 58 of Tsai's approach accordingly because each viewpoint has its own  $z$  translation  $t_{z,i}$ , but all views share the same effective focal length. We may consequently set up the following overdetermined linear equation system of  $m + 1$  unknowns:

$$\begin{bmatrix} y_{d,1,1} & 0 & \dots & \dots & \dots & 0 & -y_{c,1,1} \\ y_{d,1,2} & 0 & \dots & \dots & \dots & 0 & -y_{c,1,2} \\ \dots & \dots & \dots & \dots & \dots & 0 & \dots \\ 0 & y_{d,2,1} & \dots & \dots & \dots & 0 & -y_{c,2,1} \\ \dots & \dots & \dots & \dots & \dots & 0 & \dots \\ 0 & 0 & \dots & y_{d,i,j} & \dots & 0 & -y_{c,i,j} \\ \dots & \dots & \dots & \dots & \dots & 0 & \dots \\ 0 & 0 & \dots & \dots & \dots & y_{d,m,n_m-1} & -y_{c,m,n_m-1} \\ 0 & 0 & \dots & \dots & \dots & y_{d,m,n_m} & -y_{c,m,n_m} \end{bmatrix} \begin{bmatrix} t_{z1} \\ t_{z2} \\ \dots \\ t_{zi} \\ \dots \\ t_{zm} \\ f_k \end{bmatrix} = \begin{bmatrix} -y_{d,1,1} z'_{c,1,1} \\ -y_{d,1,2} z'_{c,1,2} \\ \dots \\ -y_{d,2,1} z'_{c,2,1} \\ \dots \\ -y_{d,i,j} z'_{c,i,j} \\ \dots \\ -y_{d,m,n_m-1} z'_{c,m,n_m-1} \\ -y_{d,m,n_m} z'_{c,m,n_m} \end{bmatrix} \quad (62)$$

where the first numeric subscript of a variable refers to the view index, the second to the calibration mark index. Solving the above system yields the focal length that fits best to all rather than to a single point of view. This modification solves one of the key problems of Tsai's approach, namely the dependency between the estimated focal length and the spatial position of the camera and the resulting parameter instability.

*Stage 3 (Resulting parameters are all  $4 + 6m$  model parameters):* The third stage is the non-linear optimization of a cost function that involves all images and the complete parameter space. Compared to Tsai's approach, the dimensionality of this space increases from 10 to  $4 + 6m$ . However, this high dimensionality is unproblematic given the very good quality of the initial guess.

Furthermore, we propose to no longer assume the world coordinates of the calibration marks to be zero, but rather to consider their actual z coordinates during optimization, should they be known. In total, the resulting error function then becomes rather complex:

$$f(c_x, c_y, \kappa, f_k, R_{x1}, R_{y1}, R_{z1}, t_{x1}, t_{y1}, t_{z1}, \dots, R_{xm}, R_{ym}, R_{zm}, t_{xm}, t_{ym}, t_{zm}) = \sum_{i=1}^m \sum_{j=1}^{n_i} \left( f_k \frac{r_{11,i}x_{w,i,j} + r_{12,i}y_{w,i,j} + r_{13,i}z_{w,i,j} + t_{x,i}}{r_{31,i}x_{w,i,j} + r_{32,i}y_{w,i,j} + r_{33,i}z_{w,i,j} + t_{z,i}} - (x_{f,i,j}dx + y_{f,i,j}dy - c_x dx - c_y dy) \left( 1 + \kappa dx^2 (x_{f,i,j} - c_x)^2 + \kappa dy^2 (y_{f,i,j} - c_y)^2 \right) \right)^2 \quad (63)$$

An existing implementation of Tsai's algorithm can be extended with comparatively little effort to include the presented modifications. It is important to note that the latter do not take away from the ease of use of Tsai's original method; it suffices to move and rotate the calibration target by hand as nothing needs to be known or is assumed of the unknown positions as long as they are somewhat different from each other. Even if they are not, the algorithm degrades to Tsai's original approach and still converges to an acceptably accurate solution. Respectively, if high calibration accuracy is not called for, it explicitly permits a single view approach. The only disadvantage of the proposed approach is the slightly longer time span needed to acquire several images instead of only one and to deal with the now significantly higher dimensionality of the search space, respectively to pursue both branches of the algorithm. As verified experimentally below, this effort is well invested given that the accuracy of the calibration method increases in many cases by a factor of about 10.

In the following, we add the subscript c to the obtained model parameters of the camera (e.g.  $\mathbf{R}_c$  for its rotation matrix) to avoid confusion with the results of the projector calibration discussed next.

#### 4.3.1.3 Projector Calibration

For projector calibration, a pattern containing marks of precisely known slide coordinates is projected and the world coordinates of each projected mark are established. Consequently, projector calibration is in a way the opposite of camera calibration as in its case the slide coordinates are known and the world coordinates have to be established. It is not practical to measure the latter off-line as with a camera calibration target. Rather, the projector illuminates the planar camera calibration target with its calibration pattern, where it is assumed that the position of the target relative to the camera is known from a previous camera calibration. Next, the camera image coordinates of the projected marks are established. This yields for each mark a camera line of view in the world coordinate system, intersecting this straight line with the well-known calibration target plane yields the 3D world coordinates of the projected marks. In combination with their known slide coordinates, this approach again produces 5-dimensional observation vectors.

Employing a colored calibration pattern, e.g. one made of red squares on a white background, would simplify the task of distinguishing between the marks of the camera calibration target and the projected ones. As shown in figure 28, the implementation of this work nevertheless uses the corners of black squares as fiducial marks; while doing so makes detecting the projected marks more challenging, it also works with black-and-white cameras, i.e. represents a more general approach to projector calibration.

Once the observation vectors are obtained, projector calibration is in no significant way different from camera calibration. So the multi-view calibration technique discussed in the previous section can be applied without any changes. In the following, we add the subscript p to the resulting projector model parameters (e.g.  $\mathbf{R}_p$  for its rotation matrix) to avoid confusion with the results of the camera calibration.

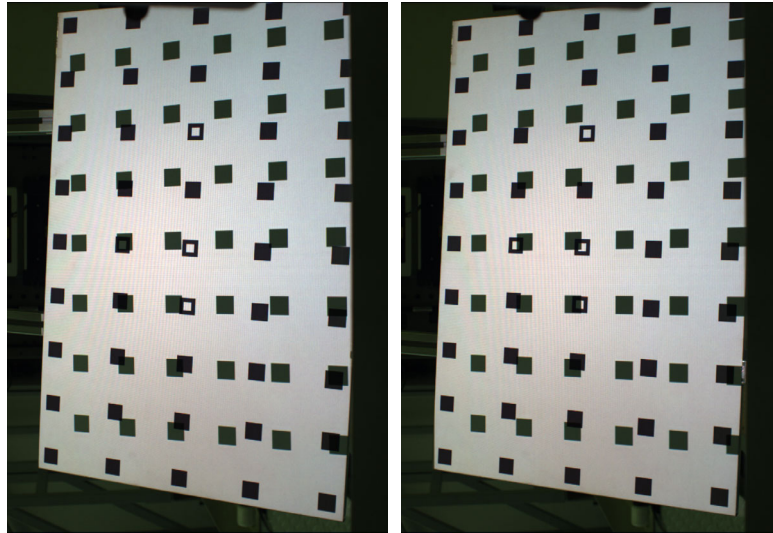


Figure 28: Exemplary images of the camera calibration target illuminated by the calibration pattern as they are used for projector calibration.

It is important to note that – other than most projector calibration approaches reported in the literature (e.g. [Mc Ivor 1994], [Monks 1994] or [Teschner 1996]) – the proposed technique also determines the distortion of the projector lens. Section 4.3.4.2 shows that considering the projector distortion is necessary to obtain an accurate structured light system but for the unlikely cases in which the projector lens does not exhibit significant distortion.

As camera and projector share the same world coordinate system, coordinates can be transformed easily from the camera to the projector coordinate system and vice versa. E.g. the conversion of projector coordinates  $(x_p, y_p, z_p)$  to camera coordinates  $(x_c, y_c, z_c)$  is done as follows:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \mathbf{R}_c \left( \mathbf{R}_p^{-1} \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} - \mathbf{R}_p^{-1} \mathbf{t}_p \right) + \mathbf{t}_c = \mathbf{R}_c \mathbf{R}_p^{-1} \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} - \mathbf{R}_c \mathbf{R}_p^{-1} \mathbf{t}_p = \mathbf{R}_{p2c} \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} + \mathbf{t}_{p2c} \quad (64)$$

The only disadvantage of the proposed method of projector calibration is that the error of the camera calibration propagates into the projector model. However, the next section shows that the camera calibration produces very accurate results. The modeling error for points on the calibration plane has a standard deviation in the area of 0.03 – 0.06 mm in object space over the targeted large working space; the systematic component of this error seems to be significantly smaller. So this dependency is unproblematic; this assessment is confirmed by the high ranging accuracy of the triangulation system evaluated in chapter 5.

#### 4.3.1.4 Experimental Results

This section describes calibration experiments with real data. As the execution time of the algorithm is in the area of one second on a standard PC, and as calibration needs to be done only once at system set-up, the topic of efficiency is rather irrelevant for our purposes and neither evaluated nor discussed in more detail here. So the sole purpose of the experiments of this section is evaluating the accuracy of the proposed calibration methods, absolutely as well as relative to the state of the art.

The camera used for the experiments is a Basler 302fc off-the-shelf single-chip color (Bayer pattern) camera of a resolution of 780 by 580 pixels. It transmits its data digitally via an IEEE 1394a (fire wire) interface. The employed lens is a standard TV lens of 12.5mm focal length (Cosmicar Pentax 12.5mm). Forster [2005] lists experiments with other camera-lens combinations that show that the results of this section are representative.

The calibration pattern is made up of 5 by 9 black squares on a white background of size 600 mm by 400 mm. The resulting 140 square corners serve as fiducial marks for calibration. The squares have a side length of 20 mm and are spaced 70 mm apart along the longer, 90 mm apart along the shorter side of the plate. The squares were printed with an ink jet printer on white paper, which was then put on a glass plate. The exact positions of the square corners were measured with an accuracy of  $\pm 0.01$  mm. It turns out that the deviations created by the limited accuracy of the ink jet printer go systematically up to 0.5 mm, the ones due to the non-flatness of the glass, respectively the glued on paper range from -0.10 mm to +0.06 mm; they are systematic as well.

With a pinhole camera, a straight line in 3D space is imaged as 2D straight plane. Square edges of black squares on white background can be detected easily. With standard lenses, the effect of radial distortion is about constant over a few (10-30) pixels. In combination, this allows to least-square fit straight lines to the square edge segments detected in the image. Intersecting the resulting straight lines yields the sub-pixel square corner position. This approach locates black corners on a white background with a sub-pixel accuracy of ca. 0.05 to 0.1 pixels standard deviation using the above mentioned single-chip color camera/lens combination (the stated accuracy refers to the typical about 30° tilt of the calibration plane as this accuracy depends to some extent on the angle of view). With black-and-white cameras, the results are notably better, indicating the technique is in line with the state of the art with respect to target location accuracy.

The proposed extension of Tsai's algorithm was implemented, where the non-linear minimization problem is solved by a modified Levenberg-Marquardt algorithm with a Jacobian calculated by forward-difference approximation. Such an algorithm is part of the well-known Minpack-1 package [Moré 1980]; this Minpack-1 implementation of a modified version of the Levenberg-Marquardt algorithm has been used for all experiments of this section.

We describe several distinct experiments with Tsai's algorithm (Wilson's freely available implementation), Zhang's method (Intel Computer Vision library implementation) and our own implementation of the method proposed in this work. To evaluate the quality of each method, we take the following approach: We consider a calibration attempt as measurement of the model parameters. It is, as any measurement, affected by a statistical error, which is assessed by repeating the same measurement several times and by analyzing the distribution of the measured values. In this context, we use the following indicator:

- The (*sample*) *standard deviation* of parameters over repeated calibration attempts. To make this indicator more readable, we normalize it by the mean value, i.e. we specify the standard deviation as percentage of the sample mean. Its acronym is NSDev.

If a measurement method has a small statistical error, it is also called precise. More important than the statistical is the systematic error. A measurement approach affected by a negligible systematic error is considered accurate, even if it is imprecise. Unfortunately, the systematic error is difficult to assess with the task at hand because the ground truth, the set of correct parameter values, is unknown.

Following a practice commonly accepted in the literature on camera calibration, we employ the following substitute indicators for the overall accuracy of a calibration method, i.e. the one including both the stochastic and systematic error:

- The *modeling* or *image plane error* is the distance, specified in image pixels, between the observed image plane location of a calibration mark and that predicted by the calibrated model. The non-linear optimization minimizes the **root of the mean squared (rms)** error of this type.

	Attempt 1	Attempt 2	Attempt 3	Attempt 4	Attempt 5	NSDev [%]
Image Plane Center	400.43	399.04	400.60	395.17	401.34	0.552
$C_x, C_y$ [pix]	305.46	303.53	305.24	301.09	305.13	0.542
Focal Length [mm]	12.382	12.407	12.387	12.443	12.376	0.196
Radial Distortion $\kappa$ [ $1/\text{mm}^2$ ]	$4.768 \cdot 10^{-4}$	$4.769 \cdot 10^{-4}$	$4.760 \cdot 10^{-4}$	$4.808 \cdot 10^{-4}$	$4.748 \cdot 10^{-4}$	0.442
Translation	223.43	224.08	223.57	225.36	223.33	0.334
$T_x, T_y, T_z$ [mm]	566.30 498.03	566.94 498.88	566.35 498.41	567.74 499.89	566.42 497.85	0.096 0.146
Rotation	-22.848	-22.937	-22.850	-23.077	-22.845	0.392
$R_x, R_y, R_z$ [deg]	27.849 11.688	27.891 11.740	27.845 11.700	28.015 11.804	27.814 11.697	0.253 0.367
Image Plane Error rms, max [pix]	0.057 0.248	0.058 0.281	0.059 0.274	0.058 0.244	0.060 0.267	
Object Space Error rms, max [mm]	0.033 0.148	0.034 0.157	0.035 0.153	0.036 0.156	0.036 0.183	
Angle bet. Planes [deg]	35.432	35.515	35.430	35.691	35.404	0.296

Table 1: Results of Tsai camera calibration repeated for 5 images from the same point of view taken under slight variations of theoretically irrelevant parameters such as the exposure time.

	Attempt 1	Attempt 2	Attempt 3	Attempt 4	Attempt 5	NSDev [%]
Image Plane Center	391.69	397.97	379.66	371.87	396.54	2.618
$C_x, C_y$ [pix]	308.54	305.20	304.20	288.52	308.74	2.468
Focal Length [mm]	12.501	12.416	12.446	12.683	12.015	1.766
Radial Distortion $\kappa$ [ $1/\text{mm}^2$ ]	$4.916 \cdot 10^{-4}$	$4.807 \cdot 10^{-4}$	$4.877 \cdot 10^{-4}$	$4.838 \cdot 10^{-4}$	$4.741 \cdot 10^{-4}$	1.239
Image Plane Error rms, max [pix]	0.060 0.326	0.046 0.208	0.050 0.239	0.075 0.380	0.092 0.378	
Angle bet. Planes [deg]	45.378	38.221	26.412	29.014	17.546	

Table 2: Internal parameters resulting from repeating Tsai’s methods for 5 distinct points of view.

- The *object space error* is the distance of closest approach between a calibration point in 3D space and the line of sight formed by back-projecting the measured 2D coordinates out through the camera model.
- The *3D measurement error* is the distance between correct and measured position in 3D space, obtained by calibrating two cameras or a camera and a projector and by determining the 3D coordinates of certain test points via stereo vision or the coded light approach. If the true position of the test points is unknown, it can be estimated by e.g. by fitting a suitable model to the data first.

In a first experiment, we apply Tsai’s technique to five images taken from the same viewpoint. Table 1 lists the resulting parameters and accuracy indicators. Clearly all parameters should remain stable; that they do have a sample standard deviation of between 0.2% and 0.7% of the sample mean indicates a reasonable precision.

In a next step, we apply Tsai’s technique to five images taken from distinct viewpoints. This time, only the internal parameters should remain stable; however, the results (table 2) show that they have a standard deviation between 1.2% and 2.6% of the sample mean value, indicating a serious stability and thus precision or accuracy problem. Repeating the camera calibration several times for one of the viewpoints does not significantly change the resulting parameters; e.g. for the view of attempt 4, the focal length is consistently about 12.7 mm over several calibration attempts using distinct images, while it remains in the area of 12.0 mm for the view of attempt 5. Necessarily, at least one of the two results for the focal length is systematically erroneous. So we conclude that the technique is inaccurate. This is despite the fact that in all but one case the target plane has been sufficiently tilted with respect to the image plane

	Attempt 1	Attempt 2	Attempt 3	Attempt 4	Attempt 5	NSDev [%]
Number of Views	5	4	5	6	5	
Image Plane Center $C_x, C_y$ [pix]	371.78 289.40	371.74 289.68	371.29 289.62	371.58 289.77	372.12 289.90	0.072 0.057
Focal Length [mm]	12.684	12.686	12.689	12.684	12.680	0.023
Radial Distortion $\kappa$ [ $1/\text{mm}^2$ ]	$4.865 \cdot 10^{-4}$	$4.877 \cdot 10^{-4}$	$4.810 \cdot 10^{-4}$	$4.827 \cdot 10^{-4}$	$4.806 \cdot 10^{-4}$	0.597
Translation $T_x, T_y, T_z$ [mm]	232.95 571.65 508.14	233.16 571.56 508.46	233.30 571.56 508.65	233.18 571.49 508.42	233.05 571.50 508.35	0.051 0.010 0.033
Rotation $R_x, R_y, R_z$ [deg]	-23.785 28.792 -12.132	-23.769 28.793 -12.132	-23.775 28.812 -12.136	-23.772 28.800 -12.131	-23.753 28.780 -12.124	0.044 0.036 0.032
Image Plane Error rms, max [pix]	0.056 0.266	0.059 0.286	0.060 0.276	0.058 0.260	0.059 0.269	
Object Space Error rms, max [mm]	0.034 0.158	0.036 0.158	0.036 0.153	0.036 0.145	0.036 0.161	
Angle bet. Planes [deg]	36.684	36.676	36.693	36.682	36.656	0.034

Table 3: Results with the proposed camera calibration method repeated for 5 distinct images from the same point of view with slight variation of e.g. the exposure time (same images as for table 1).

Finally, we determine the 3D measurement error. To that end, we move the calibration target to a position far away from the one it had during calibration. Next, we acquire a depth map of the target with a coded light system calibrated with Tsai’s method. We then perform a relative 3D measurement by LS-fitting a plane in 3D space to the obtained surface points. The mean orthogonal distance of the measured coordinates to the fitted plane is about 0.6 mm in 3D space. The spatial distribution of the distance values shows that the measurement differs systematically from the ground truth: whole areas in the corners of the image deviate by about -1mm from the plane while the central region of the image has an about constant distance of +0.6mm to the plane. With the improved calibration method described below, the error is reduced to a mean deviation of about 0.1 mm and a spatially uniform distribution of the distance values.

In sum, Tsai’s coplanar technique gives results that model the image formation very well considering points on or near the target plane. This is indicated by the very small image plane and object space error for the calibration points. However, its estimates of the fixed focal length vary from 12.0 to 12.7 mm over five attempts and remain systematically at 12.0, respectively at 12.7 mm when repeatedly calibrating from certain viewpoints. Also, a coded light system calibrated with Tsai’s technique produces non-negligible systematic ranging errors. That is, the technique is not accurate in the above sense. This insight comes as quite a surprise because the rms image plane error is small, which seems to point toward a high calibration precision and accuracy.

We repeat the above experiments with the new calibration method. For each image, 3 to 5 additional images from other unknown viewpoints are used to stabilize the calibration. The key results for repeating a calibration 5 times for the same point of view (table 3) are as follows: The parameters remain quite stable with a standard deviation between 0.01% and 0.07% of the mean value (excluding radial distortion). That is the factor 10 in precision compared to Tsai’s technique. Even more striking is that the actual results differ strongly between Tsai’s and our proposed extension. E.g. the mean focal length of the former technique is 12.412 mm, while our method yields a mean of 12.682 mm. This suggests at least one of the two methods is inaccurate.

All the above effects are even more pronounced for the distinct-points-of-view experiment (table 4): here the parameters remain again very stable with a standard deviation between 0.01% and 0.07% of the mean (excluding radial distortion), which represents an improvement of about the factor 50 compared to Tsai’s technique.

	Attempt 1 (1,2,3,4)	Attempt 2 (1,2,3,5)	Attempt 3 (1,3,4,5)	Attempt 4 (1,2,4,5)	Attempt 5 (2,3,4,5)	NSDev [%]
Number of Views	4	4	4	4	4	
Image Plane Center $C_x, C_y$ [pix]	371.84 289.96	371.42 289.90	371.34 290.01	371.83 289.96	371.34 289.31	0.063 0.090
Focal Length [mm]	12.680	12.683	12.683	12.680	12.685	0.015
Radial Distortion $\kappa$ [ $1/\text{mm}^2$ ]	$4.793 \cdot 10^{-4}$	$4.801 \cdot 10^{-4}$	$4.811 \cdot 10^{-4}$	$4.787 \cdot 10^{-4}$	$4.816 \cdot 10^{-4}$	0.225
Image Plane Error for the single view rms, max [pix]	0.063 0.330	0.044 0.198	0.050 0.224	0.075 0.380	0.093 0.401	
Angle bet. Planes [deg]	46.410	39.394	27.039	29.026	18.769	

Table 4: Internal camera parameters resulting from repeating the proposed multi-view extension of Tsai’s camera calibration for five distinct points of view (same images as for table 2).

	Focal Length	$C_x$	$C_y$	RMS over all 5 images
Zhang	832.50 pix ( $f_x$ )	303.96 pix	206.59 pix	0.335 pix
Proposed Technique	831.65 pix	303.14 pix	206.32 pix	0.176 pix

Table 5: Comparison between Zhang’s and our calibration technique, based on the data set made available by Zhang. Zhang’s results are taken from his paper [1998].

	Attempt 1 (1,2,3,4)	Attempt 2 (1,2,3,5)	Attempt 3 (1,3,4,5)	Attempt 4 (1,2,4,5)	Attempt 5 (2,3,4,5)	NSDev [%]
Number of Views	4	4	4	4	4	
Image Plane Center $C_x, C_y$ [pix]	376.31 289.57	378.07 290.11	375.71 290.27	378.07 290.11	373.81 290.25	0.424 0.087
Focal Length [mm]	12.690	12.687	12.685	12.692	12.700	0.040
Radial Distortion $\kappa$ [ $1/\text{mm}^2$ ]	$4.913 \cdot 10^{-4}$	$4.799 \cdot 10^{-4}$	$5.177 \cdot 10^{-4}$	$5.262 \cdot 10^{-4}$	$5.621 \cdot 10^{-4}$	5.588
Image Plane Error for the single view rms, max [pix]	0.093 0.469	0.084 0.384	0.082 0.341	0.132 0.608	0.109 0.494	

Table 6: Results with Zhang’s calibration technique given the data also used for tables 2 and 4.

We may for that reason safely conclude that our method is notably more precise. Also the 3D measurement error (evaluated in chapter 5) is much smaller and non-systematic with our approach. We hypothesize consequently that the results of our multi-view calibration are more accurate as well. Should that be the case, it is interesting to note that each of the then significantly erroneous parameter sets of tables 1 and 2 gives about the same rms image plane error than the corresponding about correct one of tables 3 and 4. This implies Tsai’s algorithm does not have a chance to find the latter as noise in combination with interdependency of parameters lets the wrong set appear as good or better according to its cost function. That is, the underlying problem is the data that can be extracted from a single point of view: it seems to be principally insufficient for accurate calibration.

In our opinion, this result casts doubt on the concept of accurate single-view calibration based on a planar target. Especially since the observed data is as noise-free as it can realistically get with reasonable effort and a low-cost camera: Neither the accuracy of the detected image coordinates of the fiducial marks nor that of their world coordinates leave much room for improvement. We conclude that the results given by Tsai’s coplanar method should not be considered highly accurate but for points close to the target plane.



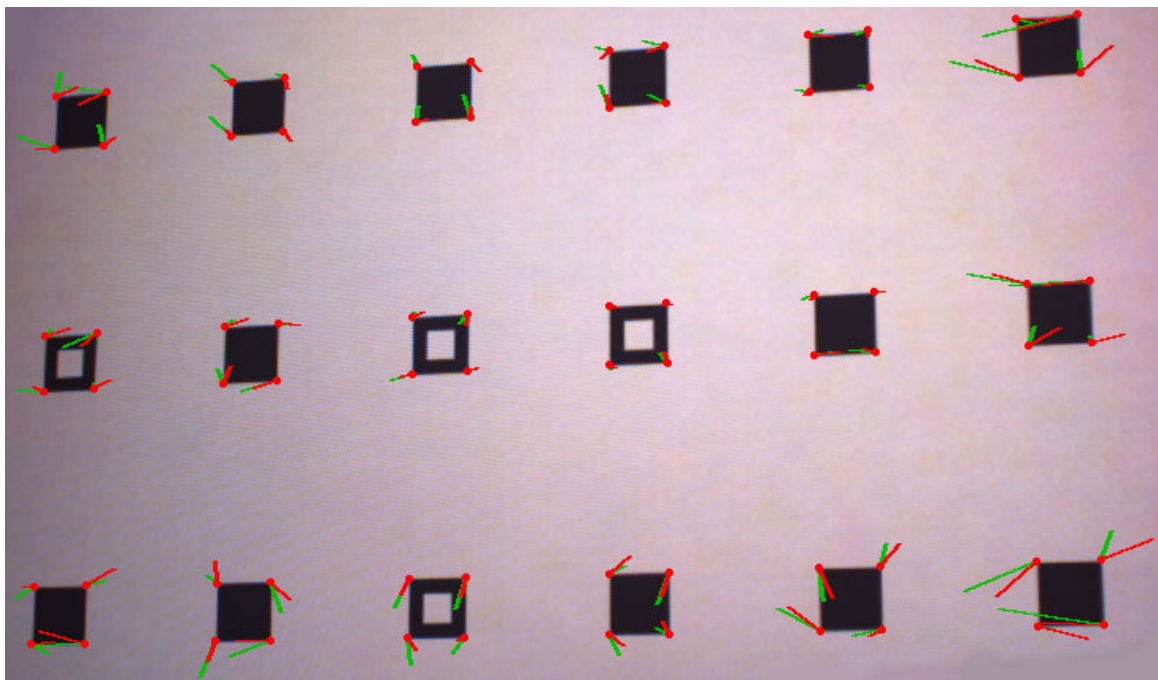


Figure 29: Visualization of the image plane error by drawing a line from the detected image position of a calibration mark into the direction of the one predicted by the model. The length of a line segment is proportional to the squared image plane distance between these two positions. The red lines are the errors of the proposed (table 4, attempt 2), the green lines of Zhang's method (table 6, attempt 2). The clipping shows the central part of the camera calibration target (rotated by  $90^\circ$ ).

We repeat the above experiment with a second realistic scenario, namely that the position of the calibration marks has not been determined up to  $\pm 0.01\text{mm}$  as in our case; so the printer and non-coplanarity problem would go unnoticed, and at least the former makes the world-coordinates systematically erroneous. The calibration results are not listed here in detail due to lack of space. As an exemplary result, the average focal length for the five views of table 2 is computed as 11.99 mm with Tsai's method, while it is 12.625 mm with the modified one. Under the assumption that the correct value is 12.683 mm, our above conclusion becomes the more pronounced, the less accurate and precise the available data is.

Next, we compare the proposed method with the state of the art in form of Zhang's approach. Zhang makes one set of observation vectors (image plane and world coordinates of the fiducial marks for five images and calibration results) publicly available. This set is rather unsuited for our approach because the target plane subtends angles of 9, 10, 11, 11 and 24 degrees with the camera's retinal plane, respectively, which is far away from the required 30 degrees inclination. Nevertheless table 5 shows that the proposed method yields – given the identical observation vectors as input – almost the same values for the internal parameters as Zhang's. It improves the rms image plane error to 0.176 pixel, i.e. by a factor of about 2 compared to the result published by the latter. This seems to indicate that it represents an improvement over Zhang's method.

This judgment is confirmed by a second experiment; in its case the observation vectors as used for table 4 are fed into Zhang's ICV implementation, i.e. the two algorithms again operate on the same data. The corresponding calibration results are listed in table 6. They indicate that the proposed multi-view Tsai technique is more precise, that is to say its results are significantly more stable (as mentioned above, more experiments to substantiate this claim are described in Forster [2005]): The improvement factor ranges from 1 to 20, depending on the parameter considered. The root-mean-square image plane error over the five calibration attempts is 0.10 pixel with Zhang's, 0.065 pixel with Tsai's modified technique, also indicating a better overall accuracy of the latter.

	Attempt 1	Attempt 2	Attempt 3	Attempt 4	Attempt 5	Attempt 6
Number of Views	1 (Tsai)	1 (Tsai)	1 (Tsai)	4	3	4
Image Plane Center $C_x, C_y$ [pix]	518.04 761.08	461.28 785.35	491.47 758.85	516.41 764.43	509.60 756.73	509.52 757.46
Focal Length [mm]	39.754	40.720	40.538	39.779	39.789	39.870
Radial Distortion $\kappa$ [ $1/\text{mm}^2$ ]	$5.028 \cdot 10^{-5}$	$3.587 \cdot 10^{-5}$	$5.145 \cdot 10^{-5}$	$5.150 \cdot 10^{-5}$	$4.791 \cdot 10^{-5}$	$4.751 \cdot 10^{-5}$
Image Plane Error for the single view rms, max [pix]	0.132 0.621	0.259 1.602	0.134 0.616	0.133 0.625	0.269 1.801	0.132 0.620
Angle bet. Planes [deg]	28.055	46.783	29.338	28.104	45.429	28.867

Table 7: Exemplary projector calibration results (internal parameters only).

Figure 29 visualizes the image plane error associated with attempt 2 of tables 4 and 6 by drawing a line from the detected image position of a calibration mark into the direction of the one predicted by the model. The length of a line is proportional to the square of the image plane distance between these two points. The red lines visualize the image plane error after calibrating with the proposed, the green lines after calibrating with Zhang's technique. Both techniques yield the expected apparent random error pattern that is most likely largely due to the error of the square corner detection.

Finally, table 7 gives results for projector calibration obtained with Tsai's original approach (single point of view, left three entries) and the one of this work (3 to 4 points of view, right three entries). The results are consistent with the camera calibration results – again the parameters are much more stable with the modified version – and for that reason not discussed in detail. The only notable differences between camera and projector calibration results are the asymmetric position of the optical slide plane center and the stronger radial distortion that reaches up to 10 pixels in the projector's y coordinate. Also the results are less precise, respectively less accurate than in the case of camera calibration. We attribute this mostly to the relatively poor quality of the projected calibration marks (compared to the printed ones), caused by depth-of-field limitations and chromatic aberration, which leads to a larger error in the detected image coordinates.

All in all, the results of this section indicate that the calibration of the camera and, even if somewhat less so, of the projector are precise and accurate. So we expect the ranging error due to the calibration error to be negligible. Of course, this needs to be confirmed in the course of an evaluation via actual 3D measurement experiments; this is done in chapter 5.

Certain interesting aspects of camera and projector calibration are not covered in this chapter; e.g. simulated results would help understanding how robust the algorithms are against various types of noise. A comparison of real and simulated results obtained in the linear stages of the algorithms would highlight their respective qualities regarding the initial estimates and potentially allow deriving further improvements. Another interesting question is whether a more sophisticated distortion model would enhance the quality of the proposed method; during the experiments conducted so far, a model considering a second coefficient of radial distortion as well as tangential distortion increased the execution time of the algorithm significantly, yet the accuracy only slightly). As the lack of space does not permit addressing these non-central questions in this work, they are presented in Forster [2005].

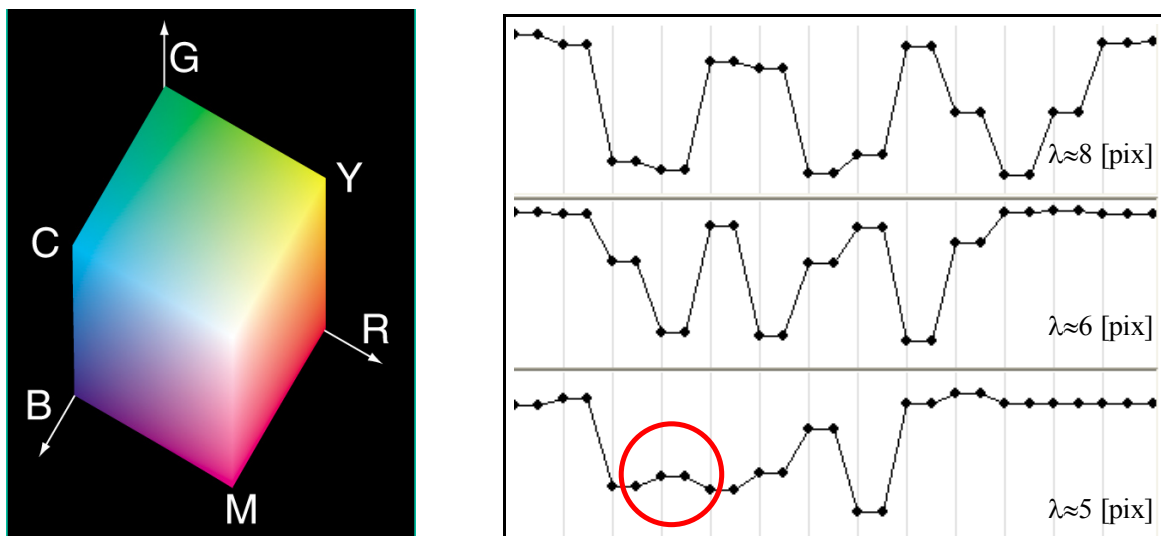


Figure 30: The eight corners of the RGB-cube (left). Red-signals along an image row resulting from imaging black-and-white line patterns of wavelength 8, 6 and 5 pixels, respectively (right).

### 4.3.2 The Choice of Colors

A key decision with a CL system is the choice of pattern colors, respectively gray levels. It is closely entwined with other core aspects such as the choice of encoding, the hardware used, the scene limitations imposed, etc. In our case, the proposed, noise-susceptible color edge approach dictates the selection of colors: to create edges of maximal contrast, only two signal levels per color channel can realistically be used. The pattern colors are accordingly the eight corners of the RGB-cube (figure 30), that is the colors black, red, green, blue, yellow, magenta, cyan, and white.

### 4.3.3 Encoding – The Projection Pattern

#### 4.3.3.1 Required Pattern Resolution

How many light planes should the pattern encode? Of course as many as possible, to achieve a maximal non-interpolated relative spatial resolution of the depth map. Then again, the camera has to be able to resolve the reflection of the projection pattern into its separate light stripes (for simplicity, we assume a stripe pattern in this section). For a comparable field of view, the resolution of cameras is typically lower than that of projectors: e.g. the granulation of a standard slide corresponds roughly to a lateral resolution of 4000 by 4000 pixels, while a CCIR camera has one of ca. 780 by 560 pixels. So for this and certain other reasons, the camera tends to be the limiting factor.

We obtain a rough estimate (the many unknown factors such as the interfering scene texture or the blurring introduced within the imaging chain prevent a more precise modeling) of the camera's stripe resolution by considering a pattern imaged as set of equidistant vertical white lines on a black background, i.e. one whose image changes along the rows only. The sampling theorem states that the pattern has to be sampled at least at twice its highest frequency. In practice, a higher factor of at least 2.8 is recommended [Albertz 1991]. The most common color pixel layout with single-chip color cameras, which is also used in this work, is the well-known Bayer pattern. In its case, at most every second pixel samples a given color component for a given row and component. This implies the stripes should be at least 6 pixels apart in the camera image if we want to at least get the base frequency of the projection pattern right (ignoring aliasing). That is, with the above resolution each channel resolves at most around 130, respectively 90 lines (depending on their orientation).

A simple experiment confirms that this coarse estimate is about right: While all lines are reconstructed fairly well when imaging line patterns with wavelengths of ca. 8 and 6 pixels, some lines go missing (red circle) in the channel image of a line pattern of wavelength 5 pixel (figure 30).

It would be theoretically possible to phase-shift the color channels relative to each other to achieve a higher total resolution over all three channels, but for reasons outlined below this is not practicable. So in our case the camera resolves in total about 130 (90) color lines. This implies the projection pattern should contain at most that many color lines (assuming camera and projector have the same field of view). Simple geometric considerations show that the imaged frequency of the projected pattern also depends on the surface orientation of the scene and can be higher than the one projected. For this reason, a frequency somewhat below the upper camera limit, in our case about 120 (80) lines, represents a good compromise. Considering we can equate a line, but also the interstice between two lines with a light plane, the pattern should then encode about  $2 \cdot 120 = 240$ , respectively  $2 \cdot 80 = 160$  light planes. Of course, this represents a minimum requirement; the encoding should in principle be suited for larger numbers of light planes for cameras of higher resolution.

#### 4.3.3.2 Towards Optimal Patterns for Spatially Encoded Light

This section discusses the properties a CL projection pattern (in this and the following sections the term CL exclusively refers to coded light based on spatial encoding) should ideally have besides the required resolution. To derive them, we look at CL systems from a digital communication, or, rather, coding system perspective. From this viewpoint the projector corresponds to a sender that channel-encodes a message and transmits it via the scene, the transmission channel, by modulating the illumination. The transmitted signal is composed of certain units, namely the pattern primitives. Between them and the code symbols exists (not necessarily, but in our case) a one-to-one correspondence. The camera represents a receiving unit that demodulates and channel-decodes the incoming message, the reflection of the projected pattern. The following aspects characterize a CL system as coding system:

- **High Error Probability:** Empirically, the likelihood that a transmitted code symbol  $p_1$  is interpreted by the receiver as a different symbol  $p_2 \neq p_1$ , i.e. that a *symbol error* occurs, is high, often in the area of 0.1 - 0.2 and greater. The corresponding probability distribution tends to be non-uniform: the probability of a given symbol being mistaken for a certain other depends strongly on the two symbols considered. This is obvious e.g. with colored symbols: clearly some color pairs (e.g. white and yellow) are more likely to be confused than others (e.g. yellow and blue) due to their respective distance in the signal space.
- **Bursty Transmission Channel:** If a surface patch (i.e. a part of the transmission channel) exhibits a low reflectivity for one or several color bands, the reflectivity smoothness assumption implies its neighbors are likely to do so as well. Consequently, errors often occur in bursts.
- **Two-Dimensional Transmission Channel:** The channel transmits a two-dimensional signal  $s(x, y)$  rather than the conventional one-dimensional signal  $s(t)$ .
- **Synchronization Problems:** With most data transmission systems, the issue of synchronization does not affect the (discrete channel) encoder and decoder; it is solved reliably at the modulation layer. For that reason, coding theory deals almost exclusively with symbol errors. With CL, it is unavoidable that some parts of the transmission are irreversibly lost. Equally inevitable are *ghost symbols*, i.e. code symbols that are received even though they have never been sent. No practical approach to modulation can solve this problem in all cases, e.g. if a part of the pattern is projected on a surface that is occluded from the camera's view. In sum, with CL *synchronization errors* (also called *clocking errors*) occur potentially frequently and have to be taken into account during (de)coding. In this context, it is important to note that the minimal distance of a code tells nothing about its robustness with respect to them: the two codewords  $c_1 = 0101$  and  $c_2 = 1010$  have a maximal Hamming distance; assume  $c_1$  is part of a longer message, e.g. ... 1111 0101 0111 ... and that its leading 0 is lost due to synchronization issues. Then the sequence 1111 1010 111 ... is received, containing  $c_2$  in place of the sent  $c_1$ . In short, already a single synchronization error can result in an undetectable decoding error even if the two codewords involved are maximally distant according to their Hamming distance.

- **The Information is in the Position Only:** With CL, the transmitted message is known a-priori; the spatial position of primitives in the received message is the sought-after information. Very importantly, this underscores again that synchronization is the key problem with CL: as long as it is possible to synchronize projected and received signal, that is to determine which received matches which transmitted primitive, in principle an arbitrary number of symbol errors can be corrected.
- **Compact Codes are Good Codes:** According to classical coding theory, the probability of an undetected symbol error can be made arbitrarily small by increasing the codeword length of a suitable code (see e.g. Shannon's Theorem in [van Lint 1982]). This applies in principle also to CL patterns; yet in their case also the likelihood of irresolvable synchronization problems, e.g. due to jump edges of the scene, increases with the word length due to the resulting larger sub-pattern size. A precise formulation of this interdependency requires knowledge of the scene, which is by definition unavailable with an all-purpose range sensor. Then again, we need some kind of guideline how to balance the above conflicting aspects. For this reason, we take on in the following a world model according to which objects tend to have a mostly continuous surface (along the lines of "matter is cohesive" as by Marr and Poggio [1976]). In this world, the likelihood of a depth jump over a given image/slide area is roughly proportional to the squared sum of its maximal x and maximal y extent (for a given finite interval that includes all areas of practical interest). The exact point at which this effect makes more compact, less redundant subpatterns better than longer ones with better error detection capabilities depends on the parameterization of the world model. In any case, the model implies that the subpatterns should be as compact as possible.
- **Image Processing/Computer Vision Requirements:** With CL systems, image processing, respectively computer vision aspects are crucial: The transmitted/projected primitives and their spatial adjacency relationships have to be recognized in the pattern image. With other words, the task of demodulation, i.e. of converting the received signal into words over the (channel) code alphabet, has to be solved by image processing/computer vision means. The encoding/modulation has to make this task possible in the first place by creating a pattern that contains suitable visible features and should support it as much as possible.
- **Incomplete Decoding Permissible:** Even though it is impossible to request a retransmission, incomplete decoding is preferable over erroneous decoding. With other words: rather no data than incorrect data. Of course, incomplete decoding should happen as rarely as possible.
- **Ample Processing Power:** In contrast to typical coding systems, en- and decoding efficiency is not a central issue with CL systems, nor do algorithms have to be implemented on minimal hardware resources. This is obvious with respect to encoding. With respect to decoding, a powerful processing unit is needed for the CL algorithm in any case and is available for decoding. In short, there is no tight limit on the complexity of en- and decoding techniques.

We conclude that a CL projection pattern should ideally cope with, respectively exploit all the above points. Clearly most of them are specific to the case of CL, and we have to find our own way of doing so. This is the topic of the following sections.

#### 4.3.3.3 A New Kind of Projection Pattern

This section introduces a new kind of projection pattern for spatial encoding. Its design is guided by the aspects discussed in the previous sections of this chapter. The first of those is that the resolution of the pattern should be about  $m = 160$  by  $n = 240$ , potentially greater. Furthermore, its code should have a large minimal distance to cope with the high symbol error probability and with the problem of burst errors. At the same time, its subpatterns should be compact and made up of only eight colors. Evidently these requirements can only be met with 1D encoding, considering that 2D encoding would require a code of at least about  $160 \cdot 240 = 38400$  distinct codewords.

