Technische Universität München
Chair for Computer-Aided Medical Procedures & Augmented Reality

Augmented Reality Tools for Digital Plant Engineering

Pierre Fite-Georgel

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende:		Univ.-Prof. G. J. Klinker, Ph.D.
Prüfer der Dissertation:		
	1.	Univ.-Prof. Dr. N. Navab
	2.	Prof. Dr. M. Pollefeys Eidgenössische Technische Hochschule Zürich / Schweiz

Die Dissertation wurde am 13.10.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 02.02.2011 angenommen.

Abstract

In this thesis, we tackle the engineering problem of *Discrepancy Check*: verifying the geometric correctness of a built object against its virtual model. Nowadays, manufactured products are first designed using Computer-Aided Design (CAD) Software. The created CAD model is a virtual mock-up of object to be manufactured and it is used for the construction as the primary planning information. After the construction, the CAD model is used as a geometric documentation of the manufactured object. For this documentation to be reliable it needs to be verified. Systems exist to verify small and medium size objects such as car components. But no scalable solution exists for large compound such as a power plant.

We propose to use *Augmented Reality* (AR) to verify the model against the built state. Acquired images of the plant are aligned with the model to offer *Mixed Views*, which helps engineers to detect clashes and act upon them by judging the geometric quality of a construction. Using AR, engineers can infer the quality of the manufactured product to create an as-built documentation rather than a costly model re-engineering. The as-built documentation that mixes images and virtual design inform engineers about the correctness of the model, not explicitly modeled elements and modifications of the original design, which are common throughout the lifecyle of such plants.

The first part of this thesis presents a complete set of tools to perform discrepancy checks using AR. We develop interaction methods to ease the visualization of CAD models and registered images. We propose a transposition of the 2D zoom and pan interaction to manipulate a Mixed View for AR and algorithms to browse images using their geo-localization. We also present a registration method to align images of civil structures to their models, which is based on reliable civil components: *anchor-plates*. We demonstrate the practicality of the method to detect discrepancies on two different real applications: an in-service nuclear reactor and a power-plant building site.

The second part of this thesis focuses on the problem of aligning images to a 3D model using a pre-registered image. We study the complete registration pipeline from image features extraction to pose estimation. First we develop a method to infer the geometric precision of multi-scale features used in stereo registration. Then we present a cost function, which deals with the imprecision of these features by correcting their localizations based on intensity information. This results allow extending the gold standard for stereo reconstruction to take into account the intensity information which is otherwise only used during the feature detection and carried it over in the final optimization. Finally we introduce a method to extend the relative pose between two images to a full pose, which relates the target image to the CAD model. We demonstrate the usefulness these methods for registration in the context of discrepancy check.

Keywords:
Industrial Augmented Reality, 3D Computer Vision, Camera Registration, 3D User Interface

Zusammenfassung

In dieser Arbeit beschäftigen wir uns mit dem Problem des Discrepancy Check, darunter versteht man die Verifizierung der geometrischen Korrektheit eines hergestellten Objekts anhand seines virtuellen Modells. Zu fertigende Produkte werden heutzutage zunächst mittels Computer-Aided Design (CAD) Software geplant. Das dabei erzeugte CAD Modell kann als virtuelles Modell des zu fertigenden Modells angesehen werden und wird als primärer Bauplan für den Bau verwendent. Nach dem Bau wird das CAD Modell als geometrische Dokumentation für das gebaute Objekt verwendet. Um die Zuverlässigkeit dieser Dokumentation zu gewährleisten ist eine Verifikation nötig. Es existieren bereits Systeme um Objekte kleinerer und mittlerer Größe, wie z.B. Autos, zu verifizieren, jedoch keine skalierbaren Systeme welche auch auf große Objekte, wie z.B. ein Kraftwerk, anwendbar sind.

Wir schlagen deshalb vor, Augmented Reality (AR) zu verwenden um den Baufortschritt gegen das Modell abzugleichen. Aufgenommene Bilder werden dabei am Modell ausgerichtet um gemischte Ansichten (Mixed Views) zur Verfügung zu stellen, welche es dem Ingenieur erleichtern Abweichungen zu erkennen und anhand der geometrischen Qualität der Konstruktion zu Beurteilen wie darauf zu reagieren ist. Mittels AR können Ingenieure die Qualität eines hergestellten Produkts ableiten um eine Dokumentation zu erstellen die den tatsächlichen Bauzustand wieder gibt (as-built Dokumentation) anstatt ein kostenintensives Reengineering durchzuführen. Die as-built Dokumentation informiert Ingenieure über die Korrektheit des Modells, nicht explizit modellierte Elemente und Ãnderungen des Originalplans, wie sie über die Lebensdauer eines solchen Kraftwerks üblich sind.

Der erste Teil dieser Arbeit präsentiert ein komplettes Set an Tools um Discrepancy Check mittels AR durchzuführen. Hierfür entwickelten wir interaktive Methoden um die Visualisierung von CAD Modellen und registrierten Bildern zu erleichtern. Wir schlagen eine Transposition des 2D Zooms und Schwenkinteratkionen vor um die Mixed View für AR zu verändern und präsentieren Algorithmen um Bilder anhand ihrer Geo-Lokalisierung zu durchsuchen. Wir präsentieren außerdem eine Registrierungsmethode um Bilder von zivilen Strukturen mit ihren Modellen zu registrieren, welche auf zuverlässigen Komponenten basieren, sogenannten Anchor-Platten. Wir demonstrieren die praktische Einsetzbarkeit der Methode zum Erkennen von Abweichungen anhand von zwei verschiedenen realen Anwendungen: einem bereits in Betrieb befindlichen Nuklearreaktor und einer Baustelle für ein Kraftwerk.

Der zweite Teil dieser Arbeit konzentriert sich auf das Problem Bilder mittels eines bereits registrierten Bildes an einem 3D Modell auszurichten. Wir untersuchen die komplette Registrierungs-Pipeline von der Extrahierung von Bild-Features bis zur Posen-Bestimmung. Als erstes entwickeln wir eine Methode um die geometrische Präzision von Multi-Scale Features, welche in Stereo-Registrierung verwendet werden, abzuleiten. Anschließend präsentieren wir eine Kostenfunktion, welche die Ungenauigkeiten dieser Features behandelt indem sie ihre Position basierend auf Bildintensitäts-Informationen zu korrigieren. Diese Ergebnisse erlauben es den Gold-Standard für Stereo-Rekonstruktion so zu erweitern dass er die Bildintensitäts-Informationen mit berücksichtigt, welche sonst nur für die Feature-Erkennung verwendet werden, und bringt sie in die finale Optimierung mit ein. Zum Abschluss führen wir eine Methode ein um die relative Pose zwischen zwei Bildern zu einer vollen Pose zu erweitern, wodurch das Zielbild mit dem CAD Modell in Verbindung gebracht wird. Wir demonstrieren die Brauchbarkeit dieser Methoden für die Registrierung im Kontext des Discrepancy Checks.

Schlagwörter:
Industrielle Erweiterte Realität, 3D Mashinelles sehen, Kameraregistrierung, 3D Benutzerschnittstelle

ACKNOWLEDGMENTS

During someone's thesis, there is a time for everything: reading, researching, implementing, writing, sleeping (rarely), defending, mindless and endless paperwork. Now comes the time I have been looking forward to the most: the time to acknowledge for the help and support of countless people.

The first thanks goes to Nassir for his continuous trust and belief that everything will work out in the end. I know of no adviser that leaves freedom as Nassir Navab does. This is at times overwhelming but taught me a lot about independent research. I want to thank (or maybe apologize to) Jan-Michael Frahm for hiring me even if my thesis was not completely written and giving me a bit of time off to finish it, it took some time but here it is. I also want to mention my committee members Marc Pollefeys and Gudrun Klinker for accepting the task of reading and reviewing this thesis. This was an honor to defend in front of you.

I want to thank Mirko Appel for his help and his belief that industrial AR was worth pursuing. I learned so much from you about how to run a project. I am still unorganized but it was worse before, so thanks for that. A special thank goes to our partners at AREVA NP Ralf Keller, Stefan Schöter, Martin Neuberger, Erwin Rusitschka, Matthias BOTT, Ghislain Airieau, Tobias SCHWAB and at Siemens CT Reiner Mueller, Hagen Kaiser and Todor Denev. Without them this thesis would not have been the same. They provided incredible data and invaluable feedback. The expectations were high but seeing my work being used still amazes me. I want to thank the students who worked on the project: Xavier Fernandez, Sumit Paranjape, Xinxing Feng, Falk Eisenbeiss. Special thanks go to Jürgen Sotke for taking a research prototype and moving it to a usable software. Last but not least the one and only Pierre Shroeder, the man behind most of the source code of VID. The guy I learned a lot from and to whom I hope I taught something too. I still remember interviewing him thinking I was just picking a student when in reality I found a life savior and a true friend.

This thesis would not have been the same without the work of countless reviewers, program committee members and editorial board, to you the anonymous I say thank you. I would like to thank Selim Benhimane for teaching me a lot and always being available to discuss new and not always great ideas. A special thought goes to Richard Hartley for proofreading one of my papers, I will always remember the discussion we had then. To Adrien Bartoli and Julien Peyras: "next time, lets skip CVPR and go directly drink a beaujolais". I want to thank Stuart Holdstock for proofreading my journal on short

notice. Now to the army (Nicolas Padoy, Olivier Pauly, Stefan Holzer (thanks for the German abstract), Richard Steffen, Jan-Michael Frahm, Virginie Fite and Peter Sturm) that proofread this 264 page document, I am in debt to all of you and I hope I can pay you back one of these days. I would like to thank Oliver Pauly (again) not only for submitting my thesis but also for going through the whole process in less than a day, "chapeau l'ami." I also want to mention Andreas Keil for his constant support and quick translation service, merci. A final thank you goes to José Gardiazabal who carried this thesis for the last yard in order to be filed in and forever collect dust on a shelf somewhere in Munich.

The work performed during my thesis would not have been possible without the help of Martina Hilla, who always had the hardest work at the Chair (taking care of a group of irresponsible children), and of Martin Horn, who has the second hardest job (enforcing the general happiness by insuring that everything was working all the time). To everyone at the Chair, it was great working with you. CAMP is the best group I have ever been with, you were more than colleagues to me, you were my friends. You made what could have a painful journey feel like a nice stroll with some friends. I will never forget these years. I am sure that I am forgetting someone. If you are this someone, I am sorry and I apologize. I am sure I can pay you a drink for compensation.

I cannot finish this acknowledgment without mentioning the students I worked with: Carlos Acero and Bernhard Zeisl, whom I supervised with the amazing help of Florian Schweiger. It was an honor working with you. I hope I taught you something, I surely learned a lot from you. I also want to thank three professors (Guy Wallet, Renaud Keriven and Ross Whitaker) who one day took the time to talk to an insignificant student, you all gave me in your own way the thirst for mathematics, vision and research.

But the biggest thank of all goes to my best friend and life partner who during this thesis became my wife and after the defense the mother of my child. She had to endure much stress, frustration and absences. To you Virginie, there is not enough words in the book to express my gratitude. I would also like to thank my family for their continuing support. Especially my mother who always believed in my abilities when no one thought much of them.

As someone I know wrote one day : "now comes the time after the thesis" [Appel, 2005]. I am hoping for half the paperwork and twice the fun. But it is late and I can hear my Charly crying, I guess it is time to save this document and finally submit it.

CONTENTS

III Advances in Feature based Registration for Augmented Reality 101

LIST OF ABBREVIATIONS

AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality
IAR	Industrial Augmented Reality
CV	Computer Vision
CAD	Computer Aided Design
GIS	Geographic Information System
HMD	Head Mounted Display
SLAM	Simultaneous Localization and Mapping

Part I

Introduction

In this part, we introduce the concepts that are developed in this thesis. In Chapter 1, we motivate our work by explaining the industrial problem we tackle. In Chapter 2, we review the techniques that are employed to obtain the registration in AR systems. Finally in Chapter 3, we present the notation and some background knowledge in 3D Computer Vision useful to apprehend the ideas develop later.

CHAPTER

ONE

MOTIVATION

The 90's were a milestone for the product design process, when Boeing produced the first aircraft only conceptualized using Computer Aided Design (CAD) software: the 777 [Sims, 1994]. Since then the design of a virtual model or virtual mock-up is the first step for large goods manufacturing such as aircraft, cars, ships, power-plants, etc. In most industries, the production happens in-house[1] therefore any modification of the product's design during its real prototyping can be easily integrated in the virtual/CAD model. For example, in car manufacturing a loop process happens between CAD engineers who define the virtual design, and prototype manufacturers who try to build the new model. Manufacturers often discover the presence of problems with the current plan. Theses issues are assessed and solved, this leads to a new virtual mock-up. This process iterates between designers and manufacturers until a satisfactory prototype is created and can be sent to production [Bazizin et al., 2009]. Unfortunately, for companies using subcontractors for production this iteration is more complicated. Sometimes, subcontractors are not enforced to produce an object matching the virtual mock-up but having the same functionality and still fulfilling the safety requirement. This is especially true in the civil construction and power generation area where it is rare to have capability to create prototypes. In this case, the first try is the prototype [AREVA Press Office, 2003]. This might seem like an inconsequential problem because the manufactured product is "functioning" or is at least matching the specifications.