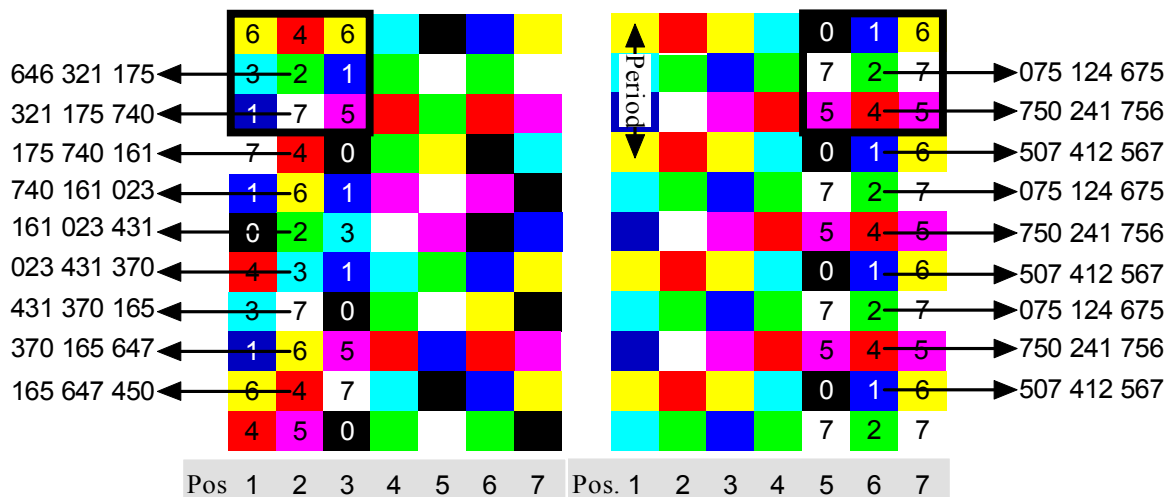


Figure 31: Example of a PRA (left) and the proposed pattern type (right), both with 3 by 3 windows as subpatterns. The array entries are in both cases uniform squares that occur in 8 distinct colors. The numbers within the squares are for illustration only, i.e. not part of the pattern.

Clearly the most difficult of the pattern design aspects is to ensure that pattern primitives, and consequently the code symbols, can be located reliably and accurately by image processing means. Respectively, that it is possible to detect if one or several symbols are missing or ghost symbols occur. As pointed out before, this task can only be solved optimally by taking into account modulation, image processing (demodulation) and coding aspects at the same time, even if doing so incurs a certain loss of generality. We consider for that reason in the following the intended type of (de)modulation during code design and vice versa. Doing so is by no means uncommon in coding theory; a corresponding approach is e.g. typically taken with orthogonal codes [Sweeney 1991].

As motivated in the previous sections, we propose employing edges as pattern primitives, more accurately edges of a RGB projection pattern that contains only the eight corners of the RGB cube as colors. With a conventional projector, we cannot directly specify an edge pattern, only indirectly via a color pattern that gives rise to the wanted edges. For this reason, we need to detail in the following the color pattern  $I_p$  along with the resulting edge pattern  $I_p'$ , even though we are only interested in the latter. Here as in the following symbols with the superscript ‘ should be associated with the edge pattern, those without a superscript with the conventional color pattern. Evidently, if an edge pattern is encoded, so is the color pattern that generates it, since identical color subpatterns give rise to identical edge subpatterns (but not vice versa). That implies that the intended approach automatically creates an encoded color pattern  $I_p$  with code  $C$  as well as the sought-after encoded edge pattern  $I_p'$  with its code  $C'$ . Both are contained in the same physical transmission, but composed of dissimilar pattern primitives and, on the coding layer, of distinctly sized alphabets.

We first discuss the color pattern. Its primitives are eight uniform, distinctly colored squares, implying  $q_p = 8$ . For reasons explained above, the eight different colors are black, red, green, blue, yellow, magenta, cyan, and white. Its subpatterns are windows of  $v$  by  $w$  primitives, where a certain minimal size ( $v \cdot w \geq 4$ ) and layout ( $w \geq v \geq 1$ ) is assumed. We further require  $w$  to be odd if  $v = 1$ . An extension to other, non-rectangular subpattern geometries such as four-neighborhoods is straightforward, but not discussed here for the sake of simplicity. We form a codeword  $c = q_1 q_2 \dots q_s$  ( $q_i \in Q_p$ ) of length  $s = v \cdot w$  from a  $v$  by  $w$  window by reading its symbols from top to bottom, then from left to right, or, with other words, by concatenating its transposed columns from left to right. Formally, this is expressed via a codeword function that reads a local  $v$  by  $w$  window in this order, i.e. by defining  $\sigma(k) = (-w/2 + (k - 1)/v, -v/2 + (k - 1) \bmod v, 1)$ , where  $1 \leq k \leq s = v \cdot w$ . The slide margin has consequently the size  $w/2$ , i.e.  $d = w/2$ .









Pattern Primitive		Code Symbol	Shorthand Symbol
Black Square		$(0\ 0\ 0)^T$	0
Blue Square		$(0\ 0\ 1)^T$	1
Green Square		$(0\ 1\ 0)^T$	2
Cyan Square		$(0\ 1\ 1)^T$	3
Red Square		$(1\ 0\ 0)^T$	4
Magenta Square		$(1\ 0\ 1)^T$	5
Yellow Square		$(1\ 1\ 0)^T$	6
White Square		$(1\ 1\ 1)^T$	7

Table 8: Mapping of pattern primitives to code symbols.

With its  $v$  by  $w$  windows, the pattern resembles a PRA as described in the literature; there are, however, several key differences, an obvious one being that it represents – other than a PRA – a 1D encoding. As its windows need to be unique with respect to the horizontal slide position only, we may periodically repeat the first  $v$  pattern rows in the vertical direction. As a result, exactly  $v$  codewords are associated with each central slide  $x$ -coordinate  $i_p$ . Since the vertical row period is  $v$ , it follows that if  $q_1 \dots q_s$  is a codeword,  $q_2 \dots q_v q_1 q_{v+2} \dots q_{v+1} q_{2v+2} \dots q_{s-1}$  is the codeword of its lower neighbor, and so on up to the codeword of the  $(v - 1)$ th lower neighbor (equivalent to the first upper neighbor)  $q_v q_1 \dots q_{v-1} q_{2v} q_{v+1} \dots q_{2v-1} q_{3v} \dots q_{s-v+1}$ . A second consequence is that the last  $s - v$  symbols of a given codeword are the first  $s - v$  symbols of its right neighbor. In this context, we call a code word  $c$  the right (left, upper ...) neighbor of a given codeword  $\hat{c}$  if  $\hat{c}$  is the codeword of the slide position  $(i_p, j_p)$  and  $c$  the one of  $(i_p + 1, j_p)$  (respectively  $(i_p - 1, j_p)$ ,  $(i_p, j_p - 1) \dots$ ), where vertical positions are read modulo  $v$ . That is, if e.g.  $v = 2$ , the upper neighbor is the lower one as well.

All in all, the proposed approach trims down the number of required color pattern codewords from about  $m$  by  $n$  to  $v$  by  $n$  while still making use of both available degrees of freedom for encoding. As typically  $m \approx n$ , but  $v \approx 1$ , this represents a reduction to the square root of the number necessary with a PRA. Figure 31 illustrates and compares both approaches for a pattern with a window size of  $v = w = 3$ . In the figure, the eight distinct primitives are formally represented by the numbers 0 to 7, i.e.  $Q_p = \{0, 1, 2, 3, 4, 5, 6, 7\}$ . The left side shows a corresponding PRA as described in the literature. Its 3 by 3 windows each identify a position within the array, e.g. 631 427 615 is the unique signature of the top left window, respectively the slide coordinates  $(2, 2)$ . The right side displays the equivalent version of the new pattern type. In its case,  $v = 3$  distinct words, which are repeated with a period of  $v = 3$ , encode each horizontal position, e.g. the three words 075-124-675, 750-241-756 and 507-412-567 all encode the horizontal slide position  $i_p = 6$ .

As mentioned above, the color pattern determines the resulting color edge pattern. We now discuss the relationship between color and edge pattern in detail. We employ a common code alphabet  $Q_p$  to formally describe the two types of patterns and their associated codes. This alphabet corresponds to the set of elements of the vector space  $GF(3)^3$ , where  $GF(3)$  is the finite Galois field with the three elements  $\{0, 1, -1 (= 2)\}$ . The non-surjective mapping of the color pattern primitives to code symbols is defined in table 8; the table also specifies a shorthand symbol for each primitive, which is e.g. used in figure 31.

With the obvious rules for addition and multiplication,  $\text{GF}(3)^3$  becomes a three-dimensional vector space over the scalar field  $\text{GF}(3)$ . That is, adding and subtracting code symbols is a well-defined operation, e.g. the symbols  $(1 \ 0 \ 1)^T$  and  $(-1 \ -1 \ 1)^T$  add up to the symbol  $(0 \ -1 \ 0)^T$ . We further introduce the subsequent definitions:

- We call a vector of the  $\text{GF}(3)^3$  *positive* if it does not have a component with the value -1.
- We call the number of nonzero elements of a vector, respectively code symbol, its *weight*.
- We assign each codeword  $c$  of the color pattern a function  $f_c: \{1, \dots, w\} \times \{1, \dots, v\} \rightarrow \text{GF}(3)^3$  that maps the window positions on the symbol displayed at this position, i.e. on the field  $\text{GF}(3)^3$ . As with any vector-valued function, its component functions  $f_{c1}$ ,  $f_{c2}$  and  $f_{c3}$  are defined via  $f_c(x, y) = (f_{c1}(x, y), f_{c2}(x, y), f_{c3}(x, y))$ .
- In the general case ( $v > 1$ ), we define the derivative of a function  $f_c$  of the above type as the function  $f'_c: \{1, \dots, 2w-1\} \times \{1, \dots, v\} \rightarrow \text{GF}(3)^3 \cup \{e\}$  where

$$f'_c(x, y) = \begin{cases} \frac{\partial f_c(n, y)}{\partial x} & \text{if } x = 2n \\ \frac{\partial f_c(n, y)}{\partial y} & \text{if } x = 2n - 1 \wedge y < v \\ e & \text{if } x = 2n - 1 \wedge y = v \end{cases} \quad \text{for } n \in N_0 \quad (65)$$

and where the directional x-derivative of a projection-pattern related function  $g(x)$  is given by

$$\frac{\partial g(x, y)}{\partial x} = \frac{g(x+1, y) - g(x, y)}{x+1-x} = g(x+1, y) - g(x, y) \quad (66)$$

and its directional y-derivative by:

$$\frac{\partial g(x, y)}{\partial y} = \frac{g(x, y+1) - g(x, y)}{y+1-y} = g(x, y+1) - g(x, y) \quad (67)$$

In the case of a stripe pattern ( $v = 1$ ), we define the derivative of a function  $f_c$  of the above type as the function  $f'_c: \{1, \dots, w-1\} \rightarrow \text{GF}(3)^3$  where  $f'_c(x) = f(x+1) - f(x)$ .

- The preceding two constructions map each codeword  $c$  via the function  $f_c$  on a regular  $v$  by  $2w - 1$  window specified by the function  $f'_c$ . We again form a word  $c'$  from this window by concatenating its transposed columns from left to right, this time omitting window elements with the value  $e$ . All in all, this uniquely defines the *derived codeword*  $c' \in \mathbb{Q}_p^{2vw-v-w}$  of a given codeword  $c$  and consequently the derived code  $C'$  of a color pattern  $I_p$  with code  $C$ .
- We associate two derived symbols with a central slide position  $(i_p, j_p)$ , namely the value of the directional x- and of the directional y-derivative (assuming  $v > 1$ ) of the slide function  $I_p$  at  $(i_p, j_p)$ . We call the former (*symbol of the vertical edge segment* at  $(i_p, j_p)$  or edge segment from  $(i_p, j_p)$  to  $(i_p + 1, j_p)$ ), the latter (*symbol of the horizontal edge segment* at  $(i_p, j_p)$  or edge segment from  $(i_p, j_p)$  to  $(i_p, j_p + 1)$ ).
- If a component of the directional x-derivative of the slide function  $I_p$  has the value 1 at  $(i_p, j_p)$ , it follows that the corresponding component of the color pattern has a value of 0 at  $(i_p, j_p)$  and of 1 at  $(i_p + 1, j_p)$ . Similarly, a value of -1 allows concluding a component value of 1 at  $(i_p, j_p)$  and of 0 at  $(i_p + 1, j_p)$ . This is an immediate consequence of the definition of the code symbols in conjunction with the definition of the directional x-derivative. Analogous inferences can be made given the value of the directional y-derivative.



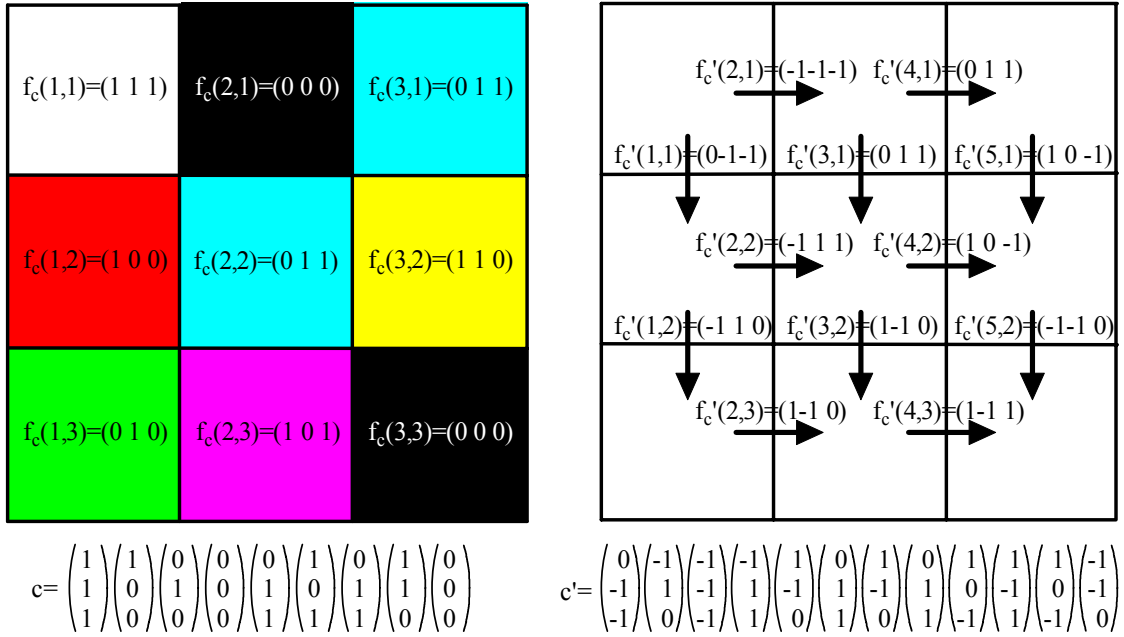


Figure 32: Relationship between color code and derived code for the parameters  $v = w = 3$ . The 9-symbol color codeword read from a given subpattern, i.e. a 3 by 3 window, is displayed to the left; the corresponding derived codeword of 12 symbols is shown to the right.

We have thus established the relationship between an encoded color pattern and its edge pattern, an example of which is visualized in figure 32. It is important to note the following aspects of this relationship:

- The mapping of codewords to derived words is not injective: different codewords can have the same derived codeword (short: derivative). More precisely, two codewords have the same derivative if and only if they differ by a constant, i.e. if one of them can be created by adding a certain symbol  $q \in Q_p$  to each of the other's symbols. This implies that a codeword can be determined given its derived codeword up to a constant. It also means that the edge pattern of an encoded color pattern is not necessarily encoded itself.
- In the general case ( $v > 1$ ), the last  $2vw - 3v - w + 1$  symbols of a given derived codeword are the first  $2vw - 3v - w + 1$  symbols of its right neighbor. With a stripe pattern, the last  $w - 2$  symbols of a derived word are the first  $w - 2$  symbols of its right neighbor.
- In the general case ( $v > 1$ ), a derived codeword has more symbols than its codeword, namely  $2vw - v - w$  versus  $vw$ . This is a key aspect as it allows creating derived codes that have about the same physical subpattern size, yet a notably larger minimal distance than the code itself.

With the derived pattern  $I_p'$ , neighborhood-relationships are more complicated than in the case of the color pattern  $I_p$  because traditional concepts such as four or eight neighborhoods do not apply. We introduce for that reason the following definitions: For a vertical edge at  $(i_p, j_p)$ , we call the vertical edges at  $(i_p, j_p - 1)$  and at  $(i_p, j_p + 1)$  its *non-consecutive neighbors*; analogously, the non-consecutive neighbors of a horizontal edge at  $(i_p, j_p)$  are the horizontal edges at  $(i_p - 1, j_p)$  and at  $(i_p + 1, j_p)$ . Moreover, we call two edge segments *consecutive (neighbors)* if one of them is an edge from  $(i_{p1}, j_{p1})$  to  $(i_{p2}, j_{p2})$  and the other one is located at  $(i_{p2}, j_{p2})$ . All in all, an edge segment has consequently six neighbors, two non-consecutive and four consecutive ones. Figure 33 illustrates these definitions for the vertical edge segment at  $(i_p, j_p)$ .

In the case of a stripe pattern, the neighborhood relationship is simple: there are only vertical edges, which have two (consecutive) neighbors, a left and a right one but for the left- and rightmost edge.

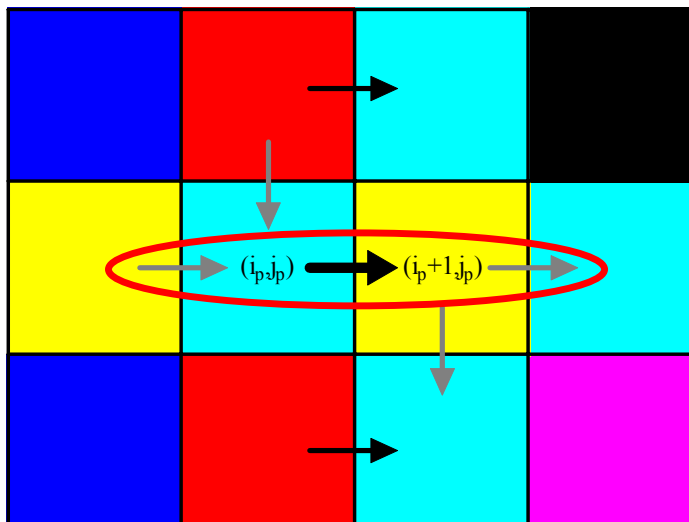


Figure 33: Neighborhood of a vertical edge segment located at  $(i_p, j_p)$ , marked with the bold arrow: Its two non-consecutive neighbors are marked with a black arrow, its four consecutive ones with a gray arrow. In the case of a stripe pattern, the situation is as within the encircled area: there are only vertical edges, which have two (consecutive) neighbors, a left and a right one.

Finally, we need to discuss the issue of *intrinsic edges*, i.e. edges that are not projected, but part of the scene. They are more problematic than noise because they have the same appearance as pattern edges and can consequently alter local subpatterns systematically. So the en- and decoding has to take special care not to be confused by them. Precisely which effect can an intrinsic edge have on a subpattern? Clearly it can overwrite projected edge segments (and cause symbol errors) and/or add new edge segments (and cause synchronization errors). We can neglect intrinsic edges resulting in local edge segment patterns that deviate drastically from the expected roughly quadrilateral-based one and that can be detected for this reason. The vast majority of the remaining intrinsic edges are of one of the following two types visualized in figure 34:

- **Type I:** An intrinsic edge that divides a color primitive into two roughly quadrilateral parts and thus splits a projected edge segment into two neighboring segments of the same symbol.
- **Type II:** An intrinsic edge that overwrites at least two neighboring edge segments with a certain symbol  $q \in Q_p$ .

Using the definitions introduced so far, we propose a projection pattern  $I_p/I_p'$  that meets the following requirements R1-R5:

- (R1) The color pattern  $I_p$  is 1D encoded over the **positive** vectors of  $GF(3)^3$  such that each  $v$  by  $w$  window, respectively the resulting codeword, uniquely identifies a **horizontal** position (i.e. an  $i_p$  coordinate) within the slide.
- (R2) Each derived codeword uniquely determines a **horizontal** position within the slide.
- (R3) Each symbol of a derived codeword has a weight greater than 1.
- (R4) Two codewords of the color code that refer to different horizontal positions within the array have at least a Hamming distance  $h > 1$ .
- (R5) Two derived codewords that refer to different horizontal positions within the array have at least a distance  $h'$  of
 

$(w - 1)/2 + 1$	for $v = 1$
$w$	for $v > 1$

The above definition is redundant for the sake of readability: E.g. R5 implies R2 and part of R1.

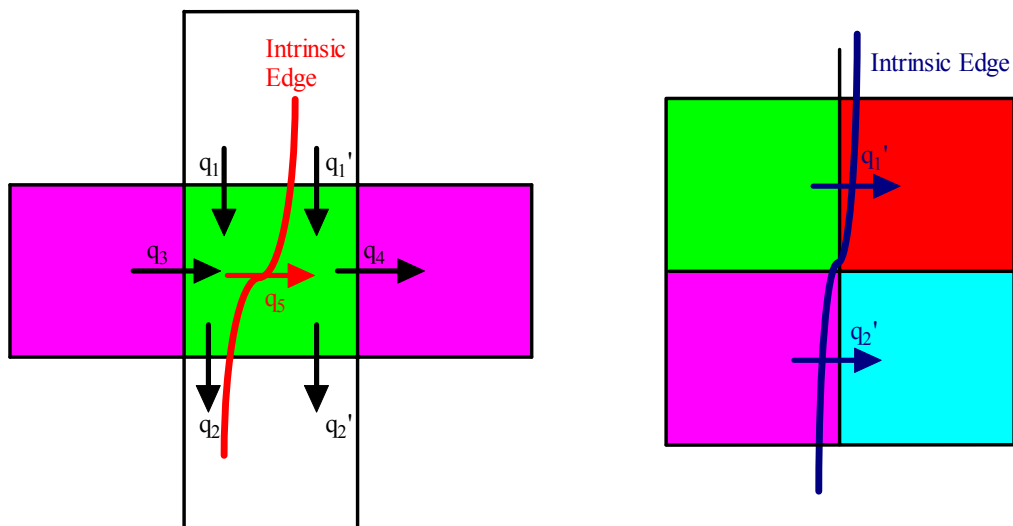


Figure 34: Two common types of intrinsic edges: Type I, displayed to the left, divides a color pattern primitive (in the figure the central green square) into two roughly quadrilateral parts such that two edge segments of the same symbol result ( $q_1 = q_1'$  or  $q_2 = q_2'$ ). Type II (right) overwrites at least two neighboring edge segments with a certain symbol  $q \in Q_p$ , i.e. such that  $q_1' = q_2' = q$ .

We now show that the **derived** code of a pattern that complies with the above requirements has the following error detection capabilities (which, unless they contain the term *always*, implicitly assume the described error is the only one affecting the codeword):

- (P1) In the general case ( $v > 1$ ), at least  $w - 1$  symbol errors are detected per derived codeword, with a stripe pattern at least  $(w - 1)/2$ .
- (P2) A single missing or a single ghost symbol per derived codeword is detected.
- (P3) Ghost symbols of weight 1 are always detected, irregardless of their number. So are symbol errors resulting in received symbols of weight 1.
- (P4) Errors due to edges of type I (provided the whole subpattern is visible in the pattern image) and type II are detected.
- (P5) Be  $q_1$  and  $q_2$  the symbols of two edge segments that are consecutive within the encoded pattern. There is at most one ghost symbol  $x$  such that both  $q_1, x$  and  $x, q_2$  are consecutive that is not detected under all circumstances. The symbols  $q_1, q_2$  and  $x$  each have weight 2.
- (P6) Even if all four edge segments forming a square are misidentified, no undetected decoding error results.

The motivation for the above capabilities is obvious but for P5 and P6. P5 concerns a problem that particularly affects stripe edge patterns. Ghost symbols necessarily occur between two stripe borders as on the left side of figure 34 where the derived symbol  $q_5$  appears between  $q_3$  and  $q_4$ . Of course, P2 guarantees that the resulting synchronization error can be detected, if it is the only one. But already a single further error might cause an undetectable error. P5 states that this is very unlikely to happen. First of all,  $q_3, q_4$  and  $q_5$  all need to have weight of 2; even if they all do,  $q_5$  needs to happen to be the one symbol out of the 12 of weight 2 that fits in-between  $q_3$  and  $q_4$ .

The motivation for P6 becomes obvious when considering two 3 by 3 windows of the color code that differ in their central square only. Then their derived codewords have a Hamming distance of 4, so in principle an undetectable error might result if all edges of the central square are misclassified. Clearly, it is advantageous to avoid such constellations and to spatially distribute the redundancy over all squares of a subpattern. P6 states that this is guaranteed with the proposed code.

The subsequent remarks and insights are useful for proving the properties P1 to P6.

Be  $q_1 = (a_1 \ b_1 \ c_1)^T$  and  $q_2 = (a_2 \ b_2 \ c_2)^T$  symbols of consecutive edge segments. First, assume both symbols have a value of 1 in a certain component, say the first:  $a_1 = a_2 = 1$ .  $q_1$  is the symbol of the edge from  $(i_{p1}, j_{p1})$  to  $(i_{p2}, j_{p2})$ . Be  $(r_1 \ g_1 \ b_1)^T$  the color symbol at  $(i_{p1}, j_{p1})$ ,  $(r_2 \ g_2 \ b_2)^T$  the color symbol at  $(i_{p2}, j_{p2})$ . It follows from  $r_2 - r_1 = 1$  that  $r_2 = 1$  and that  $r_1 = 0$ .  $q_2$  is by definition the symbol of the edge from  $(i_{p2}, j_{p2})$  to  $(i_{p3}, j_{p3})$ . Be  $(r_3 \ g_3 \ b_3)^T$  the color symbol at  $(i_{p3}, j_{p3})$ . It follows from  $r_3 - r_2 = 1$  that  $r_3 = 1$  and that  $r_2 = 0$ , which conflicts with the previous result stating that  $r_2 = 1$ . This proves that the case  $a_1 = a_2 = 1$  is not possible. The analogous reasoning shows the case  $a_1 = a_2 = -1$  is equally impossible. Of course, the proof applies to all three components.

Each symbol of a derived codeword has a weight of 2 or 3. With other words, at most a single component of one of its symbols is zero. Then, for any two such symbols, there is always a component that is nonzero with both of them. Consequently, for two consecutive edge segments of symbols  $q_1$  and  $q_2$ , there is always a component that has the value +1 with  $q_1$  and -1 with  $q_2$ , respectively -1 with  $q_1$  and +1 with  $q_2$  (the previous section rules out the remaining cases +1 and +1, respective -1 and -1). This proves that symbols of consecutive edge segments are never equal.

Be  $q_1 = (a_1 \ b_1 \ c_1)^T$  and  $q_2 = (a_2 \ b_2 \ c_2)^T$  symbols of non-consecutive neighbors, where  $q_1$  is the one of the vertical edge at  $(i_p, j_p)$  and  $q_2$  the one of its upper neighbor, i.e. the vertical edge at  $(i_p, j_p - 1)$ . Assume  $q_1 = q_2$ . A symbol of a derived codeword has at least two nonzero components; we assume without loss of generality  $a_1 = a_2 \neq 0$  and  $b_1 = b_2 \neq 0$ . This uniquely determines the corresponding components of the color-code symbols at  $(i_p, j_p)$  and at  $(i_p, j_p - 1)$  and implies they are equal in those two components. Consequently, the directional y-derivative at  $(i_p, j_p - 1)$  has at most weight 1, which conflicts with the pattern definition. This proves that the symbols of two non-consecutive, neighboring vertical edges are never equal as well. By rotating the pattern by  $90^\circ$ , we can reuse the proof for the remaining case of corresponding horizontal edges.

All in all, we have proven that the symbols of neighboring edge segments are never equal. In the one-dimensional case ( $v = 1$ ), this can be expressed as  $I_p'(i_p) \neq I_p'(i_p + 1)$ . This inequality does not only apply to the derived code, but also to the color code: two adjacent color symbols that are identical produce the derived symbol  $(0 \ 0 \ 0)^T$ , which cannot be part of a valid derived codeword. These results are helpful for the following proofs of the error detection properties P1-P6:

- **Proof P1:** This is an immediate consequence of the minimal distance of the code.
- **Proof P2:** In the one-dimensional case, a missing symbol  $q_j$  of a derived codeword  $c$  occurs either in the left or right half of a word ( $w-1$  being even, these two halves are well-defined). In the former case, the symbols  $q_j q_{j+1} \dots q_{w-1}$  are the beginning of  $\hat{c}$ , the  $j$ th right neighbor of  $c$ .  $q_{j-1}$  and  $q_j$  are the symbols of neighbors and consequently different. So  $q_{j-1} q_{j+1} \dots q_{w-1}$  cannot be part of a derived codeword, as then one of its right neighbors would have to have the symbol sequence  $q_{j+1} \dots q_{w-1}$  in common with  $\hat{c}$ , i.e. a Hamming distance of less than  $(w - 1)/2 + 1$ . This proves that the error is detected. If  $j$  occurs in the right half, it shifts the positions of less than  $(w - 1)/2$  symbols and consequently causes at most  $(w - 1)/2$  symbol errors, which are detected according to P1. Exactly the same reasoning applies to the case of a ghost symbol; this would not be the case, however, if  $w-1$  was odd, because then a ghost symbol occurring before the central symbol could potentially go unnoticed. For example, if  $w = 4$  and if the ghost symbol  $x$  appears within the derived codeword  $abc$ , the resulting codeword  $axb$  is not guaranteed to be detected. In the two-dimensional case, a missing symbol or ghost symbol can cause maximally  $v$ , respectively  $w$  symbol errors as in both cases at most a single row or a single column of symbols is shifted. The synchronization error is consequently detected according to P1.
- **Proof P3:** By definition, symbols of derived codewords have a weight of at least 2.
- **Proof P4:** Edges of type I result in non-consecutive neighbors that are equal. Edges of type II give rise to consecutive neighbors that are equal. Consequently, both types of intrinsic edges result in an error that can be detected.

- **Proof P5:** Be  $q_1$  and  $q_2$  symbols of consecutive edges. Be  $x$  the symbol of a ghost edge occurring between them such that all three edges are consecutive and that the synchronization error is not guaranteed to be detected. In each component where both  $q_1$  and  $q_2$  are nonzero, the ghost symbol needs to have a value of 0, because consecutive symbols can never have the same value in a certain component. We proved above there is at least one such component; clearly there may not be more for the ghost symbol to achieve the minimum weight 2. Consequently, all three symbols have a weight of 2. In one of the two remaining components,  $q_1$  has a value of zero and  $q_2$  of a  $\neq 0$ . The only permissible value for the corresponding component of the ghost symbol is  $-a$ . In the other remaining component, it is just the other way round. So  $q_1$  and  $q_2$  uniquely determine  $x$ , which proves there is only one possible type of problematic ghost symbol.
- **Proof P6:** This is guaranteed by R4 – two derived codewords that differ only in the four edge segments part of a single color square belong to color code codewords that have a Hamming distance of only 1.

The proof of P2 shows that in the case of  $w$  being even a ghost symbol occurring before the central symbol of the codeword could theoretically cause an undetectable error. In combination with P5, this case is rather negligible in practice. For this reason, we consider in the following also choices where  $w$  is even, e.g. the case  $w = 4$  and  $h' = 2$ .

We conclude: this section introduced a type of coded light pattern whose design is guided by the aspects of the previous sections, most notably the ones of section 4.3.3.2. It differs consequently strongly from existing work. With the proposed pattern type, image processing requirements receive the highest priority; they lead to the key decisions such as the choice of edges as pattern primitives. Nevertheless the remaining aspects are considered as well: the pattern type permits combining a high relative lateral resolution of the pattern with compact subpatterns. This is in part due to the fact that it is able to exploit both dimensions of the transmission channel for encoding. In reaction to the high error probabilities, it offers the capability to detect a large number of symbol errors as they occur e.g. with burst errors. Its design addresses in particular the importance of synchronization, respectively the problem of synchronization errors. This includes the problem of systematic synchronization errors due to intrinsic edges.

This section considers exclusively the topic of error detection; to be more precise, the subject of error detection in the most difficult case, namely the one where merely a single subpattern is reflected back into the pattern image. Of course, in most cases a larger coherent pattern clipping will be visible in the pattern image – each individual edge segment or symbol is then part of up to  $v \cdot w$  distinct subpatterns, respectively codewords. Clearly this opens up numerous additional error detection and error correction possibilities. How they can be exploited is discussed in the subsequent sections, primarily in section 4.3.4.1.

To describe and develop the pattern, we made extensive use of the color pattern  $I_p$  and its code; doing so simplifies the formal description, respectively guarantees that there is a physically viable color pattern behind the sought-after color edge pattern. For the rest of this chapter, i.e. for the purpose of demodulation and decoding, we exclusively work with the derived code  $C'$ . As pointed out before, this is because the pattern primitives and thus the symbols of the color code  $C$  cannot be directly distinguished from the scene reflectivity. Only the primitives of the color edge pattern can be retrieved to some extent by computing the derivative of the pattern image, irregardless of the scene color. Consequently, in the following terms such as codeword or code symbol always refer to the derived code.

Of course, simply defining a code as the one of this chapter does no good unless we are able to generate instances of it that have a sufficient number of words. How this is accomplished is discussed in the next section.

#### 4.3.3.4 Pattern Generation

The construction of good codes is a central topic of coding theory and related areas of science such as cryptography. For certain simple encoded projection patterns, coding-theoretic results prove to be very useful: E.g. De Bruijn sequences can be generated by feedback shift registers for all choices of parameters that are practically relevant for CL [Golomb 1967]. Removing a single zero from the all-zero word of a De Bruijn sequence yields a pseudo-random sequence, i.e. well-known techniques to create pseudo-random sequences exist as well. There are straightforward methods to construct binary perfect maps for parameter sets subject to certain simple necessary conditions, see e.g. [Paterson 1994]. Even for certain types of perfect maps based on non-binary alphabets such generation methods exist: for instance, Griffin et al. [1992] describe a method to construct non-error-detecting 2D-encoded pattern of the dimensions  $q_p^3$  by  $q_p^2$  with unique four-neighborhoods. Their pattern is consequently optimal, as each of the  $q_p^5$  possible codewords of length 5 over the  $q_p$  symbols occurs exactly once, i.e. it represents a perfect map (rather something equivalent given each possible four-neighborhood rather than rectangular  $v$  by  $w$  window occurs exactly once).

Yet in the general case, i.e. with complex non-binary and error detecting coded light patterns, the task is much more difficult, and there seems to be no straightforward way to apply advanced results from coding theory or related fields. This applies in particular to any attempt to deterministically construct a code meeting the complex requirements outlined in the previous section. For this reason, we employ a non-deterministic method, i.e. a pseudo-random algorithm, to find instances of the proposed code/projection pattern. In principle, the following simple algorithm solves this task:

```

Set largest pattern found  $\mathbf{I}_{pmax}$  and maximal pattern width both to 0
Beginning of outer loop:
Randomly generate  $\mathbf{I}_p = (q_{ij})$  of size  $v$  by  $w$  (trivial as only R3 applies)
  Beginning of inner loop:
  Exit if termination criterion is met (e.g. time limit)
  Randomly generate  $v$  symbols  $\mathbf{q}_{new} = q_1 \dots q_v$  until  $\mathbf{I}_p \mathbf{q}_{new}^T$  meets R1-R5
  If such a column vector can be found within a fixed number of tries
    Set current pattern to  $\mathbf{I}_p \mathbf{q}_{new}^T$  ( $\mathbf{I}_p := \mathbf{I}_p \mathbf{q}_{new}^T$ )
    If current pattern width > largest width found so far
      Set largest width found so far to current pattern width
      Set largest pattern  $\mathbf{I}_{pmax}$  to  $\mathbf{I}_p$  ( $\mathbf{I}_{pmax} := \mathbf{I}_p$ )
    Go to beginning of inner loop
  If no such symbol vector can be found, go to beginning of outer loop
Return  $\mathbf{I}_{pmax}$ , the largest pattern found

```

An implementation of the above straightforward approach yields good results. The computational most costly step of the pseudo-random algorithm is checking whether adding a new word preserves the required minimal distance of both code and derived code. Doing so by computing the distance of the new word to each existing word requires about  $2v \cdot w \cdot n_c$  comparisons, where  $n_c$  is the current number of codewords. For larger codes, this step is computationally expensive and slows down the algorithm considerably.

The following two corollaries allow implementing the above step and thus the whole algorithm more efficiently. They use the concept of a *subword* of a codeword. Such a subword is formed by striking out letters from a given codeword: e.g. for the codeword 123, its subwords of length 2 are 12, 13 and 23. Two subwords are of the same type if they are created from codewords by removing letters from identical positions. Clearly subwords of the same type have the same length, too.

E.g. for the two codewords 123 and 456, the subwords 12 and 45 are of the same type, but 1 and 12 or 13 and 56 are of a different type. Consequently, the number of distinct types of subwords of length  $m$  of words of length  $s$  corresponds to the number of  $m$ -combinations of an  $s$ -set, i.e. the binominal coefficient of  $s$  and  $m$ , denoted in the following by  $bc(s, m)$ .

As a result, we may re-formulate the concept of the minimal distance of a code as follows:

**Corollary 1:** A (block) code  $C$  of word length  $s$  has the minimal distance  $h$  if and only if each subword of length  $s - h + 1$  occurs at most once among all same-type subwords of the codewords.

Proof: First, assume two subwords of length  $s - h + 1$  of a code of word length  $s$  and Hamming distance  $h$  are identical. We can assume without loss of generality that both were formed by striking out the last  $h - 1$  letters of two codewords  $c_1$  and  $c_2$ . However, this results in the following contradiction

$$h \leq h(c_1, c_2) = \sum_{i=1}^s \delta(q_{1i}, q_{2i}) = \sum_{i=s-h+1}^s \delta(q_{1i}, q_{2i}) + 0 < h + 0 < h \quad (68)$$

Conversely, assume a code has the above subword property. Be  $c_1$  and  $c_2$  two distinct codewords of distance  $\hat{h} < h$ . That is,  $s - \hat{h} \geq s - h + 1$  of their letters are identical. This conflicts with the assumption that each subword of length  $s - h + 1$  occurs at most once.

For a given projection pattern, we define its color-pattern subwords of a certain type to be the set of all subwords of this type of its associated color pattern codewords. If the subword type involves striking out all leftmost  $v$  symbols of a word, we also include the subwords of imaginary codewords whose  $s - v$  rightmost symbols are the  $s - v$  symbols of a leftmost codeword into this set.

**Corollary 2a:** Adding a new column to a projection pattern of the proposed type preserves its minimal distance  $h$  if and only if each length  $(vw - h + 1)$  subword (formed from the  $v$  new color-pattern codewords) that involves at least one of the  $v$  new color symbols does not occur among the existing color pattern's subwords of the same type.

Proof: Assume the extended color pattern does not have the minimal distance  $h$ ; according to corollary 1, this implies there are two codewords  $c_1$  and  $c_2$  that have identical subwords  $c_1'$  and  $c_2'$  of length  $vw - h + 1$ . One of them has to be an existing codeword, say  $c_1$ , and the other one,  $c_2$  has to be a new codeword. Assume  $c_2'$  does not involve any of the  $v$  new symbols; then  $c_2'$  is already a subword of the left neighbor of  $c_2$  (also of length  $vw - h + 1$ , but of another type). Analogously,  $c_1'$  is a subword of the latter type of the left neighbor of  $c_1$ . According to the corollary 1, this would imply that the previously existing code does not have the minimal distance  $h$ . Consequently,  $c_2'$  involves at least one of the  $v$  new symbols. The other direction of the proof is an application of corollary 1.

For a given projection pattern, we define its edge-pattern subwords of a certain type to be the set of all corresponding subwords of its associated edge pattern codewords. If the subword type involves striking out all leftmost  $2v - 1$  symbols of a word, we also include the subwords of imaginary codewords whose  $2vw - 3v - w + 1$  rightmost symbols are the  $2vw - 3v - w + 1$  symbols of a leftmost codeword into this set.

**Corollary 2a:** Adding a new column to a projection pattern of the proposed type preserves its minimal distance  $h'$  if and only if each length  $(2vw - v - w - h + 1)$  subword of the  $v$  new edge-pattern codewords that involves at least one of the  $2v - 1$  new edge segment symbols does not occur among the corresponding edge-pattern subwords of the existing pattern.

Proof: The proof is identical to the one for the color code.

Parameters	Lowest Known Upper Bound on Word Count	Maximal Word Count Generated
$v=1, w=4, h=2, h'=2$	116	110
$v=1, w=5, h=2, h'=3$	116	114
$v=3, w=3, h=2, h'=5$	-	$702 = 234 \cdot 3$
$v=3, w=3, h=2, h'=6$	-	$471 = 157 \cdot 3$
$v=3, w=3, h=2, h'=7$	-	$285 = 95 \cdot 3$

Table 9: Exemplary results of the code generation algorithm for some practically relevant choices of parameters.

In combination, the above two corollaries allow checking whether growing a pattern by one column preserves its minimal distances very efficiently. To that end, a look-up-table (LUT) is allocated for each possible subword type of length  $s - h + 1$  that takes subwords of this type as key. If the look-up returns 0 for a given subword, the word does not yet occur among the corresponding subwords of the previously existing codewords; otherwise, it does occur. Then, if the LUT is initialized to 0 and properly updated whenever new codewords are added, it allows checking whether a given subword is already used in  $\Theta(1)$ . Since there are  $bc(s, s - h + 1) - bc(s - v, s - h + 1)$  types of subwords, this allows an efficient implementation of the proposed algorithm if either  $s - h$  or  $h$  is small.

E.g. for the parameters  $v = 1, w = 5, h = 2, h' = 3$ , checking whether a new candidate preserves the minimal distance requires three look-ups to check  $h'$  (there are three types of subwords of length two involving the new candidate; the color code distance  $h$  does not have to be checked because it follows from  $h' = 3$  that  $h > 1$ ). With the straightforward implementation, each check would require about 400 comparisons (most of the checking is done for codes that have a large number of words, because only then finding a candidate requires many attempts; with the given example, the algorithm does most of its checking against codes of about 100 words). Consequently, the proposed approach incurs in this case a significant speed-gain by a factor of about 100. This increase is relevant because finding a good code takes a couple of days even with the improved version.

Furthermore, several other heuristically motivated extensions are made to the above algorithm. Namely, the improved algorithm is given a certain amount  $t$  of time for backtracking: whenever it hits a dead end and still has some of this time left, it returns to the last junction that still has an unexplored path and follows it. Whenever it finds a code of a new record length, the time-limit  $t$  is increased exponentially. That is, an improvement resulting in a code whose length is close to the theoretical or some user-defined upper limit yields a very large time bonus, one far below this threshold only a small one. Finally, several heuristics attempt to improve the branching decisions over a purely random choice.

The above algorithm has been implemented in LISP. Table 9 lists exemplary results for the practically most relevant choices of parameters, obtained on a Pentium IV 2.4 Ghz. Clearly the algorithm is able to generate projection patterns of the proposed type that have a reasonably high resolution and minimal distance. This also proves that such patterns exist, a fact that is by no means self-evident.



#### 4.3.4 Data Processing

This section discusses how to convert an image of a scene illuminated with a pattern of the proposed type into a depth map.

##### 4.3.4.1 Demodulation and Decoding

The task of demodulation, i.e. of detecting the projected color edges and of establishing their spatial relationship, is in principle straightforward: Edge detection in gray level images is one of the classical and best understood problems of image processing. There are accordingly a large number of established techniques, most of which are based upon differentiation in one or the other form. This also applies – even if to a somewhat lesser extent – to multi-spectral images with their vector-valued image functions. A common way to deal with them (besides simply converting them to gray level images) is to treat their components individually. Doing so, however, gives rise to “the problem of how to combine them into one output” [Machuca and Phillips 1983]; e.g. in the case of a gradient based method, the gradients of the separate components will often point into different directions. For this reason, alternative approaches introduce some kind of real-valued local measure of directional multi-spectral contrast or discontinuity, e.g. for a given point the squared Euclidean distance of its image value and the one at unit displacement in the direction of interest ([Cumani 1991]). With the latter definition, for each pixel a well-defined direction  $\mathbf{w}$  of maximal contrast exists barring degenerate cases; pixels for which the multi-spectral contrast exhibits a local directional maximum in the direction  $\mathbf{w}$  are then considered edge points.

However, all state-of-the-art edge detection approaches have the following disadvantages for the purpose of demodulating patterns of the proposed type.

- Classical approaches detect all edges present; we are, however, only interested in certain types of edges, namely the projected ones.
- Techniques based on multi-spectral contrast combine the information contained in the separate channels to a single result. Consequently, significant changes in one channel are able to drown out edges of small contrast in another channel. This effect is a major problem e.g. with surfaces of high reflectivity for one and low reflectivity for another band. With the proposed pattern, different channels behave distinctly by design; recovering this dissimilar behavior is crucial for demodulation and the effects such as the described need to be avoided.
- For correct demodulation, the precise type of edge needs to be known, i.e. it is relevant which channels change in which way. Three channels, each of which can rise, fall and stay unchanged, yield 26 different classes of edges. Such an edge classification is not part of any of the classical approaches.

For these reasons, primarily the first one, we develop our own approach to edge detection. It splits up the task in two parts, namely *edge pixel detection* (the pixel-wise classification of all pixels of a given image into the two sets “part of an edge” and “not part of an edge”) and edge segment or contour detection (the classification of the edge pixels into disjunct sets according to their physical origin, in our case more specifically according to the projected edge that caused them, including the class “intrinsic edge that does not originate from a projected edge”).

The proposed approach to edge detection is based on derivation; it is well known that edge detection via derivation is an ill-posed problem in the sense of Hadamard; for that reason the image needs to be regularized with a suitable filter, typically a Gaussian, preceding differentiation. Accordingly, the algorithm smoothes in a first step the three separate components  $I_l(x, y)$ ,  $1 \leq l \leq 3$ , of the color image  $I(x, y)$  with a Gaussian filter. As stated by Torre and Poggio [1986], the strong regularization properties of this filter guarantees the existence and continuity of the derivatives of the smoothed components  $I_l(x, y)$ . In the following  $I(x, y)$ , respectively  $I_l(x, y)$  refers to the filtered image.

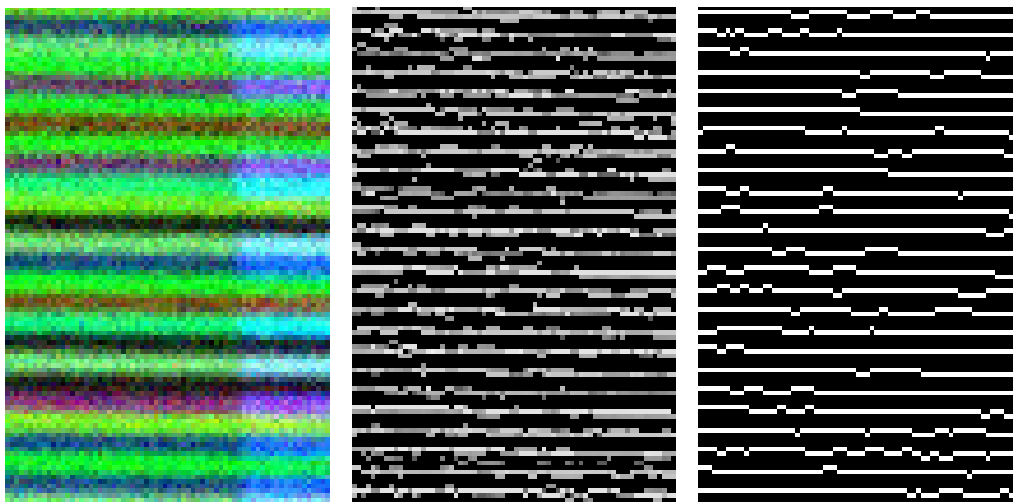


Figure 35: Color image (left) – Image formed by combining the three single-channel local extrema images via logical AND (middle) – Traced ridges of correctly identified edge pixels (right).

With respect to further processing, we distinguish between a one- and a two-dimensional projection pattern. We start with the simpler former case.

**Edge Pixel Detection:** After smoothing the algorithm performs the first stage of edge pixel detection in each of the three monochromatic channel images separately. In a given channel image  $I_l$ , it establishes the local orientation  $\mathbf{v}$  of the pattern stripes. There are two approaches to this task; the first is to compute the gradient direction  $\mathbf{w}$  and to assume  $\mathbf{v}$  to be orthogonal to  $\mathbf{w}$ ; the local orientation is thus well-defined in the 2D image plane as long as the gradient does not vanish. The second is to simply assume  $\mathbf{v}$  to be the orientation of the projected light stripes, e.g. vertical for a pattern of vertical stripes. The orientation  $\mathbf{w}$  is then set to the horizontal vector  $(1 \ 0)^T$ . With the latter approach, the algorithm might miss projected color edges whose imaged orientation deviates strongly from the projected one, i.e. edges projected on surfaces with certain position/orientation combinations; at the same time, the second approach is computationally more efficient and, more importantly, not disturbed by intrinsic edges whose orientation differs strongly from  $\mathbf{v}$ .

In any case, the following vectors and functions are well-defined almost everywhere in the image:

$$\mathbf{w}_0 = \frac{\mathbf{w}}{\|\mathbf{w}\|}, \quad I_{hw}(t) = \frac{\partial I_l(x, y)}{\partial \mathbf{w}_0}, \quad I'_{hw}(t) = \frac{\partial^2 I_l(x, y)}{\partial^2 \mathbf{w}_0} \quad (\mathbf{w} \neq \mathbf{0}) \quad (69)$$

The derivative  $I_{hw}(t)$  of  $I_l(x, y)$  in the direction of  $\mathbf{w}$  is computed and its local extrema, the transversal zero-crossing of  $I'_{hw}(t)$ , are established. The sub-pixel location of an extremum is chosen as the vertex of a parabola fitted to the extremum and its two unit-distance neighbors along  $\mathbf{w}$ . Only these local extrema are kept for further processing, a step known as *non-extrema suppression*. Performing the above procedure for each channel independently yields three images that contain only local extrema, called *single-channel local extrema* in the following. Figure 35 illustrates this step.

In a next stage, the three separate channel images are combined into a single one. To that end, single-channel local extrema are grouped to *multi-channel (local) extrema* of the composite color signal: two or three single-channel local extrema, each from a distinct channel, are combined if they share the same direction and are spatially sufficiently close. The sub-pixel position of such a multi-channel local extremum is computed using a weighted average of the positions of its components. The weight-factors are determined on the basis of the goodness of fit to the expected model and the signal's distance to the noise level. Also, each multi-channel local extremum is classified into one of the 20 color edge classes that are of the multi-channel type.

Only multi-channel extrema are classified as edge pixels; single-channel local extreme values that are not part of a multi-channel one are ignored during further processing. With other words, a second non-extrema suppression takes place. Since projected edges are known to have a minimum weight of two, i.e. to affect at least two color channels, and since the probability of a missed pattern related local extreme value is rather low (only a very small threshold is used), this step is a very effective first measure to filter out noise without losing any of the sought-after information.

**Edge Segment Detection:** Next, spatially adjacent edge pixels of the same class are traced to obtain edge segments. A classical tracing step would filter out all remaining unwanted edge pixels except ones that form segments themselves, e.g. ones due to reflectivity edges. To cope with them, the algorithm solves the identification problem before tracing: It establishes the spatial adjacency relationship of edge pixel by determining sequences of  $w - 1$  multi-channel extrema that share the same orientation  $w$  and lie along a line parallel to  $w$ . It then attempts to decode the resulting words, i.e. it checks whether the edge pixels, interpreted as code symbols according to their edge class, form a codeword when read from left to right along  $w$ . As proven in section 4.3.3.3, most errors can be detected, i.e. if a codeword is found, it is very unlikely to represent an undetected error.

Only edge pixels that are part of a valid codeword are hypothesized to be the location of a projected color edge orthogonal to  $w$ ; only they are used as starting points of the tracing operation. A well-known problem with edge segment detection via tracing is *streaking*, the “breaking up of an edge contour caused by the operator output fluctuating above and below the threshold along the length of the contour” [Canny 1986]. A popular solution to overcome streaking is *hysteresis*, where two thresholds, a low and a high one, are used. Edge pixels above the high threshold are accepted, ones below the low threshold are rejected, and ones between the two thresholds are accepted only if their segment is connected to high-threshold edge pixels in both directions.

In our case, streaking is less a question of the threshold, more one of fluctuating between a valid and an invalid codeword. So the algorithm uses an analogous approach where edge pixels of the same edge class as the starting point pass the low threshold and those also part of the same codeword pass the high threshold. A third threshold is used to pick up occasional edge pixels misclassified into a class close (according to a suitable signal space metric) to the sought one. Beginning and end of a segment need to pass the high threshold as often, e.g. at the boundary between two objects of distinct height, edges of the same class, but originating from different projected edges, join seamlessly (see e.g. the boundary between hand and color chart in figure 36). Obviously two such edges cannot be distinguished by considering the segment by itself; the high threshold is crucial for resolving such situations. It is important to note that the proposed type of tracing is effectively an error correction step that is more reliable than replacing words that contain a detected error with the codeword that is closest according to some code space metric.

Only edge pixels part of a ridge that exceeds a small minimal length are used for further processing. This can be seen as another error detection step that even detects errors that result in seemingly valid codewords, i.e. ones that are undetectable in the classical sense: Undetected errors are unlikely per se, and it is even much more improbable that several of them form an edge segment.

All things considered, the algorithm effectively determines color edges in the pattern image, yet not by a standard direct approach to edge detection, but rather using an approach designed to detect only color edge segments originating from the projected pattern. It does so by combining demodulation and decoding into one operation rather than solving one after the other; this seems to be necessary to solve the difficult task reliably even under noisy real-world conditions. Clearly the proposed approach solves two key problems commonly associated with color-coded light: neither background illumination nor the scene color pose a major problem. Both of them represent a constant factor, which has no impact on the derivative (but for decreasing the signal-noise ratio) and accordingly on the resulting edge segments. Figure 36 illustrates the ability to cope with colored scenes; it shows the output of the algorithm given a hand in front of a color chart as scene. It can be seen that almost all projected edge segments are recognized correctly.

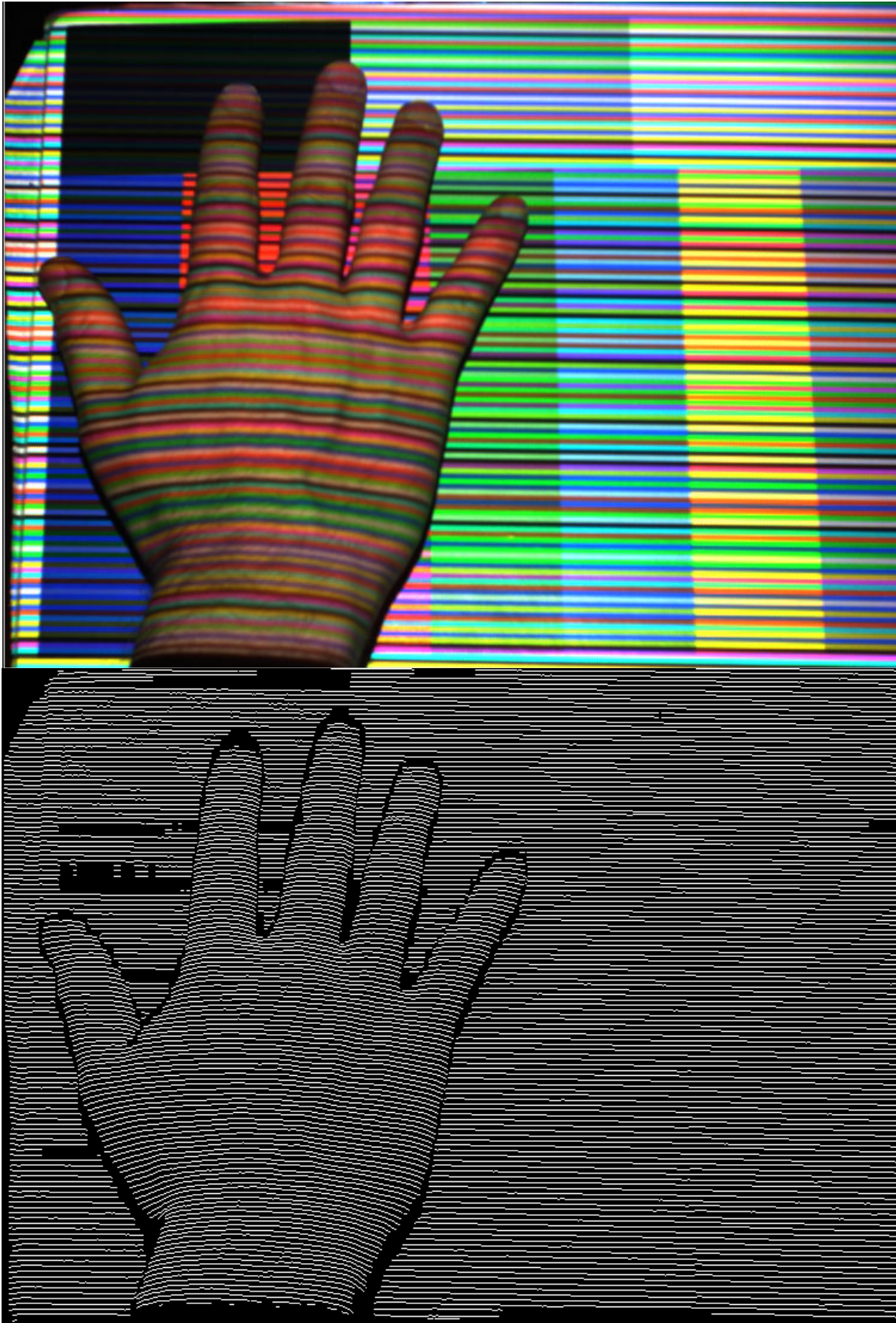


Figure 36: Detected and correctly identified projected color edges (bottom) given an image of a color chart as input (top).

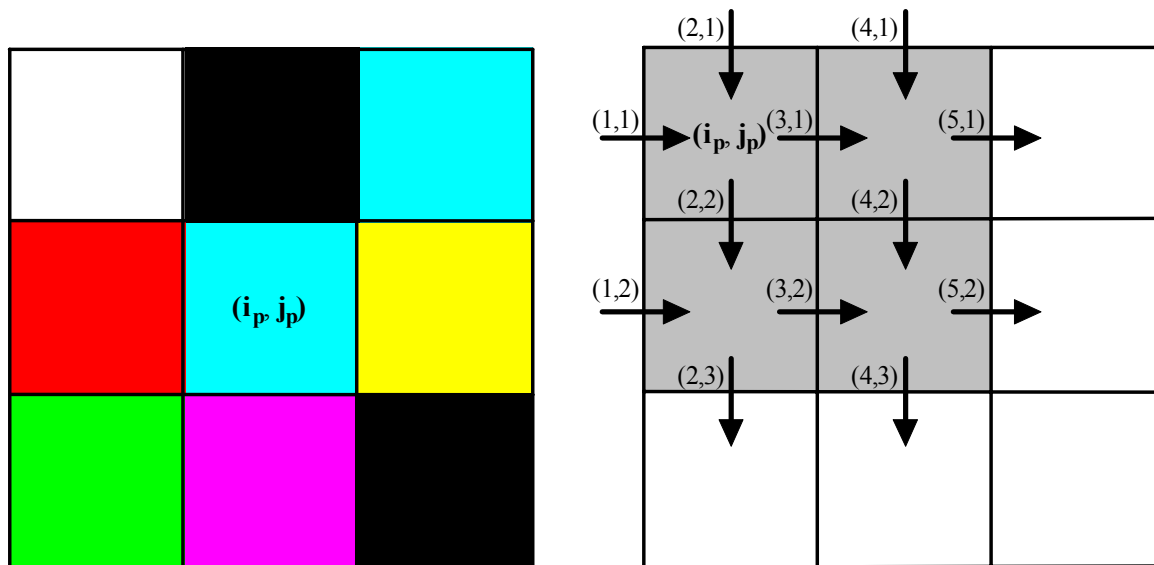


Figure 37: Remodeled subpattern layout to obtain closed contours for the exemplary parameters  $v = w = 3$ .

In the case of a two-dimensional pattern ( $v > 1$ ), the algorithm is necessarily more complex; the underlying basic idea is to attempt to detect closed contours in the image. It is for that reason advantageous to slightly modify the subpattern layout proposed in figure 32 to one that forms a closed contour. The slightly remodeled version is displayed in figure 37; in its case a subpattern of the derived pattern forms a closed contour of  $v - 1$  by  $w - 1$  primitives. It can be shown that none of the code's error detection properties is lost because of the modification; its only disadvantage is that a  $v$  by  $w$  color primitive subpattern no longer uniquely defines its color edge subpattern, only in conjunction with its left neighbor window. While this is formally somewhat awkward, it causes no practical problems.

Despite the increased complexity, the 2D pattern algorithm operates similarly to the 1D one. It is again structured into the tasks of edge pixel and edge segment detection.

**Edge Pixel Detection:** Edge pixel detection is virtually unchanged compared to the 1D pattern. The algorithm also performs non-extrema suppression in each color channel and subsequently combines single-channel local extrema to multi-channel ones according to their orientation and spatial distribution. Only the latter are considered edge pixels and classified into one of the 20 multi-channel edge classes. Figure 38 illustrates this operation.

**Edge Segment Detection:** The edge segment detection starts by detecting small segments of edge pixels of the same class. All detected segments are classified as either horizontal or vertical; as it can be shown that projected horizontal segments remain approximately horizontal in the camera image (assuming the vertical separation of camera and projector is negligible), this classification is uncomplicated. Points at which at least two segments cross are considered corner points. While this step is in principle straightforward as well, the details of implementing it are not, because edge segments are rounded off at the square corners, crossing segments intersect in several pixels or in none at all due to the discrete nature of the image, the four segments that make up a typical corner intersect at many different image positions rather than one etc. The algorithm solves these issues by replacing each segment by a straight line segment that is a few pixels longer than the original segment, whose pixels are all neighbors in the four-neighborhood sense and that has a unique id. This effectively eliminates the case of missed intersections. Next, the typically quite many associated crossing points of each edge segment are reduced to only two corner points per segment, where each reduction is communicated to all participating segments.

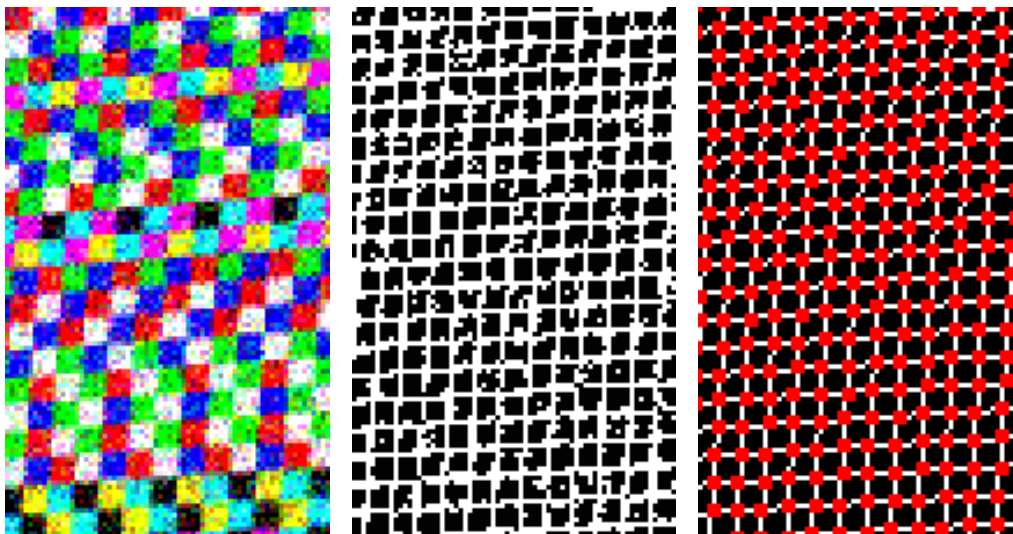


Figure 38: Color image (left) – Image formed by combining the three single-channel local extrema images via a logical AND operation (middle) – Corner points and closed contours of correctly identified edge pixels (right).

As a result, each edge segment is associated with one or two corner points, respectively each corner with two to four edge segments. This permits detecting the segments that form a closed contour. The neighborhood relationship of closed contour follows immediately given that each segment will typically be part of two such contours. Decoding then simply implies forming the word resulting from a subpattern, i.e. from blocks of  $v - 1$  by  $w - 1$  closed contours with their  $2vw - v - w$  symbols, and comparing it with the codewords. In a last step, the decoding information is propagated from correctly identified subpatterns to nearby unidentified closed contours that meet certain criteria (closed contour of about the same size, small code space distance to the expected symbol, etc.). With other words, error correction is again done by exploiting the depth or disparity continuity constraint rather than via the classical coding-theoretic operation of error correction. Figure 38 shows an example of the detected contours of correctly identified edge pixels.

When comparing the two types of patterns, the following conclusions can be drawn:

- The algorithm for a 1D pattern is simpler to implement and computationally more efficient; its resolution on the basis of reasonably compact subpattern is limited, but suffices for the objectives of this work. Its error detection capabilities are adequate for the scenes targeted in this work.
- The algorithm for a 2D pattern is more difficult to implement and computationally less efficient, even though a well-optimized implementation should be able to keep the practical impact of this drawback minimal. The 2D approach opens up the way to very high resolutions, respectively a higher redundancy with compact subpatterns. It is well suited for exploiting the abilities of high-resolution color cameras such as today's megapixel consumer cameras.
- For a given camera, the achievable non-interpolated range image resolution of the 2D pattern is somewhat lower than that of the 1D pattern, because the squares need to have a side length of about 5 to 6 pixels to obtain a meaningful edge segment length, whereas with the 1D pattern the theoretical maximal resolution of about 3 to 4 pixels per stripe (see section 4.3.1.1) can be actually achieved in practice.

With respect to further processing, there are no significant differences between the two types of patterns; for that reason, the remainder of this chapter does not distinguish between them.

Number of Iterations	Mean	Sample Standard Deviation	Min	Max	Number of Samples
0	+0.05 mm	+0.69 mm	-2.27 mm	+2.58 mm	838911
1	-0.01 mm	+0.15 mm	-2.11 mm	+1.92 mm	838911
2	-0.01 mm	+0.15 mm	-2.11 mm	+1.92 mm	838911

Table 10: Statistical parameters of the  $z_w$ - distribution in a depth map of the  $z_w = 0$  plane for various numbers of iterations of the iterative approach to compensating the projector's distortion. The depth map after 0 iterations is shown in figure 39, the one obtained after one iteration in figure 40.

#### 4.3.4.2 Triangulation

A SL system computes range values by intersecting projected light planes with lines of view of the camera, i.e. via plane-ray triangulation. At a first glance, this step seems to amount to one of the simplest problems of analytic geometry. Figure 39 illustrates that the task is more complicated. It shows the (world coordinate) depth image of the planar calibration target of size 600 by 400 mm. The target defines the  $z_w = 0$  plane in 3D space, consequently the map should have a value of 0 everywhere but for small, zeromean deviations attributable to the unavoidable ranging noise. This is clearly not the case; in some regions the depth values go systematically up to 2mm, which represents an unacceptably large measurement error. Many range images that are acquired with a structured light system and that are displayed in the literature show a similar effect (e.g. [McIvor 1994]). The reason for this type of systematic ranging error is that the above ray-plane intersection approach ignores the radial distortion of the projector.

Even if the projector's optical slide center and its coefficients of radial distortion are known, there is no simple way to correct this distortion: Only the illuminating plane, that is the single projection slide coordinate  $i_p$  is known, but not the ray (i.e.  $j_p$  is missing); both coordinates would be needed to determine the radius and to compensate for radial distortion. We propose to consider the projector's lens distortion with the following iterative approach. In a first step, a sort of average radial distortion is considered by taking the plane defined by slide coordinates  $(i_p, m_p/3)$  and  $(i_p, 2m_p/3)$  as the projected one. Simple plane-ray intersection with this plane yields preliminary camera coordinates  $(x_c, y_c, z_c)$  of the imaged scene point. These are converted via world to projector coordinates  $(x_p, y_p, z_p)$ . Back-projecting the latter on the slide plane results in ideal undistorted slide coordinates, which can be transformed to distorted slide coordinates  $(i_p', j_p')$ . Ideally,  $i_p$  is close to  $i_p'$  and  $(i_p, j_p')$  represents a very close approximation to the illuminating ray. This permits revising the plane equation by considering the radial distortion present at slide coordinates  $(i_p, j_p')$ . A second ray-plane triangulation then yields – typically much more accurate – 3D-space coordinates.

There are several alternatives to the above approach: e.g. the epipolar constraint could be exploited by computing the epipolar line on the projection slide defined by the known camera image coordinates; in that case no first depth estimate is needed. Intersecting the epipolar line with the ideal straight line (respectively distorted curve) corresponding to the identified light plane again yields a good estimate of the sought-after projection slide coordinate  $j_p$ . In practice, all approaches give about the same results; it depends on the implementation which one is more efficient.

Table 10 and figure 40 show that the above iterative approach enhances the ranging accuracy significantly: already after a single iteration the resulting depth map has the expected properties (about zeromean, small root mean square error of  $\sigma = 0.15$  mm). A second iteration does not improve things further. The remaining nonzero mean does not necessarily mean there still is an accuracy problem as the calibration target is known to be somewhat uneven. Also the calibration target, which is shown in figure 27, consists of black squares on a white background. Their contrast represents a worst case scenario for the algorithm that occasionally manages to disturb the location of stripe edges significantly. This explains the large maximal depth error of  $14\sigma$ .

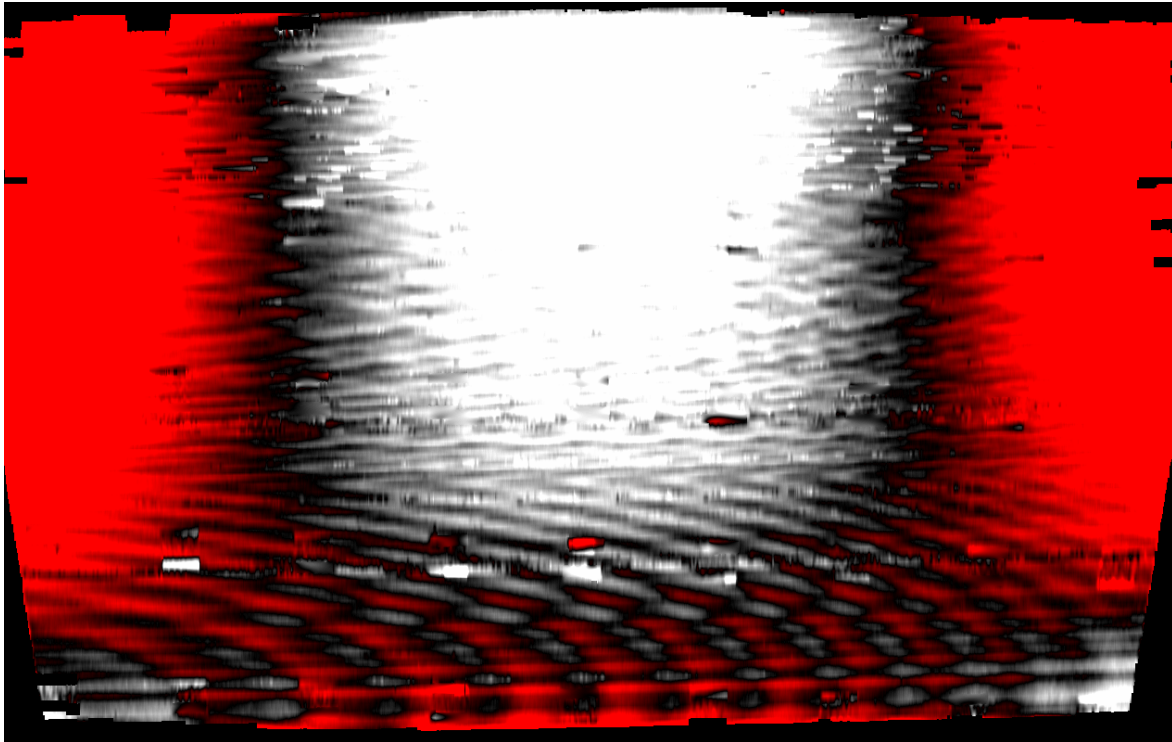


Figure 39: World-coordinate depth map of the  $z_w = 0$  calibration plane obtained without considering the projector's radial distortion. Positive  $z_w$ -values are mapped linearly to red-shades (0.00 mm as black, 0.70 mm as red), negative ones analogously to gray levels (0.00 as black, -0.70 mm as white).

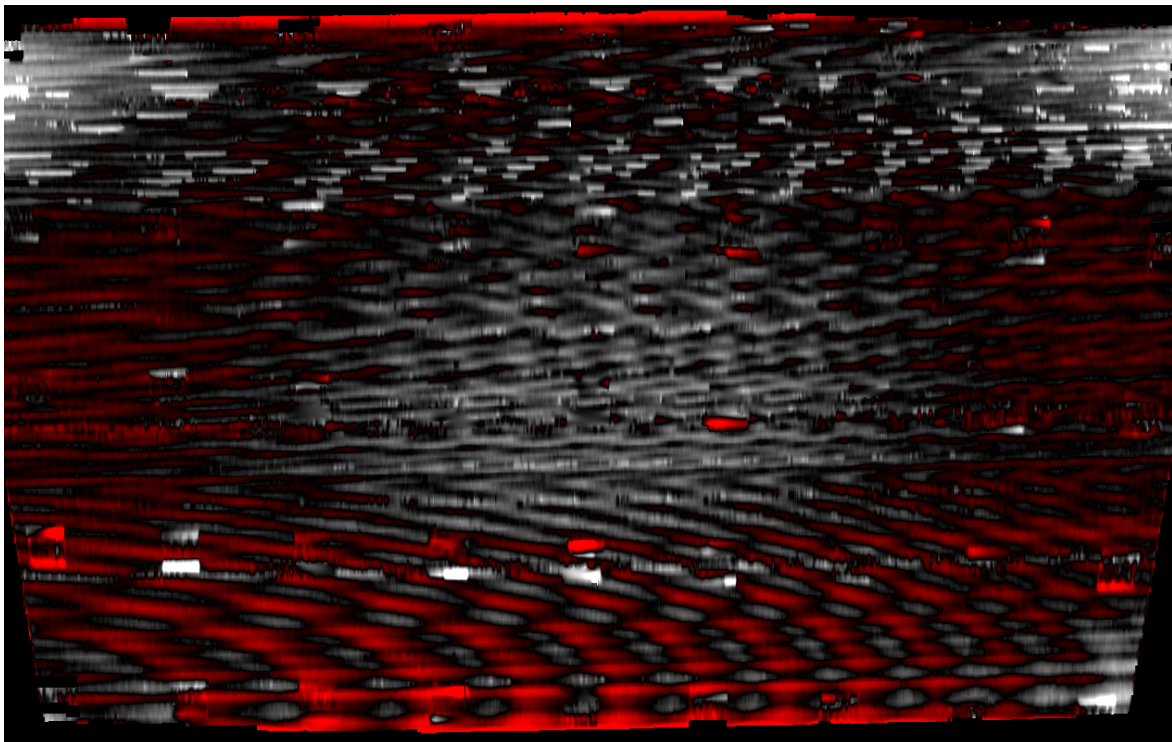


Figure 40: World-coordinate depth map of the  $z_w = 0$  calibration plane obtained by considering the radial distortion of the projector (one iteration). Visualization is identical to figure 39.



#### 4.3.4.3 Interpolation

As discussed before, computing depth values via triangulation is only possible for identified edge segments. There will necessarily be certain interspaces between adjacent edge segments. To obtain a dense range map, the depth needs to be interpolated for all remaining pixels part of an identified subpattern. It is important to note that the interpolation is only performed between edges that belong to the same projected stripe or square, i.e. that represent its left and right border. With other words, it is applied only to surface patches that are known to be continuous. Consequently, it does not cause the problems normally associated with interpolation; neither does it introduce artifacts into the range data, e.g. in the form of smooth transitions over depth gaps, nor wipe out small objects in the foreground.

The distance between two consecutive stripe edges is typically 2 to 4 pixels; this very small gap is bridged with linear interpolation. The main motivation for choosing this simple method is its efficiency. As performing linear interpolation is straightforward, it is not discussed here in detail.

### 4.4 The Stereo Subsystem

As motivated in section 4.2, we intend to employ a second camera, i.e. a set-up as with active stereo systems (figure 41) to obtain range values for the image areas for which the coded light step fails and to improve the data quality in general. In order to do that, we first need to understand the potential weaknesses of the coded light step. Generally spoken, it breaks down if it cannot recognize the reflected subpatterns in the pattern image. There are eight major reasons that might lead to this situation:

- **Surface Discontinuity:** If the continuity constraint is violated, e.g. with small objects too small to reflect the subpatterns integrally, with depth jumps or at object borders.
- **Reflectivity Discontinuity:** If the reflectivity smoothness constraint is violated. Reflectivity edges are for instance caused by certain types of texture.
- **Strong Crosstalk:** If the crosstalk between the color channels is strong. In that case, edge pixels can still be detected, yet their classification and as a consequence the decoding fails systematically.
- **Limited Depth of Field:** If the scene is outside of the depth of field of camera or projector. Especially the latter case occurs quite frequently as most projection devices have a rather limited depth of field given they are designed to illuminate planar surfaces. If the circle of confusion becomes too large, the projected edges become too blurred to be detected.
- **Occlusion:** If the imaged surface is occluded from the projector's field of projection.
- **Orientation:** If the imaged pattern frequency becomes too high, i.e. aliasing occurs. This occurs with certain surface orientations and positions. As a result, the color edges are imaged as high-frequency, high amplitude noise.
- **Oversampling:** If the imaged signal is saturated. This often happens in industrial environments where specular reflection is predominant.
- **Noise:** If the reflection of the projection pattern does not exceed the noise level. In this context, the term noise level refers to the reflection of the projected pattern, not to the overall reflection; e.g. with a strong background illumination, the absolute strength of the imaged signal might be well above noise level even though its modulation due to the projected pattern is imperceptible.

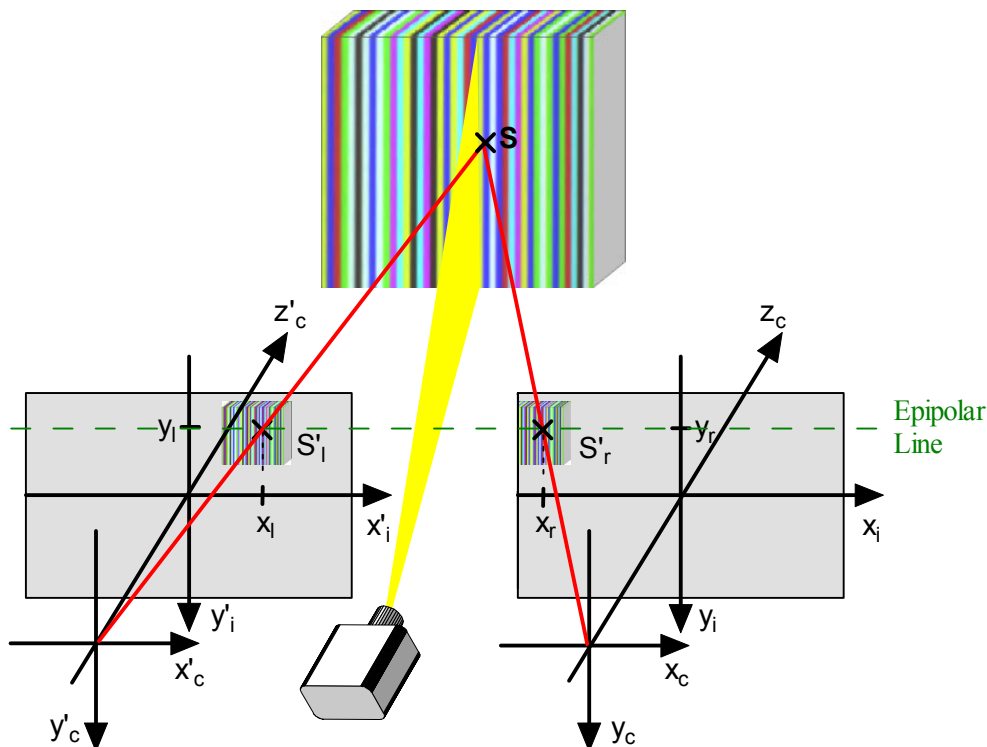


Figure 41: The principle of the combined coded light/stereo vision approach.

We hypothesize in the following that the areas where the above problems occur make up only a small part of the scene. At least the first three causes give rise to a pronounced optical structure in the pattern image, and one unlikely to be regular due to the pattern illumination. In the other cases, this is less likely, but nevertheless quite possible. So the idea of employing a stereo algorithm for the problematic areas is plausible. Its implementation is straightforward:

Initially, each camera-projector combination acts as separate CL system and acquires a range image. Then the two systems combine their respective results. A stereo algorithm identifies the problematic areas where both CL systems could not obtain range values and attempts to compute depth estimates for them. It is followed by an algorithm that establishes dense sub-pixel correspondence. All these steps are discussed below in detail.

There seems to be no comparable system described in the open literature: combining coded light and stereo vision has nominally been proposed before, but in a way very distinct from the above one and with other objectives in mind. For instance, Chen et al. [1997] project the pattern proposed by Boyer and Kak [1985] on a scene imaged by two color cameras. However, their only motivation for doing so is to assign the scene a texture that accommodates the needs of stereo vision algorithms. They explicitly do not attempt to decode the projected pattern, but employ a classical stereo algorithm based on dynamic programming. Consequently, the projector does not even have to be calibrated and, conceptually, their technique is an active stereo vision approach. Scharstein and Szeliski [2003] also employ a set-up with two cameras plus pattern projector and combine elements from coded light and stereo vision. However, their objective is to obtain the ground truth for stereo image pairs of complex scenes. This information is needed to evaluate the performance of stereo vision algorithms. To determine the ground truth, they use the classical temporally encoded Gray-code approach. Accordingly, they solve the identification problem, but only as means to solve the correspondence problem: if it is known that a certain light plane illuminates the scene patch imaged at  $x_l$  in the left image and the one imaged at  $x_r$  in the right image (for a given fixed image row), it follows that  $x_l$  and  $x_r$  form a conjugate pair. Again, the projector does not have to be calibrated for this task.

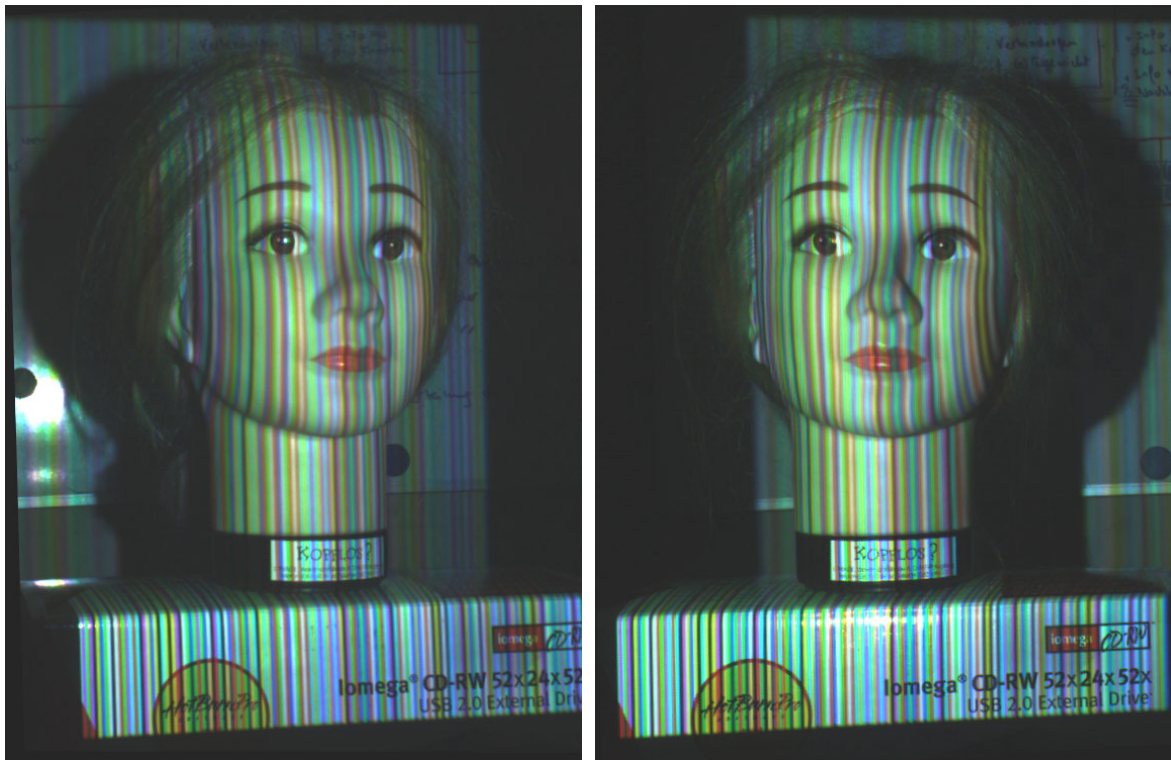


Figure 42: Rectified images of an exemplary scene as seen from the left camera (left) and from the right camera (right) with an set-up as shown in figure 41.