The presence of discrepancies would be truly inconsequential if the virtual model of the product would then be discarded and never to be used as a correct geometric document. But this is not the case; virtual mock-ups are very often used as as-built geometric documentation of the delivered product. For example in the case of power-plants, upgrades are planned using the current documentation, which can include annotated blue prints, surveyed measurements, and the 3D virtual model. When a defective valve has to be exchanged, first engineers consult the available information, using this documentation repairs are planned. If necessary new parts are ordered. Then the maintenance crew goes on site. At this point in time, the reality comes back and clashes are discovered. For example, the valve in the model could be different than the one on-site or that another

[1] It is built/produced by same company that is responsible for the design.

discrepancy might exist between his documentation and the plant. Similar workflow can be found in Nuclear Power-Plant (NPP) maintenance [Ishii et al., 2004] and decommissioning [Ishii et al., 2009]. The reason for the model incorrectness may have different sources. During construction, an error can occur when ordering the parts, which might not be available anymore at the time of purchase, maybe a subcontractor found a cheaper equivalent, or design model given to subcontractors was outdated therefore not matching the design. During the plant life-cycle, it cannot be expected that the 3D model gets updated because of the complexity of updating a 3D model. Additionally, 3D models are often given to the operator in some legacy format that are hard to handle, and when a part is replaced by a different component they need access to its 3D model to make the update. Therefore most of the annotation happens on floor-maps (orthographic projection of CAD) or isometry (non-realistic rendered view of the CAD) [Schall et al., 2008].

This discrepancy between 3D mock-up and reality might create a delay in the maintenance. These delay have great consequences especially in the power generation business where plants need to be regularly checked. For example in Japan, NPPs have to be taken offline to be completely checked every 13 month [Shimoda et al., 2005]. This weakens the power-grid and might cause some instability leading to blackouts. To get an idea of the related cost, a nuclear power-plant when offline in France costs 1 m.€ per day [de Halleux, 2009]. With power plants getting older, their maintenance might require more time. And we can expect the frequency of maintenance to be increased if the decision to double the NPPs' life span, which is by law limited to 32 years, was implemented [Seingier, 2009].

Discrepancies between documentation and product are not only consequential for power plants but also for offshore platform. For offshore structures, maintenance is planned "on-shore" using virtual mock-ups and replacement parts are ordered in advance. If there is a problem with the planned maintenance task because of a discrepancy or an undocumented modification of the system, it will force the maintenance crew to fly back on-shore to plan a new process that can be, this time, implemented.

In this thesis we present new methods to bring back some reality to virtual mock-ups. We create a better documentation for civil work such as power plants. This new documentation merges CAD models and still images that represent the built state. This allows performing discrepancy check; documenting the CAD model with un-modeled components and modification of the design. We create this documentation using the Augmented Reality paradigm. An exemplary augmentation obtained with our solution on power plant images can be seen in Figure 1.1.

1.1 Industrial Documentation

In civil industry, documents available for engineers can take several forms. 2D drawings are the traditional documents that will be find on building sites and are therefore briefly discussed in section 1.1.1. Section 1.1.2 describes modern 3D CAD models and systems.

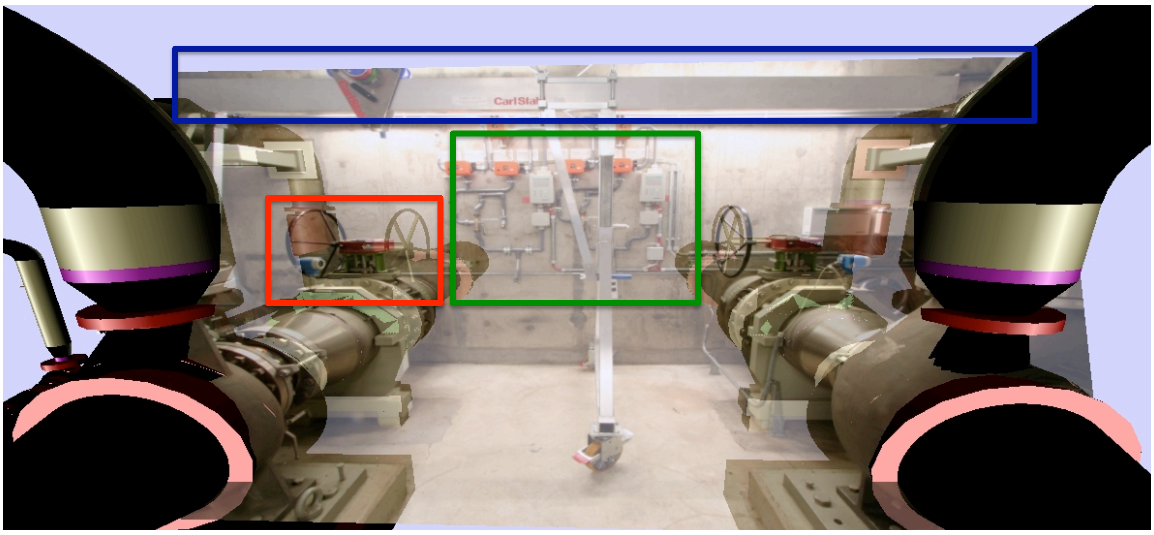


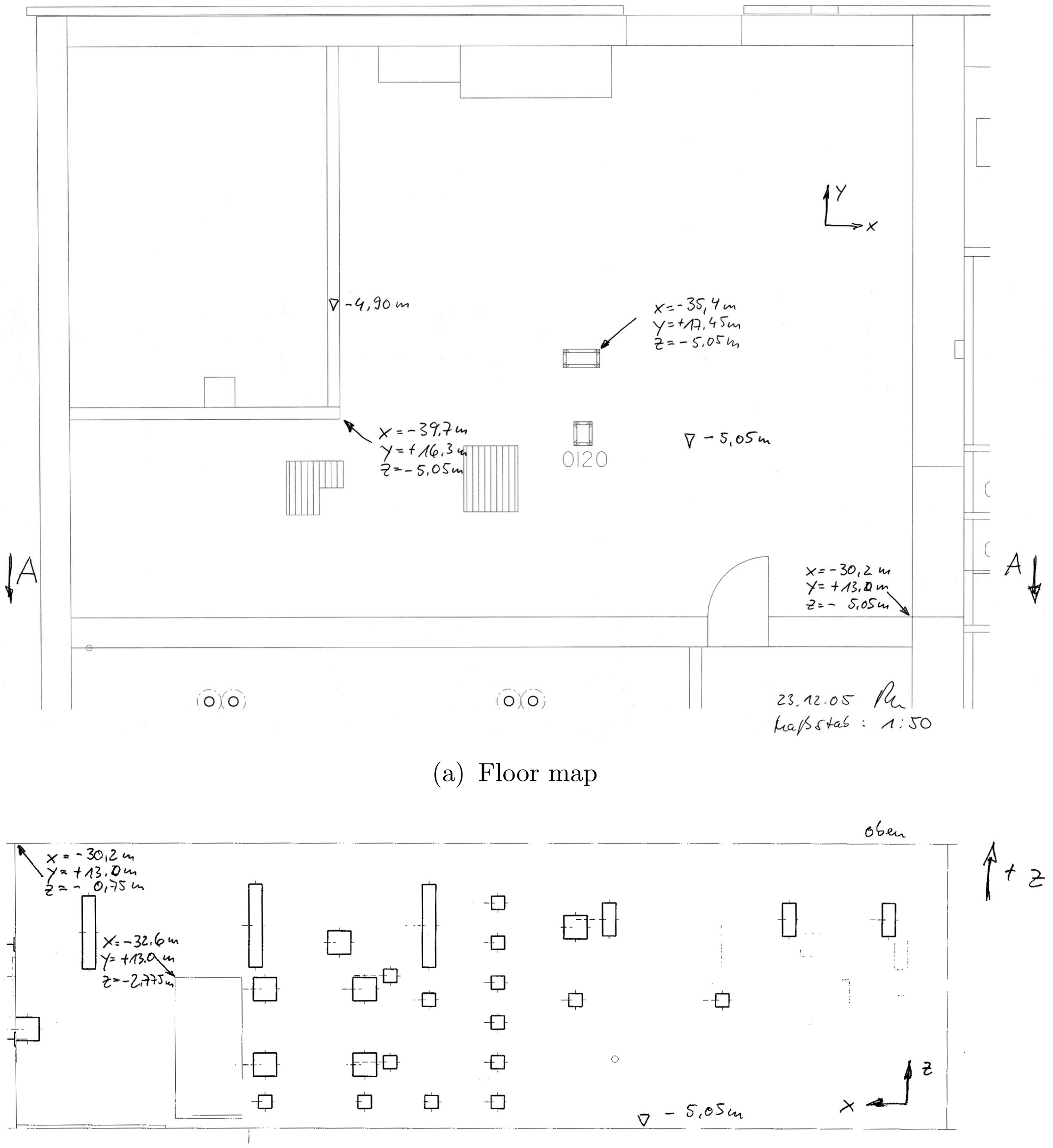
Figure 1.1: **Augmentation of an Image** brings new information to the CAD model. In the area marked in red a discrepancy, a valve has been switched. In green undocumented features, Electrical installation are visible in the image but not present in the CAD model. In Blue plant alteration, metallic structure was added to the original design.

1.1.1 2D Documents

The first of them is Geographic Information System (GIS) [Mendez et al., 2008]. These are maps that can include information such as cadastre, underground structures layout of water pipes and electric cables... The traditional document for civil engineers used to be industrial drawings or floor/wall maps, sometimes called blueprints [Appel and Navab, 2002]. These are orthographic projections of the item. Figure 1.2 shows some 2D drawings of an industrial compound. They are still common on building sites. For large complex, non-realistic representation of the model are also avaible. Isometries are able to represent a complete system. These are non-metric document: schematics that inform about angle and distance. They are extremely useful to map long pipes. They are often annotated by engineers themselves [Schall et al., 2008] to document errors in construction or modification of the design. Nowadays in the industry, design is mostly performed directly in 3D, though some companies did not finished this switch [Compain and Lancesseur, 2009].

1.1.2 3D CAD Systems

The 3D design is composed of geometric primitive (triangle, rectangle, cylinder, etc) agglomerated together to represent the physical form of the objects, see figure 1.3. In order to organize large projects, the models are often categorized by component types, which are concrete structure (walls, stairs, floors, etc), pipes, ventilation systems (heating, cooling, gaze evacuation, etc), machines (pumps, monitoring system, etc), etc. These large models are usually stored in some kind of database that have a complex structure and comport many reporting capabilities. All these 3D components are often linked with



(a) Floor map

(b) Wall map

Figure 1.2: **2D Industrial Drawings** represent orthographic views of the 3D CAD models. They are still the document of choice for construction workers. It is also used to store information as it can be easily annotated.

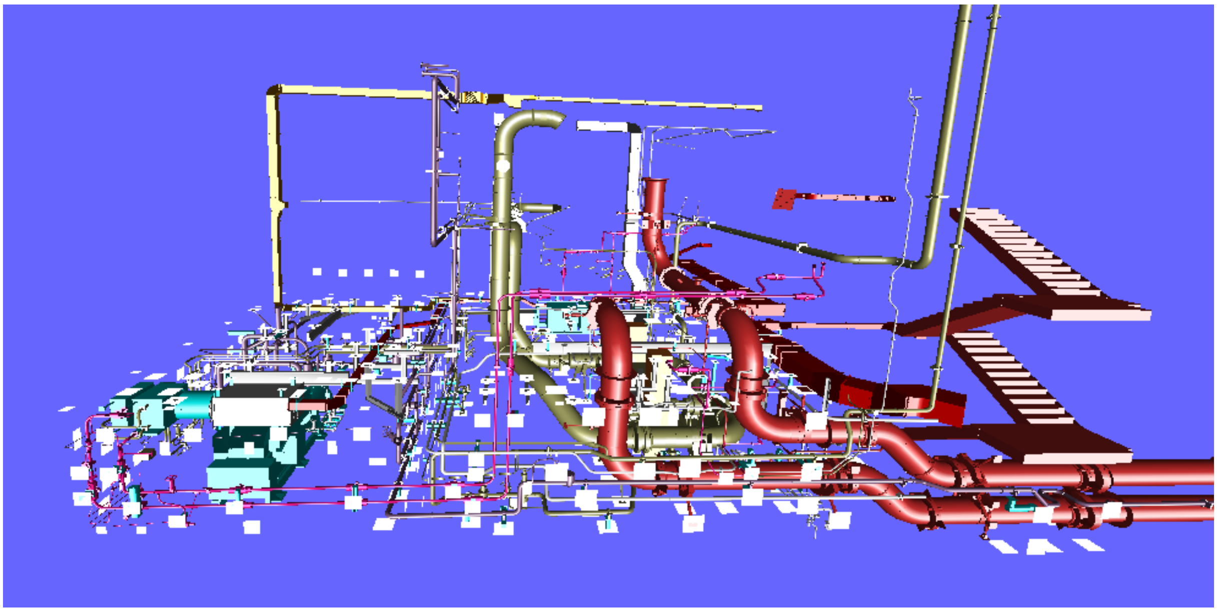


Figure 1.3: **3D CAD Models** can represent large and complex structures. This view represents all the components that belongs to three rooms of a large power-plant.

meta-data such as information on their last maintenance, model number, history of sensor readings that can be accessed upon request through the data storage system [Goose et al., 2003].