#### 4.4.1 Rectification

After their acquisition the images are rectified, that is they are re-projected on a common virtual image plane such that their rows and epipolar lines coincide. This virtual plane is not uniquely defined; almost any plane parallel to the baseline can be used. The rectification methods discussed in the literature accordingly deal with the question of how to choose this plane. A common approach, which is also taken on in this work, is to pick out the virtual plane that minimizes the distortion of the projected images as well as their scale change. In any case, with a typical stereo set-up the differences between the various plausible choices are rather negligible in practice. As the problem of rectification is effectively solved, it is not discussed further in this work. Figure 42 shows an exemplary pair of rectified images acquired with a geometric set-up as shown in figure 41, that is with one camera to the left, the other to the right of the projector, implying a horizontal-only separation.

#### 4.4.2 Mutual Update

The purpose of the mutual update is for the two CL systems to share their respective results obtained so far. For the sake of efficiency, the algorithm exchanges information about identified stripe edges rather than about obtained depth values: Given an identified stripe edge  $E$  of image  $A$ , it projects the corresponding 3D space edge into image  $B$ . If there is no identified stripe edge at or near this computed position, it adds  $E$  to image  $B$ . This step can introduce range values into image  $B$  that are correct – there is a surface at this position in 3D space – but that might not be visible from the point of view of camera  $B$ . The added stripe edges are for that reason marked with a flag that indicates that they might belong to surfaces invisible from the camera's point of view.

The mutual update is a very effective measure to overcome occlusion-related problems. Nevertheless, in the general case it cannot solve problems related to subpatterns that are visible, but whose reflection is corrupted due to one of the reasons mentioned above: most causes for failure such as reflectivity discontinuity are likely to affect both viewpoints.

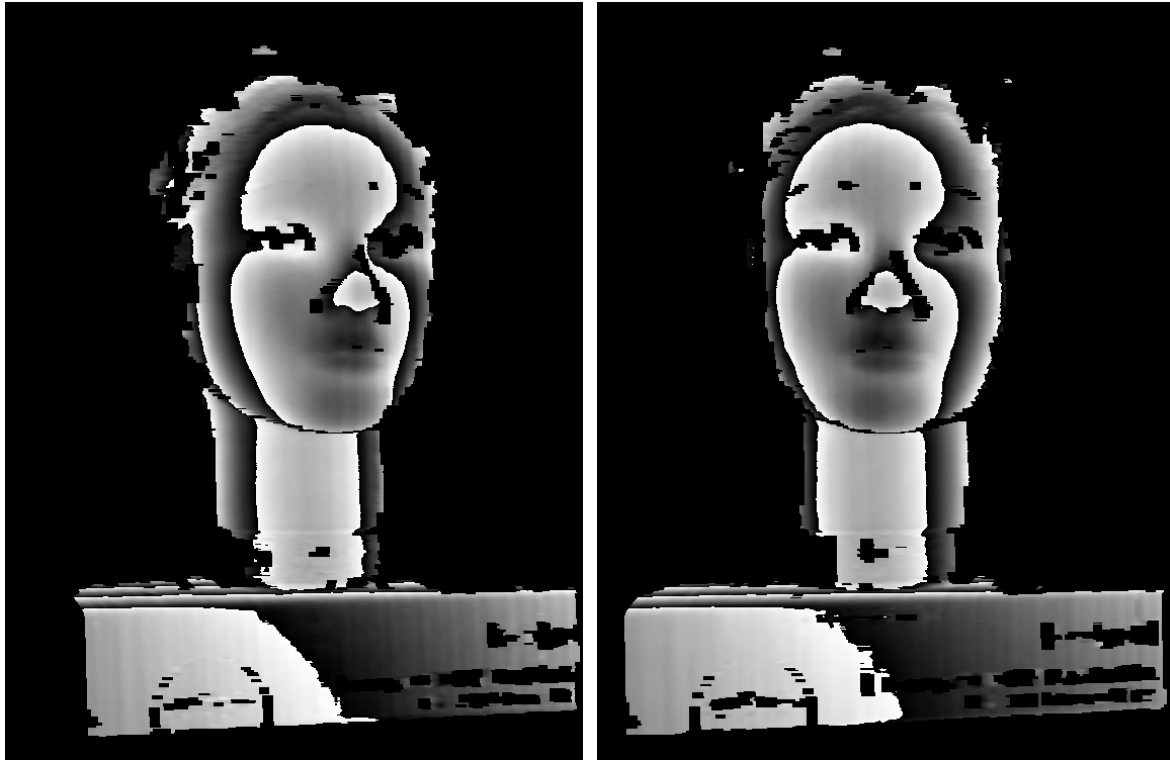


Figure 43: Output of coded light step for the left image of figure 42 (left) and the right image of figure 42 (right). The range data is visualized by mapping the depth value of a given image point, measured in  $10^{\text{th}}$  of millimeter, to the gray value range of 0 to 255:  $g = (z_c \cdot 10.0) \bmod 256$ . One gray level period then corresponds to 25.6 mm. Points for which no depth value could be acquired are mapped to zero. The depth corresponds to the  $z_c$  coordinate within the coordinate system of the rectified cameras.

#### 4.4.3 Stereo Step

At the beginning of this step, the algorithm has already computed two more or less complete depth maps from two distinct points of view. Figure 43 shows an example for such a range image pair, computed from the pattern image pair of figure 42. Of course, any of the established types of stereo vision algorithm could now be used to complete the range image acquisition. For instance, it would be straightforward to use the results obtained so far as initial guess for a global optimization or layered approach to stereo vision as described in section 3.3.9. Tackling the task in this manner would, however, miss the point: the challenge to the stereo algorithm is to conserve both the reliability and efficiency of the coded light algorithm.

For this reason, we employ our own approach to stereo vision that is specifically targeted to the task at hand. For example, it should rather output no than erroneous range data. It operates as follows: It first performs contour tracing in each of the two depth maps using a special tracing routine that considers the depth value of adjacent points of the depth map. This yields two sets of closed contours for each image:

- **Inner Contours:** Contours that enclose areas for which no depth values could be obtained. They delineate holes in a depth map, for example the eyes of the head of figure 43.
- **Outer Contours:** Contours that enclose areas for which depth values could be obtained. They delineate objects in a depth map, for example the outline of the head without the neck of figure 43.

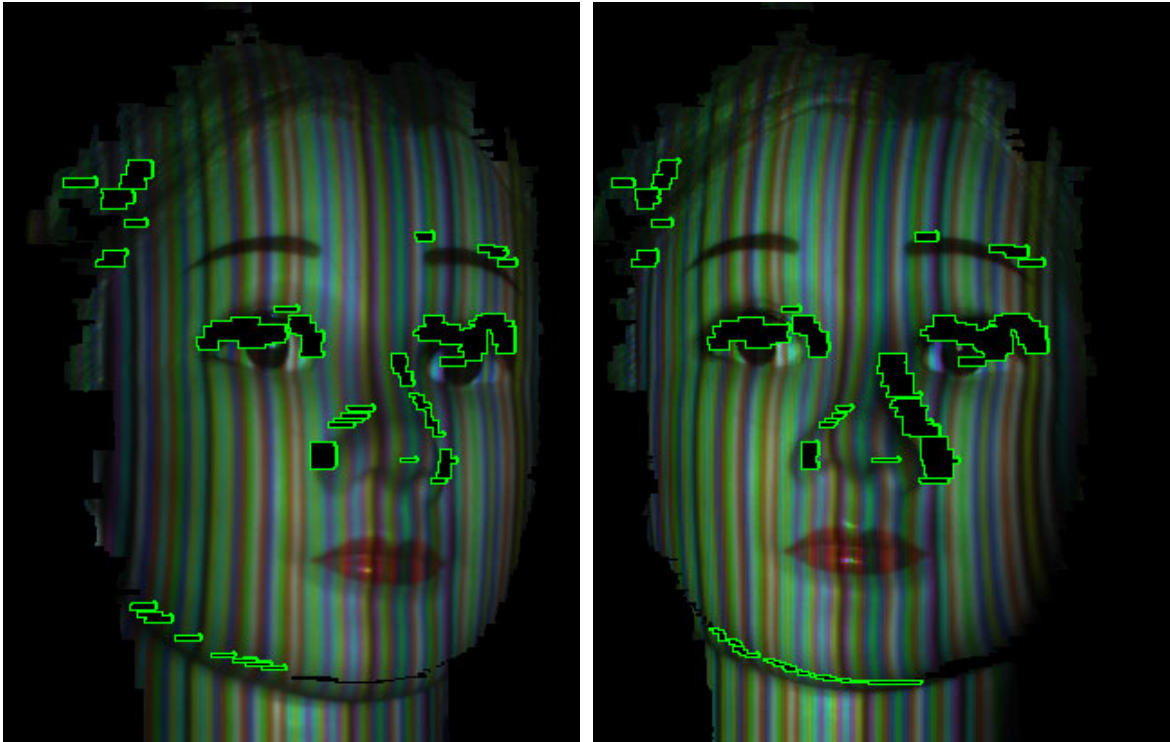


Figure 44: Example for the inner contours found in the left and right depth map of figure 43.

#### 4.4.3.1 Area-Based Stereo

The purpose of the area-based stereo algorithm is to compute range values for these parts of continuous surfaces for which one of the above-mentioned causes interrupted the CL algorithm. The candidates for such areas are the inner contours resulting from the above tracing operation. Accordingly, the matching is first done on the contour level: Given an inner contour  $C_A$  of image A, the algorithm attempts to find its corresponding contour in image B. So it computes for all inner contours of image B that meet the epipolar constraint a score that quantifies its similarity to  $C_A$ . This score considers aspects such as the two contour's difference in position, circumference and the contour moments including the area. For each contour of the reference image, the one of the other image with the highest similarity score is selected as match, unless the score is below a certain threshold, in which case the contour is left without a match.

Given two conjugate contours, a second matching is done on the pixel level. This low-level matching step is based on the hypothesis that the areas enclosed by inner contours are continuous, i.e. that the ordering constraint applies to them. Consequently, dynamic programming is well-suited to efficiently solve the correspondence problem – one image row after the other – for all pixels within two such conjugate areas. The details of this dynamic programming step are described in section 4.4.4 in the context of a related task.

If the total pixel matching cost (normalized by the number of enclosed pixels) for two supposedly conjugate contours is above a certain threshold, the areas are not matched at all. The same is done if the intensity variance within a contour is below another threshold – in most cases that means that there is an actual hole in the scene surface. A typical example for the latter case would be a bore-hole. In its case, there will be no range values in both CL images, the two resulting inner contours will be very similar and their uniformly dark intensity distributions will match perfectly. Without the above threshold, the bore would accordingly be mistaken for a continuous surface.

Experimentally, it turns out that the matching on the contour level is very simple, reliable and fast, which is mainly due to the fact that the overall number of contours tends to be in the hundreds only with a typical scene. After considering the epipolar constraint, the number of candidate contours is typically in the tens at most. Figure 44 illustrates this fact. It shows all inner contours found within the head area for the two depth maps of figure 43: Clearly there are only a few of them, most of which have a distinctive shape, which is at the same time very similar for two conjugate contours.

#### 4.4.3.2 Feature-Based Stereo

The feature-based stereo step attempts to match edge segments formed by edge pixels that could not be identified as projected color edges. Accordingly, it first detects such edge segments by tracing edge pixels of the same class that are not part of any identified color edge. For each image, this yields a set of color edge segments.

As with most classical feature-based stereo algorithm, a similarity measure is computed for each pair of segments. It considers factors such as the edge class, segment position and segment form. The correspondence of segments is then computed in the manner of a classical feature-based stereo algorithm. Such an algorithm is part of the state-of-the-art and for example described in section 3.3.9.4. It is for this reason not discussed here in more detail.

A more interesting aspect of the feature-based stereo is its attempt to improve the outer contours: the outlines of objects as computed by the coded light step are somewhat frayed because they necessarily correspond to the borders of the last identified subpattern, not to the actual object borders. The feature based-stereo attempts to determine the latter by searching for and matching the edges due to the object boundaries.

#### 4.4.4 Dense Correspondence Step

The coded light algorithm computes depth values for projected color edges only. Due to the active illumination, there will be many edges in the image, but by no means will every pixel of an image be an edge pixel. To achieve the targeted high, non-interpolated relative spatial resolution, the proposed algorithm establishes the dense correspondence for pixels between neighboring identified edge segments that belong to adjacent projected edges and that are only a few pixels apart. It is able to safely assume that the depth varies smoothly between two adjacent segments; otherwise the two adjoining projected edges would not be nearby neighbors in the pattern image.

Formally, the task at hand can be expressed as follows: Given the column interval  $[0, a]$ , which is defined by two edges in the first image, and the corresponding column interval  $[0, b]$  of the other image, find the function  $d(x): \{1, \dots, a-1\} \rightarrow \{1, \dots, b-1\}$  such that  $d(x) \leq d(x+1)$ , i.e. such that the ordering constraint holds, and such that the following cost function becomes minimal:

$$\sum_{l=1}^{a-1} \sum_{k=1}^3 (I_{k1}(l, y) - I_{k2}(d(l), y))^2 \quad (70)$$

where  $y$  is the considered image row and  $I_{kn}$  refers to the  $k$ th color component of the  $n$ th image. The above task can be solved efficiently via the well-known technique of dynamic programming. The resulting function  $d(x)$  is then considered to be the solution to the corresponding problem.

In practice, the described approach is modified slightly: the cost of matching two pixels is not computed over the squared intensity difference of the two single pixels only as in equation 70, but rather over two 5 by 5 window centered at the respective pixels. Also, to achieve sub-pixel accuracy, the actually found interval  $[0, b]$  is inflated by a constant factor, where currently a factor of 4 is used.

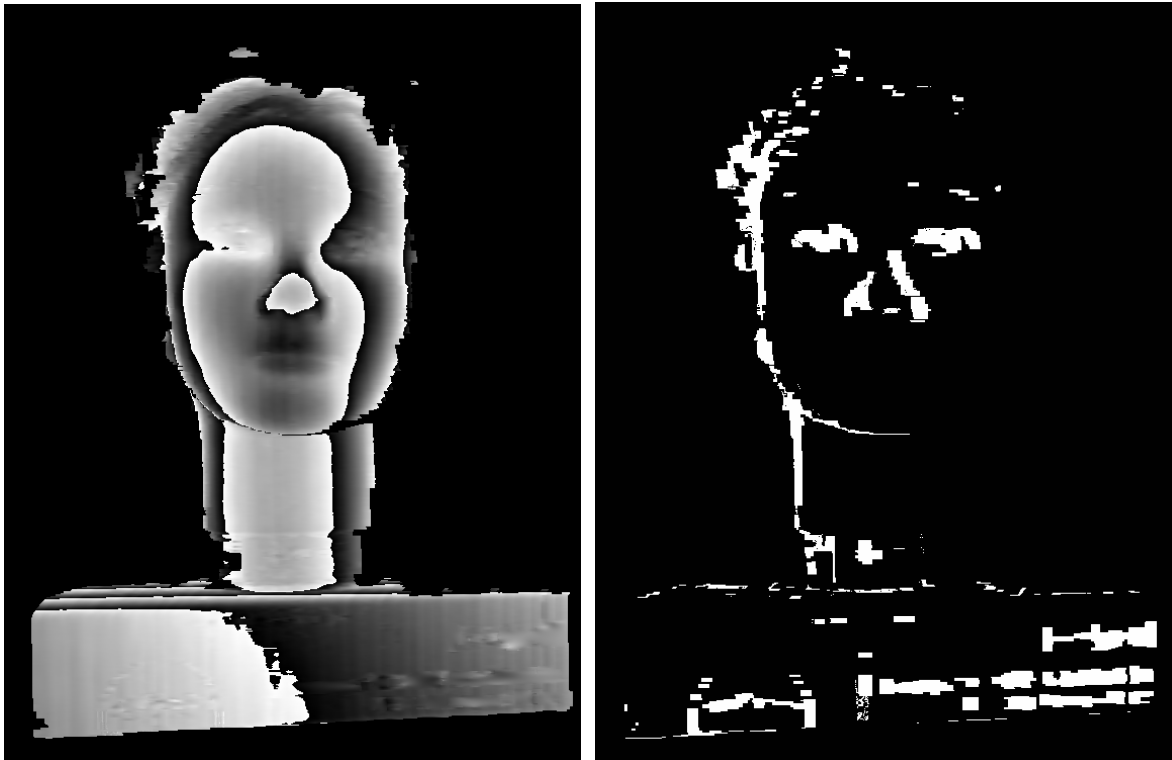


Figure 45: Final depth map of the right camera (left) – Improvement over the depth map computed by the coded light step as shown on the right side of figure 43 (right).

The dense correspondence step does not result in a major improvement with the components used in this work because in their case the gaps between edge segments are typically only about 2 to 4 pixels wide. It does, however, open up a way to efficiently exploit cameras with a high resolution such as today's affordable megapixel consumer cameras. With them, first a sub-sampled image is processed as usual; given these results, the dense correspondence is established using the original image of much higher resolution. This approach represents an efficient method to taking advantage of the current trend to ever higher camera resolutions, especially if no slide with a resolution that directly exploits the capabilities of such camera is available.

Figure 45 illustrates the output of the system after the stereo vision step; respectively, it highlights the points for which the stereo step could obtain a range value while the coded light could not. Clearly the improvement over the coded light step is rather small, a result that can be considered fairly representative with simple scenes such as human heads. One underlying reason is that the stereo algorithm is currently configured very conservatively in order not to introduce false matches. However, the additionally acquired range values mostly refer to geometrically characteristic parts of the scene surface such as its singularities in 3D space. Obtaining them might be crucial for certain tasks; it depends on the application whether this gain is worth the additional hardware requirements and increased computation time that accompanies the stereo vision step.

## 4.5 Optional System Components

### 4.5.1 Scene Reflectance Compensation

The objective of the scene reflectance compensation or reflection normalization step is to remove the effect of the scene's reflection properties on the pattern image. This effect is often significant: e.g. a red scene reflects incoming white light as red (see e.g. figure 36) and abrupt changes in the surface orientation introduce unwanted edges into the pattern image. After the compensation the pattern image should appear as if the scene itself had had a uniform and spectrally constant reflectivity function, i.e. each pixel is supposed to have the color and intensity proportional to the ray of projection illuminating the imaged scene patch. The compensation step needs an image taken under an all-white illumination (in the following called *reference image*) to determine the reflectance properties of each imaged scene patch. With other words, it requires a projector capable of switching between two distinct projection patterns and a scene that is approximately static between the acquisitions of the two images. It is for that reason an optional component of the proposed approach. This section builds on results obtained by Caspi et al. [1998] to solve a related problem with their approach to time-sequential color coded light using several distinct projection slides.

Since almost all off-the-shelf color cameras use RGB-filters, we specify colors in the following within the RGB color space introduced in chapter 2, where we normalize the components of RGB tristimulus vectors to the interval  $[0, 1]$ . As before, we employ the subscripts c and p to distinguish between camera and projector-related quantities, respectively to express that both the projector and the camera implicitly use their device-dependent approximations of the CIE RGB color space.

Let's assume a constant projection pattern; such a pattern can be described via a single vector of the  $RGB_p$  color space. We represent the spectral irradiance of a given scene patch S resulting from the pattern  $I_{pl}$  whose color is the  $l$ th unit vector of the  $RGB_p$  as  $E_{pl}(\lambda)$ . Then the irradiance with the pattern  $(0, \dots, x_l, \dots, 0)$ ,  $0 \leq x_l \leq 1$  is proportional to  $E_{pl}(\lambda)$ . Ideally, this relationship is linear, i.e. the proportionality factor is  $x_l$ , but in general this will not be the case. We consequently introduce for each channel a monotonous function  $h_l(x)$  that maps the interval  $[0, 1]$  onto itself; it allows describing the spectral irradiance radiated on S with the pattern  $(0, \dots, x_l, \dots, 0)$  as the product of  $h_l(x_l)$  and  $E_{pl}(\lambda)$ . We consequently approximate the irradiance  $E_p$  given the projection color  $(x_r, x_g, x_b)$  as:

$$E_p(\lambda, x_r, x_g, x_b) = \sum_{l=r,g,b} h_l(x_l) \cdot E_{pl}(\lambda) \quad (71)$$

As discussed in chapter 2, an image value is proportional to the radiance of the imaged surface patch S into the solid angle of the lens. With coded light, we model this radiance as reflection of the projector illumination  $E_p$  and the background illumination  $E_0$  radiated on S. Substituting this into the sensor response equation 21 yields for the channel image  $I_m$  the expression:

$$I_m = k \cdot \int s_m(\lambda) \cdot r(\lambda) \cdot \left( \sum_{l=r,g,b} h_l(x_l) \cdot E_{pl}(\lambda) + E_0(\lambda) \right) d\lambda \quad (72)$$

where  $r(\lambda)$  describes the normalized, wavelength-dependent reflectivity of the imaged scene patch. Caspi et al. [1998] propose to assume the reflectivity  $r$  to be constant within the support of the spectral responsivity of each camera channel filter. Doing so yields three corresponding constants  $r_r$ ,  $r_g$  and  $r_b$ . With them, equation 72 can be rewritten as:

$$I_m = \sum_{l=r,g,b} h_l(x_l) \cdot r_l \cdot k \int s_m(\lambda) \cdot E_{pl}(\lambda) d\lambda + k \int s_m(\lambda) \cdot r(\lambda) \cdot E_0(\lambda) d\lambda \quad (73)$$

Each integral of the left sum is independent of the projection pattern and the reflectivity of S; it can consequently be determined via an off-line color calibration: For the patterns  $I_{pr}$ ,  $I_{pg}$  and  $I_{pb}$ , a scene



with known reflectivity, e.g. a gray level chart, is imaged. With each pattern  $I_{pl}$ , the expression  $h_l(x_l)$  is nonzero only if  $l = l'$ . If there is no background illumination during calibration, all but one integral vanishes from equation 73. The equation can then be solved for this integral, its only remaining unknown. This process yields 9 constants  $a_{ij}$  that describe the coupling between a certain projected color and the response of a certain color channel of the camera. With these constants and ignoring the background light  $E_0$  for the moment, the expression for  $I_m$  simplifies to:

$$I_m = \sum_{l=r,g,b} h_l(x_l) \cdot r_l \cdot a_{ml} \quad \text{with } a_{ml} = k \cdot \int S_m(\lambda) \cdot E_{pl}(\lambda) d\lambda \quad (74)$$

Or in matrix form:

$$\begin{pmatrix} I_r \\ I_g \\ I_b \end{pmatrix} = \begin{pmatrix} a_{rr} & a_{rg} & a_{rb} \\ a_{gr} & a_{gg} & a_{gb} \\ a_{br} & a_{bg} & a_{bb} \end{pmatrix} \begin{pmatrix} r_r & 0 & 0 \\ 0 & r_g & 0 \\ 0 & 0 & r_b \end{pmatrix} \begin{pmatrix} h_r(x_r) \\ h_g(x_g) \\ h_b(x_b) \end{pmatrix} \quad (75)$$

The matrix  $\mathbf{A} = (a_{ij})$  is approximately diagonal since  $a_{rr}$ ,  $a_{gg}$ , and  $a_{bb}$  are much greater than its other entries for any properly color-balanced camera - projector pair. It is therefore invertible.

Let's assume the color of the projection pattern is unknown. Given the pattern image and the matrix  $\mathbf{A}$ , the only unknown factors of equation 75 are the pattern color and the reflectivity constants, yielding a total of six unknown factors. A single color image of the scene such as the pattern image gives only three equations; in the general case, the resulting system of equations is indeterminate and has no unique solution. This is the principal dilemma of approaches to coded light that are based on a single pattern image and need to know the projected color; even in a perfect environment without any uncontrolled illumination they still have to make assumptions regarding the reflectivity of the scene. They are therefore unable to produce acceptable results with scenes that significantly deviate from these assumptions.

So to determine the projected color, a second image is needed. Taken with known constant illumination, it yields another three equations. The best choice for this illumination is a uniform white pattern, as then each  $h_l(x_l)$  is known to take on a value of 1, and equation 75 simplifies to:

$$\begin{pmatrix} I_r \\ I_g \\ I_b \end{pmatrix} = \mathbf{A} \begin{pmatrix} r_r & 0 & 0 \\ 0 & r_g & 0 \\ 0 & 0 & r_b \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \mathbf{A} \begin{pmatrix} r_r \\ r_g \\ r_b \end{pmatrix} \Rightarrow \begin{pmatrix} r_r \\ r_g \\ r_b \end{pmatrix} = \mathbf{A}^{-1} \begin{pmatrix} I_r \\ I_g \\ I_b \end{pmatrix} \quad (76)$$

Equation 76 can be solved for the reflectivity constants of an imaged scene patch given its  $RGB_c$  tristimulus vector. The constants are substituted into equation 75 formulated for the image with the unknown projection color. If all three constants are nonzero, the values of  $h_r$ ,  $h_g$  and  $h_b$  can be determined for  $S$ . In matrix form, this can be expressed as follows:

$$\begin{pmatrix} h_r(x_r) \\ h_g(x_g) \\ h_b(x_b) \end{pmatrix} = \begin{pmatrix} r_r & 0 & 0 \\ 0 & r_g & 0 \\ 0 & 0 & r_b \end{pmatrix}^{-1} \mathbf{A}^{-1} \begin{pmatrix} I_r \\ I_g \\ I_b \end{pmatrix} \quad (77)$$

Since the functions  $h_r$ ,  $h_g$  and  $h_b$  are typically strictly monotonous, their inverses exist and the sought-after  $RGB_p$  value of the pattern can be determined. This value does usually not correspond to an actually projected one due to noise. To estimate the projected value, the computed has to be mapped to one of the possible  $RGB$ -values, e.g. using a minimal-distance operator. If only two levels per  $RGB$ -channel are used for the projected light, as with the approach introduced in this chapter, the minimal distance operator simplifies to a threshold operation.

The outlined approach assumes the ambient light can be neglected, i.e.  $E_0 \ll E_p$ , within the exposure time and the support of the camera channels. If it cannot be neglected, the effect of the background illumination can be determined by acquiring a separate image of the scene illuminated by  $I_0$  only and by subtracting it from the reference and pattern image. However,  $I_0$  has to be approximately time-invariant and the scene static within the time span needed to acquire these images.

The matrix  $\mathbf{A}$  and the functions  $h_i$  are determined via calibration routines. Clearly the actual entries of  $\mathbf{A}$  depend on the camera (settings), the projector (settings) and the patch  $S$ . This is unproblematic as long as the light projected on a surface spot other than  $S$  or with changed settings has a spectral power distribution that differs from the calibrated one by a constant factor only. In that case, the “correct” matrix  $\mathbf{A}'$  would be identical to  $\mathbf{A}$  up to a constant factor, i.e.  $\mathbf{A}' = k \mathbf{A}$ . This factor consequently cancels out in equation 77 because the inverse reflectivity matrix is implicitly scaled by  $k$  while the matrix  $\mathbf{A}^{-1}$  is scaled by  $1/k$ . It consequently suffices to determine the matrix  $\mathbf{A}$  only once; it can then be used for any surface patch of an arbitrary scene, respectively with any geometric setup and independent of camera settings such as the shutter time.

The above results lead to the following overall approach to the task of scene reflectance compensation: First, the camera-projector coupling matrix  $\mathbf{A}$  is determined off-line. To apply the compensation step on-line, two images are acquired, one with the encoded, one with an all white projection pattern. The image resulting from the latter illumination is used to obtain the three reflectivity constants of each imaged scene patch via equation 76. Given the constants, equation 77 and the described further processing is applied to each pixel of the pattern image. In each case, this yields the color of the ray that illuminates the scene patch imaged at the considered pixel. Replacing the value of each pattern image pixel with this color effectively removes the effect of the scene’s reflection properties on the pattern image as intended.

It is important to note that the above approach is based on certain assumptions besides the ones explicitly stated:

- Imaging chain noise such as transport effects, the spatial averaging over the finite CCD array or blurring caused by the lenses can be neglected.
- Projector noise such as a time-varying projection illumination or blurring caused by its lenses can be neglected.
- The conversion of a sensor charge to a digital value is linear.
- Scene dependent noise such as mutual illumination can be neglected.

These assumptions imply certain important limitations of the model and as a result of the scene reflectance compensation approach; e.g. the first two imply – among other things – that both camera and projector have to be focused properly. While this is rather obvious, other consequences are not: the assumption of a temporally constant illumination means that the compensation step cannot be used with the popular DLP-projectors that project the red, green and blue components of a pattern one after the other unless some kind of synchronization mechanism is implemented. Moreover, while a suitable correction table allows linearizing the response of a camera to some extent, effects such as color clipping due to saturation cannot be corrected. This is a significant problem with scenes that exhibit specular reflection and consequently extreme contrast; with them also mutual illumination becomes a non-negligible factor.

Figure 46 shows an example for the scene color compensation step. The topmost image shows the reference image of a color chart taken with an all-white projection pattern. The center image represents the pattern image acquired while the coded light pattern was projected. The bottommost shows the pattern image after compensating the scene’s local reflectance; it can be seen that in this image almost all imaged scene patches have the color of the illuminating projection ray but for a few patches in the blue and the black areas of the chart, i.e. in areas of very low reflectivity.

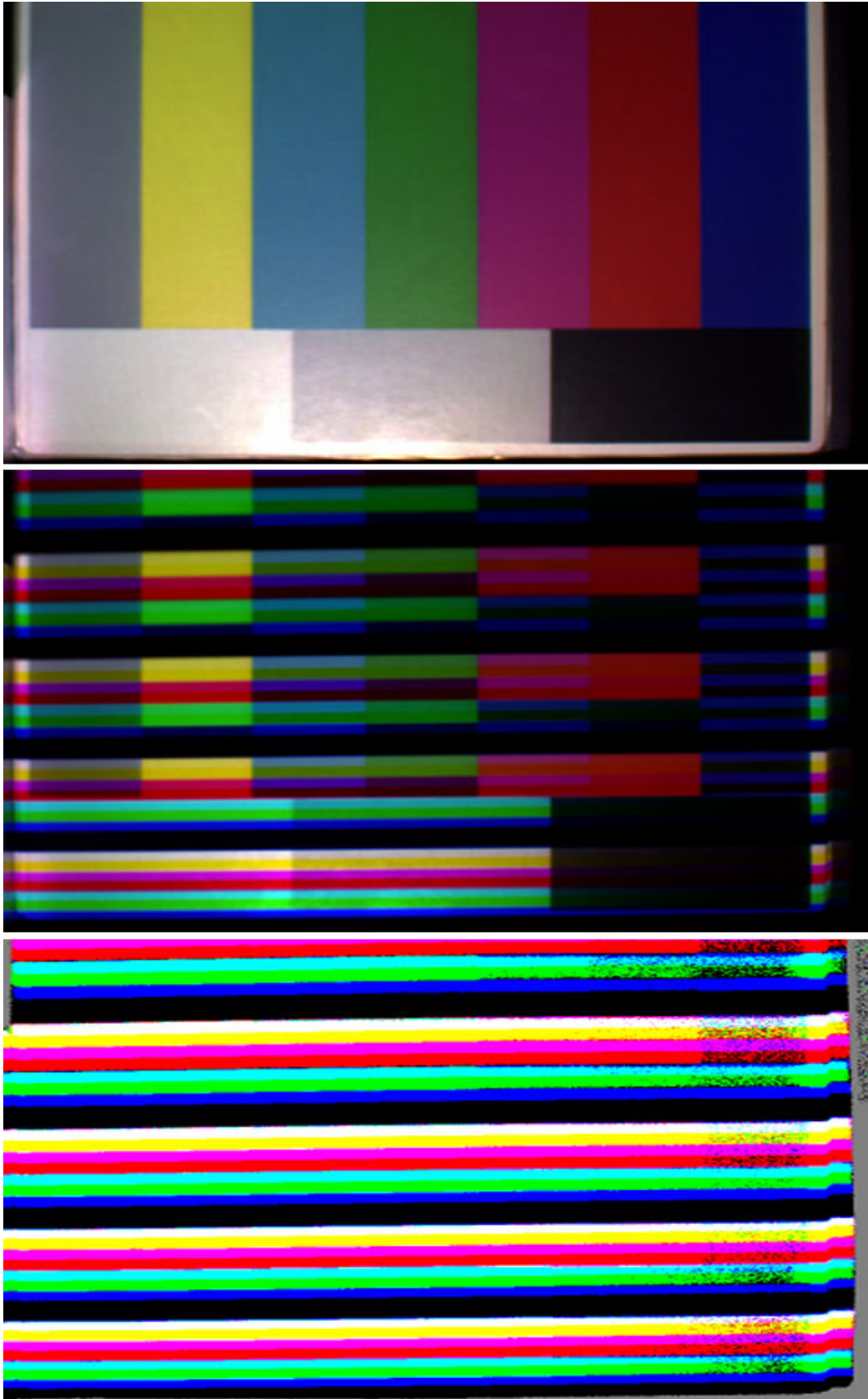


Figure 46: Scene color compensation – the intrinsic scene colors (top), the pattern image (middle) and the pattern image after compensating the scene color (bottom).

### 4.5.2 Autonomous Threshold Optimization

The correct choice of thresholds has a significant impact on the performance of the proposed ranging approach. In this context, the term threshold applies to parameters of the algorithm, e.g. the minimal amount of a local extreme value of the directional derivative to be considered an edge pixel, as well as camera settings such as its exposure time. Certain events require an adjustment of the thresholds, e.g. a major modification of the geometric set-up, the integration of a new component such as a new camera or strongly varying ambient conditions. Ideally, the system should manage its thresholds without any user interaction and in real-time.

The coded light approach of this work allows reliably measuring the quality of a set of thresholds by simply counting the number of pixels that are part of an identified projected color edge segment, in this section called edge pixels. Given the error detecting encoding and processing, it is practically impossible that a non-negligible number of false edge pixels occur. So an algorithm can use this indicator to find an optimal or at least near-optimal choice of thresholds. If camera settings are to be optimized, this requires a static scene; if only algorithm-related thresholds are to be adapted, this can be done given a single representative image.

Optimizing the parameters can be formally described as the global optimization problem of finding the set of parameters that maximizes the number of edge pixels, i.e. of finding  $\max f(\mathbf{x})$ ,  $\mathbf{x} \in \mathbf{X}$ , where  $f$  is the *objective* or *cost function* and  $\mathbf{X}$  is the *parameter space*. Since maximizing  $f(\mathbf{x})$  is equivalent to minimizing  $-f(\mathbf{x})$ , the task is in principle a classical optimization problem quite similar to the ones discussed before, e.g. in the context of shape-from-motion or static stereo vision. So all in all, the task at hand is straightforward; the only non-trivial aspect is the choice of the optimization method. In this context, the following aspects need to be considered:

- **Integral (Co-)Domain:** All parameters are integers.
- **Multidimensional (Co-)Domain:** The parameter space is  $n$ -dimensional, where  $n$  is typically somewhere between 5 and 8.
- **Constrained Optimization:** The thresholds cannot take on every value: they are all bracketed by a minimum and a maximum. In combination with the previous aspect, this implies a finite parameter space.
- **Non-Linearity:** Clearly  $f(\mathbf{x})$  is a non-linear function.
- **Efficiency:** As an algorithm needs to perform a partial evaluation of one or several images for each evaluation of the cost function, the computing time is significant. At the same time, given the only 5 to 8 dimensions of the parameter space, storage space is irrelevant.
- **Near-Optimal Solution Acceptable:** It is not necessary to find the global minimum; it suffices to find a local one where the cost function is close to its optimal value. Such a local minimum does not lead to any errors as it would in the case of stereo vision or camera calibration, only to a slightly suboptimal system performance.

The integral (co-)domain implies derivatives are not available, respectively only coarse approximations of the derivative are obtainable. For this reason, most of the classical optimization methods, e.g. gradient-based ones, are not appropriate. Instead, we choose the well-known non-deterministic method of simulated annealing (see e.g. [Press et al. 2002]) well suited for combinatorial minimization over a discrete parameter space. The annealing schedule is chosen according to the time constraints; at system start-up, when optimization is allowed to take up several seconds, the cooling is slow in accordance with the principle of simulated annealing. During operation, when optimization needs to be efficient, the cooling is much faster, even though this takes away from the strength of simulated annealing to some extent. Typically this is no problem as with a running system the results of the previous optimization tend to represent a very good initial guess.

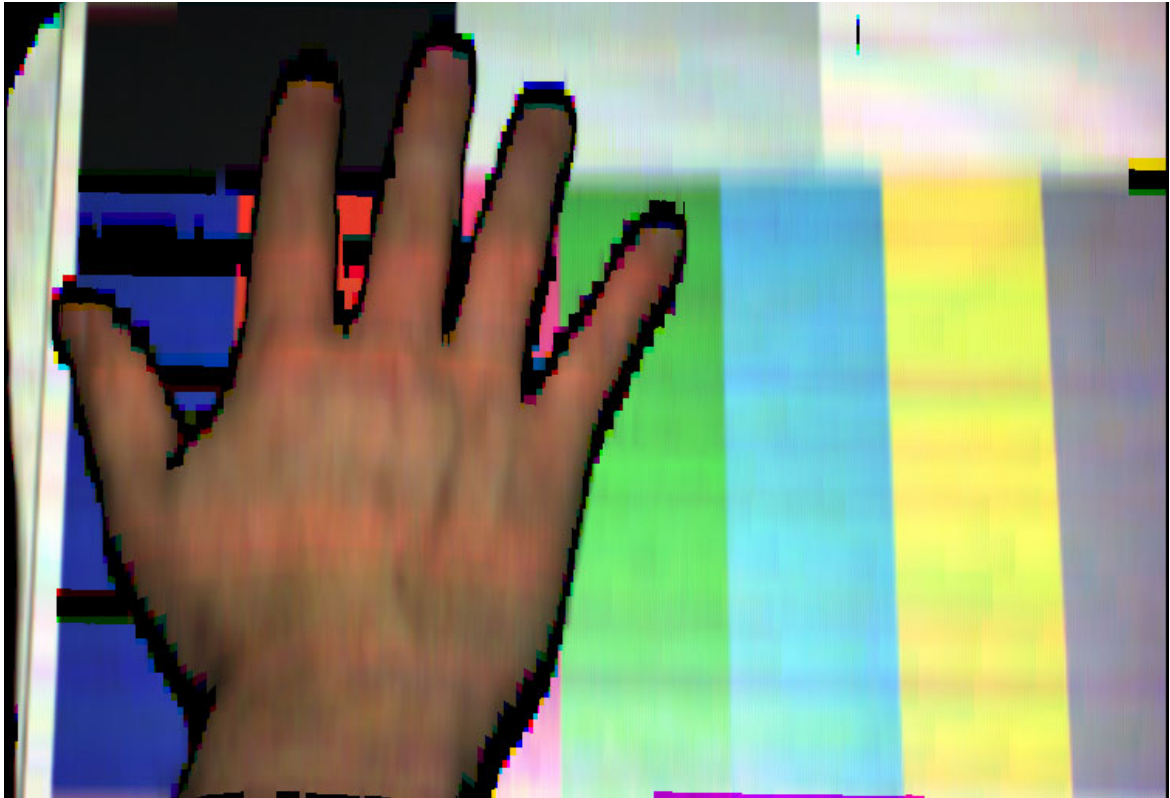


Figure 47: Scene color estimated from the pattern image shown in figure 36.

### 4.5.3 Scene Color Estimation

During the coded light step a color image of the scene is acquired, but one that is rather unsuited for further processing due to the irregular illumination. For the part of the scene for which the identification problem could be solved, the spectral composition of the projected illumination is known. This section discusses in as far it is possible to compute the local reflectivity of an imaged patch under these circumstances, i.e. to reconstruct a color image of the scene that appears as if the illumination had been uniformly white.

As discussed in chapter 2 as well as in section 4.5.1, the value of an image pixel  $I$  is proportional to the radiance of the imaged surface patch  $S$  into the solid angle of the lens. With structured light, this radiance consists of the reflection of the projector illumination  $E_p$  on  $S$  and the background illumination  $E_0$  radiated on  $S$  by any other light sources present. Substituting this into the standard sensor response equation 21 yields the following equation for the value of a channel image  $I_l$  (e.g. the red-component):

$$I_m = k \int_{-\infty}^{+\infty} s_m(\lambda) \cdot r(\lambda) \cdot (E_p(\lambda) + E_0(\lambda)) d\lambda \quad (78)$$

where  $k$  is a constant,  $s_m(\lambda)$  represents the spectral responsivity of the considered channel and  $r(\lambda)$  the unknown spectral reflectance function of the imaged scene patch, in the following informally called scene color.

Given equation 78 and a correctly decoded subpattern, the projector illumination  $E_p(\lambda)$  is known for  $S$ . The scene color  $r(\lambda)$ , actually a function of the wavelength, is modeled as a constant for a given pixel over the support of each channel filter. With a RGB camera as used in this work,  $r(\lambda)$  is again split up into three scene patch, respectively pixel-dependent constants  $r_r$ ,  $r_g$ , and  $r_b$ . Assuming an additive camera response, equation 78 can then be rewritten as:

$$I_m = r_m \cdot \int_{-\infty}^{+\infty} s_m(\lambda) \cdot E_p(\lambda) d\lambda + r_m \cdot \int_{-\infty}^{+\infty} s_m(\lambda) \cdot E_0(\lambda) d\lambda \quad (79)$$

The first integral is now scene independent; for a given camera, it depends on the projector illumination only. Consequently, the possible values of the integral can be determined via an off-line color calibration similar to the one described in section 4.5.1. For each of the eight distinct values of  $E_p$ , a scene with known reflectivity is imaged. Provided there is no background illumination during calibration, the second integral vanishes and equation 79 can be solved for the first integral. With a RGB camera, this process yields 24 constants  $k_{ml}$  that describe the coupling between a certain projected color and the response of a certain color channel of the camera. These constants are normalized by dividing them through the largest occurring constant (which is certainly nonzero). Then all constants are known to have a value between 0 and 1.

Given a pattern image pixel and a patch with known projector illumination, equation 79 is solved again, this time for the unknown scene reflectivity constant  $r_l$ :

$$r_m = \frac{I_m}{k_{ml} + \int_{-\infty}^{+\infty} s_m(\lambda) \cdot E_0(\lambda) d\lambda} \quad (80)$$

Unfortunately, this equation contains a second unknown, namely the channel response to the background illumination. Simply ignoring it distorts the estimated scene color. Clearly the disturbing influence of the background illumination is the stronger, the smaller the constant  $k_{ml}$ . For this reason, the scene color is only estimated if the constant  $k_{ml}$  is large ( $\geq 0.5$ ) for a given projection color; in that case the remaining integral of equation 80 is simply dropped. For example, while projected green will cause a strong response in the green channel ( $k_{gg}$  will be large), the responses of the red and blue channel to green will be weak ( $k_{gr}$  and  $k_{gb}$  will be small, i.e.  $< 0.5$ ). So for projected green, only the green component of the scene color is estimated; the remaining constants are interpolated from adjacent stripes or squares. This approach effectively overcomes the problem of background illumination as well as that of noise, at the cost of a low sampling frequency.