Furthermore for large system such as power plants, the model can be of great complexity and be so large that they are hard to render in their whole. For the manipulation of such models to be trackable, they are organized hierarchically. Each complex is divided in buildings that are designed for a specific task for example cooling. These systems comport redundancy for safety reasons. Each building is itself divided in levels that are separated in rooms. This structure offers multi-users access using versioning systems; someone can manipulate a part of the model without blocking the access to the complete project. Additionally, if implemented right accessing such a data-structure can be extremely efficient compare to a naive implementation, which would not be usable.

During this thesis, we had the chance to have access to design data and actual state pictures from different power plants. Using this access to the original database, we propose an integrated solution to the civil engineers in charge of the clash inspection and plant walk-down. We used the similar techniques than the one developed to handle large CAD model to keep the methods scalable for multiple users and data access. This way by keeping the original structure, we avoid loosing all the information contained in the original model. Additionally proposing an integrated system, that keeps a consistent structure with existing information system, should facilitate the training of civil engineers to use our tool and therefore its acceptance.

1.2 Existing Solutions for Discrepancy Check

To find discrepancy, other technologies than Augmented Reality could be used. The first of them, which is in use for clash inspection, is to employ a ruler and estimate the deviation for sub-part of a system. The accumulation of errors in distance and angle makes this method unreliable at best. Additionally some components, which might not be easily accessible (e.g. on the ceiling), will require more time to be measured (e.g. installation of scaffolding).

In car industry the state of the art is to compare design data and manufactured pieces by using a probing robot [Nashman et al., 1995]. A touch sensitive sensor is attached to a tracked robot arm that gives 3D information every time the sensor is in contact with the manufactured product. More recently, NDI released the Portable CMM[2] to compete with probing robots to verify manufactured parts. At the moment, these methods do not scale well to be usable in a large complex such as a power plant.

Another solution would be to re-engineers the 3D model and create an as-built model. It could be directly used as a new document or to compare with the planning data. The geometric primitives (spheres, cylenders, cubes...) of this model have to be estimated from low-level information. This information generally is a dense set of 3D points. These 3D points can be estimated using different techniques, which can be considered as state of the art. For example, photogrammetry uses images to accurately reconstruct 3D points [Heuvel, 2000]. Professional software are available to estimate this points' cloud. Devices such as laser scanners, which use a laser beam to obtain a dense cloud of points [Milroy et al., 1996], could also be used. It usually offers denser clouds than the ones obtained using photogrammetry. The 3D information required to estimate the model could aslo be retrieve using Mixed solution, which merges Computer Vision algorithms and projective display called structured light [Battle et al., 1998].

The estimation of a complex as-built model, which could then be used in CAD software, is interactive at best. And, as far as we know, there is no method to compare a complex virtual mock-up with a reconstructed model; such an algorithm would have to perform a deformable matching between 3D shapes. Even if they existed, these approaches are hardly scalable. For example a probing arm works fine for a singular industrial object but would be unusable on large models. Other methods would take too much time if applied on the whole plant. But above all it is an overkill as they require a lot of time and price. A proper documentation about correctness and discrepancies often is sufficient. Though as-built modeling should not be always discarded, but should only be considered for some hot spots where having a correct model is thought mandatory.

1.3 Augmented Reality

The concept of Augmented Reality (AR) was sketched by Sutherland when he developed the first Head Mounted Display (HMD) [Sutherland, 1965]. But it was baptized by Caudell [Mizell, 1994] in the early 90's when working at Boeing with his colleague Mizell.

[2] Coordinate Measuring Machine - http://www.ndigital.com/industrial/products-pcmm.php

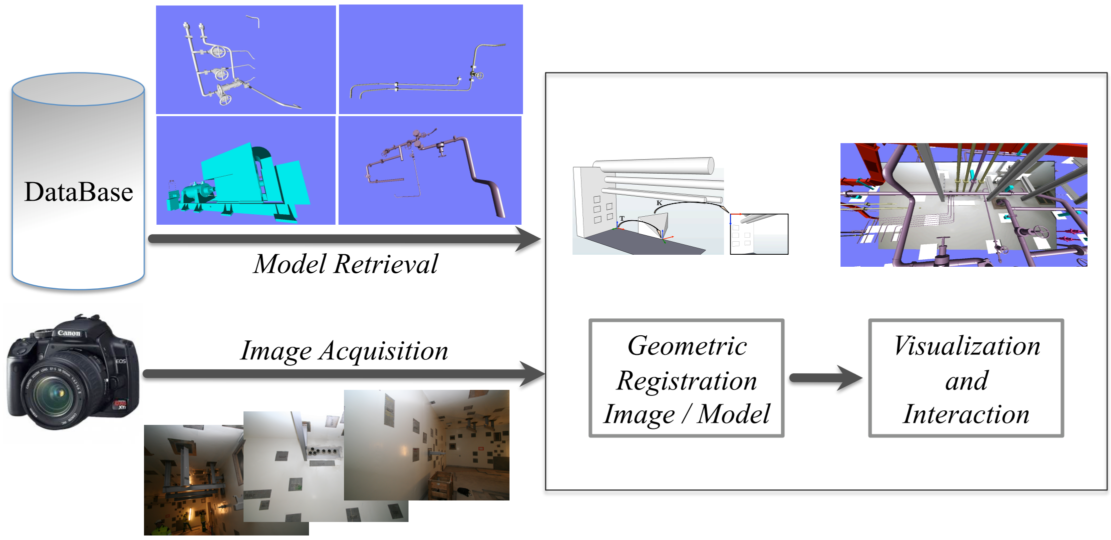

DataBase
Model Retrieval
Image Acquisition
Geometric Registration Image / Model
Visualization and Interaction

Figure 1.4: **Augmented Reality System** needs access to the real world here represented by images, and virtual information, here represented by CAD model. An AR system is composed of a registration procedure to align the virtual and real information, and a visualization module to interact with the created mixed world.

They developed the first industrial application to support the assembly of wire bundle that used AR technology. Later Milgram and Colquhoun [1999] and then Azuma [1997] tried to conceptualize AR.

The basic idea behind AR, which is the definition we use in this document, is to display a computer generated image that merges a picture of the Reality and a Virtual object. The alignment should be geometrically correct (registration), and interactive. This might be considered as a broader definition than Azuma's [Azuma, 1997] because the reality is not acquired on the fly. In this thesis we are focusing in the application of AR as a medium to support an industrial task.

1.3.1 Augmented Reality Requirements

An AR system requires specific components to be implemented. Otherwise it does not make it a complete solution. A schematic representing these modules is accessible in figure 1.4.

Data Access is the ground layer of any AR system that tries to augment a view of the real world. It needs to have information about the real world, which usually are images. Set of images can be continuous via video-streams or discrete via a set of still images, as shown in figure 1.5(a). Information to augment the world is also necessary. Though the augmentation could take any form such as olfactory or auditory. We augment the world by visually incorporating virtual models. In our scenario, these models are extracted from a CAD system database as usable 3D models, as shown in figure 1.5(b).

Image Registration is the corner stone of any AR system. This module takes care of aligning the real world (i.e. the image) to the virtual one (e.g. 3D model). In our scenario, we have to compute the perspective transformation that happens between an image and the 3D model. This is still a major topic of research especially to make it usable in an industrial context. A more theoretical introduction to this problem is given in chapter 3.

Visualization and Interaction should be concidered as a part of any AR System. They define what the user perceives and how he modifies his experience. Interactions to use this medium have to be implemented. This module depends on the task at hand. For example, the visualization module of an AR navigation should guide the user to its target for example with arrows. In our scenario the interactions should support the user to detect discrepancies.

In this thesis we discuss all the aspects of such an AR system.

1.3.2 Industrial Augmented Reality (IAR) Challenges

In 1998, during the first International Workshop of Augmented Reality (IWAR) [Behringer et al., 1999] a panel was formed to discuss possibilities, limitations and applications of AR. They felt that AR had a great potential in many areas (factories, airplanes, medicine) for trained and untrained users. They discussed about the necessary steps to build a truly useful AR system. Both academic and industrial researchers emphasized that researchers and developers should design properly their applications in collaboration with end-users and that the focus should only be given to applications where AR technologies can make a real difference. AR should limit human errors and create a mean to perform a task better. They point out the technological problems of the time, some of which are still addressed issues in current scientific publications. The two major problems are HMDs and connectivity (i.e. access to network everywhere). They unfortunately could not yet define the “killer app” where AR would have a massive impact.

ARVIKA [ARVIKA, 2001], a German project founded by the German ministry of education and research (BMBF), applied AR ideas to many application fields. It created a great excitement for the companies within this consortium (Siemens, Daimler-Chrysler, BMW, VW, Framatone ANP[3]...). They all came with problems that could be solved by Augmented Reality [Weidenhausen et al., 2003]. Unfortunately, because of technological limitations with display, tracking techniques and registration algorithms most of the demonstrations did not make it to the market. Only one prototype made it out the door: the *Intelligent Welding Gun* [Echtler et al., 2003]. A follow-up project named ARTESAS [ARTESAS, 2004] tried to solve some of ARVIKA's short comings. Even if it was a scientific success specially in term of marker-less tracking [Platonov et al., 2006, Wuest and Stricker, 2007], it did not reveal the killer application that would demonstrate the profitability of AR.

[3] Framatone ANP is now named AREVA NP. It is a consortium between Areva (owns 2/3 of the shares) and Siemens (resp. 1/3)

A more recent review [Navab, 2004], using the knowledge developed at Siemens on IAR, gives hints to develop an industrial killer application. For Navab, it should not be an overkill. The AR solution should not try to solve something that is solved better and for less with other technology. The problems that AR tries to solve have to be financially beneficial, because AR hardware is still expensive and the R&D is costly. Finally, he emphasizes on the necessity to produce scalable solution. It should not only work out of the lab but in a complete setup and should be easily reproducible to be accepted by the industry. Even though he does not give the silver bullet he offers the instruction to craft it.

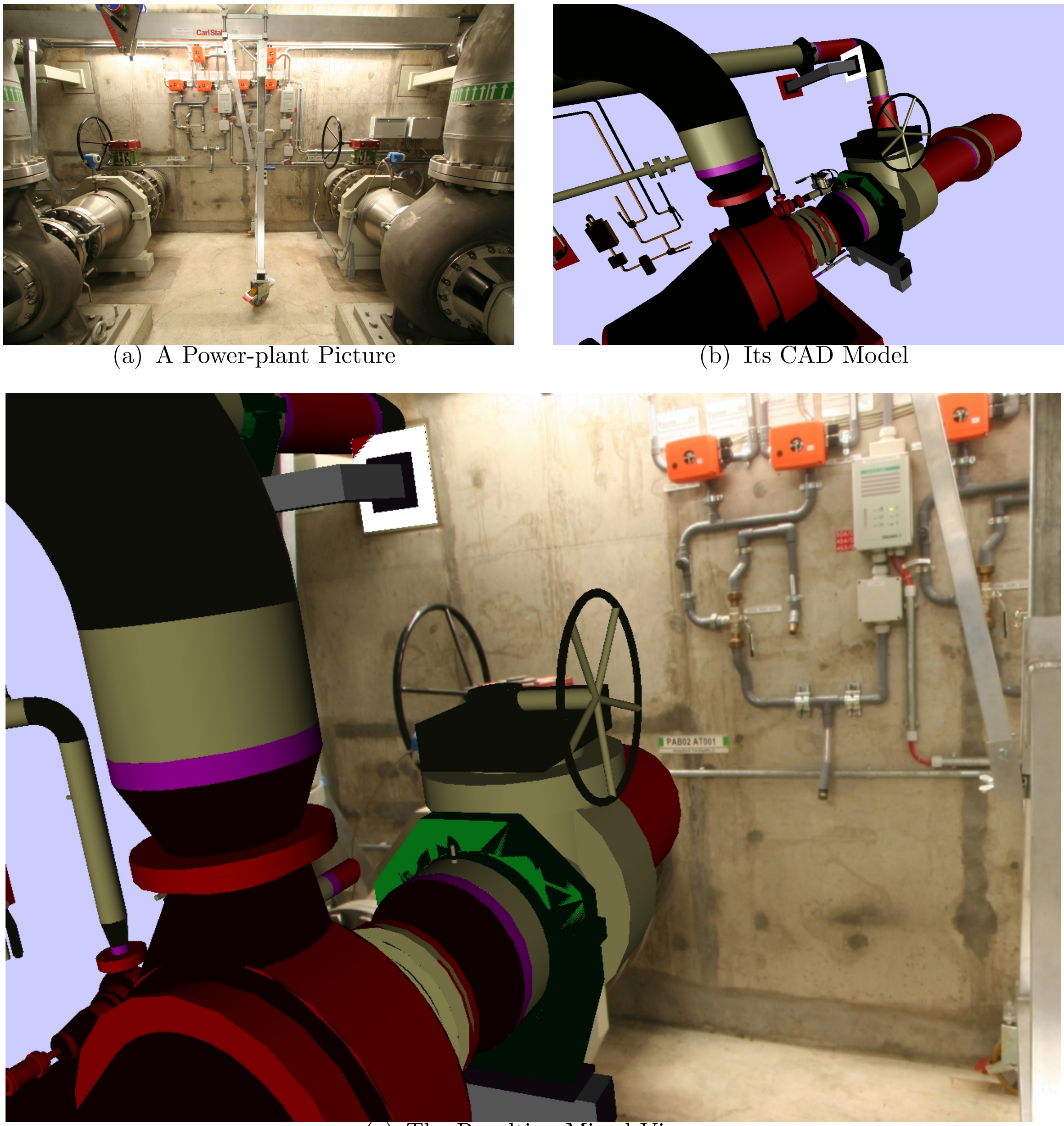
To summarize an AR application has to be:

- developed with and for the end user,
- financially beneficial,
- scalable and reproducible.

1.4 Benefits of an Augmented Reality Solution

An economic impact study has been conducted for the application of AR technology to the construction of NPPs [Blum et al., 2006, Haeberle et al., 2006], which includes discrepancy check application to support clash inspection and plant walk-down. They conclude that the introduction of this technology would be beneficial for the industry in many aspects. It would offer a faster and more precise method to evaluate discrepancy and a better document that reduces the media discontinuity. They conclude by stating that their study was based on the implementation of the system for only one power-plant, which let them believe that its application on several building sites should increase financial return. One of the co-author of this work Appel was interviewed in [Economist, 2007] and stated that “the software will reduce the cost of constructing a typical medium-sized coal-fired power plant by more than $1m”. They emphasize on the fact that such a solution can only be beneficial if employed to perform the task. It needs to be accepted by the user therefore it should be developed in collaboration with them and be similar to software solutions they are used to (e.g. CAD software).

The financial aspect is not the only advantage offered by our approach based on AR for discrepancy check. Having a verified virtual model accounting for the actual state of the plant offers side benefits such as better and more accurate maintenance and upgrade planning. New applications can also be imagined. For example, offering on-site tracking should be possible using the registered images and verified model. It could be used for AR supported maintenance or remote expert. This new document could also be used for creation of an as-built modeling [Navab et al., 1999b].



(a) A Power-plant Picture

(b) Its CAD Model

(c) The Resulting Mixed View

Figure 1.5: **The Mixed View** (c): a new document that merges **Pictures of the Plant** (a) and its **3D Model** (b). The new document links metadata from the CAD to the image, informs the user about undocumented features (e.g. electrical wires), allows the user to perform discrepancy check and offers new navigation techniques for large image sets.

1.5 Prior Work for Discrepancy Check Using Augmented Reality

In this section, we review AR applications, which support an operator to find clashes between a virtual model and the reality. A denser review of industrial AR applications can be found in Appendix A.

For the car industry, Nölle and Klinker [2006] propose to superimpose wire-frame model on to a newly manufactured component of the prototype to verify that the available CAD model and the current “real” prototype are synced. They display a checkerboard pattern, which alternating tiles represent the real and virtual world, to help the operator finding differences. A stereo-system is used to create the augmentation and a marker-based system is deployed to obtain registration between the optical system and the CAD model

METAIO also develops a solution for finding these clashes in the context of factory planning [Doil et al., 2003, Pentenrieder et al., 2007]. They present a system that could, based on markers, augment static pictures of a factory. They offer, in the augmented view, tools for precise measurements that are made possible using a carefully designed registration method. These measurements are used to verify that a new object (e.g. car) can be manufactured through the current production line. If a clash is detected, it means that the current production line needs to be modified.

Webel et al. [2007] proposes a solution to find existing discrepancies between a real and a virtual mock-up. Some companies use real mock-up to improve a design, for example for sub-marines were space is an expensive resource. In their project, they looked at optimizing the pipes layout for which a proper design is hard to find. Using the virtual model as planning information, they manufacture a real mock-up. This real mock-up is then physically modified to minimize the space requirements of the pipes layout. Using augmentation of the real mock-up, discrepancies are identified. Discrepancies can be then included directly in the virtual model by means of reconstruction made possible using a stereo system. This iterative design process is promising, unfortunately cumbersome calibration and limited reconstruction precision lower the usability of the current system.

Schoenfelder and Schmalstieg [2008] propose an AR solution for the plant walk-down. This process is performed before the delivery of a new factory. Its purpose is to verify the correctness and quality of the construction. They allow the engineers access to live augmentation of the plant on a screen that is mounted on a cart. The engineers can interact with the augmentation and when a discrepancy is found, he can easily document it. The obtained document can be used as a testimony of discrepancy to the responsible subcontractors to force them, for example, to deliver a correct 3D model. This system is designed to focus on hot-spots as it requires an external tracking system to be setup in advance.

All these approaches show the broad interest in both the AR community and the industry for discrepancy check. Unfortunately, the existing solutions are not scalable to the size of the problem we tackle as they all require some scene alteration for registration (installation of markers or of an external tracking system), which require highly trained operators to be executed. This limits their deployments for large systems such as power-plants, which would require many teams of experts to execute the check in a reasonable time. With our system, we try to introduce simple registration methods that could be used by everyone present on the construction site that have a basic knowledge of CAD system to guarantee the usability and scalability of the proposed solution.

1.6 Objectives

The work surrounding this thesis was to develop an entire solution to support civil engineers during documentation of discrepancies between a CAD model and the built state. It should include all necessary tools to perform this task; no additional software should be required. Because end-users are not expert in AR, the methods involved to register images with their virtual models have to be as simple as possible. This means that it should be understandable by civil engineers, who do not have knowledge about computer vision. Registration methods have to produce precise alignments so the superposition used for visual evaluation of discrepancy can be relied on. When possible methods should be automated and if not, users should always be guided through every computational step to limit the required training. Visual inspection of discrepancy have to be performed with the developed solution. Therefore, the solution should include necessary interactions to ease this process. Users should be able to reveal discrepancies at all stages of the plant life-cycle: construction, commissioning, servicing... Documentation of these discrepancies should be supported within the solution, therefore reporting system have to be included. This should ease communication between the different department of companies and the subcontractors. The ultimate goal of this work is to create a new document of the product that merges the virtual model and as-built information.

1.7 Contributions of this thesis

1. We introduce a new workflow for industrial documentation based on a scalable solution for discrepancy check. This can be applied to two scenarios:

   (a) For existing plants that do not have proper verified 3D model to plan a maintenance or an upgrade. Using registered images to the virtual model, engineers create an up-to-date documentation, by discovering: discrepancies, undocumented features, and alteration of the design.

   (b) For the erection of a new plant, the proposed solution allows civil engineers to follow the different steps of construction and perform the appropriate checks. If a discrepancy is discovered between models and the construction, it is assessed. This can lead to an annotation, a change in the design (remodeling) to handle the discrepancy or destruction and rebuilding the system comporting a discrepancy.

   This as-built documentation should minimize clashes and thus accelerate the construction, decrease the planning and implementation time of tasks such as maintenance, upgrade or decommission.

2. We present new methods to interact with the developed AR CAD viewer. This includes zooming functionality, 3D navigation capabilities, and component-based viewing. Along with rendering method to visualize discrepancy, this offers all necessary tools to handle such data.

3. We propose a registration method based on industrial components, which are broadly used in the civil industry. This approach is a two-steps process composed of a segmentation algorithm and matching method. The resulting interactive method allows engineers to quickly align newly acquired images to the virtual model for inspections. The registered image can then be used to develop other registration methods for other images of the same scene.

4. We study localization uncertainty of multi-scale local features, which are at the center of most algorithms dealing with registration of an image to another. We represent the underlying localization error distribution as being Gaussian and develop a method to estimate it automatically from image information. We present result on the known local features that are SIFT and SURF.

5. We present a new non-linear cost function for the estimation of the essential matrix. The essential matrix governs the geometric relation existing between two images. This method combines geometric information provided by feature points and intensity information from the neighborhood of these features. This algorithm is able to cope with badly localized points better than the gold standard method for registration of images.

6. The essential matrix yields to a relative pose of an image to another, this relative pose lacks a degree of freedom to be used for augmentation. We develop an algorithm to extend it to a full pose using 3D information that is linked to images registered to a CAD model. This algorithm is presented in our IAR framework as one of the solution to register images to a 3D model.

7. We give an up-to-date literature review of augmented reality industrial applications developed to support a product over its life-cycle (design, maintenance, etc). During the past few years we gathered some knowledge about industrial applications that we summarize here. We categorize solutions by the stage at which the solution was applied during the life-cycle of a product, rather than a purely historical narration. We try to give a critical report of proposed approaches and to explain reasons for failure and success in AR systems implementation.

Apart from the IAR literature review (contribution 7), all presented contributions have been published in peer reviewed conference proceedings or journals, contribution 4 was mainly developed during the Diplom Arbeit of Berhnard Zeisl, which was co-supervised by Florian Schweiger and myself. A list of all my publications is located in appendix C. A list of the abstract of the publications, which this thesis is based on can be found in Appendix D. Not all publications, I authored or co-authored during the period of this thesis are presented in this document. A list of extended abstracts of these articles can be found in Appendix E.

1.8 Outline

This thesis is divided in four distinct parts and a set of appendixes.

In the remaining of the Part I , we give an introduction to existing solutions to register images to a CAD model for IAR in Chapter 2 and we describe the notation and some background in 3D Computer Vision in Chapter 3.

In Part II, we present our discrepancy check application with first a description of the proposed workflow in Chapter 4, followed by a presentation of two 3D interaction methods designed for augmented CAD model in Chapter 5 and concluded by a description of a pose estimation process based on reliable CAD component in Chapter 6.

In Part III, we propose some advances for computer vision, first on localization uncertainty of multi-scale features points in Chapter 7, then on the non-linear estimation of essential matrix in Chapter 8, and finally on the estimation of the translation's length between two cameras in Chapter 9.

In Part IV, we discuss the outcome of this work and describe possible future work.

In Appendix A, we study in depth prior applications of augmented reality developed to support a product throughout its life-cycle and in Appendix B we discuss implementation details on our software solution and present some of the user interfaces.

A list of my publications and patent application can be found in Appendix C, a list of abstracts of the publication described in this thesis in Appendix D and a list extended abstracts in Appendix E for the publication not discussed in the thesis .

CHAPTER

TWO

STATE OF THE ART IN REGISTRATION FOR INDUSTRIAL APPLICATIONS

In the past two decades, different strategies have been employed to compute the rigid transformation that exists between a 3D model and a perspective image. In this Chapter, we give an overview of different techniques available to perform this task. This includes automatic means of computing the pose as well manual methods.

We, first in Section 2.1, focus on Visual Marker as there the most commonly used approached to register cameras in AR systems. Then we discuss external tracking systems in Section 2.2, stereoscopic system in Section 2.3, vision methods based on 2D and 3D model in Section 2.4, algorithms based on keyframes in Section 2.5 and finally, in Section 2.6, briefly describe simultaneous localization and mapping.

2.1 Visual Marker based Registration

Fiducial 2D Markers are the most widely used technology for registration in AR. Usually tracking systems based on markers are quite basic and are often using a single camera. Each frame gets processed to detect fiducials. Theses fiducials are advanced bar-codes with which one can compute a pose. An example of such a fiducial is visible in Figure 2.1(a) with the widely known “Hiro” tag from ARToolKit [Kato and Billinghurst, 1999]. Markers are designed such that their detection is fast and reliable. Ultimately, a large quantity of distinguishable markers should be available to identify them uniquely.

In the past decade many markers have been developed. To name just a few: ARToolKit, SCR Tag [Zhang and Navab, 2000], CyberCode [Rekimoto and Ayatsuka, 2000], ARTag [Fiala, 2005], ARToolKitPlus [Wagner and Schmalstieg, 2007], Nestor [Hagbi et al., 2009], etc. Most of these algorithms have publicly available implementation, which explain their broad usage. Figure 2.1,2.2, 2.3 and 2.4 show some markers and their applications.