A second relevant aspect is the blurring introduced by the projector lenses. For instance, a white stripe between two blue stripes has a slightly bluish hue while one between two red stripes tends to have a reddish tint. For this reason, a correction factor is introduced that reflects the color of the pattern neighborhood in combination with a certain typical degree of blurring of the projected colors. Each estimated reflectivity constant  $r_l$  is multiplied by a corresponding correction factor.

Figure 47 shows that the approach of this section yields a good estimate of the low frequency scene texture while the high-frequency information is lost in the direction orthogonal to the stripes.

It is important to note that the above problem has in principle already been solved in a more general manner in section 4.5.1. Equation 75 would allow to directly solve for the scene color  $(r_g \ r_b \ r_r)^T$  given the known color of the illuminating ray of projection and the resulting camera response for the imaged scene patch. However, it turns out that the above more specific approach has several advantages over the general one: by explicitly calibrating the response for the eight pattern colors rather than deriving them from a theoretic model avoids the inaccuracies inevitably introduced by such a model. Also equation 80 is more robust against noise and background illumination than the matrix-equation approach of chapter 4.5.1.

## 5 Evaluation of a Prototype System

This chapter analyzes and evaluates a prototype system based on the proposed ranging approach with respect to various aspects. Its focus is on accuracy as the most important aspect of a range image acquisition system; with a triangulation-based system, it is also the most complicated aspect because the error in the obtained 3D coordinates depends on a large number of parameters such as the spatial position and orientation of both camera and projector relative to the scene. To deal with the resulting complexity, we introduce in the following a parameterized model of such a system (5.1). Next, we define certain important accuracy-related terms, discuss which factors cause the measurement uncertainty in the first place, and establish – quantitatively as well as qualitatively – the relationship between the error caused by these factors and the choice of parameters (5.2). We then verify through a number of experiments that the resulting model predicts the actual range measurement error reasonably well (5.3). Finally, we experimentally determine the prototype system’s frame rate (5.4) and its ability to acquire range data of arbitrary scenes (5.5).

With respect to the distinct algorithmic components and configurations of the proposed approach, the focus of the evaluation is on the coded light step (based on a stripe pattern); this is motivated by

- the circumstance that the coded light step typically acquires about 95% of the total range values (ignoring the intra-stripe interpolation of the stereo step)
- the fact that extensive polishing, without which any implementation of a very complex algorithm is off by a large constant factor from a truly representative performance, has only been applied to the coded light step due to lack of time
- the intention to keep the scope of the evaluation within reasonable limits

Unless stated otherwise, the results of this section are obtained with the following set-up similar to the one shown in figure 48: We use Basler 302fc single-chip Bayer-Pattern CCD RGB cameras to acquire color images of size 780 by 580 pixels and the digital IEEE 1394a (Firewire) bus and interface to transfer them to a PC. Bayer-decoding of the camera-supplied data to RGB data and all other processing steps are done in software. The implementation runs on an off-the-shelf PC with a Pentium IV 2.4 Ghz processor (for the speed test, also one with a 3.2 GHZ processor is used) based on the Windows XP operating system without any special hardware. An Epson LCD 710 multimedia projector with a native resolution of 1024 by 768 pixels projects the pattern on the scene.

The remaining parameters of the system are as follows: The camera’s lens is a standard TV lens. Its focal length is approximately 12.5 mm. A camera pixel is square with a side length of about 0.008 mm. The projector has roughly the same ratio of focal length to pixel size, which permits using the same values for focal length and pixel size for both camera and projector. The projector is located at (300, 0, 0) and rotated by a convergence angle of ca. 20° around the y axis towards the left camera (all values specified in the millimeter-based coordinate system of the left camera).



Figure 48: Exemplary prototype system based on the coded light step of the proposed ranging approach. A similar set-up – that is one without the housing to allow an arbitrary choice of the baseline and the integration of a second camera – is used for the experiments of this section. The dimensions of the portable system shown are about 0.45 m x 0.25 m x 0.1 m.



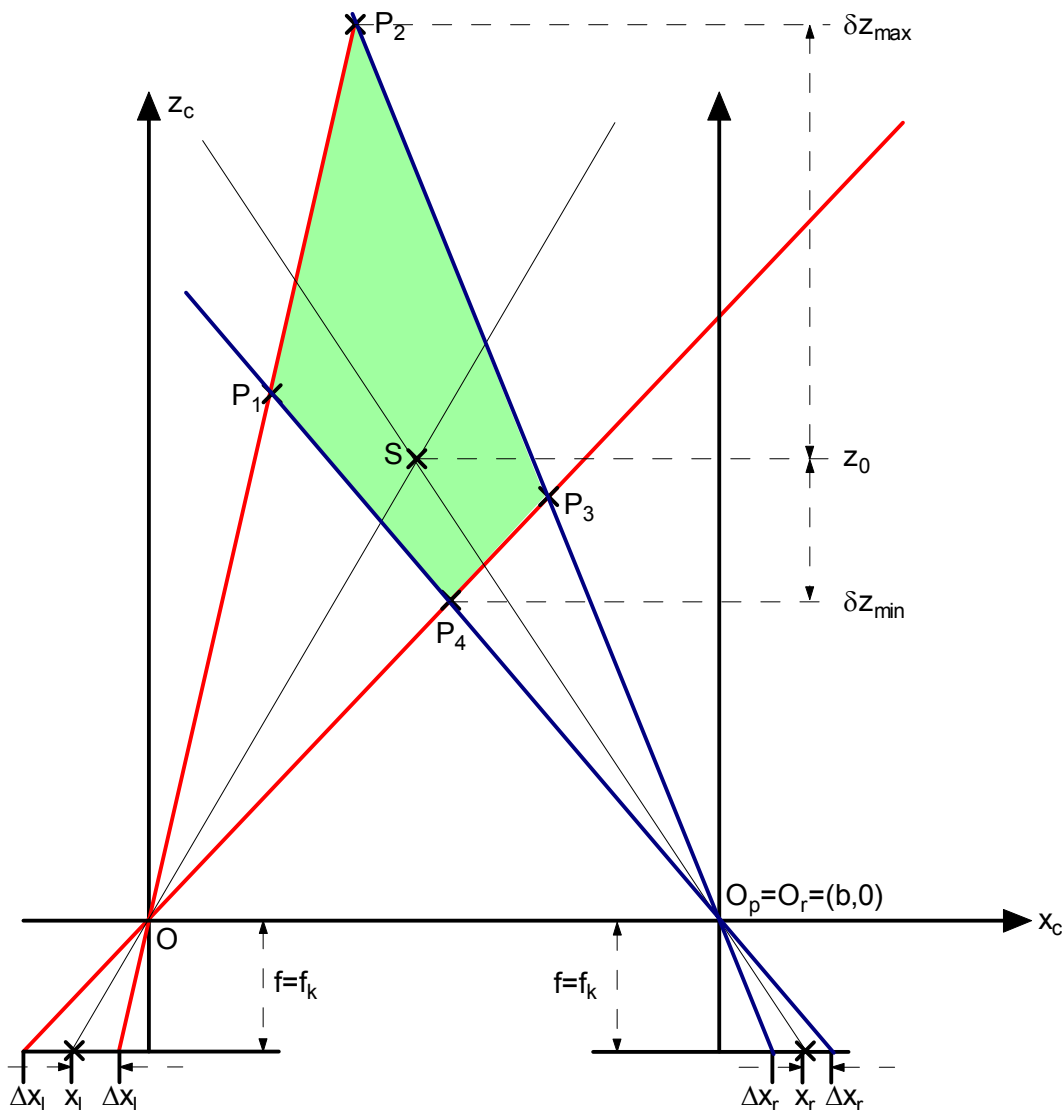


Figure 49: Parameterized model of a structured light system – the special case of parallel optical axes of camera and projector (equivalent to the standard geometry of stereo vision). The figure shows an arbitrary  $XZ_c$ -plane slice through the 3D space.

For comprehensibility, we focus in the following nominally on the case of a structured light system, e.g. by calling one component camera and the other projector. It is important to note that the subsequent sections are nevertheless on triangulation systems in general: for instance, by simply considering the projector as a second camera, we can directly apply all results to the case of stereo vision (unless explicitly stated otherwise).

## 5.1 A Parameterized Model of a Triangulation System

In this section, we develop a parameterized model of a triangulation system, respectively structured light system. The following realistic assumptions are made regarding its geometric set-up and components:

- The  $y$ -axes of camera and projector are parallel. Simple geometric considerations show that this type of set-up is optimal with respect to ranging accuracy. Also, it can be realized in practice up to an imprecision that is negligible for the purpose of an error analysis.

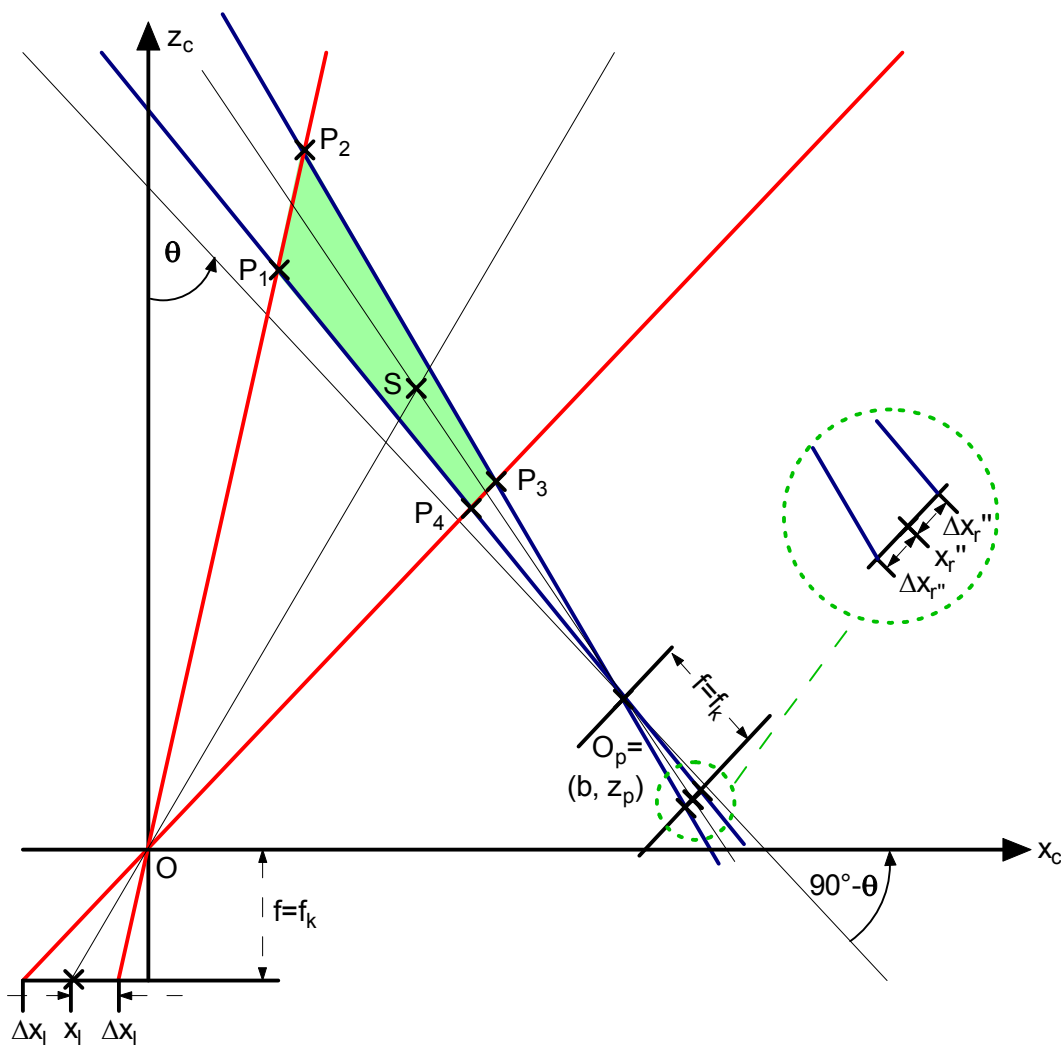


Figure 50: Parameterized model of a structured light system – the general case of non-parallel optical axes of camera and projector that intersect under an angle  $\theta > 0$ .

- The pinhole camera model is a close approximation to the behavior of the lens camera and the lens projector. This implies for instance that imaging aberrations such as radial distortion can be ignored for the accuracy analysis; e.g. the basis for this specific assumption is that such aberrations can be corrected computationally or are negligible with suitable optical components.
- The (effective) focal length  $f = f_k$  is the same for both camera and projector. This assumption is unrealistic, but also unproblematic because for a given system we can e.g. rescale the camera image to simulate the special case of its focal length being equal to that of the projector.

We use in the following exclusively the camera's coordinate system  $XYZ_c$  as the model's coordinate system seeing that it is the most intuitive choice. Respectively, with a standard geometry set-up, we use a version of the camera coordinate system that is translated by  $(b/2, 0, 0)$  in camera coordinates, namely the cyclopean coordinate system introduced in section 3.3.9. As the y-axes of projector and camera are parallel and we ignore radial distortion, we are able to limit our model to an arbitrary  $XZ_c$ -plane slice through the 3D space as shown in figures 49 and 50 (but for an analysis of the error in the y-coordinates). Within this plane slice, the projector has its optical center  $O_p$  at  $(b, z_p)$ . It illuminates the considered scene patch  $S$  located at  $(x_0, z_0)$  in the  $XZ_c$ -plane, respectively at  $(x_0, y_0, z_0)$  in 3D space, with the light plane of projector slide plane coordinate  $x_r$ .  $S$  is imaged at the pixel  $x_l$ .

Figure 49 shows a special case of this set-up: the optical axes of camera and projector are parallel, and the projector is located at  $(b, 0)$ , i.e. just shifted along the x-axis of the camera coordinate system relative to the camera. This special case corresponds to the standard geometry of stereo vision.

In the general case shown in figure 50, the two optical axes are no longer parallel, but intersect at an angle  $\theta$  (*convergence* or *triangulation angle*),  $0^\circ \leq \theta \leq 90^\circ$ . Furthermore, the projector is located at an arbitrary position  $(b, z_p)$ , i.e. it no longer resides on the camera's x axis. Close-range triangulation systems generally need to be set up with a significantly nonzero convergence angle to obtain a usable working space. With other words, unlike most of the literature on the accuracy of stereo vision, we cannot limit ourselves to the simple case of a standard geometry set-up.

## 5.2 Theoretical Analysis of the Measurement Error with Triangulation

### 5.2.1 Definition of Basic Terms

As any real-world measuring device, a rangefinder acquires data that deviates to some extent from the correct value, the so-called ground truth. This deviation is also called *measurement uncertainty*, *measurement error* or sometimes somewhat misleadingly *accuracy* (the term inaccuracy would seem to be more appropriate). To be able to actually use a range acquisition system for real-world applications, its measurement uncertainty has to be known at least approximately.

Formally, the measurement of the 3D coordinates of a given scene patch with ground-truth coordinates  $(x_0, y_0, z_0)$  represents a continuous, real-valued and three-dimensional random vector  $(x, y, z)$ . Consequently, the measurement error  $\delta$  is a continuous, real-valued and three-dimensional random vector  $(\delta x, \delta y, \delta z)$  itself, whose definition follows as:

$$\delta = (\delta x, \delta y, \delta z) = (x, y, z) - (x_0, y_0, z_0) \quad (81)$$

The measurement error can be broken down into a systematic and a statistic component. Its expected value represents the *systematic error* of the measurement; its standard deviation is sometimes called the *statistical error* of the measurement. In contrast to the systematic error, the statistical can be averaged out by repeating a measurement over and over again. In this context, Jähne [2002] defines the *depth resolution* as the statistical error of the  $z_c$ -coordinate, i.e. of the depth measurement, because one can argue that this error determines the minimal resolvable depth difference. It is important to note that this works assumes both errors are constant during the experiment and not e.g. random processes over a variable of time or temperature.

It follows from the fact that the measurement error is a continuous random vector that the objective of an error analysis is to determine its probability density function (pdf), or at least its mean value and other key distribution parameters such as its standard deviation. Even for an unknown type of probability density function, the latter could be used to derive bounds on the probability of certain measurement errors (e.g. using the Chebyshev inequality). There are, however, two factors that make this task rather difficult in practice.

The first is that multivariate random variables with their joint probability density functions are difficult to handle and not particularly easy to interpret, especially if their variables are dependent as they are in our case. To avoid having to deal with them, several one-dimensional alternatives suggest themselves. Two examples are the *absolute error*  $\delta_{abs}$  as the vector norm of the measurement error

$$\delta_{abs} = |(x, y, z) - (x_0, y_0, z_0)| \quad (82)$$

and the *relative (total) error*  $\delta_{rel}$  as the ratio of the absolute error to the norm of the ground truth vector  $(x_0, y_0, z_0)$ :

$$\delta_{rel} = \frac{|(x, y, z) - (x_0, y_0, z_0)|}{|(x_0, y_0, z_0)|} \text{ for } (x_0, y_0, z_0) \neq (0, 0, 0) \quad (83)$$

Suitable norms are e.g. the L2, which we use in the following, or the L-infinity norm of the  $\mathcal{R}^3$ . Clearly both the absolute and the relative error represent continuous real-valued random variables with an intuitive interpretation.

A further alternative is to consider the error with respect to a selected coordinate only. We define the continuous real-valued random variable  $\delta z$  as the error in the z coordinate (the depth or range error); the random variables  $\delta x$  and  $\delta y$  be defined accordingly. In the following, we mostly investigate the variable  $\delta z$ . This will be justified a posteriori because we show that the other two variables  $\delta x$  and  $\delta y$  – and consequently the absolute and the relative error as well – depend approximately linearly on  $\delta z$  in our case.

The second complicating factor is that with an area-scan, triangulation-based range acquisition system (and in particular one with a convergent geometric set-up) the accuracy changes significantly over the working space. This conflicts with the intention to keep things simple, i.e. to be able to characterize a range acquisition system with an un-parameterized random vector  $\delta z$  or one that depends on a few key parameters describing the system geometry and components only. There are several strategies to deal with this situation:

- The most direct approach is to minimize the number of parameters as far as possible, but to nevertheless model the measurement error as a random process over a parameter space that (ex- or implicitly) includes the spatial position of the considered scene patch. That is, to treat the error as a mapping of the parameter space to a set of random variables, each of them with its individual pdf. Then, given measured surface coordinates, one can hypothesize that they are reasonably close to the ground truth and accordingly obtain the pdf, statistic error etc. specific to the considered scene patch.
- An alternative is to attempt to derive suitable approximations for certain standard set-ups, which consequently yield magnitude-of rather than precise values.
- A third way is to choose the worst case uncertainty, defined e.g. as the position within the working space for which the standard deviation of  $\delta z$  is maximal, as the representative one; in fact, this approach needs to be taken if a certain application has strict accuracy requirements that need to be guaranteed over the whole working space.

In this chapter, we focus on the first two strategies: we first derive an exact formula for the measurement error and then simplify it via suitable approximations that apply to certain practically relevant set-ups. In any case, before we can set out to determine the accuracy, we have to understand which factors cause the now well-defined measurement error in the first place.

### 5.2.2 Causes of Inaccuracy

In our case, the following factors contribute to the uncertainty of range measurements:

- **Localization, Quantization, Image Processing or Image Resolution Error:** An algorithm cannot locate light planes (here the border between two projected stripes or squares) with infinite accuracy on the image plane, but only with a certain localization error  $\Delta x_l$  (specified in camera coordinate system units, i.e. in our case in millimeter, not in pixel units), even if it uses sub-pixel arithmetic. See figures 49 and 50 for an illustration of this definition. Among the many reasons for this localization error are the loss of the high frequency information due to the discrete sampling of the scene radiance (sampling theorem), blurring introduced through the camera lens (in particular with today's consumer megapixel cameras with their small sensors in combination with low-cost lenses), and electronic noise occurring within the imaging chain. With the proposed approach, the scene texture can as well contribute to this error.
- **Projection Error:** A light plane emitted by the projector does not correspond to a 2D plane in 3D space due to the non-negligible size of the slide pixels, rather to a cone. As previously discussed, we are able to circumvent this problem to some extent by using only the border between two slide pixels for triangulation as it approximates a light plane. However, even this border is not clearly defined because e.g. the projection slide is not flawless, the projector optics introduce a certain amount of blurring, distortion and in particular chromatic aberration, and the irradiance of the light bulb is not constant, but itself a random variable that changes with the solid angle, over time etc. We can specify the location of this border on the projector's retinal plane only with a certain uncertainty of  $\Delta x_r$ , specified in projector coordinate system units, i.e. in our case in millimeter. See figures 49 and 50 for an illustration of this definition.
- **Modeling Error:** The calculation of 3D coordinates of a scene point is based on two ideal device models, in this work on the lens camera and lens projector model. These models are only an approximation of the real-world devices actually used (or vice versa, the real-world devices are only approximations of the ideal devices their manufactures intended to produce). The unavoidable difference between model and reality causes an error in the range measurement.
- **Calibration Error:** Even without a modeling error, the (e.g. lens camera) model parameters can be determined with limited accuracy only with the known calibration methods since the calibration target is imperfect, the fiducial marks cannot be localized with infinite precision, the algorithm does not find the global minimum of the cost function etc.
- **Misidentification or False Match Error:** A misidentified light plane results in 3D coordinates that are almost arbitrarily off.
- **Other Error Causes:** Several other effects are able to introduce additional errors. E.g. movement within the scene tends to introduce motion blur, especially if the shutter time is non-negligible relative to the speed of movement. Changes in temperature or mechanical strain (including a simple push or somebody walking by) might affect the set-up and introduce a potentially time-varying, aggravating calibration error. This type of error is very relevant for any real-world application, especially as its effect is systematic and can become unlimitedly large. Furthermore, ambient illumination can cause errors, let alone software bugs or operating errors of the user.

In the following analysis, we consider only the projection and localization error. This is motivated by our experience that those two dominate the resulting error over the modeling and calibration error, the misidentification error does practically not occur with the proposed system and the other error causes do not lend themselves to a generic analysis.

### 5.2.3 Exact Formulas for the Measurement Error

We first derive the error in the separate coordinates resulting from a given localization error  $\Delta x_l$  and projection error  $\Delta x_r$ . We start out with the simple parallel set-up as shown in figure 49 and obtain – via equation 48 – the following expression for the depth error  $\delta z$  resulting from these two causes

$$\delta z = \frac{-bf}{x_l + \Delta x_l - x_r - \Delta x_r} - \frac{-bf}{\underbrace{x_l - x_r}_d} = \frac{bf(\Delta x_l - \Delta x_r)}{d \left( d + \underbrace{\Delta x_l - \Delta x_r}_{\Delta x} \right)} \stackrel{*}{\approx} \frac{z^2 \Delta x}{bf} \quad (84)$$

Analogously, we obtain from equation 48 for the error in the measured x and y coordinates,  $\delta x$  and  $\delta y$ , the two expressions:

$$\begin{aligned} \delta x &= \frac{b(x_l \Delta x_r - x_r \Delta x_l)}{d(d + \Delta x_l - \Delta x_r)} \stackrel{*}{\approx} \frac{z^2 (x_l \Delta x_r - x_r \Delta x_l)}{bf^2} \\ \delta y &= \frac{-y_l bf \Delta x}{f d (d + \Delta x)} = \delta z \frac{y_l}{f} \stackrel{*}{\approx} \frac{z^2 \Delta x y_l}{bf^2} \end{aligned} \quad (85)$$

For the above formulas, we use the estimate  $d(d + \Delta x) \approx d^2$ , whose application is marked by the symbol \*. It is in fact a very good approximation in our case: e.g. for the typical set-up used throughout this work, the range values vary between 700 and 1200 mm and the baseline has a size of about 300 mm. As mentioned above, the pixel side length is about 0.008 mm for both camera and projector. Disparity values are consequently in the area of 3.0 mm or greater. A reasonable upper bound for the localization and projection error is about half a pixel, i.e. in camera coordinate units about 0.004 mm. For the minimal disparity value of 3.0 mm, the exact range error resulting from a combined localization error  $\Delta x$  of 0.008 mm is 3.324 mm, the approximated one is 3.333 mm. Only for much more remote scene points, i.e. ones resulting in a disparity of no more than a few pixels, the above approximations would become notably inaccurate. Area-scan SL systems are intrinsically limited to close-range acquisition; consequently such remote points are principally not of interest with them and we may safely apply the above approximation.

To analyze a non-standard set-up as shown in figure 50, we first introduce an imaginary projector located at the same position ( $b, z_p$ ) as the actual projector, but whose optical axis is parallel to the one of the camera, i.e. one that would lead to a convergence angle of  $0^\circ$ . Be  $x_r''$  the slide coordinate of the real rotated projector and  $x_r'$  the one of its imaginary non-rotated counterpart. Be  $(x_{cr}'', z_{cr}'')$  and  $(x_{cr}', z_{cr}')$  the corresponding coordinates in the coordinate system of the actual and of the virtual projector, respectively. We obtain the slide coordinate  $x_r'$  by converting  $(x_{cr}'', z_{cr}'')$  to  $(x_{cr}', z_{cr}')$ , i.e. by rotating  $(x_{cr}'', z_{cr}'')$  by the convergence angle, and by projecting the obtained point  $(x_{cr}', z_{cr}')$  on the retinal plane of the imaginary projector. Mathematically, this is expressed as follows:

$$x_r' = -f \frac{x_{cr}'' \cos \theta - z_{cr}'' \sin \theta}{x_{cr}'' \sin \theta + z_{cr}'' \cos \theta} = -f \frac{-\frac{x_r'' z_{cr}''}{f} \cos \theta - z_{cr}'' \sin \theta}{z_{cr}'} = \frac{z_{cr}''}{z_{cr}'} (x_r'' \cos \theta + f \sin \theta) \quad (86)$$

Alternatively, we obtain the virtual slide coordinate  $x_r'$  by transforming the slide plane position  $(x_r'', -f)$  into coordinates of the imaginary projector, and by again projecting the resulting point on its retinal plane. This step yields a different, but of course equivalent expression for  $x_r'$ , which has the advantage of involving slide coordinates only:

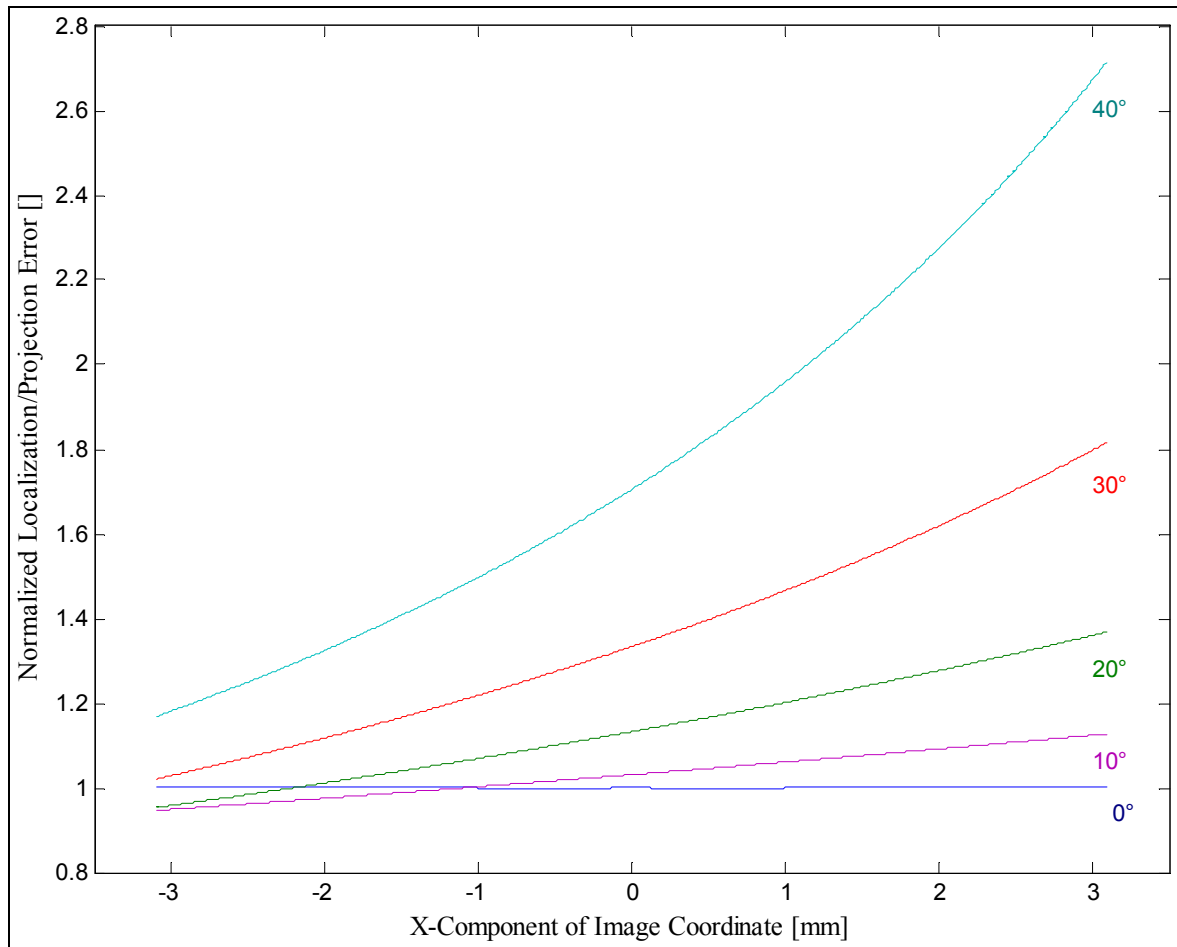


Figure 51: Effect of rotating the projector on the projection error (for a representative triangulation system as used in this work).

$$x'_r = -f \frac{x''_r \cos \theta + f \sin \theta}{x''_r \sin \theta - f \cos \theta} \quad (87)$$

Equation 87 allows computing the non-convergent projection error  $\Delta x'_r$  of the imaginary projector at  $(b, z_p)$  that is equivalent to the error  $\Delta x''_r$  in the rotated slide plane:

$$\Delta x'_r = f \frac{x''_r \cos \theta + f \sin \theta}{x''_r \sin \theta - f \cos \theta} - f \frac{(x''_r + \Delta x''_r) \cos \theta + f \sin \theta}{(x''_r + \Delta x''_r) \sin \theta - f \cos \theta} \quad (88)$$

The term  $(x''_r + \Delta x)$  in the right denominator cannot be approximated by  $x''_r$ , which makes equation 88 rather complex. Nevertheless the equation allows a scene-independent analysis of the effect of rotating the right camera, respectively the projector on the measurement uncertainty. Figure 51 shows the normalized localization, respectively projection error, defined as the ratio  $\Delta x'_r / \Delta x''_r$ , for a set-up as used in this work and for several distinct convergence angles. It can be seen that from a convergence angle of 20° degrees on the effective error is no longer even approximately uniform, but increases with the x-coordinate and reaches up to the 2.7 fold for a convergence angle of 40°. That is, a given projection error of the rotated projector is equivalent to up to the 2.7 fold error of a non-rotated projector. We conclude that in the case of a rotated projector the ranging error no longer depends on the z-coordinate only as with a standard geometry set-up, but on the x slide (and consequently x world) coordinate as well.

All in all, the effect of the rotation on the accuracy is mostly a negative one (the ratio  $\Delta x_r'/\Delta x_r''$  mostly takes on a value  $> 1$ ) that worsens with increasing x-slide coordinate; with a set-up as in figure 50, the measurement uncertainty is accordingly the largest on the left (considering a  $z_c = \text{const.}$  plane in 3D or straight line in 2D space). Simple geometric considerations show that this effect is not particular to our specific set-up, but applies to most practically relevant combinations of projection slide size and focal length. The reason for it is the change in the effective lateral resolution of the rotated projector relative to a non-rotated one.

The above effect implies for example that increasing the baseline by moving the projector along the x-axis does not necessarily have the expected strictly linear positive effect on the ranging accuracy as suggested by equation 84; this is because increasing the basis will typically require a stronger rotation of the projector to obtain a suitable working space, which has the above-mentioned mostly negative impact on the accuracy.

As equation 88 allows reducing the case of a rotated projector to a set-up with a convergence angle of zero, the remaining case is the one of a non-convergent set-up with the projector located at  $(b, z_p)$ , where  $z_p \leq z$  (the case  $z_p > z$  is of virtually no practical interest due to occlusion effects). With such a set-up, we obtain the following two formulas for the relationship between image, respectively slide and camera coordinates:

$$x_l = -\frac{xf}{z} \wedge x_r' = -\frac{x'f}{z'} = -\frac{(x-b)f}{z-z_p} \quad (89)$$

Equation 89 is the equivalent of the standard-geometry equation 47. By resolving it for the unknown  $z$ , we obtain

$$z = \frac{-bf - z_p x_r'}{d'} \quad (90)$$

Equation 90 is the equivalent of equation 48 for a standard geometry set-up. From it, the formula for the range error due to  $\Delta x_l$  and  $\Delta x_r'$  (respectively, via equation 87 the error due to the actual projection error  $\Delta x_r''$  of the rotated and translated projector) is derived as:

$$\delta z = \frac{-bf - z_p x_r' - z_p \Delta x_r'}{x_l + \Delta x_l - x_r' - \Delta x_r'} - \frac{-bf - z_p x_r'}{d'} = \frac{bf \Delta x_r' + z_p x_r' \Delta x_r' - z_p d' \Delta x_r'}{d'(d' + \Delta x_r')} \quad (91)$$

While equation 91 allows numerically computing the range error  $\delta z(x_l, x_r, b, z_p, f, \Delta x_l, \Delta x_r'', \theta)$  from a given localization and projection error, it is no particular help in understanding the relationship between the separate parameters and the accuracy as it is quite difficult to interpret. For this reason, the next section introduces several approximations that help to develop a better understanding of this relationship.



### 5.2.4 Simple Estimates for the Measurement Error of a Structured Light System

To obtain a simpler model of a SL system, this section introduces the approximation that the projection error is negligible ( $\Delta x_r \approx 0$ ). Its application is in the following marked by \*\*. In our experience, this assumption is more or less justified with most projection devices conventionally used for area-scan SL (as opposed to point- or line-scan SL systems that mechanically sweep a ray or plane of light over the scene). It applies for instance to today's multimedia projectors that are manufactured with very high precision despite being consumer products. It typically holds with specially designed filters that occasionally serve as SL projection slides and in particular with custom-made LCD projectors as used with high-end SL systems.

Quantitatively, the assumption  $\Delta x_r \approx 0$  simplifies the expression for the error  $\delta x$  in the x-coordinate given a standard geometry set-up to:

$$\delta x \approx \frac{z^2 (x_l \Delta x_r - x_r \Delta x_l)}{bf^2} \approx \delta z \frac{x_r}{f} \quad (92)$$

Next, it implies we may neglect the loss of lateral resolution that is caused by rotating the projector. For a projector located at  $(b, z_p)$ , the expression for the depth error  $\delta z''$  then simplifies to:

$$\delta z'' = \frac{bf\Delta x' + z_p x_r' \Delta x' - d' z_p \Delta x_r'}{d'(d' + \Delta x_r')} \approx \frac{z^2 \Delta x_l}{z_p x_r' + bf} = \delta z \frac{bf}{z_p x_r' + bf} = \frac{-\delta z}{\frac{z_p}{b} \cdot \frac{x_r'' \cos \theta + f \sin \theta}{x_r'' \sin \theta - f \cos \theta} + 1} \quad (93)$$

where the symbol  $\delta z$  represents the depth error that would result with a corresponding normal geometry set-up.

A geometrical interpretation of the consequence of negligible projection error, respectively an analytical interpretation of equation 93 leads to the following result: As far as the projector is concerned, the slope of the ray of projection illuminating the considered scene patch is the sole factor that determines the measurement inaccuracy; this is because it controls the angle under which line (or rather cone) of view and ray of projection intersect. With increasing ratio  $z_p/b$ , respectively the flatter the slope of the ray is, the smaller is the resulting range error (all other things being equal). This relationship becomes very apparent in figure 52, which shows the error in the depth coordinate resulting from a given localization error for a given scene point and several distinct values of  $z_p$ , respectively of the ratio  $z_p/b$ .

The latter point of view leads to another, even simpler approximate expression for the ranging error of a SL set-up that does not adhere to the standard geometry: let's consider a scene patch whose illuminating ray of projection subtends an angle  $\alpha$  with the camera's z-axis. Given the ray is known to pass through the projector's optical center  $(b, z_p)$ , it follows that it intersects the camera's  $x_c$ -axis at the point  $(b + z_p/\tan(90^\circ - \alpha), 0) = (b + z_p \tan(\alpha), 0)$ . So from a ray-of-projection and consequently accuracy standpoint, the projector might as well be located at  $(b + z_p \tan(\alpha), 0)$ .

On the basis that projector lenses are almost never wide-angle lenses, we approximate the scene-point dependent angle  $\alpha$  with the convergence angle  $\theta$  to obtain an even more general expression. With other words, we hypothesize that a projector located at  $(b, z_p)$  is more or less equivalent to one located at  $(b + z_p \tan(\theta), 0)$ . So we may rephrase that positioning the projector at a nonzero z-coordinate  $z_p$  corresponds to increasing the basis  $b$  by an offset of  $z_p \tan(\theta)/b$ , respectively by a factor of  $1 + z_p \tan(\theta)/b$ .

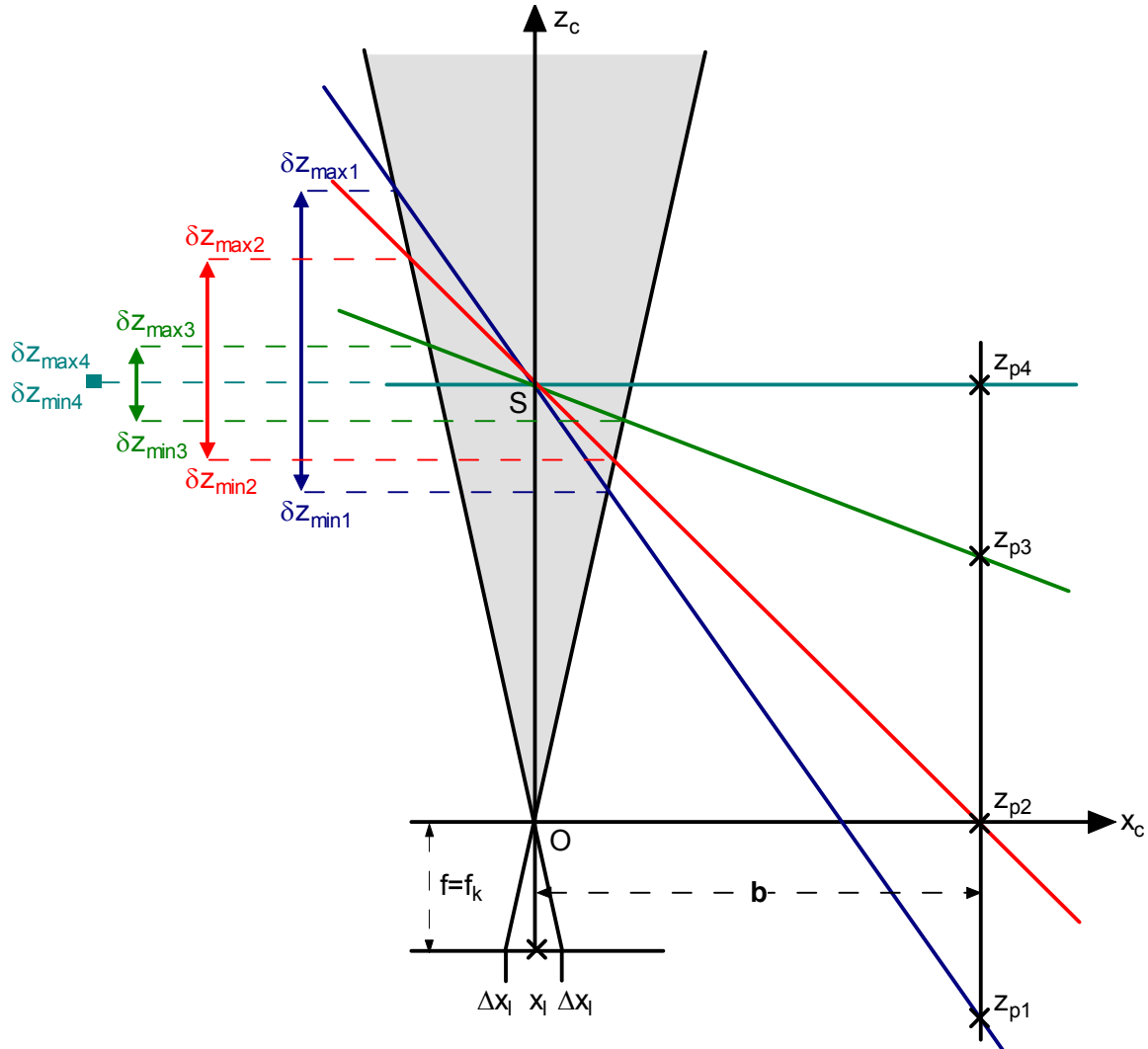


Figure 52: Influence of the ratio of the projector's  $z_p$  to the base length  $b$  on the measurement error for several distinct values of  $z_p$  and consequently of the ratio  $z_p/b$ . It can be clearly seen that – all other things remaining unchanged – the depth error shrinks with growing ratio  $z_p/b$ .

All in all, we obtain the following estimate of the range error  $\delta z''$  with a non-standard SL setup:

$$\delta z'' \approx \frac{z^2 \Delta x}{f(b + z_p \tan \theta)} = \frac{z^2 \Delta x}{bf \left(1 + \frac{z_p \tan \theta}{b}\right)} = \delta z \frac{1}{1 + \frac{z_p \tan \theta}{b}} \quad (94)$$

It contains only the depth of the considered scene patch as working-space related parameter. We have consequently derived an approximation that allows reducing the complex case of a non-standard SL set-up to the straightforward case of a standard geometry SL set-up via a simple correction factor.