A marker allows relating a user to the marker coordinate system, which can be related to a 3D model with pre-registration. Boeing, in the first IAR application, uses circular fiducials stuck on their generic foam board for easing bundle wire assembly [Curtis et al., 1998]. The ECRC developed their own markers [Koller et al., 1997a,b] and used them t

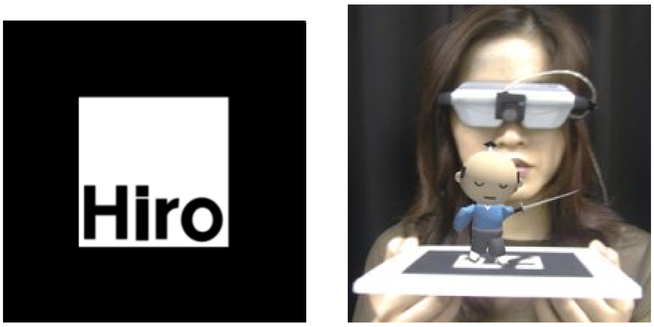


Figure 2.1: **ARToolKit Markers**: on the left the well-known "Hiro" tag and on the right a demonstration of ARToolKit to find the transformation between the marker and the user's HMD. *Courtesy of* ***http://www.hitl.washington.edu/artoolkit/***.

t

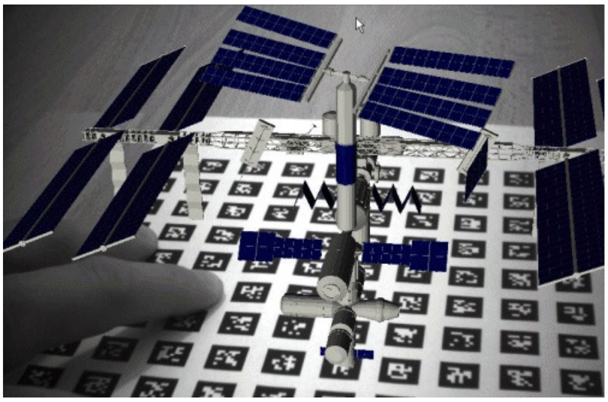
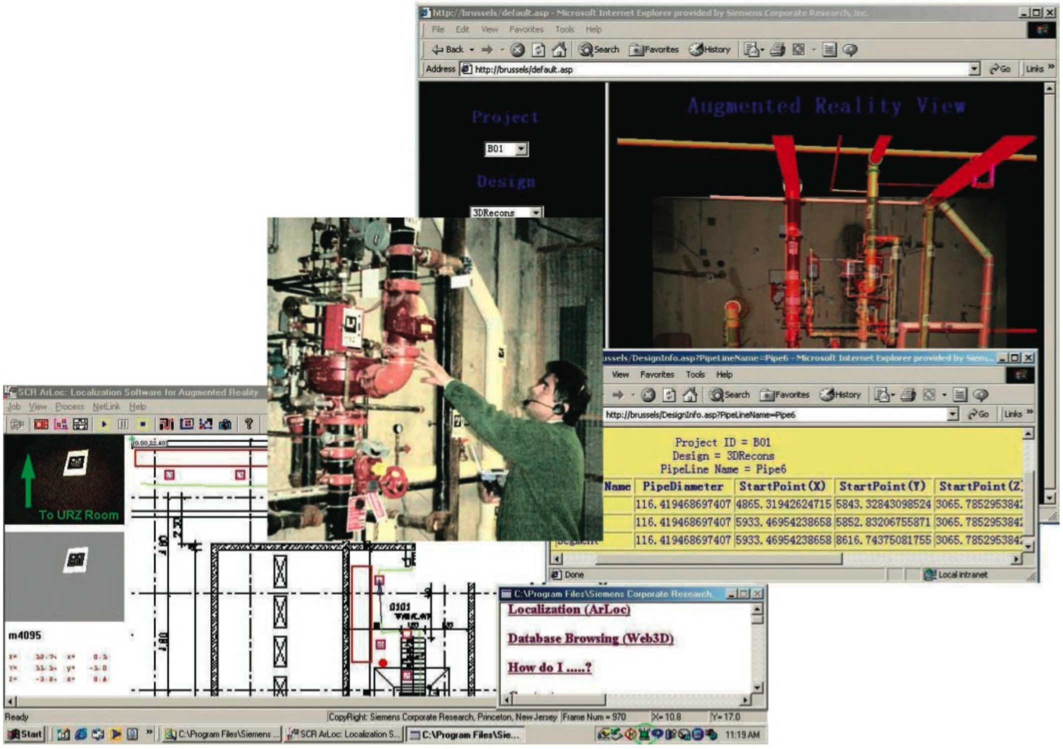


Figure 2.2: **ARTags** are advanced fiducial markers that offer a large set of distinctive markers. *Courtesy of* ***http://www.artag.net/***.


http://brussels/default.asp - Microsoft Internet Explorer provided by Siemens Corporate Research, Inc.
Augmented Reality View
Project
B01
Design
3DRecons
ussels/DesignInfo.asp?PipeLineName=Pipe6 - Microsoft Internet Explorer provided by Siem...
http://brussels/DesignInfo.asp?PipeLineName=Pipe6
SCR ArLoc: Localization Software for Augmented Reality
To URZ Room
m4095
Ready
C:\Program Files\Siemens Corporate Research
Localization (ArLoc)
Database Browsing (Web3D)
How do I .....?
CopyRight Siemens Corporate Research, Princeton, New Jersey
Done
Local intranet

Project ID = B01
Design = 3DRecons
PipeLine Name = Pipe6

Name	PipeDiameter	StartPoint(X)	StartPoint(Y)	StartPoint(Z)
	116.419468697407	4865.31942624715	5843.32843098524	3065.78529538
	116.419468697407	5933.46954238658	5852.83206755871	3065.78529538
	116.419468697407	5933.46954238658	8616.74375081755	3065.78529538

Figure 2.3: **Siemens SCR in the SEAR Project** uses markers for AR supported maintenance. The markers offer navigation guidance and contextual information. *Courtesy of Nassir Navab [Goose et al., 2004].*

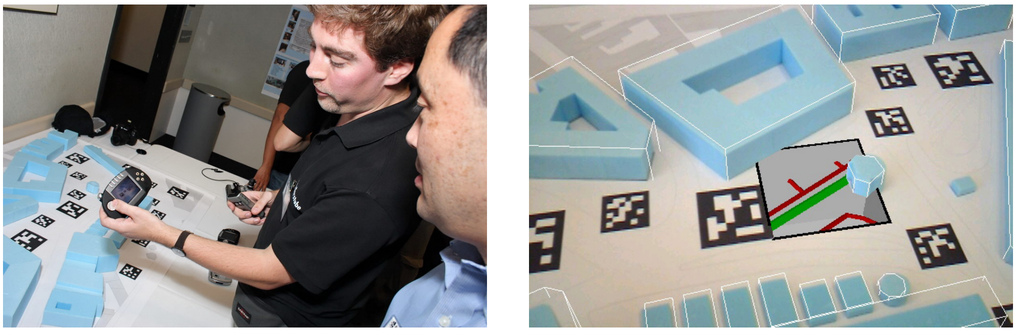


Figure 2.4: **Demonstration of ARToolKit+ at ISMAR 2007**: Markers for registration on handheld devices such as cellphones. These pictures show a demonstration, which uses the real mock-up of a city to offer X-Ray vision into underground structures. *Courtesy of Daniel Wagner* ***http://studierstube.icg.tu-graz.ac.at/handheld_ar/media_press.php.***

for various scenarios such as interior design. In [Kobayashi et al., 2001], markers are used in a welding training application. They were installed on the welding gun and the working plane. They are tracked using the camera installed within the mask. Siemens SCR, in [Goose et al., 2003], takes advantages of visual markers to support a maintenance task. Using the marker they can localize the user with respect to the factory. Using this geo-localization information they propose a mixed view composed of a pre-registered high-resolution image with the virtual mock-up. The ARTHUS project [Broll et al., 2004] used markers [Liu et al., 2003] to localize of Head-Mounted Displays for collaborative design. The robot manufacturer Kuka [Bischoff and Kazi, 2004], when prototyping a new user interface paradigm for robot control using AR also employs them. For decommissioning of power plants, [Ishii et al., 2009] use self-developed linear bar code markers. Since they cannot directly provide a pose, as a regular 2D marker would, they use them in combination [Shimoda et al., 2005]. METAIO and Volkswagen, in [Pentenrieder et al., 2007], demonstrate that markers can provide industrial grade precision, if installed with care and properly related to the 3D Model. In their application they use drill holes of the car body to mount the markers, thus guaranteeing a precise positioning.

Marker technology is not only used for absolute pose estimation between the user and the world coordinate system. Markers have also been employed as tangible interaction tools. They are typically attached to an object manipulated by the user. The marker informs us of the relative transformation between the camera and the object. For example, in the URP project Ben-Joseph et al. [2001] use barcode-like markers to manipulate tangible objects on an augmented table; a similar approach can be found in [Kato et al., 2003]. Gausemeier et al. [2002] also use the tangible property of markers to organize a floor shop. They scoop 3D models from an AR catalog and deposit them in their virtual factory; the markers give the geometric relation between the models. For a similar application, Doil et al. [2003] propose that each marker represents a single piece of hardware that has to be organized within the floor map. Dunston et al. [2002] use them to visualize complex pipe layouts; the model is attached to the marker and moving the marker as the viewpoint thus allows the user to understand the layout in a natural way. Molineros and Sharma [2001] use home grown markers, which are a set of white dots to encode the data, to support assembly workers.

Markers were used in countless prototypes of AR system as shown in [Regenbrecht et al., 2005]. Here is a list of the papers discussed in Appending A that are based on marker technology: [Fujiwara et al., 2000] to help visualization on remote building cites, [Reiners et al., 1998, Zhong et al., 2003, Zauner et al., 2003, Hakkarainen et al., 2008] to support or train for an assembly task, [Klinker et al., 2001a, 2004, Lipson et al., 1998, Neumann et al., 1999, Siltanen et al., 2007] to help maintenance workers, [Klinker et al., 2002, Nölle and Klinker, 2006, Regenbrecht et al., 2002, Schumann et al., 1998] for collaborative design inspection.

The simplicity of markers had a massive impact on AR. Since they only require to be printed and related to a 3D model, a vast majority of AR prototypes are based on them. Their algorithmic complexity made it possible to use them on handheld devices [Pasman and Woodward, 2003] and [Riess and Stricker, 2006].

Even if marker technology seems to be successful in printed medias, where they offer additional features for articles and advertising [Magazines, 2009]; only few applications tried to use markers in an industrial context, even fewer in a real setup because the solution is hardly scalable. It requires having many distinctive markers. One would need to register each of them to a model for example using [Klopschitz and Schmalstieg, 2007]. Their geometric integrity has to be guarantied over time in order to be reusable, which is more than unlikely for example in an ever evolving factory floor. Finally, the obstruction of the workspace that they create is unfortunately their biggest major drawbacks. Markers failed to be accepted for a broad use, as the German project ARVIKA (based mainly on markers) has demonstrated [Weidenhausen et al., 2003]. But even if they cannot be used on a large scale, they can be beneficial in some specific cases and therefore should not be discarded all the time, as they are at the moment by the industry maybe because researchers pushed their use too much forward when other technologies were available.

2.2 External Tracking Systems

Some applications use external tracking systems to estimate the pose of a camera or a display at any times. For this, these systems have to be registered to the 3D model. All of the systems described in this section are available as off-the-shell solutions, which is one of the explanations for the wide use in prototypes and industrial applications. They do not require any re-implementation to obtain a high-quality tracking. These systems can be separated into two categories, local systems that are limited in their working volumes and global systems that are deployed on a larger scales.

2.2.1 Local Systems

2.2.1.1 Magnetic Tracking Systems

Magnetic trackers are commonly used in medical applications. These systems are composed of a transmitter, which creates the magnetic field and a sensor. This sensor is a passive coil, which under the influence of the magnetic field generates a current that can be used to determine its current position. For example, [Kaufmann et al., 2000] use a 6 DoF tracker to estimate the transformation between the virtual world and the user HMD, for AR guided assembly of geometric shapes.

2.2.1.2 Ultrasound Tracking System

Ultrasound tracking systems have been widely used in Augmented Reality especially in the 90's because it was one of the only viable solution then. It is composed of beacons, the object to be tracked, that emits ultrasounds and a set of microphones that triangulates the position of the beacons. Several beacons can be identified by the different frequencies they produce. It was used in [Webster et al., 1996] to track the user's HDM in 3D space to reveal invisible structure of buildings. Newman et al. [2001] combines ultrasonic trackers and inertial trackers to fellow the movements of the user's HMD.