### 5.2.5 Stochastic Analysis of the Measurement Error

The previous sections establish the relationship between a certain localization, respectively projection error and the measurement error for any given set-up. Their results allow a straightforward derivation of the probability density function  $f_{\delta z}$  of the depth error from the density of the two er-

rors because the depth error is a monotonous function of the difference of the two (for a constant depth). As commonly accepted throughout the literature on stereo vision (e.g. [Blostein and Huang 1987], [Rodriguez and Aggarwal 1990] and [Chang et al. 1994]), we assume the localization error is approximately uniformly distributed over an interval  $[-dx/2, dx/2]$ . As suggested by the symbol, a typical choice for  $dx$  is the size of a pixel, respective sensor element; or, with sub-pixel arithmetic, a corresponding fraction thereof, sometimes called the *effective pixel size*. From the distribution hypothesis, the probability density function of the localization error follows immediately as  $f(x) = 1/dx$  for this interval, zero otherwise. The expected value of the localization error is consequently zero, its variance  $dx^2/12$ .

For the reason mentioned above, we focus on the case of a structured light system. So we may exploit the assumption of a negligible projection error introduced in the previous section. Then the combined localization and projection error  $\Delta x$  simplifies to the localization error  $\Delta x_r$ , i.e.  $\Delta x = \Delta x_r$ . So given a ground truth of  $z = z_0$ , we obtain for the smallest, respectively largest possible error in the  $z$ -coordinate,  $\delta z_{\min}$  and  $\delta z_{\max}$ , according to the above distribution hypothesis:

$$\delta z_{\min} = -\frac{dx z^2}{2bf}, \quad \delta z_{\max} = +\frac{dx z^2}{2bf} \quad (95)$$

With these two constants, the probability density function of the depth error for a given depth  $z$  follows as the stair function  $f_{\delta z}$ :

$$f_{\delta z}(\delta z) = f_{\Delta x} \left( \frac{bf \delta z}{z^2} \right) \cdot \frac{bf}{z^2} = \frac{bf}{z^2 dx} \quad \text{for } \delta z_{\min} \leq \delta z \leq \delta z_{\max}, \quad 0 \text{ otherwise} \quad (96)$$

The expected value of the depth error is consequently zero, and its variance, respectively standard deviation follows as

$$\text{var}(\delta z) = \frac{(dx)^2 z^4}{12b^2 f^2}, \quad \sigma(\delta z) = \frac{dx z^2}{\sqrt{12}bf} \quad (97)$$

It is now straightforward to determine e.g. the absolute measurement error of a standard-geometry SL system as

$$\delta_{abs} = |(\delta x, \delta y, \delta z)| = \left| \delta z \left( \frac{x_r}{f}, \frac{y_r}{f}, 1 \right) \right| = |\delta z| \sqrt{\frac{x_r^2}{f^2} + \frac{y_r^2}{f^2} + 1} \quad (98)$$

The formula for the relative error follows analogously.

With the approximation of the previous section, the formulas for a non-standard set-up differ only by a constant factor from the above results; we obtain e.g. for the probability density function

$$f_{\delta z''}(\delta z'') = \frac{f(b + z_p \tan \theta)}{z^2 dx} \quad \text{for } \delta z''_{\min} \leq \delta z'' \leq \delta z''_{\max}, \quad 0 \text{ otherwise} \quad (99)$$

and the variance, respectively standard deviation of the depth error

$$\text{var}(\delta z'') = \frac{(dx)^2 z^4}{12f^2 (b + z_p \tan \theta)^2}, \quad \sigma(\delta z'') = \frac{dx z^2}{\sqrt{12}f(b + z_p \tan \theta)} \quad (100)$$

In the following, we use equation 100 to estimate the standard deviation of the depth error of a coded light system given its set-up parameters.

### 5.2.6 Conclusions Regarding the Accuracy – Set-Up Relationship

The results of the previous sections allow a number of general conclusions regarding the relationship between ranging accuracy and set-up of a triangulation-based acquisition system:

- With a typical close-range system set-up, i.e. one of a rather small convergence angle, the error in each separate coordinate is about proportional to the squared depth of the considered scene patch.
- With a typical close-range system set-up, the relative error in each separate coordinate is approximately linearly proportional to the depth of the considered scene patch.
- With a typical close-range system set-up, the error in the z-coordinate dominates the error in the x and in the y coordinate, given the sensor size is usually much smaller than the focal length and consequently the ratio  $x_r/f$ , respectively  $y_r/f$  tends to be much smaller than one. Accordingly, we may approximate the absolute error with the one in the z-coordinate. As the convergence angle becomes large, this relationship is reversed: from a certain point on, the error in the x-coordinate dominates; however, this situation is much less important in practice as corresponding set-ups are rather useless due to occlusion effects.
- In all cases, the error in the depth measurement is proportional to the total localization error. Increasing the relative spatial resolution – by either using a camera of higher resolution or a larger focal length – consequently reduces the depth error roughly linearly.
- The error in the separate coordinates decreases linearly with the length of the baseline; from a certain point and with certain components, the effect of the often necessary stronger rotation of the right camera/projector needs to be considered, which tends to offset the accuracy gain for large baselines to some extent.
- Positioning the projector of a SL system closer to the scene and rotating it towards the camera to an convergence angle of  $\theta$  has approximately the same effect as increasing the baseline by a factor of  $(1 + z_p \tan(\theta)/b)$ .
- With a SL system, the ellipsoid within which the measured value lies within a certain fixed probability (for a given ground truth point in 3D space) tends to have a distinct direction, namely it is oriented along the ray/plane of projection illuminating the considered scene point. So with a typical close-range system set-up, this ellipsoid is oriented along the  $z_c$  axis, and the error in the z-coordinate dominates; with increasing convergence angle this ellipsoid rotates, and from a certain large convergence angle on, the error in the x-coordinate becomes predominant over the one in the z-coordinate. Again, the latter case is rather irrelevant in practice.
- The negligible projection error gives SL an edge over comparable stereo vision systems as with the latter both coordinates  $x_l$  and  $x_r$  are affected by non-negligible uncertainty.

It is important to note that all the above conclusions hold only within certain reasonable limits: of course neither a system with infinite baseline nor a SL system with a convergence angle of  $90^\circ$  will produce error-free depth data. Moreover, it does not suffice to simply consider accuracy by itself, because doing so leaves aspects such as the working space or occlusion effects out of the equation: a choice of parameters that is optimal with respect to accuracy often leads to unacceptable occlusion effects. The high-level relationship between these aspects is as follows:

**Working Space:** The working space is defined as the intersection of the field of view of the camera and the field of projection of the projector. The working space can theoretically be infinite. However, in practice it will always be finite as the ratio of scene irradiance of the projector to ambient illumination decreases with the square of the distance scene-projector. Consequently, the reflection of the projection pattern back into the camera quickly converges to a value within the image's noise level. As a rule of thumb, the accuracy decrease with increasing volume of the working space, that is accuracy and volume of working space tend to be inversely related.

**Occlusion:** We define occlusion as the percentage of image pixels for which the corresponding scene patch has not been illuminated by the projected light and no range value could be obtained for this reason. Occlusion effects are completely scene dependent and cannot be assessed in the course of a generic analysis. With flat objects, there will be no occlusion effects; at the same time, given the necessary nonzero baseline of a triangulation-based range acquisition system, it is always possible to construct a scene that is totally occluded. Obviously occlusion effects increase along with convergence angle and baseline; the more divergent the two lines of view, the more likely and severe the missing data problem. With other words, accuracy and occlusion are inversely related.

### 5.3 Experimental Evaluation of the Range Error

In this section, we present the experimental evaluation of the accuracy of the prototype ranging system. To begin with, we experimentally determine the main factor that causes the measurement uncertainty of the system, namely the localization error (5.3.1).

Next, we analyze the range error itself. As mentioned above, we break down the measurement error into the statistical and the systematic error. We quantify the former error by comparing many different range images of a given static scene, i.e. by repeating the same measurement several times and analyzing the scatter of the range measurements. This is discussed in section 5.3.2.

Quantifying the total error that includes a systematic component as well is much more difficult. To this end, we need suitable reference or ground truth data; the uncertainty regarding this data should – as a rule of thumb – be 5 to 10 times smaller than the measurement error of the system to be evaluated [Luhmann 2000]. So given that we expect the latter to be in the area of 0.2 mm for the specified working space, the reference data should be known to within  $\pm 0.02$  mm. In most cases, there is no large and complex test body available whose shape is known up to such a small measurement uncertainty. Accordingly, the literature almost universally takes on an alternative approach and uses less realistic simple bodies of elementary geometry, in most cases planar objects (e.g. [Vuylsteke and Oosterlinck 1990]), for accuracy evaluation. For practical reasons, we adopt this approach as well and use a simple planar object to obtain numeric accuracy results. This is the topic of the remaining subsections 5.3.3 and 5.3.4.

#### 5.3.1 Experimental Determination of the Localization Error

To determine the localization error experimentally, we position a planar object approximately parallel to the projector's retinal plane, i.e. in a way such that it has an about constant depth in the projector coordinate system. The projector illuminates the plane with the encoded pattern, which is effectively a pattern of equidistant stripe edges. Then the projected stripe edges are approximately equidistant on the target plane. Next, we acquire an image of the object. In general, the planar object and the camera image plane subtend an angle  $\theta$ , that is the camera coordinate depth changes over the image plane and the stripe edges are not equidistant in the image due to perspective distortion. Nevertheless, the distance between imaged edges is locally about constant provided the  $z$ -distance between camera and plane is large and the angle  $\theta$  is not too large. This circumstance permits a statistical analysis of the unknown random variable "localization error". Figure 532 shows the results of two corresponding experiments, more precisely the relative frequency (unit-less from 0 to 1) of certain edge distance values (specified in pixel units) as they occurred during each experiment. The first sample consist of 11580 measured values, which have a mean distance of 3.38 pixel und a sample standard deviation of 0.14 pixel. The second sample consists of 5124 distance values, which have a mean distance of 4.46 pixel and a sample standard deviation of 0.18 pixel.

Without sub-pixel arithmetic, one would expect a standard deviation of ca. 0.29 pixel. So we conclude from the observed standard deviation values (which include several other samples not listed here) that the employed sub-pixel arithmetic leads to an effective pixel size of about half the physical pixel size. We use this result over the next sections to predict the accuracy of the prototype.

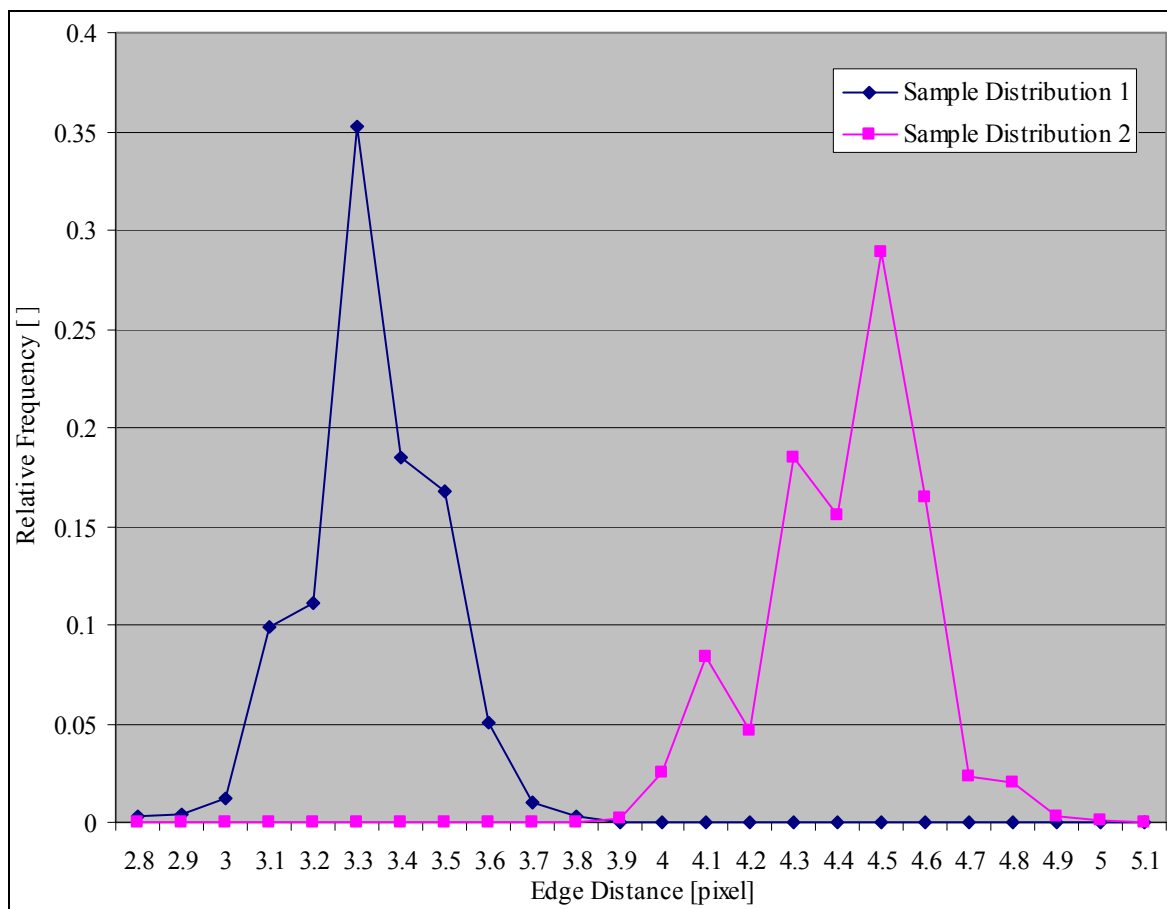


Figure 53: Relative frequency of imaged edge distance values for two large samples.

We hypothesize that each sample mean represents the respective ground truth; there is no reason to suspect that the localization error as such (i.e. not the one for a given fixed pixel-scene combination, but the many samples of the localization error over the whole image) has a major systematic component. The distribution of the sample values does not contradict the hypothesis of a uniform density function of the error. At the same time, contrasting the predicted with the actual distributions shows that there are certain factors such as the fixed-point sub-pixel arithmetic or the fact that the camera is a single-chip color camera that cause the actual results to deviate notably from the simple theoretical model. In this context, it is important to note that a more complex model that would seemingly explain the observed data better would not significantly change the key results of this chapter that are in any case estimates only; it would only make them harder to obtain.

### 5.3.2 The Precision or Repeat Accuracy

We quantify the statistical error of a range acquisition system by acquiring several range images of a given static scene, i.e. by repeating the same measurement over and over, and analyzing the scatter of the depth measurements. Conducting corresponding experiments with the prototype system yields an error of a standard deviation in the area of 0.01 mm to 0.04 mm (for several distinct scenes, levels of background illumination etc.). This small error is due to the fact that for a given set-up and static scene the localization error has a large systematic component when considering isolated and fixed pixel/scene patch combinations. In this case, only the imaging noise and other minor factors such as the ambient illumination represent a statistical component that causes the observed very small scatter in the z-coordinate. So the statistical error in the classical sense is not a very informative quantity with the proposed system and accordingly not discussed in more detail.

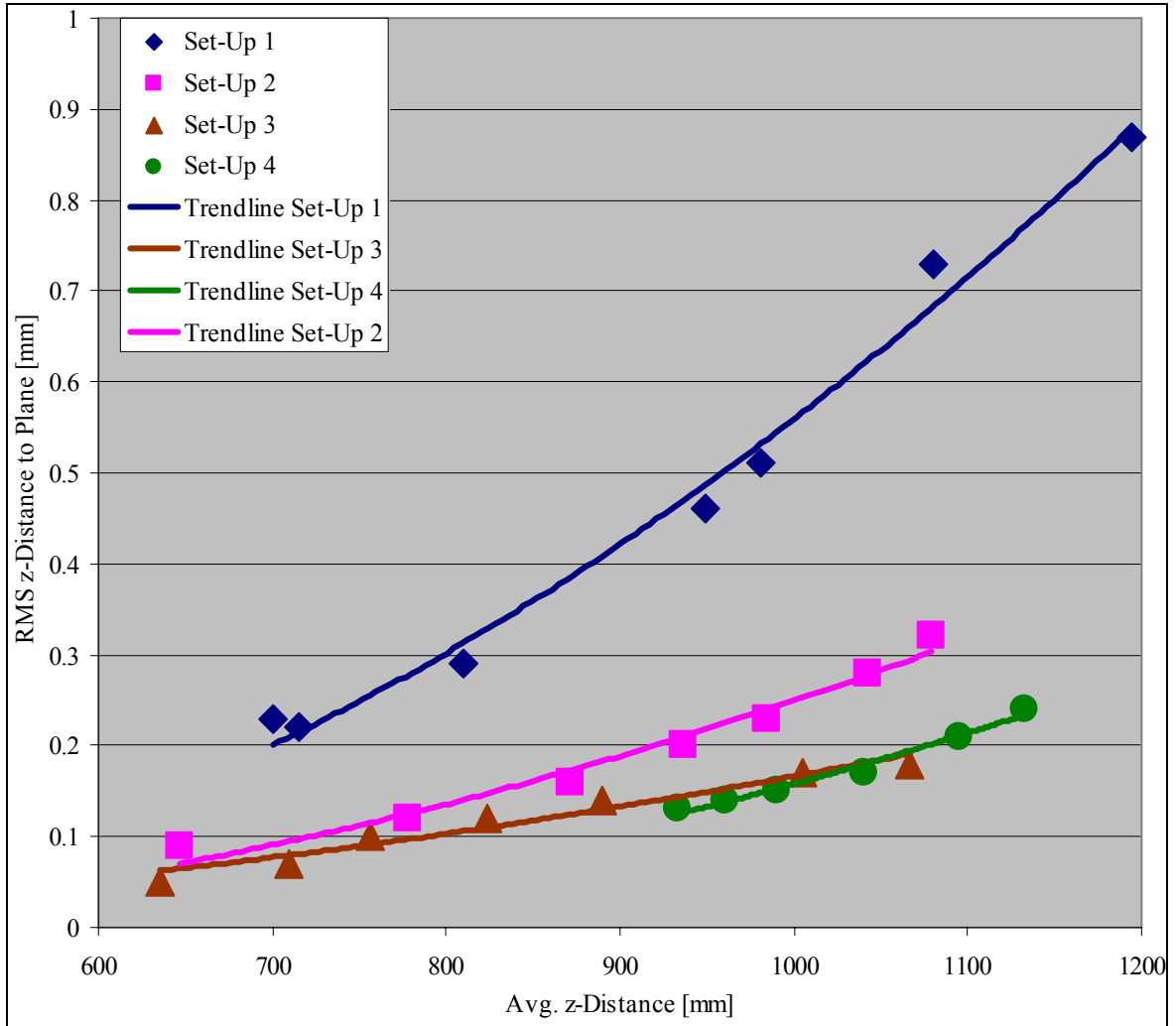


Figure 54: The standard deviation of the z-distance between measured coordinates of a planar object to that of the plane with the minimal squared z-distance to the measured data (for several distinct set-ups and distances, where each sample consists of about 200 000 measured values).

### 5.3.3 Measured Shape of a Planar Object

For the experiments of this section, we acquire a depth map of a planar object that is approximately parallel to the camera's image plane, i.e. one located at an about constant depth. Subsequently, we perform a least-square fit of the acquired surface data to a 3D space plane  $z = Ax + By + D$  by determining the set of parameters  $(A, B, D)$  that minimizes the following sum over all  $n$  obtained 3D coordinates:

$$\sum_{i=1}^n \underbrace{(Ax_{ci} + By_{ci} + D - z_{ci})^2}_{d_i} \quad (101)$$

We then analyze the scatter of the z-distances  $d_i$  of the measured coordinates relative to the resulting plane (a spot check using orthogonal rather than z-distances yielded about the same results). The planar object used is the calibration target, i.e. a sheet of glass of size 600 mm by 400 mm with a glued-on checkerboard pattern of 5 by 9 black squares on a white background. The target is known to be planar to within  $\pm 0.05$  mm. A fixed region of the image (corresponding to about 200 000 points in 3D space) that is visible in all images over all distinct set-ups is used to deter-

Set-Up 1			Set-Up 2			Set-Up 3			Set-Up 4		
Mean [mm]	RMS [mm]	Pred. RMS [mm]	Mean [mm]	RMS [mm]	Pred. RMS [mm]	Mean [mm]	RMS [mm]	Pred. RMS [mm]	Mean [mm]	RMS [mm]	Pred. RMS [mm]
700	0.23	0.29	647	0.09	0.13	635	0.05	0.08	933	0.13	0.18
715	0.22	0.30	778	0.12	0.18	710	0.07	0.10	960	0.14	0.19
810	0.29	0.39	871	0.16	0.23	756	0.10	0.11	990	0.15	0.20
949	0.46	0.53	936	0.20	0.27	824	0.12	0.13	1040	0.17	0.22
981	0.51	0.57	984	0.23	0.29	890	0.14	0.15	1095	0.21	0.24
1080	0.73	0.69	1043	0.28	0.33	1005	0.17	0.19	1133	0.24	0.26
1194	0.87	0.84	1079	0.32	0.35	1066	0.18	0.22			

Table 11: Actual and predicted statistical parameters of the  $z_c$ -distribution in a depth map of a planar object for various distances and geometric set-ups (relative to a least-square fit 3D space plane).

mine the plane equation; no point within this region is excluded from the plane fit to verify the claim that the proposed approach avoids false matches, that is misidentifications of light planes.

We repeat this experiment for a number of different distances between plane and acquisition system and with four distinct set-ups, where calibration and choice of optical settings is done only once for each set-up:

- Set-Up 1: With the first set-up, the projector is located at the camera coordinates  $x_p \approx 160$  mm,  $z_p \approx 0$  mm, i.e. the separation of camera and projector is about 160 mm. The convergence angle is roughly  $10^\circ$ .
- Set-Up 2: With the second set-up, projector is located at the camera coordinates  $x_p \approx 310$  mm,  $z_p \approx 0$  mm, i.e. the separation of camera and projector is about 310 mm. The angle between the optical axes of camera and projector is roughly  $20^\circ$ . This set-up corresponds roughly to the one used for the recording of the face database.
- Set-Up 3: With the third set-up, projector is located at the camera coordinates  $x_p \approx 490$  mm,  $z_p \approx 0$  mm, i.e. the separation of camera and projector is about 490 mm. The angle between the optical axes of camera and projector is roughly  $30^\circ$ .
- Set-Up 4: With the fourth set-up, the projector is located at the camera coordinates  $x_p \approx 310$  mm,  $z_p \approx 270$  mm, i.e. the separation of camera and set-up is 310 mm. The angle between the optical axes of camera and projector is roughly  $30^\circ$ .

Figure 54 and table 11 show the sample standard deviation (or **Root-Mean-Squared (RMS)** distance) of the  $z$ -distance values of the measured coordinates to the fitted plane. The presented experimental results lead to the conclusion that the depth accuracy is all in all rather high, typically in the area of a standard deviation of 0.1 to 0.4 mm. According to section 5.2, the error in the  $x$ - and  $y$ - is much smaller than the one in the  $z$ -coordinate, with our set-up only about 1/3 of it and less; with other words, it can be expected to be below 0.1 mm standard deviation. So we conclude (in this case only theoretically) that the prototype's overall measurement accuracy over all three coordinates is rather high.

The experimental results also imply that the calibration accuracy is at least fairly high; otherwise it would not be possible to fit planes to the range data with such a low root-mean-square error as listed in table 11 (over the whole large working space and with a system calibration that remains unchanged over all plane fits of a given set-up).



	Mean	Sample Standard Dev.	Min	Max
Set-Up 1, $z = 810$ mm	-0.02 mm	+0.34 mm	-1.58 mm	+1.36 mm
Set-Up 2, $z = 871$ mm	-0.01 mm	+0.17 mm	-0.84 mm	+0.77 mm
Set-Up 2, $z = 935$ mm	-0.02 mm	+0.23 mm	-1.32 mm	+2.30 mm

Table 12: Statistical parameters of the  $z_w$ -distribution in a depth map of the  $z_w = 0$  plane for a few representative set-ups.

Table 11 also lists the predicted standard deviation according to equation 100. In all cases, the error predicted by our model is close to, but consistently better than the actual one; this gap is barely noticeable with large  $z$ -distances.

We interpret the latter result as follows: First of all, the model estimates the depth error reasonably well. In particular, the approximate formula for the accuracy of a convergent set-up with  $z_p \neq 0$  seems to be quite useful: e.g. the trendlines for set-up 3 and 4 are more or less identical, as predicted by the formula. Next, the data processing employs several low-pass filter operations (e.g. convolution with a 3 by 3 Gaussian filter) that happen to have a positive effect on the accuracy with a planar test target. Things would be different with an object of high-frequency geometry, or, with other words, the use of a planar test target overstates the accuracy of the prototype somewhat. Finally, the model’s main limitations – mostly due to the intention to make it scene-independent – are that it neither takes defocusing nor the scene texture into account. The amount of the former is significant with the fixed-large-aperture projector used for the experiments and explains why the distance between predicted and actual error shrinks significantly with large, i.e. out-of-focus distances. The effect of the texture is completely scene dependent, yet not too problematic with the calibration target used for the above experiments. While its high-contrast black-and-white edges occasionally shift the position of a detected color edge quite a bit (leading to a large minimal and maximal error), there are simply not enough squares to have a noticeable effect on the standard deviation.

### 5.3.4 Reconstruction of the Calibration Target

The previous section evaluated the prototype’s relative accuracy; in this section we use the calibration target for testing the absolute accuracy of the system. It is the natural choice for this task given that it defines the  $XY_w$  plane of the world coordinate system and its (world coordinate) depth map should for that reason map all pixels to 0.

The experiments of this section use the same images as acquired for the experiments of the previous section to permit comparing the respective results. For each image, an external calibration is performed; afterwards the calibration target defines the world-coordinate system  $XY_w$  or  $z_w = 0$  plane. This circumstance permits directly analyzing the scatter of the measured  $z$  world coordinates. The difference to the experiment of the previous sections is that the measured values are not transformed in any way; in particular no plane fit is performed. Figure 40 shows an exemplary resulting range image. Table 12 gives the corresponding statistical parameters; given the results are consistent with those of table 11, it lists only a few representative examples.

We conclude from table 12: The sample mean is about zero as it should be; the slightly nonzero sample mean is unproblematic because the calibration target is known to be not exactly planar. In each case, the sample standard deviation is somewhat larger than the one shown in table 11. This might be due to the fact that for table 12 the whole image, that is the complete calibration target, was used as compared to only half of it for table 11, i.e. to the known slight systematic deviation of the test plane from planarity. In any case, the difference is in the area of 0.01 to 0.05 mm and consequently rather negligible. The reason for the comparatively large minimal and maximal error of table 12 is given in the previous section.

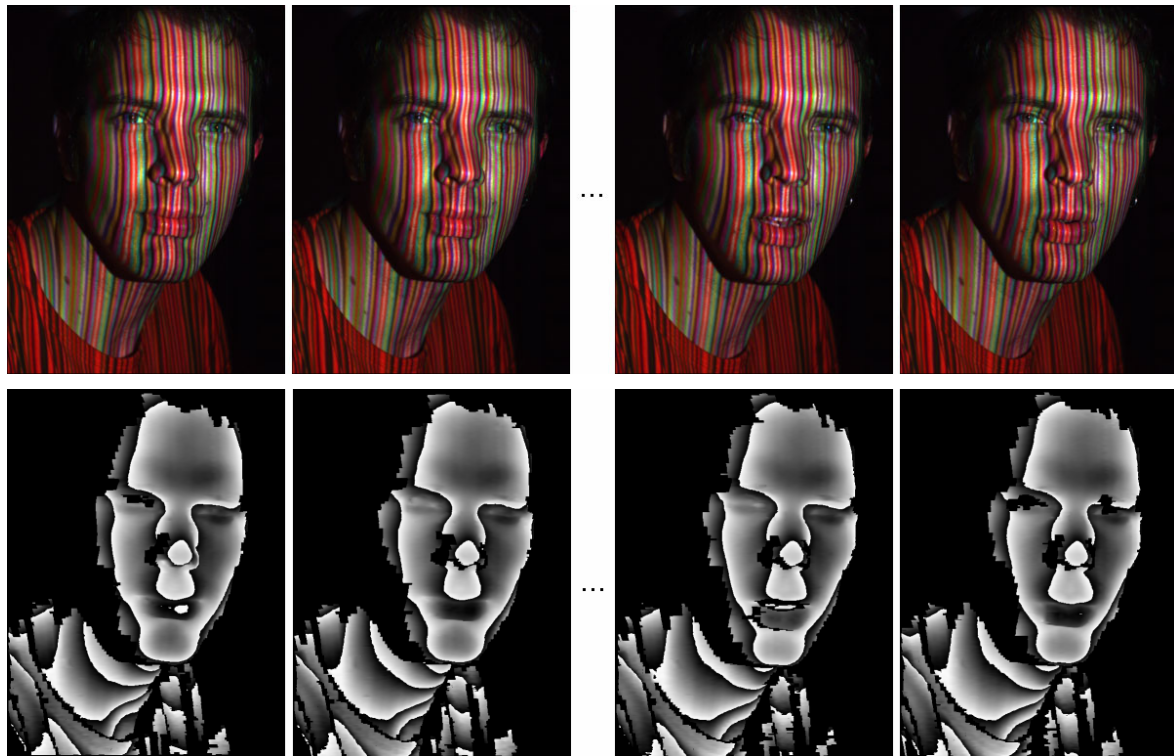


Figure 55: Test sequence talking head. The mapping of depth to gray level values is done as in figure 43, the only difference being that the coordinate system of the camera is used.

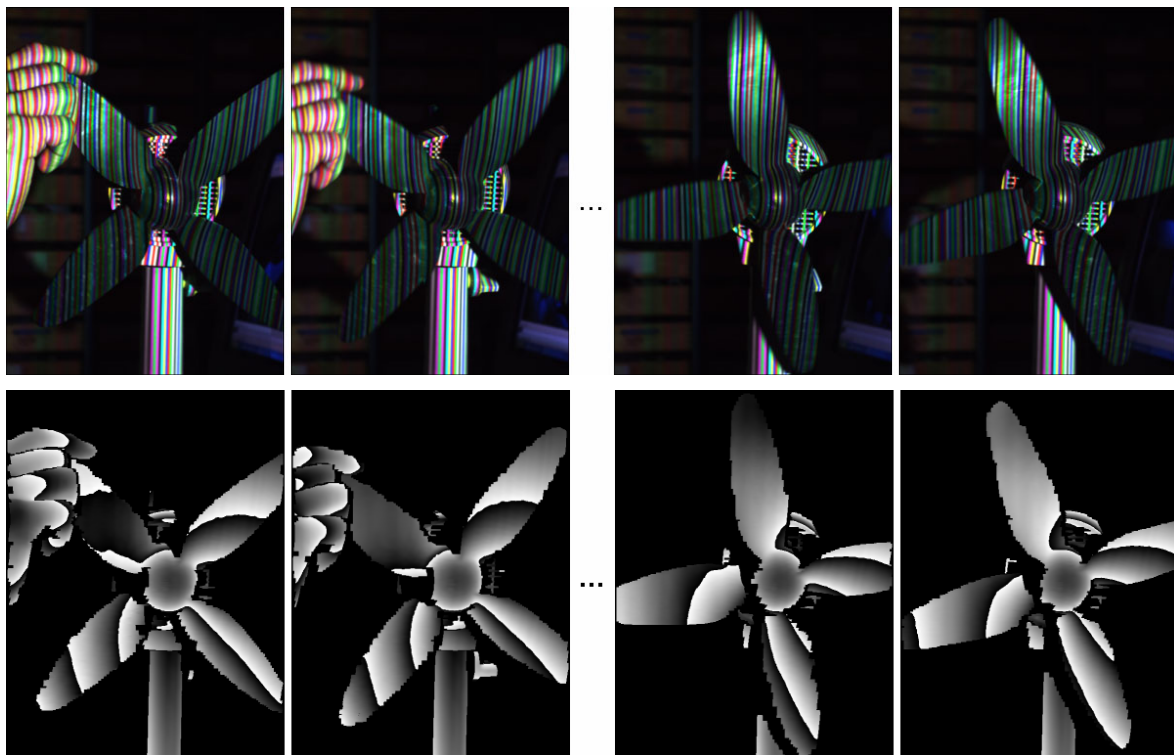


Figure 56: Test sequence rotating fan. The mapping of depth to gray level values is done as in figure 55.

Sequence	Head	Fan	Gesture
Number of images	10	10	10
Frames per second on 2.4 Ghz PC	12.3	14.5	16.0
Frames per second on 3.2 Ghz PC	18.3	21.3	23.7
Avg. nr of range values	192 000	142 000	83 000

Table 13: Frame-rate (in frames-per-second) of the prototype system for several exemplary scenes.

## 5.4 Experimental Evaluation of the Frame Rate

This section discusses the frame rate of the 3D acquisition system. Even with a fixed choice of system components, this rate is not a fixed number, but depends to some extent on several additional factors, the primary one being the size of the scene in the camera image. Table 13 describes the frame rate achieved with the color coded light step (without the optional scene color compensation step) for a set of representative, pre-recorded image sequences on an off-the-shelf low-cost 2.4 and 3.2 Ghz PC. Figures 55 and 56 show exemplary color and range images from these two sequences. According to table 13, the color coded light step misses the targeted 25 fps only by a narrow margin if the scene is rather small as in the case of the gesture sequence; if the scene fills most of the image, as e.g. with the talking head sequence, it still achieves about 18 fps (unless stated otherwise, all values refer to the results with the 3.2 Ghz PC).

These exemplary results are consistent with our experience that the current prototype system delivers between 17 and 25 frames of dimension 780 by 580 per second, with 20 fps being a representative, 17 fps being the practical worst-case frame rate. The implementation is currently not consistently optimized for speed; there is certainly significant room for improvement.

If the optional scene color compensation is used, the frame rate of the prototype drops considerably, namely to a value between 3.5 and 4 fps. This decrease is almost exclusively due to the fact that the software has to wait about 100 ms for the LCD projector to switch between the two distinct projection patterns. This problem could be solved easily by employing a more advanced synchronization mechanism.

The implementation of the complex stereo algorithm has currently known efficiency limitations which could not be remedied in the course of this work due to lack of time; as a consequence, the full system, i.e. the one including the stereo step, currently achieves a frame rate of about 1-2 fps only.

Empirically, the frame rate scales fairly linearly with the clock rate of the processor, as can be seen to some extent from table 13. We conclude that building a real-time system in the sense of section 4.1 based on the proposed approach is possible and could be realized with some more effort already today, respectively by simply waiting for the next generation of processors in the near term. Of course, simply reducing the frame size would immediately turn the current prototype into a real-time system. A corresponding experiment shows that the system consistently reaches a frame rate of over 30 fps if the image resolution is cropped to 640 by 400 pixels.

## 5.5 Experimental Qualitative Evaluation: Exemplary Scenes

This section presents several exemplary range images acquired with the prototype of the proposed ranging technique. The purpose is to give a qualitative impression; no numeric evaluation is attempted.

- Figure 57 displays the rectangle-based 3D mesh of a human face with the matching pattern image to the right of it. The data was acquired with the coded light module plus the optional color compensation step.
- Figures 58 and 59 show several point clouds of human faces as used for the 3D face recognition application described in chapter six. In each case, the underlying range image was acquired with the coded light module plus the color compensation step. Each view is generated by rotating the point cloud corresponding to a single frontal range image of the person.
- Figure 60 shows the rotated point cloud of a human hand acquired with the coded light step without the color compensation module in an outdoor environment on a sunny day. It demonstrates that the prototype is capable of operating in real-world environments, including uncontrolled outdoor scenes.
- Figure 61 shows the rotated point cloud of a human face acquired with the coded light module without the color compensation module. It shows the scene reflectivity as reconstructed from the pattern image with the approach of section 4.5.3.
- Figure 62 shows a complete 3D surface model of a dwarf. The 360° model was put together from several separate range images acquired with the coded light module plus the color compensation module. As the points of view were known for each image, the 3D model could be created by simply transforming the separate point clouds into a common coordinate system and computing a triangle mesh from the resulting point cloud. The latter step was carried out by Dehning [2004].
- Figure 63 shows a (nearly) complete 3D surface model of an ear-impression as used for manufacturing custom-made hearing aids. The scan is made with a low-cost system based on the proposed ranging technique. It employs a consumer camera for image acquisition and a LED and a single interference slide to generate the encoded pattern. The system is described in [Forster et al. 2003b]. The 360° model was put together from several separate range images acquired with the coded light module without the color compensation module. The combination of the several views to a single model was done by Pagoda Systems [2004].
- Figure 64 shows the image of a car wheel illuminated by the coded light projection pattern. Figure 65 displays the depth map resulting from this pattern image by evaluating it with the coded light algorithm without the color compensation module. Both images are acquired in a factory environment. They represent an example of a real-world application of the technique proposed in this work, the so-called task of wheel alignment [Forster et al. 2003c]. Its purpose is the adjustment of the angles of the wheels of a car, most importantly toe and camber, so that the wheels and the surface of the road, respectively the vehicle's axis of symmetry, form certain angles determined at the undercarriage design stage. The objective of these adjustments is maximum tire life and a car that tracks straight and true with a centered steering wheel, respectively one that travels only where it is steered.

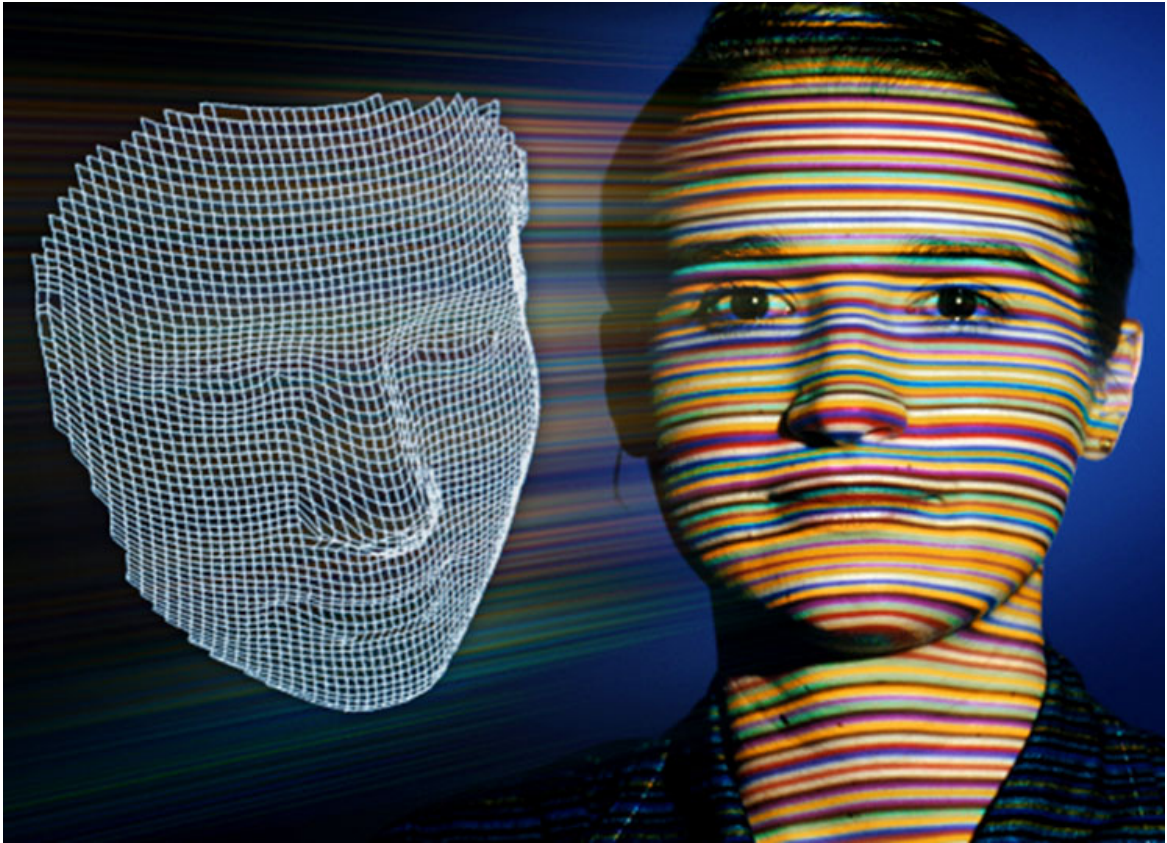


Figure 57: Rectangular mesh as computed from the image of a face illuminated by the proposed encoded pattern.



Figure 58: Several views of a 3D point cloud that represents a human face; the cloud and all views are generated from the same frontal range image.



Figure 59: Several point clouds of human faces corresponding to frontal range images. The top four views are all generated from the same range image.

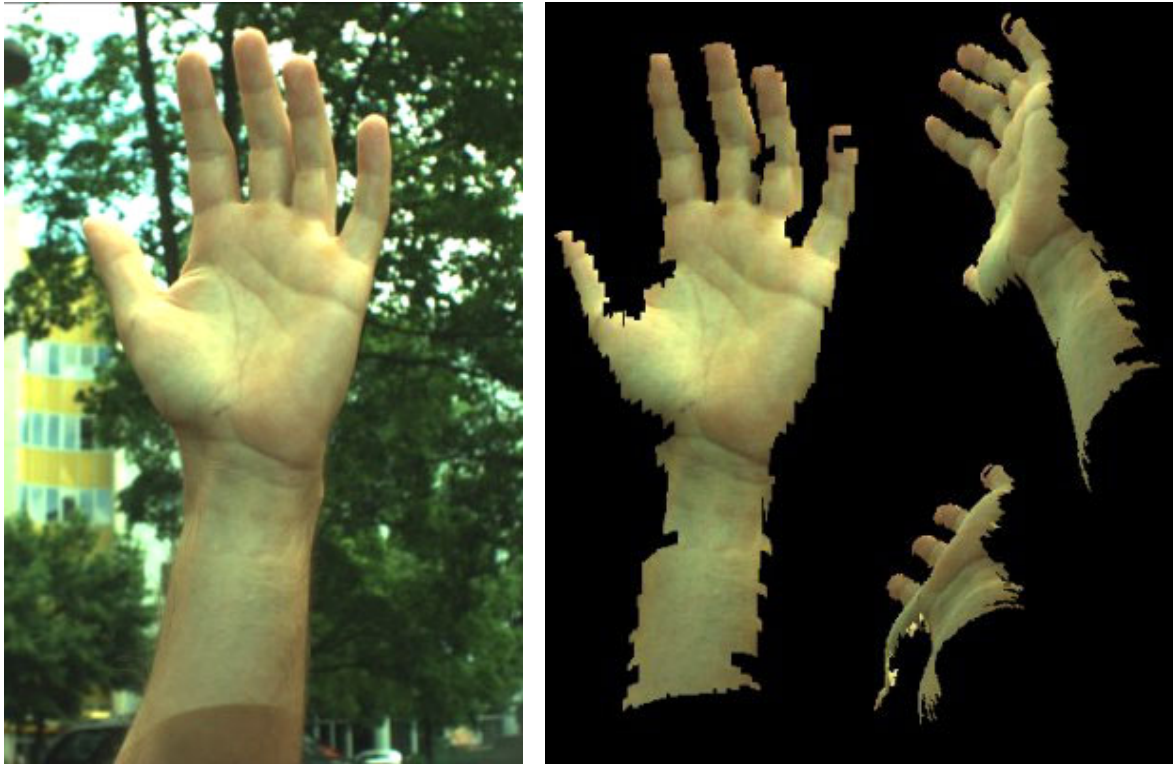


Figure 60: Several point clouds of a human hand corresponding to a range images acquired in an outdoor scenario on a sunny day. The three views are all generated from the same range image.

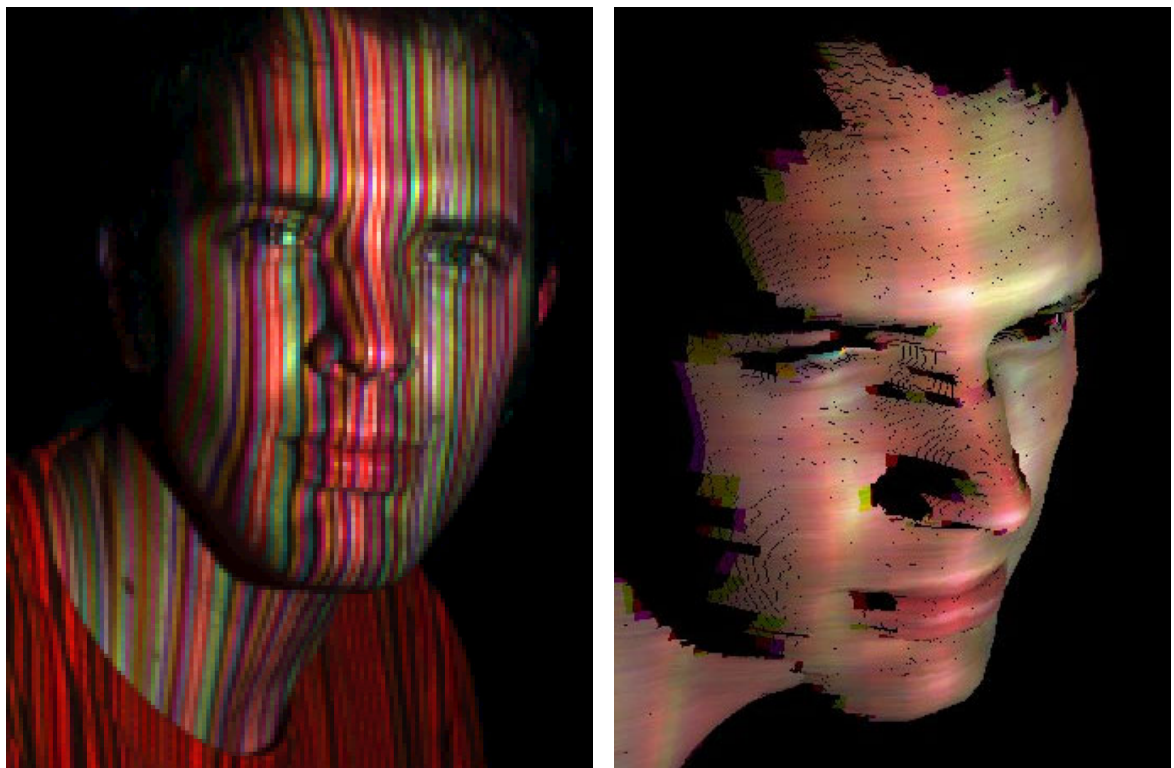


Figure 61: Pattern image of a human head (left); rotated point cloud generated from this pattern image (right). The texture of the point cloud is reconstructed from the pattern image with the approach of section 4.5.3.

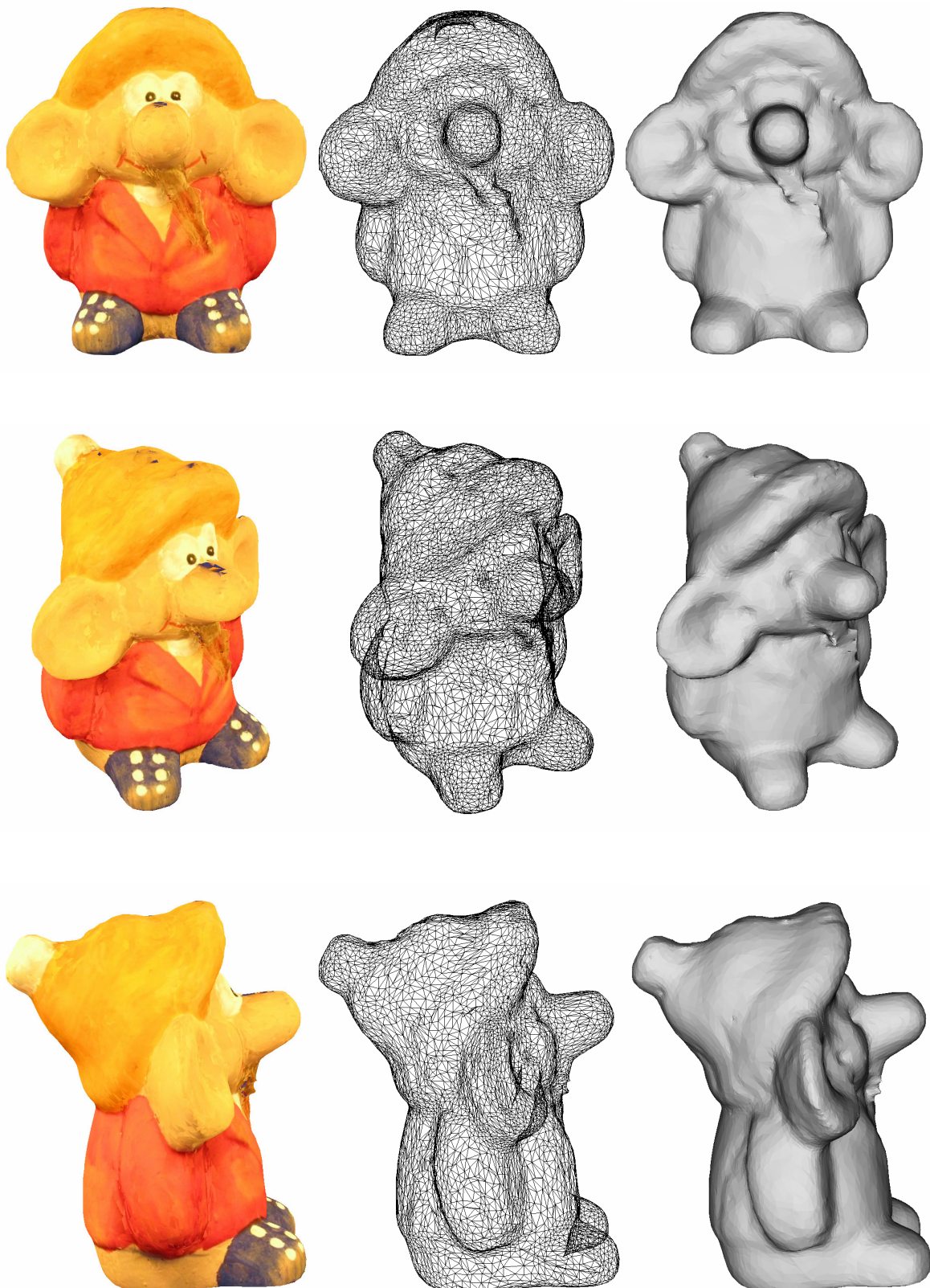


Figure 62: Three distinct views of a complete 3D model of a dwarf obtained by combining several range maps acquired with the prototype (left: 3D model with superimposed texture, middle: triangle mesh, right: shaded, texture-less 3D model).



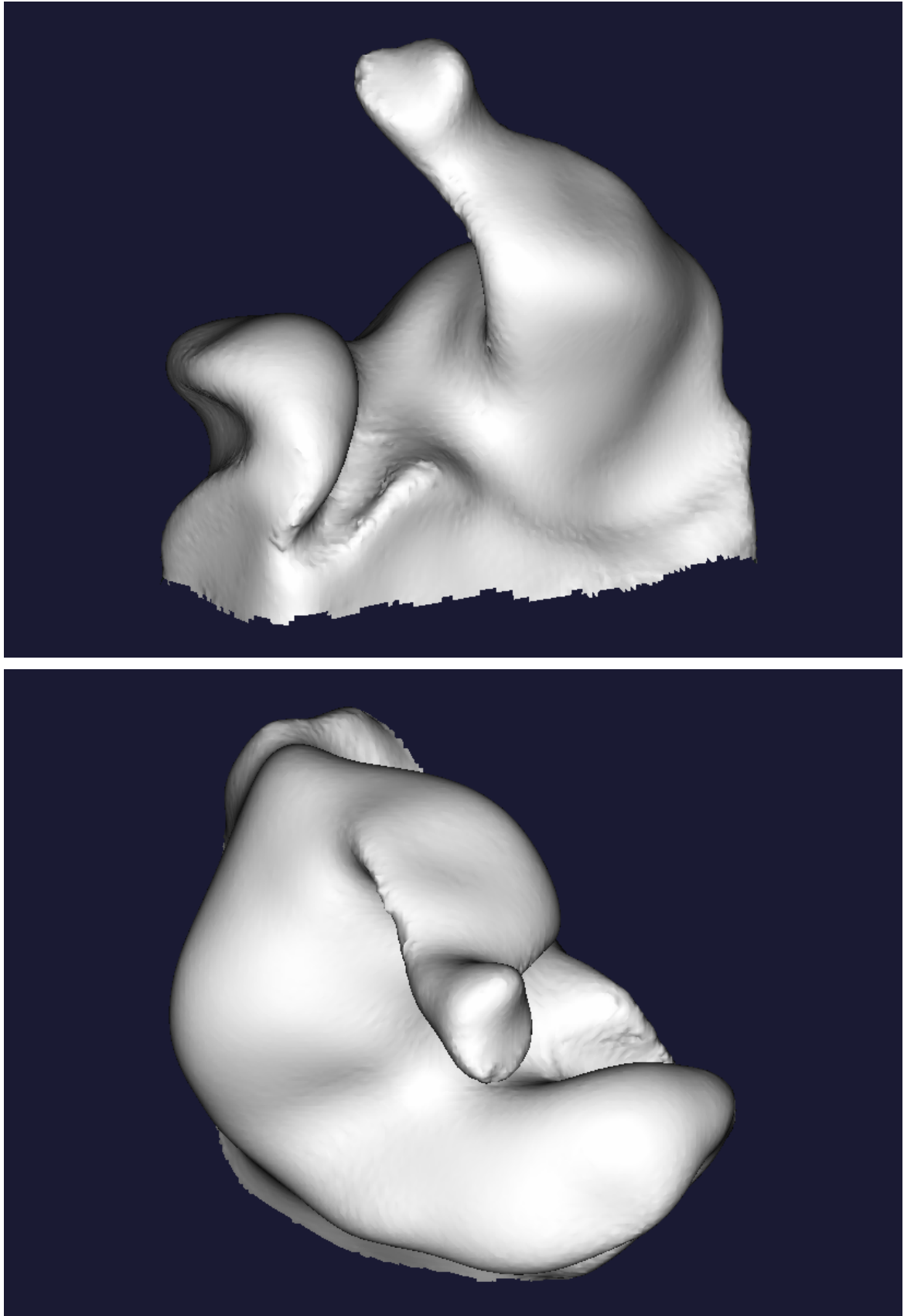


Figure 63: Two views of a 360° scan of an ear impression.



Figure 64: Image of a car wheel illuminated by the projection pattern.

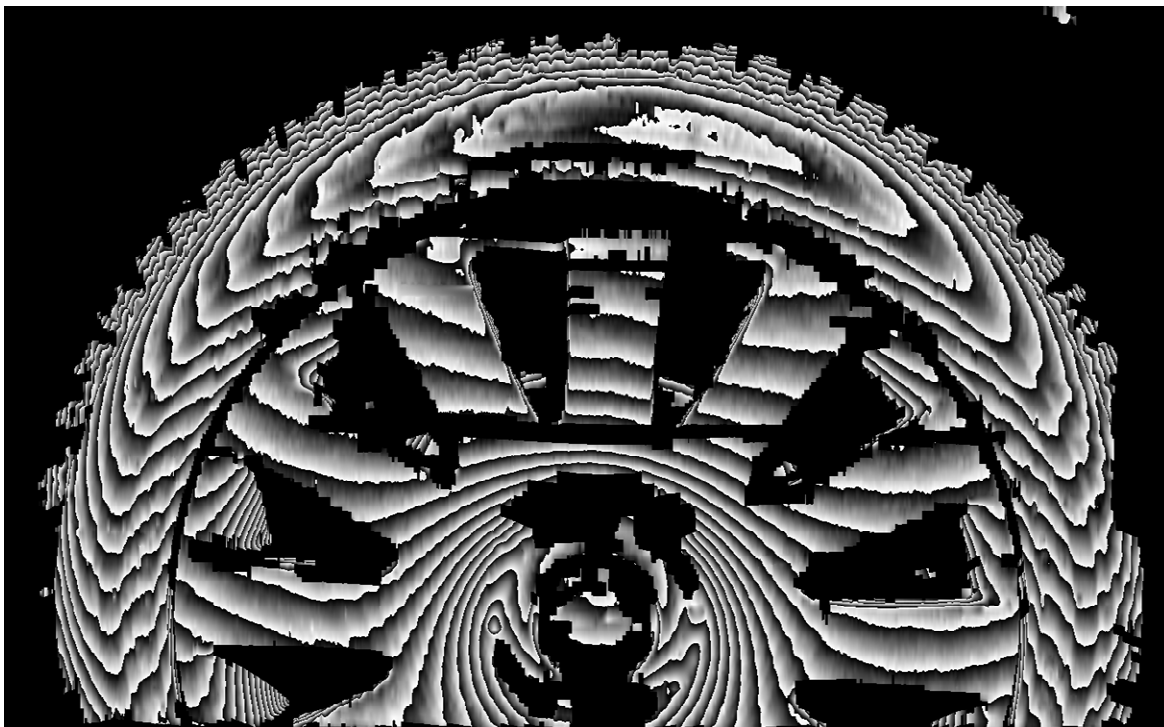


Figure 65: The depth map resulting from figure 64. The range data is visualized by mapping the depth value of a given image point, measured in  $100^{\text{th}}$  of millimeter, to the gray value range of 0 to 255:  $g = (z_c \cdot 100.0) \bmod 256$ . One gray level period then corresponds to 2.56 mm.

## 6 Face Recognition: An Exemplary Application

This chapter describes the exemplary application of the proposed ranging technique to the task of 3D face recognition, more precisely face verification. The focus of this chapter is on describing experiments that validate the idea of using range data for human-machine interfaces; no technical details on the face recognition algorithm are presented, given the underlying algorithm is considered state-of-the-art and its development was in any case not part of this work, but carried out by Tsalakanidou et al. [2004]. Thorough knowledge of biometry- and face recognition-related terms and concepts is assumed.

We first motivate – on a high level – why it might be favorable to use 3D data instead (or rather, in addition to) color or gray level images of a human face (6.1). We then sketch the technique employed for face verification (6.2). Section 6.3 describes a 3D database acquired for the evaluation of the algorithms, followed by the results of the evaluation (6.4).

### 6.1 Motivation for 3D Face Recognition

Why use range data for face recognition? From a high-level point of view, the following general arguments are well-established throughout the literature on face recognition:

- A recognition system based on range data cannot be deceived by photographs; an impostor needs to get hold of and reproduce the facial geometry associated with the claimed identity.
- Range data simplifies the task of face detection and localization.
- Range data is not affected by ambient illumination.
- Range data is not affected by the pose of the user (but for occlusion effects).
- Range data is not scaled by the unknown distance user-camera (but for a change of lateral resolution and accuracy of the data, depending on the ranging technique and set-up used).
- Range data gives access to 3D space-metric information not or not explicitly present in intensity images such as distance, area, volume or curvature data. Clearly this information is very useful for face recognition as the human face has a very pronounced and characteristic geometry, e.g. the nose or the cavities of the eyes exhibit a very characteristic curvature. The classic intensity-based approaches are at most able to exploit image or relative distances, respectively need to rescale a face image such that e.g. the eyes have a certain norm pixel distance.

The above advantages are not minor, but crucial ones; e.g. Zhao et al. [2003] conclude in their recent literature survey on face recognition that illumination and pose represent “two key problems for any face recognition system”.

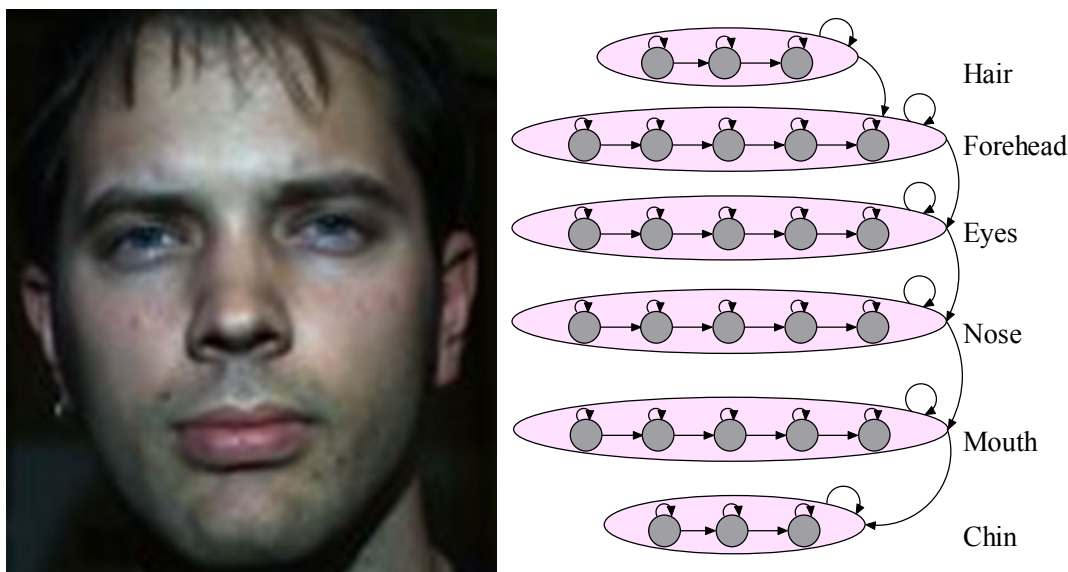


Figure 66: Topology of an embedded HMM for modeling the human face.

## 6.2 The Method Used for Face Recognition

*Hidden Markov Models* (HMMs, see e.g. [Rabiner [1989] or [Rabiner and Huang 1993]]) represent a popular statistical machine learning/ pattern recognition technique. They are traditionally used to model and recognize time-sequential or, more generally, one-dimensional data such as speech. For two-dimensional data such as images, so-called *pseudo-2D*, *planar* or *Embedded HMMs* (EHMM) have been proposed [Kuo and Agazzi 1994]. *EHHMs* are a generalization of HMMs where each state of a standard HMM, then called super-state, is itself a HMM. As transitions between embedded states belonging to different super-states are forbidden for computational complexity reasons, EHMMs do not have a truly two-dimensional structure, which explains the attribute *pseudo-2D*.

Samaria [1994] and subsequently [Nefian 1999] proposed employing (E)HMMs for face recognition, motivated by the stable sequential structure of human faces: for frontal views, the significant facial features – hair, forehead, eyes, nose, mouth and chin – appear in a natural order from top to the bottom. Accordingly, most EHMM-based face recognition systems choose the super-states to correspond to these six regions, i.e. to model the face along the vertical axis, and the embedded states to model it along the respective horizontal axes. Transitions between super-states are only permitted in a top-down manner, between embedded states only from left to right; skipping a state is illegal with most implementations. Figure 66 shows an example of a corresponding type of EHMM for modeling the human face.

To enroll a person into a face recognition system, a new EHMM is trained with one or several images of this person. The training follows the same steps as with standard HMMs, described e.g. in [Rabiner 1989], only that the Viterbi, the segmental k-means, the forward-backward and the Baum-Welch re-estimation algorithm are replaced by respective extended versions able to deal with embedded HMMs. Nefian [1999] describes the latter in detail. The observation sequence is generated from an image by scanning a sampling window with a certain, typically large overlap over it (from left to right and from top to bottom). From each extracted two-dimensional block a one-dimensional observation vector is formed, by appending either its pixel values [Samaria 1994] or some coefficients resulting from a suitable transformation of the block. Popular examples for the latter case are Karhunen-Loeve [Nefian 1999] or **D**iscrete-**C**osine-**T**ransformation (DCT) coefficients [Nefian 1999]. Given the trained EHMMs, verification is straightforward: it amounts to computing the probability that the EHMM associated with the claimed identity leads to the presented face image, or rather to the observation sequence extracted from it. To calculate this likelihood, in most cases the embedded extension of the well-known Viterbi algorithm is used.

Today the use of EHHM for face recognition is part of the state of the art; this is e.g. reflected by the fact that the popular Intel Computer Vision library [Intel 2001] includes corresponding functionality. We consequently limit our description to the most relevant aspects of the evaluated implementation.

Without doubt, its single most important aspect is that it is able to exploit range data. This proves to be helpful for the tasks of face detection and segmentation already; in fact, due to working space limitations of the acquisition system, typically only head and torso appear in the recorded range images. So the only non-trivial segmentation task is the one of separating the two. The described implementation solves it by separating the range image into two 3D space clusters and choosing the upper one to be the head (or, more formally, by taking advantage of a-priori knowledge of the body geometry). Given the segmented head, the algorithm locates the horizontally oriented axis of bilateral symmetry between the eyes. To that end, it introduces a measure of symmetry for two image points by comparing their respective intensity gradients, i.e. a measure that does not consider the range data. It then computes for each plausible axis a total symmetry measure by summing up the individual symmetry measure over all points pairs

- that lie on a certain straight line orthogonal to the candidate axis
- where both pixels have a certain fixed distance to the candidate axis

The axis that receives the highest total symmetry measure is taken to be the sought-after one. Next, the algorithm detects the corresponding vertical axis in an analogous way. It determines the center of the face as the intersection of these two axes. In a subsequent step, it centers a window of constant aspect ratio at this point and scales the window's content according to the point's depth.

This scaled face window with the face centered at a certain fixed position is then supplied to the actual face recognition engine, which is implemented as follows: it is based on an EHHM structured into the six super-states listed above and 3, 7, 7, 7, 7 and 3 embedded states, respectively. As with most continuous HMMs, the observation probability density function of each state is approximated by a mixture of (in this case four) distinct Gaussian distributions; the covariance matrices are chosen to be diagonal. To extract an observation sequence from a given RGB face window, the algorithm rescales it to a resolution of 128 by 128 pixels and transforms it into the YCrCb color-space; the motivation for choosing this color space is that it yielded the best results in a number of experiments. It then scans a sampling window of 12 by 12 pixels with an overlap of 10 by 10 pixels over the YCrCb image. To transform a block into an observation vector, the algorithm applies the 2D DCT transformation to each block, where each color component is treated separately. For each block and component, only the 3 by 3 low-frequency DCT coefficients excluding the DC-component, i.e. a total of eight coefficients, are kept. The observation vector of length 24 resulting from a 12 by 12 YCrCb block is formed by appending the DCT coefficients of the three color components. The depth map is processed analogously by simply treating it as a grayscale image.

For each enrolled person, two EHHMs are trained, one with the color and one with the depth data. To estimate the probability that a given image belongs to a certain identity, the likelihood (or an equivalent score) is computed that the observed color sequence occurs with the color EHHM trained for this identity. The same is done for the corresponding depth EHHM and the depth observation sequence. The two individual scores are then combined to a single one via the formula

$$Score_{tot} = w_{col} \cdot \log_{10}(Score_{col} + 1) + w_{depth} \cdot \log_{10}(Score_{depth} + 1) \quad (102)$$

where  $w_{col}$  and  $w_{depth}$  are the relative weights of the color and depth scores. The choice of the fusion technique as well as of the relative weights is motivated experimentally. The total score is the output of the face recognition engine.

As mentioned before, a more detailed description of the algorithm can be found in [Tsalakanidou et al. 2004].

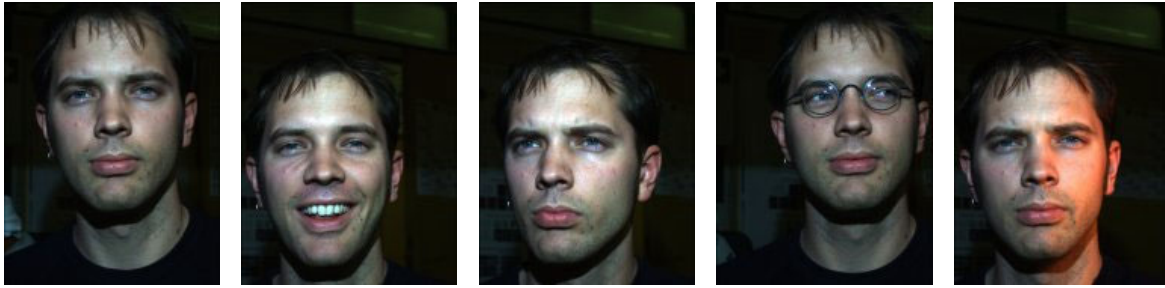


Figure 67: Exemplary images from the database showing images of the type frontal, expression, rotation, glasses and illumination, in that order, for the same person.

### 6.3 The Face Database

To evaluate both the 3D face recognition and the proposed range acquisition technique, a database was recorded in the course of this work. The specification criteria for this database orient to international guidelines such as the ones issued by the Biometrics Working Group ([BWG 2000, 2002]). The evaluation database contains 2,711 recordings of 20 individuals. A recording consists of a color image (571 x 752 pixels) and the corresponding depth map (571 x 752 pixels). The images were captured by the acquisition system of this work in an indoor office scenario; as the stereo step (section 4.4) was not yet functional at the beginning of the recording and, much more importantly, the coded light step suffices to acquire high-quality range images of human faces, the database was acquired with a single-camera, coded light-only system.

The database contains two recording sessions per person, where the time span between the sessions was several weeks. The illumination conditions were somewhat different during each session due to daylight influences (via the office windows). Each session comprises the same 20 male individuals. Their age varied from 28 to 60. On average, 135 images per person were acquired. For each person, the database contains the following types of images:

- **Frontal Images:** The database contains several “ideal” images taken under a pose angle of  $0^\circ$ , i.e. with an expressionless user looking directly in the camera, under standard illumination and without glasses.
- **Pose:** The database contains at least two images with pose angles of  $+20^\circ$  and  $-20^\circ$ . The other recording settings are equal to the ones for frontal images.
- **Expressions:** The database contains at least three so-called expression-images, where the user grins (with a closed mouth), laughs (with an opened mouth) or closes his eyes, respectively. The other recording settings are equal to the ones for frontal images.
- **Illumination:** The database contains several images with significantly changed background illumination. To that end, common head-lamps were fixed on the left and right of the person. The other recording settings are equal to the ones for frontal images.
- **Eyeglasses:** The database contains several frontal images of each person wearing glasses.
- **Enrolment:** The database contains several frontal images with and without light expressions for the purpose of enrolment.

Figure 67 shows an example for each image type (but for the enrolment type) for the same individual. Table 14 lists the number of images per session and type contained in the database.

Type	Session 1	Session 2	Total
Frontal	94	132	226
Pose	228	246	474
Expression	252	368	620
Illumination	359	387	746
Glasses	95	134	229
Enrolment	204	212	416
All	1,232	1,479	2,711

Table 14: Number of images per session and type contained in the recorded database.

Clearly the database has certain deficiencies for the purpose of a full-fledged evaluation of a biometric system: it contains too few individuals, too few and too closely timed sessions and the population of exclusively Caucasian male adults is not diverse at all. However, these shortcomings are acceptable because the purpose of the database is not the evaluation of the performance of the face recognition system compared to the state of the art. Such a technology evaluation would require testing competing algorithms on the database as well, which is by far beyond the scope of this work. Rather, its purpose is exclusively the evaluation of the effect of using range data in addition to color data for an exemplary human-machine interface.

## 6.4 Evaluation Results

This section describes the evaluation of the face recognition system of section 6.2 with the database of the previous section. During the evaluation, different algorithm variation - image type combinations are assessed, namely the algorithmic variations color only, grayscale only, depth only, color plus depth and grayscale plus depth with the respective image types frontal, pose, expression, illumination and eyeglass images. The motivation for considering distinct algorithmic approaches is as follows: the ones using color or grayscale only are taken to represent the baseline of this evaluation since they correspond to standard face recognition systems. Comparing the performance of the approaches based on depth, color plus depth or grayscale plus depth to this baseline should provide an indication in as far depth images improve the performance of face recognition systems. Respectively, contrasting the results for different types of images gives certain hints as to the strength and weaknesses of the various algorithmic approaches.

The most important aspect of a biometric system – and the only one evaluated in this section – is its error rate. There are two different errors a verification system can make: it can falsely accept an impostor or falsely reject an enrolled person, a so-called *original*. The respective error rates are called **False Acceptance Rate (FAR)** and **False Rejection Rate (FRR)**. These two rates are not a fixed property of a biometric system, but depend on its parameterization.

Why this is the case becomes obvious when considering that a face recognition algorithm as described in section 6.2 outputs a score in response to an image together with a claimed identity. The verification system needs to make its accept- or reject decision on the basis of this score, which it typically does by applying a threshold to the score.

With most biometric systems, this threshold is not global, but person-dependent. Considering a single original, we obtain for each choice of its threshold a certain FAR and FRR. So for each original  $i$  out of the  $n$  enrolled ones the two functions  $FAR_i(FRR)$  and  $FRR_i(FAR)$  that specify the FAR for a certain FRR and vice versa are well defined (ignoring minor problems due to the discrete nature of all these quantities, which can e.g. be solved by interpolation).

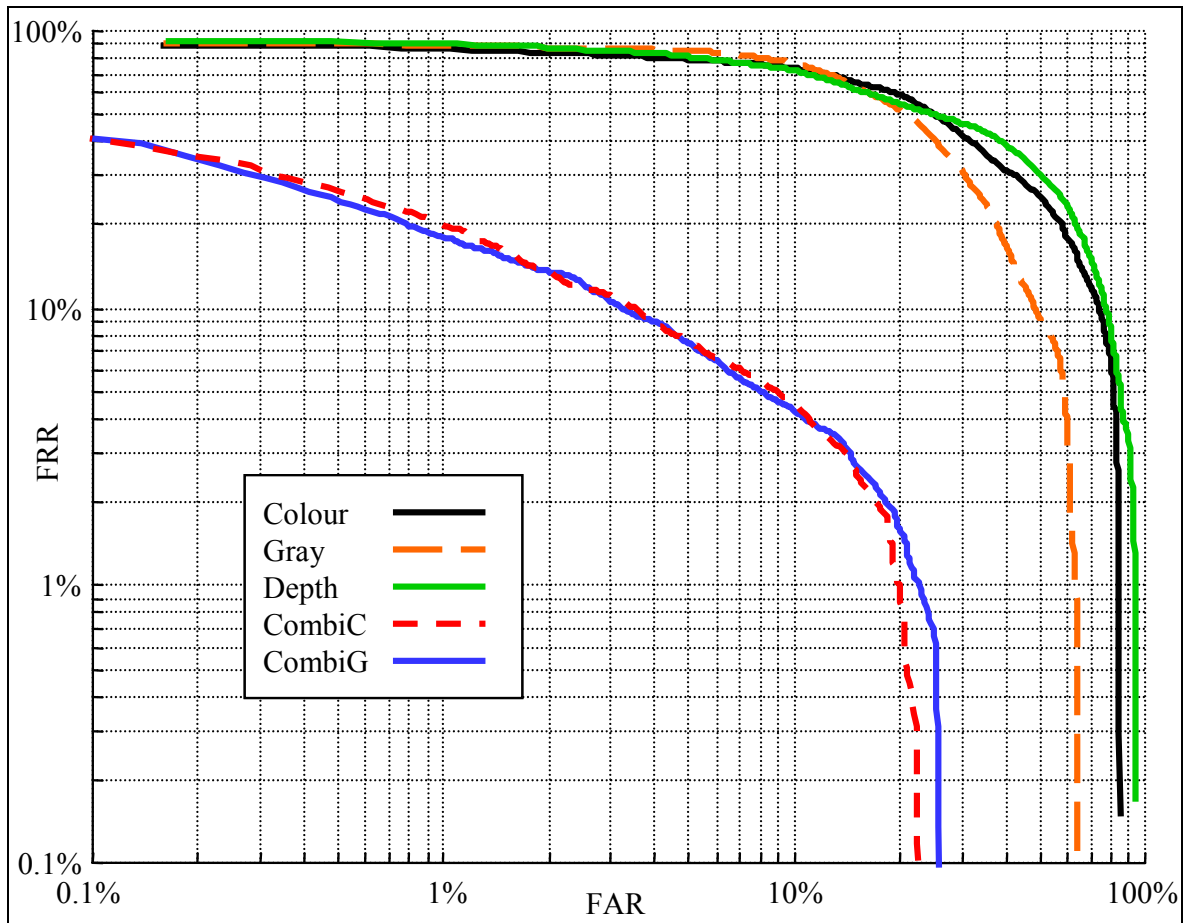


Figure 68: Characteristic curve of the face recognition system for several distinct variations of the EHHM-algorithm over all images (double logarithmic scale). Image taken from [Hiscore 2003].

The overall FAR for a given FRR is then calculated by computing the average individual FAR for this FRR over all  $n$  persons in the database:

$$FAR(FRR) = \frac{1}{n} \sum_{i=1}^n FAR_i(FRR) \quad (103)$$

The corresponding overall FRR for a given FAR value is calculated analogously by computing the average individual FRR for this FAR over all  $n$  enrolled persons:

$$FRR(FAR) = \frac{1}{n} \sum_{i=1}^n FRR_i(FAR) \quad (104)$$

The now well-defined relationship between its (global) FAR and the (global) FRR permits determining the **Receiver Operating Characteristic (ROC)** curve of a biometric system by plotting the FRR for all FAR values between 0.0 and 100 %, i.e. the FRR as function of the FAR.

Figure 68 shows the result of the evaluation in form of such ROC curves; more precisely, it displays the ROC curves for several distinct variations of the EHHM-algorithm over all images in the database. Two left- and bottommost curves represent the approaches combining range with color, respectively grayscale data, while the three right- and topmost ones are the ROC curves of the color-, gray level- and depth-only variations.



Variation	Frontal	Pose	Expression	Illumination	Glasses	All
Color Data Only	19	24	26	62	25	35
Gray Level Data Only	20	27	30	39	27	31
Depth Data Only	35	45	40	36	37	39
Combination of Color and Depth Data	2	4	3	8	6	6
Combination of Gray Level and Depth Data	2	8	4	6	7	6

Table 15: The equal-error rates of the face recognition system for several variations of the algorithm. All values are specified in %-units.

It is often preferable to represent the system quality more concisely via a single value. A widely-used corresponding indicator is the **Equal Error Rate (EER)**. It is determined as follows: for each individual  $i$  there is a threshold setting that results in the FRR being equal to the FAR; the corresponding person-dependent value of the FAR (or FRR) is denominated as  $EER_i$ . The overall EER is then computed as the average equal error rate over all  $n$  originals:

$$EER = \frac{1}{n} \sum_{i=1}^n EER_i = \frac{1}{n} \sum_{i=1}^n FRR_i \Big|_{\frac{FAR_i}{FRR_i}=1}. \quad (105)$$

Table 15 shows the equal error rate for all variations of the EHMM algorithm over all images as well as for certain image types. The evaluation results as represented by figure 68 and table 15 lead to the following conclusions:

- The overall performance of the combined approach is fair, the one of the color, gray level and depth only approaches not convincing.
- Color and grayscale data alone result in an about equal performance, where color is slightly better in most cases, but considerably worse in the case of the illumination images.
- Range data alone results in the worst overall performance, but one that is at least not affected by strong illumination changes.
- As could be expected, every algorithm variation gives the best results with frontal and the worst with glasses and illumination-type images.
- For every image type, the combination of 2D and depth information improves the equal error rate of the 2D-only approaches (color, grayscale or depth) drastically. This is the key result of the evaluation. A relevant aspect here is that the color, gray level or depth only approaches have to use alternative strategies for certain tasks, e.g. the approaches operating without access to range data cannot rely on it for segmentation or scaling. However, the key difference between the latter and the combined approaches is the score fusion. This can be seen by e.g. considering the case of the ideal frontal color images, in whose case a color- or intensity-based segmentation is straightforward due to the dark background and no significant scale changes are present.

A more detailed description of the evaluation result can be found in [HISCORE 2003]. The overall conclusions on the above results are given in the next chapter.



## 7 Conclusions

The final chapter lists the inferences reached in the course of the work (7.1). It then discusses how future work could extend on the presented results (7.2). It concludes with stating the contributions of new knowledge made by this thesis (7.3).

### 7.1 Conclusions

This work solves the problem of range image acquisition as defined in section 4.1. Respectively, given the proposed technique is able to obtain the 2D intensity/color data of a scene along with the range data, it solves the problem of simultaneously acquiring range and color images of a scene.

To this end, a new ranging approach has been developed in section four. Its centerpiece is a color coded light approach based on a single static projection pattern. As shown in chapters four and five, this approach overcomes certain key problem commonly associated with this technique. For instance, it copes well with strongly colored and textured scenes. The principal mechanisms that allow achieving this are the use of local color edge patterns for error-detecting encoding, a corresponding projection pattern design guided by image processing requirements and a specialized approach to data processing. The latter exploits the pattern's error detection capabilities to detect only projected color edges and is moreover able to correct certain errors. These ideas work quite well in practice, as does the one of adopting a pseudo-random approach to generate the necessary color edge patterns. As a result, the coded light step is well suited to obtain range images of most scenes, including the primarily targeted ones, namely human faces and hands. This inference is underscored by the many examples presented throughout this work and the fact that a large 3D face database was acquired using the coded light step only.

Nevertheless the coded light step has certain intrinsic limitations, e.g. when it comes to coping with surface singularities or some kinds of high-frequency texture. To overcome them, this work introduces a stereo algorithm to obtain range values for the parts of the scene where the coded light step fails. In its case, the basic idea is that accurate real-time stereo vision becomes possible because these problematic parts typically have a distinct optical structure and tend to make up only a small percentage of the scene. The stereo step as presented in this work does not yet fully exploit the large potential of such a combined approach; this is by and large unnecessary with the scenes relevant for human-machine communication as targeted in this work. Nonetheless it produces promising first results; we interpret them as proof that the underlying general idea is well-founded.

All in all, we conclude with regard to the presented ranging approach: its coded light step represents an improvement of an in principle well-known technique to overcome its major shortcomings. The proposed way of combining coded light and stereo represents a new idea altogether. It has several principal advantages compared to standalone coded light or stereo vision:

- It obtains range data for scene patches where (spatially-encoded) single-shot coded light cannot work due to principal limitations, e.g. with patches that do not have the necessary size in the image or for which the projected pattern is unrecognizable due to their high-frequency texture.
- It is more accurate at object borders compared to traditional single-shot coded light.
- It allows achieving a higher resolution than traditional single-shot coded light (via the sub-pixel correspondence for areas between the boundaries of pattern primitives).
- It is much faster, more accurate and more reliable – as it avoids false matches – than stereo vision algorithms; moreover, it works well with optically unstructured scenes.
- It is less affected by occlusion than standalone stereo vision or coded light because it suffices if a scene patch is visible for any two of the three system components.

Of the optional components not directly related to ranging, the scene reflectance compensation works well in practice and should be used if a high-quality color image needs to be acquired anyway. This might be the case with certain applications because in its current state the scene reflectance estimation is still quite a bit away from providing such a high-quality color image. The threshold optimization component is an interesting example of what else can be achieved by exploiting the error detection capabilities of the proposed pattern and solves an important problem relevant for any real-world application.

We further conclude that the task of camera and projector calibration is solved; extending Tsai's monoview camera calibration technique to one based on several views and applying it to projector calibration as well results in an overall calibration accuracy that should suffice for most purposes.

The range error analysis of chapter 5 yields valuable insights regarding the relationship between the set-up parameters and the accuracy. Contrasting the predicted with the experimentally determined accuracy leads to the conclusion that the derived formulas are very useful, but should be treated with certain precaution; they forecast the magnitude of the uncertainty rather than its precise values. This is because the presented analysis needs to ignore certain factors that are quite relevant for the measurement uncertainty such as the scene texture or depth-of-field influences.

The evaluation shows that the prototype system meets most objectives defined in section 4.1. Its representative ranging accuracy of 0.2 mm standard deviation over the large working space should be adequate for most tasks and could be improved easily by e.g. employing a megapixel camera. While the coded light step achieves 25 frames per second with certain scenes, its average rate is ca. 20 fps and may even drop as low as 17 fps in some cases. Including the stereo algorithm the rate is currently only about 1-2 fps. However, these are limitations of the prototype, not of the approach. We estimate that a well-optimized implementation of the algorithms would lead to an immediate speed gain by a factor of at least two to three. The evaluation also allows the conclusion that the proposed technique is well suited for arbitrary scenes. There are, of course, certain restrictions: e.g. typically no data at all is obtained in the far-away background due to depth of field limitations.

Other than the work on range image acquisition the 3D face recognition results of this thesis are mostly indicative; the evaluation shows that all variants of the presented face recognition algorithm do not represent a first-rate technique by today's standards. This is in part due to the challenging evaluation database which was strictly separated from the development database as well to the fact that it is almost impossible to compete with current commercial systems that benefit from by now hundreds of man-years of work. Nevertheless a definitive outcome of the evaluation is that the range data acquired by the prototype system is per se useful. And its key result is that the performance of a standard 2D face recognition algorithm is considerably boosted by the use of range data. Whether, or to which degree this conclusion applies to other algorithms and application as well and which way is the best to exploit the possibilities opened up by range data is a topic of further research, some examples of which are listed in the next section.

## 7.2 Future Research

How could future work extend on the results of this thesis?