2.2.1.3 Optical Tracking System

Optical tracking systems are the most commonly used technology in prototypical AR applications. These systems are composed of two or more cameras that are composed of infra-red flash and an infra-red filters. This system can easily detect infra-red reflective surface. The tracked object is usually composed of four reflective markers (e.g. spheres), each of whom can be localized in 3D using triangulation techniques, and all together creating a identifiable 3D marker with 6 DoF. A good introduction to this technology is available in [Sauer et al., 2000], where they use it in an AR guided wiring application, and [Schwald and Laval, 2003], where it was used for a maintenance task training. The KARMA system from Columbia University [Feiner et al., 1993] employed an external infra-red tracking system to localize a printer to repair and the HMD in real-time. Optical tracking systems were also combined with inertial systems to support pilots in jets [Foxlin et al., 2004]. ART[1] systems were deployed in many applications, [Echtler et al., 2003] use it to track a welding gun, which is used to calibrate the optical system to the real CAR thus offering very precise tracking and quick calibration; [Barakonyi et al., 2004] used it to prototype an AR guided maintenance application, [Schwerdtfeger and Klinker, 2008] used it for logistic picking task test fields and [Schoenfelder and Schmalstieg, 2008] deploy it in a factory for clash inspection. Other off-the-shelf systems have been used, for example the optotrak[2], as an out of the box solution such as design evaluation [Ohshima et al., 2003].

Each of these technologies had his success because of their reconfigurability. Once such a system is installed in a lab only the targets (beacons, coils or markers trees) have to be re-configured to use it for a different application. This can be made easy if a proper software architecture is used such as DWARF [Bauer et al., 2001] or CAMPAR [Sielhorst et al., 2006].

Unfortunately, none of these tracking systems is a perfect solution: magnetic trackers are perturbed by metallic masses, ultrasound systems are rather un-precise and infrared is perturbed by the sunlight. But above all, they are designed to work in a small volume and to be used in a large environment they would need to be moved and calibrated each time as explained in [Schoenfelder and Schmalstieg, 2008]. Since we want our system to work on a large scale and not on a hotspot it could not be considered as an alternative.

2.2.2 Global Systems

All global systems are based on a similar principle. They use landmarks that have known positions and that can be uniquely identified. Most of these systems were developed for the military and the navy, for example the radio direction finding system which allowed a ship to localized itself at sea but is obsolete nowadays. GPS is, as far as we know, the only technology that can offer AR grade precision. Therefore we only focused on it.

[1] http://www.ar-tracking.de/

[2] http://www.ndigital.com/

2.2.2.1 Global Positioning System (GPS)

GPS is broadly used to support field workers. A GPS client triangulates its position by estimating its distance to four or more satellites positioned on a medium orbit; the fourth satellite is used for absolute time estimation. Since a GPS only provides a location in the earth coordinate system the orientation has to be given by external sensors. The TINMITH system [Thomas et al., 1999] was deployed in [King et al., 2005], where they use the integrated GPS to localize a laptop and a compass to orient their GIS data with respect to their current video frame to augment a vineyard with soil information. Dodson et al. [2002] combine a GPS and a gyroscope in AR binocular that relates the user with virtual models; this way they can augment the binocular view with underground structures such as gas pipes. The same combination was used in [Schnädelbach et al., 2002], where they built an outdoor system to obtain a CAVE-like experience outdoor for the augmentation of touristic sites. Vesp'R [Kruijff and Veas, 2007, Veas and Kruijff, 2008] combines inertial sensors and GPS to obtain the pose of the device; it was deployed for virtual redlining [Schall et al., 2008]. [Behzadan and Kamat, 2005], using a similar setup, are able to augment real construction sites.

Global systems are often the only outdoor solution and offer some solution for AR. But these systems have been designed for outdoor use and provide no position information when used indoor or underground because of the signal disturbance by concrete structures. The property is less than desirable for our scenario that includes outdoor as well as indoor.

2.3 Stereoscopic System

Stereo systems are a commonly used technology to acquire 3D information. For example the DaVinci Robot[3], a surgical robot, uses a stereo endoscope to give a 3D feeling to surgeons. A stereo system is composed of two cameras. The optical setup is fully calibrated: known displacement and known internal parameters. If points are in relation between the two images from the cameras, the system can directly triangulate their position. This system offers real-time performance compatible with the augmentation of video-streams.

Stereoscopic systems were one of the first technology used in AR for example in robotics [Zhai and Milgram, 1991, Milgram et al., 1993]. They offer measuring capabilities and direct overlay of augmentation. More recently Nölle [2002] used a stereo head for inspection because the immersion obtained is supposedly better. [Webel et al., 2007] use a stereo system for discrepancy check between a mock-up and its virtual model; the stereo system has here additional advantages that only stereoscopic system can offer, such as reconstruction for CAD update.

Stereoscopic systems offer direct access to 3D information in the scene but when registration is required, it is mostly done manually or based on markers.

[3] http://www.davincisurgery.com/

2.4 Model Based Registration

Many tracking and registration systems are using the model of the object. It can be 2D, 3D, textured or non-textured. In this section we list different AR applications that use such methods.

Manual interaction is the first model-based method one can think of. ECRC uses, in some of their demonstrations, a manual registration system where the user selects points in the image or 3D points in the scene using a 3D pointer and then put them in relation with the 3D model to obtain the registration [Tuceryan et al., 1995]. This is a perfectly suitable solution when only single image augmentation is required and tracking is not necessary.

The 3D model of a bridge is used to obtain its pose based on edges [Chevrier et al., 1995, Berger et al., 1996, 1999] . This work is quite impressive when considering that the results are obtained using images taken at night. Drummond and Cipolla [1999], Klein and Drummond [2003] present a method that uses a non-textured 3D model to estimate the current pose of an HMD. They extract edges from the current view and from a rendered view of the 3D model; and estimate the pose by minimizing some re-projection error. Similarly Comport et al. [2003] use a visual servoing framework to register the current image to a 3D model using edges. These edge methods operate at high frame rates, which makes them suitable for portable devices as demonstrated for still images in [Riess and Stricker, 2006]. Unfortunately these methods do not support automatic re-localization. These problem was tackled by [Platonov and Langer, 2007] to automatically localized engine parts for maintenance and by [Kotake et al., 2007] for printer maintenance. Kameda et al. [2004] use points, which can be detected stably between the CAD and the image to estimate the current pose.

Using 2D models, the tracking system of BUILT-IT extract contours in the current frame to estimate the pose of planar objects [Rauterberg et al., 1998]. Navab et al. [1999b] estimate the pose of the current view by calculating homographies; the landmark used for registration are extracted manually by the user. Molineros et al. [2004] use edge detection on images from airport runways and track them in real-time using 2D maps, this to help pilot taxiing.

Some methods are based on textural information. For example, Klinker et al. [1998] uses *Reality Model* generated with different technology (stereovision...) combined with GIS data, to develop automatic algorithms for registration. Reitmayr et al. [2005] use the texture from the maps to estimate the current geometry; they use local features extracted from the current video frame and match them to a dictionary of objects (e.g. maps) that could present in the current view. Using a coarse textured 3D model, Reitmayr and Drummond [2006] estimate the pose of a mobile system by matching edgels from the rendered model to the current view using intensity information.

The model used for these systems is supposed to be correct. Some of the methods are made “robust” but only to handle occlusion. None of them is designed to handle a model that does not match the built state. Additionally the clutter present in industrial environment makes the information provided by an edge detector poor. Since the verified part of the model, we have accessed to, is limited, methods based on iterative closest point (ICP) [Rusinkiewicz and Levoy, 2001] mostly fail because of the clutter.

2.5 Keyframe Based Registration

In order to overcome some of the problems related to model based approaches, researchers started to use the concept of keyframes well described by [Lepetit et al., 2003]. Keyframes are images registered to a global coordinate system (in our approach the CAD coordinate system). These images permit to use features-based approach, which offers real-times performances making them attractive for Augmented Reality.

Bleser et al. [2005] proposed a method that uses a single keyframe and a correct CAD model for industrial AR. For maintenance support, Platonov et al. [2006] track a unique keyframe at any given time, they use the 3D model obtain the correct scaling. Other researcher focused on the use of multiple keyframes, when no global coordinate are fixed, to obtain a stable tracking for example [Chia et al., 2002]. [Stricker, 2001], also using multiple keyframes, first finds the closest keyframe to the current view and then computes an affine alignment.

Keyframes have a great potential for Augmented Reality. Unfortunately they suppose the registration of the keyframes to be given, which is often a tedious procedure. Some use computer vision software to generate the model and their keyframes [Lepetit et al., 2003] others markers [Platonov et al., 2006]. This is not suitable as we have already a model and as mentioned in 2.1 we do not want to use markers.

The registration method, which we introduce at the end of this chapter, will indeed create keyframes that are then used for registration of new images. A more detail algorithmic description of the papers presented here is given in chapter 9 where we present a method for full pose estimation from a single keyframe that applies for wide baselines.

2.6 Simultaneous Localization and Mapping

Vision based Simultaneous Localization and Mapping (VSLAM) has taken some momentum in the past five years in AR. They offer the possibility to create a 3D map of the environment that is captured by the camera using feature points. These methods can mostly be divided in two categories, either based on Extended Kalman Filters [Davison and Murray, 2002] or Bundle Adjustment [Klein and Murray, 2007]. In all these methods the camera knows its location with respect to the maps it creates; some methods are able to perform relocation when tracking is lost for example when the scene is occluded.

It has mainly been used for *remote expert* type of scenarios [Davison et al., 2003a,b, Reitmayr et al., 2007] where the relation to an absolute reference (e.g. from a CAD) is not necessary. This is at the moment the biggest limitation of VSLAM, but one can conjuncture that a VSLAM system that uses keyframes pre-registered to a model should have a great impact for the industrial applications.

Conclusion

In this chapter, we showed there are many different methods to obtain a registration. These methods are not perfect for every applications, they have advantages and disadvantages. As our system as to work indoor in a power-plant as well as outdoor we cannot used a GPS. Generally scene engineering is not practical because of the scale of the environment we need to augment. This is why we focus on component-based registration to obtain registered images that can be used for direct augmentation or as keyframe.

CHAPTER
THREE

NOTATIONS AND BRIEF INTRODUCTION TO 3D COMPUTER VISION

In this chapter, we introduce mathematical notations (Section 3.1) and computer vision concepts (Section 3.2 to Section 3.9) necessary to grasp the problem we tackle. For a complete description of geometry of multiple cameras refer to textbooks such as [Faugeras, 1993] and [Hartley and Zisserman, 2003]. The notations and formulations used here are largely inspired by [Bartoli, 2003] and [Benhimane, 2007].

3.1 Notations

symbol	**meaning**
general writing style	
$\mathbb{A}$	set
$\mathbb{A}^*$	set $\mathbb{A}$ without its null element
$\mathbb{A}^n$ (with $n \in \mathbb{N}^*$)	n-ary Cartesian power of $\mathbb{A}$
a, A	scalar value
$\mathbf{a}, \mathbf{A}$	vector
A	matrix
reserved symbols	
$\mathbb{N}$	set of all natural numbers
$\mathbb{R}$	set of all real numbers
$\mathbb{R}_+$	set of all positive real numbers
$\mathcal{M}$	point in 3D space
$\boldsymbol{\mathcal{M}}$	homogeneous coordinates of $\mathcal{M}$
$\mathbf{M}$	3D point Euclidean coordinates of $\mathcal{M}$
$\mathbf{m}, \mathbf{c}$	2D image point
$\mathbf{p}$	2D point in camera coordinate system
$\check{\mathbf{M}}$	3D virtual point
$\check{\mathbf{m}}$	2D virtual point

I	identity matrix
$\mathbf{0}$	zero vector
$\mathsf{0}$	zero matrix
$\langle \mathbf{i}, \mathbf{j}, \mathbf{k} \rangle$	canonical base of $\mathbb{R}^3$
$\mathbf{i} = [1\,0\,0]^\top$	x direction vector
$\mathbf{j} = [0\,1\,0]^\top$	y direction vector
$\mathbf{k} = [0\,0\,1]^\top$	z direction vector
K	camera calibration matrix
R	rotation matrix
$\mathbf{t}$	translation vector
T	homogeneous motion matrix
P	projection matrix
$\mathbf{y}$	vector of residuals
$\mathbf{x}$	state/parameters vector
$\overline{\mathbf{a}}$	true value of $\mathbf{a}$
$\widetilde{\mathbf{a}}$	observation vector of $\overline{\mathbf{a}}$
$\widehat{\mathbf{a}}$	estimatation vector of $\overline{\mathbf{a}}$
statistic symbols	
σ	standard deviation
Σ	covariance matrix
$N(\overline{\mathbf{x}}, \Sigma_{\mathbf{x}})$	normal distribution with mean vector $\overline{\mathbf{x}}$ and covariance $\Sigma_{\mathbf{x}}$
mathematic operators	
$\lvert a \rvert$	absolute value of a
$\lVert \mathbf{y} \rVert$	L_2 norm of the vector $\mathbf{y}$
$\lvert \mathsf{A} \rvert$	determinant of the matrix A
A^{-1}	inverse of matrix A
$\mathsf{A}^\top$	transposed matrix of A
$[.]_\times$	3×3 skew-symmetric matrix of a 3-vector (c.f. Eq. 3.1)
$[.]_v$	matrix vectorization operator (c.f. Eq. 3.2)

$[.]_\times$ is the 3×3 skew-symmetric matrix, that represents the cross product operator. $\forall \mathbf{p}, \mathbf{q} \in \mathbb{R}^3$, $[\mathbf{p}]_\times \mathbf{q} = \mathbf{p} \times \mathbf{q}$. It is defined as follows:

$$\forall \mathbf{p} = [x\,y\,z]^\top, \quad [\mathbf{p}]_\times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} . \tag{3.1}$$

$[.]_v$ is the vectorization operator, which rearranges a matrix in a vector defined as:

$$\mathsf{A} \in \mathbb{R}^{r \times c}, \ \mathbf{a} = [\mathsf{A}]_v \ \text{ s.t. } \ \mathbf{a} = [a_{1,1}, \ldots, a_{1,c}, \ldots, a_{r,1}, \ldots, a_{r,c}]^\top , \tag{3.2}$$

with $a_{i,j}$ the element of the matrix A at row i and column j.