1. A promising idea for future research is to exploit the real-time capability of the approach to improve the quality of the range data. At least with slowly moving scenes (i.e. ones with which the image position of a scene point changes only by a few pixel between consecutive frames), it should be feasible to establish correspondence between successive frames (i.e. to use dynamic stereo) and to recover the shape of objects at places where even the combined approach failed; respectively, such a dynamic stereo step could replace the static one of this work. A related topic is to combine 3D data from several consecutive range images to a more complete scene model, for instance via stitching. Another is to use the information from such consecutive frames to improve decoding; e.g. let's consider a moving surface that reflects only a very small and consequently unidentifiable fraction of a projected subpattern  $S_1$  back into the camera at the time the first frame is acquired. In the next frame, the small surface is at a slightly different position in 3D space; so it is illuminated by and reflects a distinct non-decodable subpattern fragment  $S_2$ . With slow movement,  $S_1$  and  $S_2$  must be close to each other on the pattern slide. In the best case, this information already suffices to identify the two fragments; otherwise additional frames could be consulted. Clearly such an approach is principally capable of decoding pattern fragments that are incomplete or corrupted when considering a single frame only.
2. The visible active illumination of the proposed method is inconvenient for certain purposes. Future work might attempt to use an infrared one instead, either adopting a monochromatic or a multi-spectral approach. The advantages of the former are its modest hardware requirements: a conventional black-and-white camera and a simple slide projector with suitable cut-off filter would suffice. It would be a topic of future work to adapt the encoding and data processing techniques of this work to the monochromatic case. With a multi-spectral infrared approach, the encoding and the algorithm could remain unchanged. In its case, the difficulties would lie on the hardware side instead, i.e. with camera and projector. Recently, multi-spectral consumer cameras (for three distinct spectral bands) have become available whose spectral sensitivity can in principle be adjusted to arbitrary wavelength intervals [Foveon 2004]. They could for example be used to acquire the multi-spectral infrared images besides less elegant beam-splitting approaches. The issue of how to project the pattern remains, but is clearly solvable.
3. The stereo step in its current form does not fully exploit the potential of a combined CL/stereo approach as proposed in this work. Improving it, e.g. in order to deal with specular reflection or to get an at least low-accuracy estimate of the scene background, seems to be a promising area of future research. The same applies to using more than two cameras, i.e. developing a combined CL/n-ary stereo approach.
4. The proposed approach to range image acquisition has intentionally been kept general, i.e. it makes as few assumptions about the scene as possible. With most applications, the type of scene to be expected is known beforehand. For instance, an acquisition system part of a face recognition system needs to acquire depth maps of faces only, and it is irrelevant if the system does not work well with anything that is not a face. In such a case, exploiting very specific a-priori knowledge, e.g. in the form of an application-specific scene model (such as a generic face model in the case of face recognition), or at least stronger constraints such as a global continuity assumption could significantly improve the performance of the approach.
5. An affordable high-res real-time range acquisition system operating under real-world conditions opens up the way to many applications. Examples from the field of human-machine interfaces are 3D gesture, face or hand recognition. Future work could develop corresponding applications based on the range acquisition system of this work and investigate systematically in as far range data permits overcoming the limitations of intensity-only approaches.

### 7.3 Summary of Contributions

The following lists the contributions of new knowledge made by this thesis:

1. **Range Image Acquisition:** Developed a new approach to obtain range images
  - of almost all practically relevant, dynamic scenes up to a few meters away from the acquisition system
  - under real-world conditions, in particular with non-negligible background illumination
  - with high accuracy, e.g. in case of the evaluated prototype with an exemplary standard deviation of ca. 0.2 mm from the ground truth over a cubical working space of about 0.5 m side length
  - of almost arbitrarily high relative lateral resolution, that is one which is – within certain limits – proportional to the resolution of the camera(s) used
  - in real-time, e.g. with up to 25 range images (of a resolution of 780 by 580 pixels) per second with the evaluated prototype system on a standard PC
  - that can be integrated using only low-cost off-the-shelf components such as one-chip color cameras

Achieved this by combining the existing techniques of coded light and stereo vision to a new approach that avoids most drawbacks traditionally associated with them.

2. **Coded Light Approach:** Developed a new approach to coded light; in more detail:
  - Demonstrated how color coded light based on a single static projection pattern can be used to acquire range maps of most practically relevant scenes; also that this technique is – contrary to what is stated in the literature – able to cope with strongly colored scenes, respectively significant ambient illumination.
  - Developed a suitable projection pattern to that end; more technically, concluded from an abstract scene model that encoding via local color edge patterns is a way to overcome the said limitations; established the properties a projection pattern should ideally have so that an algorithm is able to demodulate and decode it from an image even given adverse conditions. Arrived in this context at certain conclusions that differ strongly from those of existing work, e.g. that synchronization of projected and received signal is the key issue with coded light. Derived a corresponding new type of projection pattern furthermore allows a very high lateral resolution of the range data. Proved that this pattern type permits detecting if the reflection of a local color edge pattern is disrupted in virtually all practically relevant cases.
  - Contributed a pseudo-random algorithm that is capable of generating such complex codes, respectively projection patterns; established certain theoretical properties that simplify this task, respectively permit obtaining such codes in an acceptable time-span.
  - Introduced an algorithm to robustly convert a color image of a scene illuminated by the proposed pattern into a range image in real-time; achieved this not via a standard approach to edge detection, but one that exploits the pattern's error detection capabilities to detect only projected color edges and that is moreover able to correct certain errors, i.e. to decode corrupted local color edge patterns.
3. **Stereo Vision:** Established that stereo algorithms are typically well suited for obtaining range values for the parts of the scene where the coded light step fails; also that such algorithms are capable of operating in real-time if these problematic parts make up only a small percentage of the total scene. Presented a corresponding stereo algorithm to complement the coded light step, i.e. in total a two-stage ranging technique suited for almost all practically relevant scenes up to

a few meters away from the acquisition system. Introduced a technique to combine the range data obtained by two coded light systems that use different cameras, but share the same projector (i.e. with a two cameras - one projector set-up as with the proposed combined coded light/ stereo system). Described how a stereo algorithm can be used to increase the lateral resolution of a range map obtained with the coded light step by computing the dense sub-pixel correspondence in-between color edge segments.

4. **Additional Functionality of the Range Image Acquisition System:** Showed how the original scene reflectivity can be re-computed given a color image of a scene illuminated by a color stripe or color grid pattern, i.e. how to compute an image that appears as if the illuminating pattern had been all white. Respectively, starting out from previous work, how to compensate the effect of the scene's reflection properties on a color image, i.e. how to compute an image that appears as if each imaged scene patch had the color of the illuminating ray of projection. Introduced a technique that permits a fully autonomous adaptation of the algorithm thresholds and camera settings to the current conditions, respectively scene properties, by exploiting the error detection capabilities of the approach.
5. **Camera and Projector Calibration:** Developed a new camera calibration technique (by extending Tsai's well-known monoview camera calibration technique to one based on several views) that works with a simple planar calibration target; carried out experiments that indicate the technique produces better results than comparable state-of-the-art calibration methods. Showed how to extend the above camera calibration technique to the task of projector calibration, yielding an accurate projector calibration technique and in sum a method to calibrate a structured light system with high accuracy. Presented experiments that confirm this inference. Demonstrated in this context that it is necessary to consider the radial distortion of the projector's lens to obtain accurate range maps; introduced a new way to efficiently compensate this radial distortion.
6. **Theoretical Accuracy Analysis of Structured Light/ Triangulation Systems:** Presented a parameterized model of a triangulation system; listed the causes that lead to the measurement uncertainty of such a system; carried out an analysis of the measurement error for a given set of parameters, including the complex case of a non-standard geometric set-up, whose results may e.g. be used to predict the standard deviation in the separate measured coordinates for a certain choice of set-up parameters. Presented several formulas (derived via approximation) that make the effect of the key parameters of a convergent, non-standard geometry – e.g. the choice of convergence angle and projector position – intuitively understandable and simple to estimate.
7. **Experimental Evaluation:** Presented an experimental evaluation of a prototype implementation based on the proposed ranging approach that verifies most of the above claims.
8. **Face Recognition:** Acquired a 3D and color face database with 2,711 recordings of 20 individuals with the proposed ranging method that – within certain limits – adheres to international guidelines for evaluating biometric techniques. Described experiments with the above face database that indicate that the performance of a standard approach to face recognition – one based on embedded Hidden-Markov-Models – is significantly improved by the use of range along with intensity data (compared to an intensity-only version of the approach).
9. **Literature Survey:** Introduced a common framework to formally describe all coded light systems described in the literature; used the parameters of this framework to systematically classify the existing work on coded light into clear-cut categories and characterized the principal (dis)advantages of each category and their causes.





## 8 References

- [Agin and Binford 1973] Agin G. J. and Binford T.O. Computer Description of Curved Objects. Proceedings IJCAL: 629-640, 1973
- [Albertz 1991] Albertz J. Grundlagen der Interpretation von Luft- und Satellitenbildern. Wissenschaftliche Buchgesellschaft, Darmstadt. 1991
- [Altschuler et al. 1979] Altschuler M.D., Altschuler B.R. and Taboada J. Measuring Surfaces Space-Coded by a Laser-Projected Dot Matrix. Imaging Application for Automated Industrial Inspection: 1979
- [Asada et al. 1985] Asada M. and Tsuji S. Utilization of a Stripe Pattern for Dynamic Scene Analysis. Proceedings IJCAI: 895-897. 1985
- [Asada et al. 1986] Asada M., Ichikawa H. and Tsuji S. Determining of Surface Properties by Projecting a Stripe Pattern. Proceedings ICPR: 1162-1164. 1986
- [Asada et al. 1988] Asada M., Ichikawa H. and Tsuji S. Determining Surface Properties by Projecting a Stripe Pattern. IEEE PAMI 10(5): 749-754. 1988
- [Ayache 1991] Ayache N. Artificial Vision for Mobile Robots. The MIT Press, Cambridge, MA. 1991
- [Baker et al. 1998] Baker S., Szeliski R. and Anandan P. A Layered Approach to Stereo Reconstruction. ICCVPR: 434-441. 1998
- [Ballard and Brown 1982] Ballard D. and Brown C. Computer Vision. Prentice Hall, Inc. New Jersey. 1982
- [Battle et. al 1997] Battle J., Mouaddib E. and Salvi J. A Survey: Recent Progress in Coded Structured Light as Technique to Solve the Correspondence Problem. Pattern Recognition: 31(7): 963-982. 1997
- [Bergmann and Schäfer 1987] Bergmann L. and Schäfer C. Lehrbuch der Experimentalphysik, Band 3, Optik. Walter de Gruyter, Berlin. 1987 (8th Edition)
- [Beumier and Acheroy 1999] Beumier C. and Acheroy M. 3D Facial Surface Acquisition by Structured Light. Proceedings International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging: 103-106. 1999
- [Bhat and Nayar 1998] Bhat D. N. and Nayar S. K. Ordinal Measures for Image Correspondence. IEEE PAMI 20(4): 415-423. 1998
- [BWG 2000] Biometrics Working Group (2000). Best Practices in Testing and Reporting Performance of Biometric Devices. Version 1.0, <<http://www.cesg.gov.uk/assurance/iacs/itsec/documents/protection-profiles/media/BBP.pdf>>. 01.08.2000

- [BWG 2002] Biometrics Working Group (2002). Best Practices in Testing and Reporting Performance of Biometric Devices. Version 2.01. [http://www.cesg.gov.uk/technology/biometrics/media/Best\\_Practice.pdf](http://www.cesg.gov.uk/technology/biometrics/media/Best_Practice.pdf). 08.01.2002
- [Blostein and Ahuja 1989] Blostein S. D. and Ahuja N. Shape from Texture: Integrating Texture-Element Extraction and Surface Estimation. IEEE PAMI 11(12):1233-1251. 1989
- [Blostein and Huang 1987] Blostein S. D. and Huang T. S. Error Analysis in Stereo Determination of 3-D Point Positions. IEEE Pami 9(6): 752-765. 1987
- [Boyer and Kak 1987] Boyer K.L. and Kak A.C. Color-Encoded Structured Light for Rapid Active Ranging. IEEE PAMI 9(1): 14-28. 1987
- [Boykov et al. 1998] Boykov Y., Veksler O. and Zabih R. A Variable Window Approach to Early Vision. IEEE PAMI 20(12): 1283-1294. 1998
- [Boykov et al. 2001] Boykov Y., Veksler O. and Zabih R. Fast Approximate Energy Minimization via Graph Cuts. IEEE PAMI 23(11): 1222-1239. 2001
- [Canesta 2002] Canesta. Product Information. <http://www.canesta.com>. 11.12.2003
- [Canny 1986] Canny J. A Computational Approach to Edge Detection. IEEE PAMI 8(6): 679-698. 1986
- [Carrhill and Hummel 1985] Carrhill B. and Hummel R., Experiments with the Intensity Ratio Depth Sensor. Computer Vision, Graphics and Image Processing 32: 337-358. 1985
- [Caspi et al. 1998] Caspi D., Kiryati N. and Shamir J. Range Imaging with Adaptive Color Structured Light. IEEE PAMI 20 (5): 470-480. 1998
- [Chellappa et al. 1995] Chellappa R., Wilson C. L. and Sirohey S. Human and Machine Recognition of Faces. A Survey. Proceedings of the IEEE 83(5): 705-740. 1995
- [Chang et al. 1994] Chang C., Chatterjee S. and Kube P. R. A Quantization Error Analysis for Convergent Stereo. IEEE 735-739. 1994
- [Chen et al. 1997] Chen C.S., Hung Y. P., Chiang C.C. and Wu J.L.. Range Data Acquisition Using Color Structured Lighting and Stereo Vision. IJ on Image and Vision Computing 15: 445-456. 1997
- [Cochran and Medioni 1992] Cochran S. D. and Medioni G. 3-D Surface Description from Binocular Stereo. IEEE PAMI 14(10): 981-994. 1992
- [Creath 1988] Creath K. Phase Measurement Interferometry Techniques. Progress in Optics XXVI: 351-393. E. Wolf, ed. Elsevier, New York 1988
- [Cumani 1991] Cumani A. Edge Detection in Multispectral Images. Graphical Models and Image Processing 53(1): 40-51. 1991
- [Dändliker et al. 1995] Dändliker R., Hug K., Politch J. and Zimmermann E. High Accuracy Distance Measurements with Multiple-Wavelength Interferometry. Optical Engineering 34(8). 1995
- [Davies and Nixon 1994] Davies C. J. and Nixon M.S. Coloured Spots for Three Dimensional Sensing, Signal Processing VII(3): 1401-1404. 1994
- [Davies and Nixon 1998] Davies C. J. and Nixon M.S.. A Hough Transform for Detecting the Location and Orientation of 3D Surfaces Via Color Encoded Spots, IEEE Transactions on Systems, Man and Cybernetics 28 (1): 90-95. 1998
- [De Bruijn 1946] De Bruijn N.G. A Combinatorial Problem. Proceedings Nederlandsche Akademie van Wetenschappen. 49: 758-764. 1946
- [Dehning 2004] Dehning O. Volume Imprint. Private Communication. 2004

- [Ens and Lawrence 1993]      Ens J. and Lawrence P. An Investigation of Methods for Determining Depth from Focus. IEEE PAMI 15(2): 97-108. 1993
- [Faugeras 1993]              Faugeras O. Three-Dimensional Computer Vision. A Geometric Viewpoint. The MIT Press, Cambridge, Mass. 1993
- [Forster et al. 2001a]        Forster F., Rummel P., Lang M., and Radig B.: The Hiscore Camera: A Real-Time Three-Dimensional and Color Camera. Proceedings International Conference on Image Processing: 598-601 (Vol. 2). 2001
- [Forster et al. 2001b]        Forster F., Lang M. and Radig B.: Real-Time 3D and Color Camera. Proceedings International Conference on Augmented, Virtual Environments and 3D Imaging: 45-48. 2001
- [Forster et al. 2002]        Forster F., Lang M. and Radig B.: Real-Time Range Imaging for Dynamic Scenes Using Colour-Edge Based Structured Light. Proceedings International Conference on Pattern Recognition: 645-648 (Vol. 3). 2002
- [Forster et al. 2003a]        Forster F., Rummel P. und Hoffmann C.: Method And Device for Three-Dimensionally Detecting Objects and The Use of this Device and Method. International Patent Application WO 2004/010076 A1. July 17, 2003
- [Forster et al. 2003b]        Forster F., Doemens G., Rummel P. and Niederdränk T.: Selbstkalibrierender Rundum-Scanner für Ohrabdrücke. Eingereichte Patentanmeldung. Amtliches Aktenkennzeichen 103 44922.1. 2003
- [Forster et al. 2003c]        Forster F., Laloni C. and Jahn L.: Optische 3-Dimensionale Achsgeometrievermessung durch Triangulation mit Codiertem Licht. Eingereichte Patentanmeldung. Amtliches Aktenkennzeichen 103 35829.3. 2003
- [Forster 2004]                Forster F.: Vorrichtung und Verfahren zur Bestimmung von Raumkoordinaten eines Objekts. Eingereichte Patentanmeldung. Amtliches Aktenkennzeichen 10 2004 008 904.3. 2004
- [Forster 2005]                Forster F.: Accurate Calibration with A Planar Calibration Target. Submitted for publication. 2005
- [Foveon 2004]                Foveon – X3 Technology. <[http://www.foveon.com/X3\\_tech.html](http://www.foveon.com/X3_tech.html)>. 05.05.2004
- [Forsyth and Zisserman 1991]      Forsyth D. and Zisserman A. Reflections on Shading. IEEE PAMI 13(7): 671-679. 1991
- [Fusiello et al. 1997]        Fusiello A., Roberto V. and Trucco E. Efficient Stereo with Multiple Windowing. Proceedings ICCVPR: 858-663. 1997
- [Geng 1996]                  Geng Z.J. Rainbow Three-Dimensional Camera: New Concept of High-Speed Three-Dimensional Vision Systems. Optical Engineering 35 (2): 376-383. 1996
- [Geng 1995]                  Geng Z.J. Color Ranging Method for High Speed Low-Cost Three Dimensional Surface Profile Measurement. United States Patent 5 675 407
- [Gerthsen 1960]              Gerthsen C. Physik. Springer Verlag, Berlin. 1960 (6th Edition)
- [Golomb 1967]                Golomb, S. W. Shift Register Sequences. Holden-Day, San Francisco, California. 1967.
- [Goresky and Klapper 2004]      Goresky M. and Klapper A. Polynomial Pseudo-Noise Sequences Based On Algebraic Feedback Shift Registers. <<http://www.math.ias.edu/~goresky>>. 1.23.2004
- [Griffin 1958]                Griffin D.R. Listening in the Dark: The Acoustic Orientation Of Bats and Man. Yale University Press, New Haven. 1958

- [Griffin and Yee 1991] Griffin P.M. and Yee S.R. The Use of a Uniquely Encoded Light Pattern for Range Data Acquisition. *Computers ind. Engineering* 21(1-4): 359-363. 1991
- [Griffin et al. 1992] Griffin P.M., Narasimham L.S. and Yee S.R. Generation of Uniquely Encoded Light Patterns for Range Data Acquisition. *IJ on Pattern Recognition* 25 (6): 609-616. 1992
- [Grimson 1985] Grimson W.E.L. Computational Experiments with a Feature Based Stereo Algorithm. *IEEE PAMI* 7 (1): 17 – 34. 1985
- [Grum and Bartelson 1980] Grum F. And C. J. Bartelson (Editors). *Optical Radiation Measurement, Volume 2, Color Measurement*. Academic Press, New York. 1980
- [Guehring 2001] Guehring J. Dense 3D Surface Acquisition by Structured Light Using Off-the-Shelf Components. *Proceedings SPIE Vol. 4309*: 220-231. 2001
- [Hallam 2002] Hallam S. SparkNotes On Geometric Optics. <http://www.sparknotes.com/physics/optics/geom/terms.html>. 29.8.2002
- [Hall-Holt and Rusinkiewicz 2001] Hall-Holt O. and Rusinkiewicz S. Stripe Boundary Codes for Real-Time Structured Light Range Scanning of Moving Objects. *Proceedings ICCV*: 359-366. 2001
- [Halioua and Liu 1989] Halioua M. and Liu H.-C. Optical Three-Dimensional Sensing by Phase Measuring Profilometry. *Optics and Lasers in Engineering* 11: 185-215. 1989
- [Hassebrook et al. 1997] Hassebrook L.G., Daley R.C. and Chimitt Jr W.J. Application of Communication Theory to High Speed Structured Light Illumination, *SPIE Vol. 3204*: 102-113. 1997
- [Hattori and Sato 1996] Hattori K. and Sato Y. Accurate Rangefinder with Laser Pattern Shifting. *Proceedings ICIP*: 849-853. 1996
- [Häusler and Ritter 1993] Häusler G. and Ritter D. Parallel Three-Dimensional Sensing by Color-Coded Triangulation. *Applied Optics* 32 (35): 7164-7169. 1993
- [Hecht and Zajaac 1974] Hecht E. and Zajaac A. *Optics*. Addison-Wesley Publishing Company, Inc., New York. 1974
- [Hirschmüller 2001] Hirschmüller H. Improvements in Real-Time Correlation-Based Stereo Vision. *Proceedings of IEEE Workshop on Stereo and Multi-Baseline Vision*: 141-148. 2001
- [HISCORE 2003] Küpper W. and Schuster M. Deliverable D16.2. Evaluation Report – Evaluation of the Face Recognition System. IST HISCORE Project Report. 2003
- [Hobson et al. 1997] Hobson C. A., Yow K.C. and Pearson J.D. Filters and Bandwidth for 3D Shape Measurement using Structured Light. *Proceedings IPA*: 224-228. 1997
- [Horn 1968] Horn B.K.P. Focusing. Project MAC, AI Mem. 160 MIT, Cambridge, Mass. 1968
- [Horn 1977] Horn B.K.P. Understanding Image Intensities. *Artificial Intelligence* 8: 201-231. 1977
- [Horn 1986] Horn B.K.P. *Robot Vision*. The MIT Press, Cambridge, Mass. 1986
- [Horn and Brooks 1989] Horn B.K.P and Brooks M.J. (eds.). *Shape from Shading*. MIT Press, Cambridge, Mass. 1989
- [Horn and Kiryati 1998] Horn E. and Kiryati N. Towards Optimal Structured Light Patterns. *Image and Vision Computing* 17(2): 87-97. 1999

- [Horn 2000] Horn B.K.P. Tsai's Camera Calibration Revisited. <http://www.ai.mit.edu/people/bkph/papers/tsaiexplain.pdf>. 20.11.2003
- [Hu and Stockmann 1989] Hu G. and Stockmann G. 3D Surface Solution Using Structured Light and Constraint Propagation. IEEE PAMI 11(4): 390-402. 1989
- [Huang et al. 1999 ] Huang P.S. et alii. Color Encoded Digital Fringe Projection Technique for High-Speed Three-Dimensional Surface Contouring. Optical Engineering 38 (6): 1065-1071. 1999
- [Hügli and Maitre 1988] Hügli H. and Maitre G. Generation and Use of Color Pseudo Random Sequences for Coding Structured Light in Active Ranging. SPIE Vol. 1010 Industrial Inspection, 1988
- [Hülsmeier 1904] Hülsmeier C. Verfahren, um entfernte metallische Gegenstände mittels elektrischer Wellen einem Beobachter zu melden. Kaiserliches Patentamt, Patent Nr. 165546, 1904
- [Hung 1993] Hung D.C.D. 3D Scene Modelling by Sinusoid Encoded Illumination. . Image and Vision Computing 11(5): 251-256. 1993
- [Inokuchi et al. 1984] Inokuchi S., Sato K. and Matsuda F. Range Image System for 3D Object Recognition. Proceedings ICPR: 806-808. 1984
- [Intel 2001] Intel Corporation. Open Source Computer Vision Library. Reference Manual. Intel Corporation. 2001
- [Ito and Ishi 1995] Ito M. and Ishi A. A Three-Level Checkerboard Pattern (TCP) Projection Method for Curved Surface Measurement. Pattern Recognition 28(1): 27-40. 1995
- [Jähne 2002] Jähne B. Digitale Bildverarbeitung. Springer Verlag, Berlin. 2002 (5th Edition)
- [Jiang and Bunke 1997] Jiang X. and Bunke H. Dreidimensionales Computersehen. Springer Verlag, Berlin. 1997
- [Kanade et al. 1991] Kanade T., Gruss A. and Carley L.R. A Very Fast VLSI Range-Finder. Proceedings IC on Robotics and Automation: 1322-1329. 1991
- [Kanade et Okutomi 1996] Kanade T. and Okutomi M. A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiments. IEEE PAMI 16(9): 920-932. 1996
- [Kang et al. 1995] Kang S.B. et. al. A Multibaseline Stereo System with Active Illumination and Real-Time Image Acquisition. Proceedings ICCV: 88-93. 1995
- [Kellogg 1961] Kellogg W. N. Porpoises and Sonar. University of Chicago Press, Chicago, 1961
- [Klein 1993] Klein R. Ein laseroptisches Entfernungsmessverfahren mit frequenzvariabler Pseudo-Noise-Modulation. PhD Thesis, Universität Siegen. 1993
- [Klette et al. 1996] Klette R., Koschan A. and Schlüns K: Computer Vision. Vieweg Verlagsgesellschaft, Braunschweig/Wiesbaden. 1996
- [Klunker et al. 1990] Klunker G., Shafer S. and Kanade T. A Physical Approach to Color Image Understanding. IJ On Computer Vision 4: 7-38. 1990
- [Koschan et al. 1996] Koschan A, Rodehorst V. and Spiller K. Color Stereo Vision Using Hierarchical Block Matching and Active Color Illumination. ICPR: 835-839. 1996
- [Kuo and Agazzi 1994] Kuo S. and Agazzi O. Keyword Spotting in Poorly Printed Documents using Pseudo 2-D Hidden Markov Models. IEEE PAMI 16(8): 842-848. 1994

- [Lange 2000] Lange R. 3D Time-Of-Flight Distance Measurement with Custom Solid – State Image Sensors in CCD/CMOS-Technology. PhD Thesis, Universität Siegen. 2000
- [Larkin and Oreb 1992] Larkin K.G. and Oreb B.F. Design and Assessment of Symmetrical Phase-Shift Algorithms. *Journal of the Optical Society of America A* 9(10): 1740-1748. 1992
- [Le Moigne and Waxman 1984] Le Moigne J. and Waxman A. Projected Light Grids for Short Range Navigation of Autonomous Robots. *Proceedings ICPR*: 203-206. 1984
- [Lee and Rosenfeld 1985] Lee C. H. and Rosenfeld A. Improved Methods of Estimating Shape from Shading Using the Light Source Coordinate System. *Artificial Intelligence* 26: 125-143. 1985
- [Levi 1968] Levi L. *Applied Optics*. John Wiley & Sons, Inc., New York. 1968
- [Levine et. al. 1973] Levine M.D., O’Handley D.A. and Yagi G.M. Computer Determination of Depth Maps. *Computer Graphics Image Processing* 2: 131-150. 1973
- [Lin 2002] Lin M.H. Surfaces with Occlusions from Layered Stereo. PhD Thesis, Stanford University. 2002
- [Luhmann 2000] Luhmann T. *Nahbereichsphotogrammetrie: Grundlagen, Methoden und Anwendungen*. Wichmann, Heidelberg. 2000
- [Machuca and and Phillips 1983] Machuca R. and Phillips K. Applications of Vector Fields to Image Processing. *IEEE PAMI* 5: 316-329. 1983
- [MacWilliams and Sloane 1976] MacWilliams F.J. and Sloane N.J.A. Pseudo-Random Sequences and Arrays. *Proceedings IEEE* 64(12): 1715-1729. 1976
- [Malassiotis et al. 2003] Malassiotis et al.: IST-1999-10087-HISCORE. Final Project Management Report - Public Version. Project Report, Document ID HISCORE-IST-10087/SIE/R/PUD017.1/a1. 2003
- [March 1988] March R. Computation of Stereo Disparity Using Regularization. *Pattern Recognition Letter* 8: 181-187. 1988
- [Marr and Poggio 1976] Marr D. and Poggio T. A Co-Operative Computation of Stereo Disparity. *Science* 194: 283-297. 1976
- [Marr and Poggio 1979] Marr D. and Poggio T. A Computational Model of Human Stereovision. *Proceedings of the Royal Society of London B*, 204: 301-328. 1979
- [Maruyama and Abe 1993] Maruyama M. and Abe S. Range Sensing by Projecting Multiple Slits with Random Cuts. *IEEE PAMI* 15(6): 647-651. 1993
- [McIvor 1994] McIvor A.M. The Accuracy of Range Data from a Structured Light System. Industrial Research Limited Report 190. Auckland, New Zealand. 1994
- [Mengel et al. 2001] Mengel P., Doemens G., Listl L. Fast Range Imaging By CMOS Sensor Array. *Proceedings ICIP*: 169-172. 2001
- [Minou et al. 1981] Minou M., Kanade T. and Sakai T. A Method of Time-Coded Parallel Planes of Light for Depth Measurement. *Transactions of the IECE of Japan*, Vol. E64(8): 521-528. 1981
- [Monks 1994] Monks T. Measuring the Shape of Time-Varying Objects. PhD Thesis, University of Newcastle. 1994
- [Morano et al. 1998] Morano R.A. et alii. Structured Light Using Pseudorandom Codes. *IEEE PAMI* 20 (3): 322-327. 1998
- [Moré 1980] Moré J., Garbow B. S. and Hillstrom K. E.. User guide for MINPACK-1. Technical Report 80-74, Argonne National Laboratory, USA, 1980

- [Morita et al. 1988] Morita H., Yakima K. and Sakata S.: Reconstruction of Surfaces of 3D Objects by M-Array Pattern Projection Method. Proceedings ICIP: 468-473. 1988
- [Nayar et al. 1991] Nayar S. K., Ikeuchi K. and Kanade T. Surface Reflection: Physical and Geometrical Perspectives. IEEE PAMI 13(7): 611-634. 1991
- [Nayar et al. 1996] Nayar S., Watanabe M. and Noguchi M. Real-Time Focus Range Sensor. IEEE PAMI 18(12): 1186-1197. 1996
- [Nefian 1999] Nefian A. A Hidden Markov Model-Based Approach for Face Detection and Recognition. PhD Thesis, Georgia Institute of Technology. 1999
- [Negahdaripour 1998] Negahdaripour S. Revised Definition of Optical Flow. IEEE PAMI 20(9): 961-979. 1998
- [Nguyen and Huang 1992] Nguyen T.C. and Huang T.S. Quantization Errors in Axial Motion Stereo on Rectangular Tessellated Image Sensors. Proceedings ICPR: 13-16. 1992
- [NIST 2002] National Institute for Standardization. <http://physics.nist.gov/cuu/Units/current.html>. 05.05.2002
- [Notni et al. 1997] Notni G., Recknagel R. J., Gerold F. and Trefz M. Anwendungen der Weißlichtinterferometrie. 4. ABW Workshop 3D BV. 1997
- [Pagoda Systems 2004] Pagoda Systems. Combination of Several Range Maps to a Full 3D Model of an Ear-Impression. Private Communication. 2004
- [Paterson 1994] Patterson K.G. Perfect Maps. IEEE Transactions on Information Theory 40(3): 743-753. 1994
- [Paul and Stahnke 2000] Paul L and Stahnke G. Visuelle Echtzeitvermessung räumlicher Deformationsprozesse durch spektral kodierte Szenenbeleuchtung. VDI Berichte Nr. 1572: 59-66. 2000
- [Pearson 1996] Pearson J.D, Automated Visual Measurements of Body Shape in Scoliosis. PhD Thesis, Liverpool. 1996
- [Pentland 1984] Pentland A. P. Local Shading Analysis. IEEE PAMI 6(2): 170-187. 1984
- [Phong 1975] Phong, B.T. Illumination for Computer Generated Pictures. Communications of the ACM 18(6): 311-317. 1975
- [Pollard et al. 1985] Pollard S. B., Mayhew J. E. W. and Frisby J. P. PMF: A Stereo Correspondence Algorithm Using a Disparity Gradient Limit. Perception 14(4): 449-470. 1985
- [Porr et al. 2002] Porr B., Nuerenberg B. and Woergoetter F. A VLSI-Compatible Computer Vision Algorithm for Stereoscopic Depth Analysis in Real-Time. IJ on Computer Vision 49(1): 39-55. 2002
- [Potsdamer and Altschuler 1982] Potsdamer J. L. and Altschuler M. D. Surface Measurements by Space Encoded Projected Beam Systems. Computer Graphics Image Processing Vol. 18: 1-17. 1982
- [Pratt 1991] Pratt W.K. Digital Image Processing. John Wiley & Sons Inc., New York. 1991 (2<sup>nd</sup> Edition)
- [Press et al. 2002] Press W., Teukolsky S., Vetterling W. and Flannery B. Numerical Recipes in C++: The Art of Scientific Computing. Cambridge University Press, Cambridge. 2002 (2<sup>nd</sup> Edition)
- [Proesmans et al. 1996a] Proesmans M., Van Gool L.J., Oosterlinck A. One-Shot Active 3D Shape Acquisition. Proceedings ICPR: 336-340. 1996

- [Proesmans et al. 1996b] Proesmans M., Van Gool L.J., Oosterlinck A. Active Acquisition of 3D Shape for Moving Objects. Proceedings ICIP: 647-650. 1996
- [Proesmans et al. 1998] Proesmans M., Van Gool L.J., Defoort F. Reading Between the Lines: A Method for Extracting Dynamic 3D with Texture. Proceedings ICCV: 1081-1086. 1998
- [Rabiner and Huang 1993] Rabiner L. and Huang B. Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs, NJ. 1993
- [Rathjen 1995] Rathjen C. Statistical Properties of Phase-Shift Algorithms. Journal of the Optical Society of America A 12(9): 1997-2008. 1995
- [Rioux and Blais 1986] Rioux M. and Blais F. Compact 3D Camera for Robotic Applications. Journal of the Optical Society of America A 3(9): 1518-1521. 1986
- [Roach and Aggarwal 1980] Roach J. W. and Aggarwal J.K. Determining the Movement of Objects from a Sequence of Images. IEEE PAMI 2(6): 554-562. 1980
- [Roberts and Deriche 1996] Roberts L. and Deriche R. Dense Depth Map Reconstruction: a Minimization and Regularization Approach which Preserves Discontinuities. ECCV(A): 439-451. 1996
- [Rodriguez and Aggarwal 1990] Rodriguez J. J. and Aggarwal J.K. Stochastic Analysis of Stereo Quantization Error. IEEE PAMI 12(5): 467-470. 1990
- [Salvi et al. 1988] Salvi J., Batlle J. and Mouaddib E. A Robust-Coded Pattern Projection for Dynamic 3D Scene Measurement. Pattern Recogn. Letters 19 (11). 1998
- [Samaria 1994] Samaria F. Face Recognition Using Hidden Markov Models. PhD Thesis, University of Cambridge. 1994
- [Sanz 1989] Sanz J. L. C. Advances in Machine Vision. Springer Verlag, New York. 1989
- [Sato et al. 1986] Sato K., Yamamoto H. and Inokuchi S. Tuned Range Finder for High Precision 3D Data. Proceedings ICPR: 1168-1171. 1986
- [Sato and Inokuchi 1987] Sato K. and Inokuchi S. Range-Imaging System Utilizing Nematic Liquid Crystal Mask. Proceedings ICCV: 657-661. 1987
- [Scharstein and Szeliski 2002] Scharstein D. and Szeliski R. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. IJ on Computer Vision 47(1/2/3): 7-42. 2002
- [Schmallfuss et al. 2002] Schmallfuss H., Reiter M. and Simon A. Neue Sensoren für die schnelle Lichtschnitt-Datenaufnahme, Proc. 7. ABW-Workshop, Esslingen. 2002
- [Schmaltz 1936] Schmaltz G. A Method for Presenting the Profile Curves of Rough Surfaces, Naturwissenschaften 18: 315-316. 1932
- [Schubert 1996] Schubert E. Mehrfachfarbcodierte Triangulationsverfahren zur Topometrischen Erfassung und Vermessung von 3D-Objekten. PhD Thesis, Universität Siegen. 1996
- [Schwarte 1995] Schwarte R, Heinol H.G., Xu Z. and Hartmann K. A New Active 3D Vision System Based on rf-Modulation Interferometry of Incoherent Light. Proceedings of SPIE 2588: 126-134. 1995
- [Shafer 1985] Shafer S. Using Color to Separate Reflection Components. Color Research and Applications 10(4): 210-218. 1985
- [Shirai and Suwa 1971] Shirai Y. and Suwa M. Recognition of Polyhedrons with a Range Finder. Proc. 2nd Int. Joint Conference on Artificial Intelligence: 80-87. 1971
- [Shirai 1972] Shiray Y. Recognition of Polyhedrons Using a Range Finder. Pattern Recognition 4:243-250. 1972



- [Shirai 1987] Shiray Y. Three-dimensional Computer Vision. Springer Verlag, Berlin. 1987
- [Shrikhande and Stockmann 1989] Shrikhande N. and Stockmann G. Surface Orientation from a Projected Grid. IEEE PAMI 11(6): 650-655. 1989
- [Slama et al. 80] Slama C., Theurer C. and Hendriksen S. (Editors). Manual of Photogrammetry. American Society of Photogrammetry, Falls Church, Va. 1980 (4th Edition)
- [Smutny and Pajdla 1996] Smutny V. and Pajdla T. Rainbow Range Finder and its Implementation at the CVL. Technical Report Nr. K335-96-130, Czech Technical University, Prague. 1996
- [Stevens 1979] Stevens K. Presenting and Analyzing Surface Orientation. In: Artificial Intelligence: An MIT Perspective (Winston P.H. and Brown R.H., Editors). Vol. 2. The MIT Press, Camebridge. 1979
- [Strutz 1993] Strutz T. Ein genaues aktives Bildtriangulationsverfahren zur Oberflächenvermessung. PhD Thesis, Technische Universität Magdeburg. 1993
- [Sweeney 1991] Sweeney P. Error Control Coding: An Introduction. Prentice Hall. 1991
- [Tajima 1987] Tajima J. Rainbow Range Finder Principle for Range Data Acquisition. Proceedings IEEE Workshop on Industrial Applications of Machine Vision and Machine Intelligence: 381-386. 1987
- [Tajima and Iwakawa 1990] Tajima J. and Iwakawa M. 3D Data Acquisition by Rainbow Range Finder. Proceedings ICPR: 309-313. 1990
- [Takasaki 1970] Takasaki H. Moiré Topography. Applied Optics 9(6): 1457-1472. 1970
- [Takeda and Mutoh 1983] M. Takeda and K. Mutoh. Fourier Transform Profilometry for the Automatic Measurement of 3-D Object Shapes," Applied Optics 22(24): 3977-3982. 1983
- [Teschner 1996] Teschner 1996. Rekonstruktion und Adaptive Integration von Tiefenkarten mit Lichtstreifenprojektion und Schwenk-Neige-Einrichtung. Diplomarbeit, TU Berlin. 1996
- [Torrance and Sparrow 1967] Torrance L. and Sparrow E. Theory for Off-Specular Reflection from Roughened Surfaces. Journal of the Optical Society of America 57(9): 1105-1114, 1967
- [Torre and Poggio 1986] Torre V. and Poggio T. On Edge Detection. IEEE PAMI 8(2): 147-163. 1986
- [Triggs 1998] Triggs B. Autocalibration From Planar Scenes. Proceedings ECCV: 89-105. 1998
- [Tsai 1986] Tsai R.Y. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. Proceedings CVPR: 22-26. 1986
- [Tsai 1987] Tsai R.Y. A Versatile Camera Calibration Technique for High-Accuracy 3D Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. IEEE Robotics and Automation 3(4): 323-344. 1987
- [Tsalakanidou et al. 2004] Tsalakanidou F, Malassiotis S. and Srinatzis M.G. Face Localization and Authentication Using Color and Depth Images", IEEE Transactions on Image Processing, accepted for publication.
- [Valkenburg and Mc Ivor 1996] Valkenburg R.J. and Mc Ivor A.M. Accurate 3D Measurement using a Structured Light System. Industrial Research Limited Report 576. Auckland, New Zealand. 1996
- [van Lint 1982] van Lint J.H. Introduction to Coding Theory. Springer-Verlag, New York. 1982

- [Vuylsteke and Oosterlinck 1990] Vuylsteke P. and Oosterlinck A. Range Image Acquisition with a Single Binary-Encoded Light Pattern. *IEEE PAMI* 12 (2):148-164. 1990
- [Wahl 1984] Wahl F.M. A Coded Light Approach for 3-Dimensional (3D) Vision. IBM Research Report: RZ 1452. 1984
- [Wang et al. 1987] Wang Y.F., Mitchie A. and Aggarwal J.K. Computation of Surface Orientation and Structure of Objects Using Grid Coding. *IEEE PAMI* 9(1): 129-137. 1987
- [Wang 1991] Wang Y.F. Characterizing Three-Dimensional Surface Structures from Visual Images. *IEEE PAMI* 13(1): 52-60. 1991
- [Wang and Cheng 1992] Wang Y.F. and Cheng D. I. Three-Dimensional Shape Construction and Recognition by Fusing Intensity and Structured Lighting. *Pattern Recognition* 25(12): 1411-1425. 1992
- [Wechsler 1990] Wechsler, H. *Computational Vision*. Academic Press, Boston. 1990
- [Weng et al. 1992] Weng J., Cohen P. and Herniou M. Camera Calibration with Distortion Models and Accuracy Evaluation. *IEEE PAMI* 14(10): 965-980. 1992
- [Will and Pennington 1971] Will P.M. and Pennington K.S. Grid Coding: A Preprocessing Technique for Robot and Machine Vision. *Proceedings of the IJC on Artificial Intelligence*: 66-70. 1971
- [Witkin 1981] Witkin A.P. Recovering Surface Shape and Orientation from Texture. *AI* 17(8): 17-45. 1981
- [Wong and Pugh 1987] Wong A.K.C. and Pugh A. *Machine Intelligence and Knowledge Engineering for Robotic Applications*. Springer Verlag, Heidelberg. 1987
- [Wust and Capson 1991] Wust C. and Capson D.W. Surface Profile Measurement Using Color Fringe Projection. *IJ on Machine Vision and Applications* 4: 193-203. 1991
- [Wyszecki and Stiles 1982] Wyszecki G. and Stiles W.S. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons Inc, New York, 1982 (2nd Edition)
- [Yonezawa and Tamamura 1978] Yonezawa S. and Tamamura Y. Coded Grating Method for Measuring Three-Dimensional Objects. *Transactions IEICE J61-D*, 411-418 (1978)
- [Zabih and Woodfill 1994] Zabih R. and Woodfill J. Non-Parametric Local Transforms for Computing Visual Correspondence. *Proceedings European Conference on Computer Vision*: 151-158. 1994
- [Zhang et al. 1999] Zhang R., Tsai P.S., Cryer J. E. and Shah M. Shape from Shading: A Survey. *IEEE PAMI* 21(8): 690-705. 1999
- [Zhang 1998] Zhang Z. A Flexible New Technique for Camera Calibration. Technical Report MSR-TR-98-71. Microsoft Research, Microsoft Corporation, Redmond, 1998
- [Zhang 2001] Zhang Z. A Flexible New Technique for Camera Calibration. *IEEE PAMI* 22(7): 1330-1334, 2000
- [Zhao et. al. 2003] Zhao W., Chellappa R., Rosenfeld A. and Phillips P.J. Face Recognition: A Literature Survey. *ACM Computing Surveys* 35(4): 399 - 458. 2003
- [Zheng and Capella 1991] Zheng Q. and Chellapa R. Estimation of Illuminant Direction, Albedo and Shape from Shading. *IEEE PAMI* 13(7): 680-702. 1991
- [Zitnick and Kanade 2000] Zitnick L. and Kanade T. A Cooperative Algorithm for Stereo Matching and Occlusion Detection. *IEEE PAMI* 22(7): 675-684. 2000