3.2 Multi-variate Functions and Derivatives

In this section, we describe the notation, we use, for derivatives of multi-variate functions. We consider vectors and matrices as multivariate functions that can therefore be derived.

The jacobian or first derivative of the vector function $\mathbf{a}(\mathbf{b}) = [a_1(\mathbf{b}) \cdots a_{n_a}(\mathbf{b})]^\top$ with respect to the vector $\mathbf{b} = [b_1 \cdots b_{n_b}]^\top$ is defined as the $n_a \times n_b$ matrix:

$$\mathsf{J}_{\mathbf{a}}(\mathbf{b}) = \frac{\partial \mathbf{a}}{\partial \mathbf{x}}(\mathbf{x})\bigg|_{\mathbf{x}=\mathbf{b}} = \begin{bmatrix} \frac{\partial a_1}{\partial b_1}(\mathbf{b}) & \frac{\partial a_1}{\partial b_2}(\mathbf{b}) & \cdots & \frac{\partial a_1}{\partial b_{n_b}}(\mathbf{b}) \\ \frac{\partial a_2}{\partial b_1}(\mathbf{b}) & \frac{\partial a_2}{\partial b_2}(\mathbf{b}) & \cdots & \frac{\partial a_2}{\partial b_{n_b}}(\mathbf{b}) \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial a_{n_a}}{\partial b_1}(\mathbf{b}) & \frac{\partial a_{n_a}}{\partial b_2}(\mathbf{b}) & \cdots & \frac{\partial a_{n_a}}{\partial b_{n_b}}(\mathbf{b}) \end{bmatrix} . \tag{3.3}$$

We define the $n_a \times n_b$ matrix $\mathsf{M}_{\mathbf{a}}(\mathbf{b}_1, \mathbf{b}_2)$ related to the second derivative of $\mathbf{a}$ with respect to $\mathbf{b}_1$ as:

$$\mathsf{M}_{\mathbf{a}}(\mathbf{b}_1, \mathbf{b}_2) = \left[\ \mathsf{H}_{a_1}(\mathbf{b}_1)\mathbf{b}_2 \quad \cdots \quad \mathsf{H}_{a_n}(\mathbf{b}_1)\mathbf{b}_2\ \right]^\top , \tag{3.4}$$

with $\mathbf{b}_1$, $\mathbf{b}_2 \in \mathbb{R}^{n_b}$ and H_{a_i} the second derivative (also called Hessian matrix) of the multivariate scalar function $a_i(\mathbf{b})$ with respect to $\mathbf{b}$. It is defined as follows:

$$\mathsf{H}_{a_i}(\mathbf{b}) = \frac{\partial \mathsf{J}_{a_i}}{\partial \mathbf{x}}(\mathbf{x})\bigg|_{\mathbf{x}=\mathbf{b}} \begin{bmatrix} \frac{\partial^2 a_i}{\partial b_1^2}(\mathbf{b}) & \frac{\partial^2 a_i}{\partial b_1 \partial b_2}(\mathbf{b}) & \cdots & \frac{\partial^2 a_i}{\partial b_1 \partial b_{n_b}}(\mathbf{b}) \\ \frac{\partial^2 a_i}{\partial b_2 \partial b_1}(\mathbf{b}) & \frac{\partial^2 a_i}{\partial b_2^2}(\mathbf{b}) & \cdots & \frac{\partial^2 a_i}{\partial b_2 \partial b_{n_b}}(\mathbf{b}) \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 a_i}{\partial b_{n_b} \partial b_1}(\mathbf{b}) & \frac{\partial^2 a_i}{\partial b_{n_b} \partial b_2}(\mathbf{b}) & \cdots & \frac{\partial^2 a_i}{\partial b_{n_b}^2}(\mathbf{b}) \end{bmatrix} , \tag{3.5}$$

$\mathsf{H}_{a_i}(\mathbf{b})$ is an $n_b \times n_b$ symmetric matrix ($\mathsf{H}_{a_i}(\mathbf{b})^\top = \mathsf{H}_{a_i}(\mathbf{b})$) because $\frac{\partial^2 a_i}{\partial b_i \partial b_j}(\mathbf{b}) = \frac{\partial^2 a_i}{\partial b_j \partial b_i}(\mathbf{b})$.

Finally, we extend the Jacobian operator $\mathsf{J}_{\mathbf{a}}(\mathbf{b})$ to matrices:

$$\mathsf{J}_{\mathsf{A}}(\mathsf{B}) := \mathsf{J}_{[\mathbf{A}]_v}\left([\mathsf{B}]_v \right) , \tag{3.6}$$

where $\mathsf{J}_{\mathsf{A}}(\mathsf{B})$ is a matrix of dimension $r_A c_A \times r_B c_B$.

This generalization of the Jacobian operator to matrices is useful when deriving an image function represented by a matrix function.

3.3 Non-linear Least Square Optimization

In this section, we give a brief introduction to optimization of non-linear least square problems. Specifically we are focusing our attention to minimization which will be the optimization problem tackled in this thesis. For a more thorough presentation on optimization the reader is referred to [Flecher, 1987].

Let $f(\mathbf{x})$ be the function we want to minimize:

$$f: \quad \mathbb{R}^m \quad \rightarrow \quad \mathbb{R}_+ \quad , \tag{3.7}$$

which can be expressed as a sum of squares:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{y}(\mathbf{x})^\top \mathbf{y}(\mathbf{x}) \ , \tag{3.8}$$

with $\mathbf{y}(\mathbf{x}) = [y_1(\mathbf{x}) \ y_2(\mathbf{x}) \ \ldots \ y_n(\mathbf{x})]^\top$ referred to as the set of residuals.

We want to estimate a minimizer $\overline{\mathbf{x}}$ of f:

$$\overline{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}) \ , \tag{3.9}$$

Here we present local and iterative methods to estimate a minimizer of f from a current estimate $\widehat{\mathbf{x}}$. The current estimate $\widehat{\mathbf{x}}$ is updated by an increment $\Delta\mathbf{x}$ estimated by the optimizer. A classic and simple optimizer is the steepest descent where the update is computed as follows:

$$\Delta\mathbf{x} = -\alpha \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}})^\top \mathbf{y}(\widehat{\mathbf{x}}) \ , \tag{3.10}$$

with $\alpha \in \mathbb{R}_+^*$.
As $f >= 0$ and supposing that α is selected to be small enough the update obtained by the gradient descent will decrease f. If enough iteration are performed, the optimization should reach a stable point where the gradient is zero. This indicates the possibility of a minimum. Even though we can guarantee convergence with such an optimizer, the choice of α is complicated and might lead to a slow convergence. Therefore we introduce more complex methods called Newton methods which, in most cases, offer faster convergence[1].

We recall that a minimizer $\check{\mathbf{x}}$ of f has a zero gradient:

$$\overline{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}) \Rightarrow \mathsf{J}_f(\mathbf{x}) = \mathbf{0} \,. \tag{3.11}$$

By deriving Equation 3.8, we obtain the jacobian of f as:

$$\mathsf{J}_f(\mathbf{x}) = \mathsf{J}_{\mathbf{y}}(\mathbf{x})^\top \mathbf{y}(\mathbf{x}) \ . \tag{3.12}$$

A Taylor expansion of J_f around the current estimate $\widehat{\mathbf{x}}$ leads to:

$$\begin{aligned} \mathsf{J}_f(\widehat{\mathbf{x}} + \Delta\mathbf{x}) = & \ \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}} + \Delta\mathbf{x})^\top \left(\mathbf{y}(\widehat{\mathbf{x}}) + \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}})\Delta\mathbf{x} + \mathrm{O}\left\|\Delta\mathbf{x}\right\|^2\right) \\ = & \ \left(\mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}}) + \mathbf{M}(\widehat{\mathbf{x}}, \Delta\mathbf{x}) + \mathrm{O}\left\|\Delta\mathbf{x}\right\|^2\right)^\top \left(\mathbf{y}(\widehat{\mathbf{x}}) + \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}})\Delta\mathbf{x} + \mathrm{O}\left\|\Delta\mathbf{x}\right\|^2\right) \\ = & \ \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}})^\top \mathbf{y}(\widehat{\mathbf{x}}) + \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}})^\top \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}})\Delta\mathbf{x} + \mathbf{M}(\widehat{\mathbf{x}}, \Delta\mathbf{x})^\top \mathbf{y}(\widehat{\mathbf{x}}) + \mathrm{O}\left\|\Delta\mathbf{x}\right\|^2 \\ \approx & \ \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}})^\top \mathbf{y}(\widehat{\mathbf{x}}) + \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}})^\top \mathsf{J}_{\mathbf{y}}(\widehat{\mathbf{x}})\Delta\mathbf{x} + \mathbf{M}(\widehat{\mathbf{x}}, \Delta\mathbf{x})^\top \mathbf{y}(\widehat{\mathbf{x}}) \ . \end{aligned} \tag{3.13}$$

[1] In the sense of fewer iterations to obtain a minimum.

The Newton update can therefore be found as:

$$\Delta\mathbf{x} = -\mathsf{S}^{-1}\mathsf{J}_{\mathbf{y}}\left(\widehat{\mathbf{x}}\right)^{\top}\mathbf{y}\left(\widehat{\mathbf{x}}\right)\ , \tag{3.14}$$

with

$$\mathsf{S} = \mathsf{J}_{\mathbf{y}}\left(\widehat{\mathbf{x}}\right)^{\top}\mathsf{J}_{\mathbf{y}}\left(\widehat{\mathbf{x}}\right) + \sum_{i=1}^{m} y_i\left(\widehat{\mathbf{x}}\right)\mathsf{H}_{y_i}\left(\widehat{\mathbf{x}}\right)\ . \tag{3.15}$$

Newton optimizers converge quadratically[2] in the neighborhood of $\overline{\mathbf{x}}$. Unfortunately, it requires to compute the inverse of S, which might not exist because the weighted sum of Hessian matrices might not be positive definite[3]. Additionally, computing m Hessian matrices at each iteration is computationally expensive. Therefore different approximations of S are usually employed. Most of them are based on the idea that around the minimum $\overline{\mathbf{x}}$ the second term of 3.15 is negligible because residuals $\mathbf{y}$ are small.
This leads to the Gauss-Newton method, which defines its update as:

$$\mathsf{S} = \mathsf{J}_{\mathbf{y}}\left(\widehat{\mathbf{x}}\right)^{\top}\mathsf{J}_{\mathbf{y}}\left(\widehat{\mathbf{x}}\right)\ . \tag{3.16}$$

It converges almost quadratically. Unfortunately, convergence cannot be guaranteed and the term $\mathsf{J}^{\top}\mathsf{J}$ may be ill-conditioned. In order to deal with these limitations, modifications to Gauss-Newton have been proposed.
For example, Levenberg-Marquardt augments the diagonal of $\mathsf{J}^{\top}\mathsf{J}$ to force S to be definite positive. This leads to an optimizer that can switch between a gradient-descent and a Gauss-Newton with guaranteed convergence:

$$\mathsf{S} = \mathsf{J}_{\mathbf{y}}\left(\widehat{\mathbf{x}}\right)^{\top}\mathsf{J}_{\mathbf{y}}\left(\widehat{\mathbf{x}}\right) + \lambda\mathbf{I}\,, \tag{3.17}$$

where $\lambda \in \mathbb{R}_{+}^{*}$ is a damping factor chosen such that the selected step decreases f. For a given λ, an update is computed based on S: if it leads to decreasing f, λ is decreased (e.g. $\lambda \leftarrow 0.1 \times \lambda$) to go towards a Gauss-Newton else λ is increased (e.g. $\lambda \leftarrow 10 \times \lambda$). This increase of λ is repeated until an update is found that decreases f; this modifies the optimizer behavior towards a steepest descent.

Optimizers presented here are iterative. They iterate between computing the step $\Delta\mathbf{x}$ and updating the current estimate $\widehat{\mathbf{x}}$ until convergence. We use two different stopping criteria. The first one is related to the update size (i.e. $\|\Delta\mathbf{x}\| < \epsilon$), which indicates a small gradient of f. This is a necessary condition for a minimum. Unfortunately it could also indicate a saddle point [Flecher, 1987]. The second condition is a maximum on the number of iterations in order to avoid an infinite loop when oscillating or diverging.

To illustrate the property of these different optimizers, we try to find the minimum of the sum of square cost function shown in Figure 3.1(a). This cost function has a local minimum closer to the global minimum located at $\mathbf{0}$. As mentioned before, the gradient descent (Figure 3.1(b)) convergences towards the closest minimum and therefore can get stuck in a local minimum. Additionally, the number of iterations is high and the update

[2] We say that an optimizer has quadratic convergence when $f_{(t+1)}/f_{(t)}^{2} \leqslant a$ [Flecher, 1987]. It can be seen as how quick an optimizer minimize f.

[3] A is definite positive if and only if $\forall \mathbf{x} \in \mathbb{R}_{+}^{n} \Rightarrow \mathbf{x}^{\top}\mathsf{A}\mathbf{x} > 0$

step decreases the closest it gets to the minimum because the gradient of f decreases. Therefore the computational time of a gradient descent can be large to obtain a precise result. Newton optimizer (Figure 3.1(c)) offers in this case the best results but requires extra computation. Gauss-Newton (Figure 3.1(d)) diverges when the matrix S is badly conditioned. This often happens when the optimizer is badly initialized. But it offers a fast convergence close to the minimum. The Levenberg-Marquardt method (Figure 3.1(e)) offers a good trade-off between the guaranteed convergence of the gradient descent and the fast convergence of the Gauss-Newton method close to the minimum.

In this thesis, when minimizing a non-linear sum of square cost function we use either a Gauss-Newton or a Levenberg-Marquardt optimizer.

3.4 Special Euclidean Group - $\mathbb{SE}(3)$

In this section, we introduce the special Euclidean group denoted $\mathbb{SE}(3)$, which will be used in this thesis to parametrize the motion of cameras. For a more detailed description to $\mathbb{SE}(3)$, its associated Lie algebra and their applications to 3D Computer Vision, the reader is referred to [Benhimane, 2007] and [Klein, 2006].

The Special Euclidean group $\mathbb{SE}(3)$ is the group of rigid body motions, which includes translations and rotations. In a neighborhood of I, $\mathbb{SE}(3)$ can be parametrized using its associated Lie algebra $\mathfrak{se}(3)$ using the matrix exponential:

$$\begin{array}{rccl} \exp: & \mathfrak{se}(3) & \rightarrow & \mathbb{SE}(3) \\ & \mathsf{A} & \mapsto & \exp(\mathsf{A}) = \sum_{i=0}^{\infty} \frac{1}{i!} (\mathsf{A})^i \ , \end{array} \tag{3.18}$$

A is a 4×4 matrix.

We defined the generators of rotations of $\mathbb{R}^3$ as:

$$\mathsf{A}_1 = \begin{bmatrix} [\mathbf{i}]_\times & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}, \mathsf{A}_2 = \begin{bmatrix} [\mathbf{j}]_\times & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}, \mathsf{A}_3 = \begin{bmatrix} [\mathbf{k}]_\times & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}, \tag{3.19}$$

and the generators for translations:

$$\mathsf{A}_4 = \begin{bmatrix} \mathsf{0} & \mathbf{i} \\ \mathbf{0}^\top & 0 \end{bmatrix}, \mathsf{A}_5 = \begin{bmatrix} \mathsf{0} & \mathbf{j} \\ \mathbf{0}^\top & 0 \end{bmatrix}, \mathsf{A}_6 = \begin{bmatrix} \mathsf{0} & \mathbf{k} \\ \mathbf{0}^\top & 0 \end{bmatrix}. \tag{3.20}$$

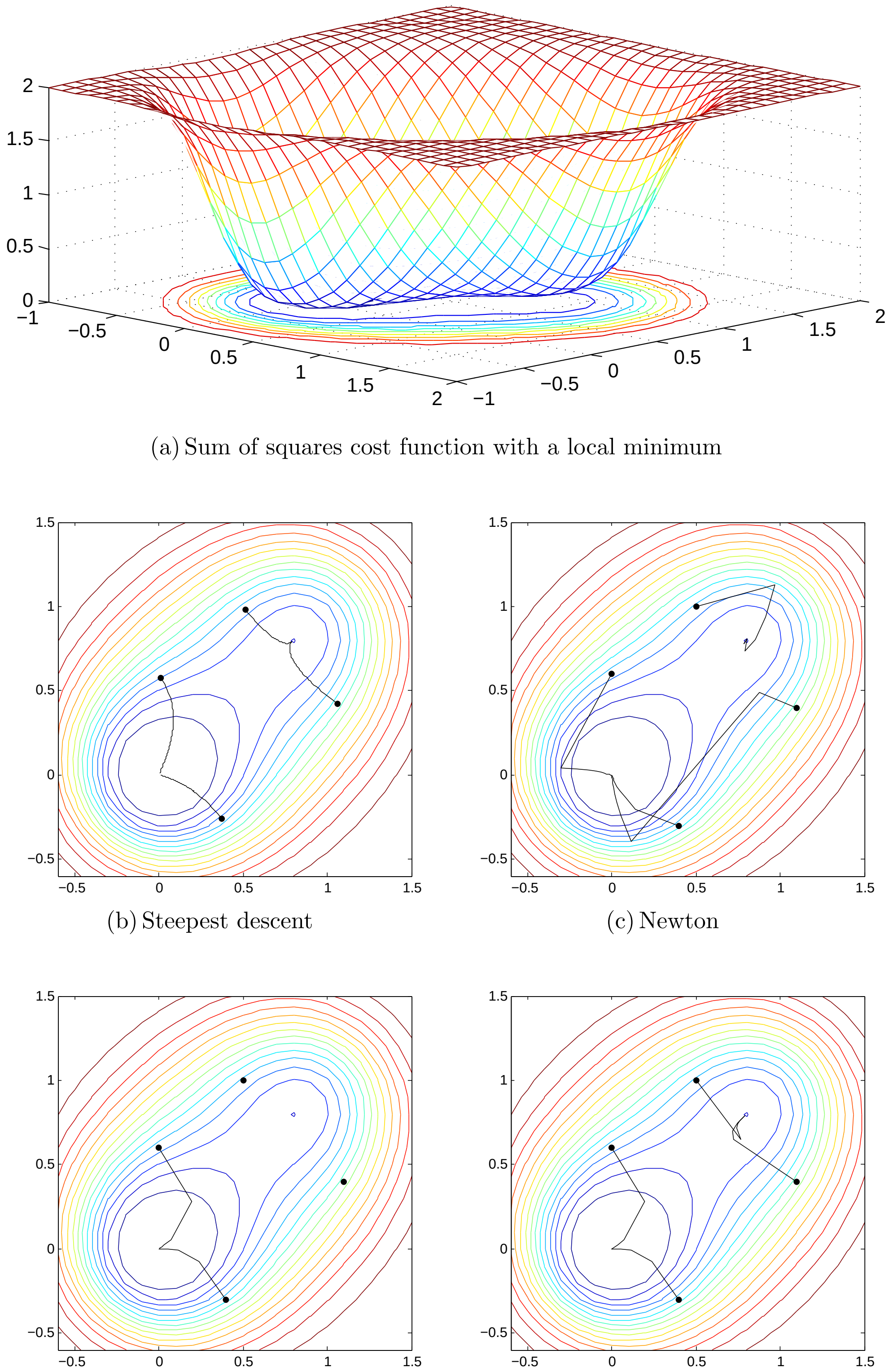
We define by $\mathbf{x} = [x_1, x_2, \cdots, x_6]^\top$ the linear coefficients of $\mathsf{A} \in \mathfrak{se}(3)$ such that:

$$\mathsf{A}(\mathbf{x}) = \sum_{i=1}^{6} x_i \mathsf{A}_i = \begin{bmatrix} [\mathbf{u}(\mathbf{x})\theta(\mathbf{x})]_\times & \boldsymbol{\beta}(\mathbf{x}) \\ \mathbf{0}^\top & 0 \end{bmatrix}, \tag{3.21}$$

with $\theta(\mathbf{x}) := \|\mathbf{x}_\mathsf{R}\|$, $\mathbf{u}(\mathbf{x})\theta(\mathbf{x}) := \mathbf{x}_\mathsf{R} := [x_1\, x_2\, x_3]^\top$, $\boldsymbol{\beta}(\mathbf{x}) := \mathbf{x}_\mathbf{t} := [x_4\, x_5\, x_6]^\top$.

It is possible to express $\mathsf{T} \in \mathbb{SE}(3)$ as a function of the coefficients $\mathbf{x}$, if it is in a neighborhood of I as follows:

$$\mathsf{T}(\mathbf{x}) = \exp(\mathsf{A}(\mathbf{x})) = \begin{bmatrix} \mathsf{R}(\mathbf{x}) & \mathbf{t}(\mathbf{x}) \\ \mathbf{0}^\top & 0 \end{bmatrix}, \tag{3.22}$$



(a) Sum of squares cost function with a local minimum

(b) Steepest descent

(c) Newton

(d) Gauss-Newton

(e) Levenberg-Marquardt

Figure 3.1: **Minimizer results on a function with a local minimum.**

where rotations are parametrized using the Rodriguez formula:

$$\mathsf{R}(\mathbf{x}) = \mathsf{I} + \sin(\theta(\mathbf{x}))\,[\mathbf{u}(\mathbf{x})]_\times + (1 - \cos(\theta(\mathbf{x})))\,[\mathbf{u}(\mathbf{x})]_\times^2 \ , \tag{3.23}$$

and translations are parametrized as follows:

$$\mathbf{t}(\mathbf{x}) = \left(\mathsf{I} + \frac{1 - \cos(\theta(\mathbf{x}))}{\theta(\mathbf{x})}\,[\mathbf{u}(\mathbf{x})]_\times + \left(1 - \frac{\sin(\theta(\mathbf{x}))}{\theta(\mathbf{x})}\right)[\mathbf{u}(\mathbf{x})]_\times^2\right)\beta(\mathbf{x})\ . \tag{3.24}$$

So $\mathbf{x}$ is a parametrization of the rigid body movement of $\mathbb{R}^3$ (expressed from an isomorphism of projective space $\mathbb{P}^3$) around I. This parametrization is used throughout this thesis to represent camera motion.

3.5 Basic Image Geometry

In this section, we describe the camera motion and projective model used in this thesis. Let $\mathcal{M}$ be a point in 3D space and $\mathcal{C}$ the canonical coordinate system of the 3D space. In $\mathcal{C}$, $\mathcal{M}$ has for coordinates $\mathbf{M} = [x\,y\,z]^\top \in \mathbb{R}^3$. Let $\mathcal{C}_i$ be an another coordinate system of the 3D space, in $\mathcal{C}_i$ $\mathcal{M}$ is defined by $\mathbf{M}_i = [x_i\,y_i\,z_i]^\top$. The change of coordinate system between $\mathcal{C}$ and $\mathcal{C}_i$ is achieved by a rotation $\mathsf{R}_i \in \mathbb{SO}\,(3)$ ($|\mathsf{R}_i| = 1$ and $\mathsf{R}_i^\top \mathsf{R}_i = \mathsf{I}$) and a translation $\mathbf{t}_i \in \mathbb{R}^3$. $\mathbb{SO}\,(3)$ is the super-orthogonal group of dimension 3. This leads to the following change of coordinate system equation:

$$\mathbf{M}_i = \mathsf{R}_i\mathbf{M} + \mathbf{t}_i\ . \tag{3.25}$$

This change of coordinate systems is what we describe as the *motion*.

The motion between the coordinate systems $\mathcal{C}$ and $\mathcal{C}_i$ can be expressed by the transformation $\mathsf{T}_i \in \mathbb{SE}\,(3)$, where $\mathbb{SE}\,(3)$ is the special euclidean group (*c.f.* Section 3.4). T_i has the following form:

$$\mathsf{T}_i = \begin{bmatrix} \mathsf{R}_i & \mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}\ . \tag{3.26}$$

Because $\mathbb{SE}\,(3)$ is a group and therefore has a multiplicative law:

$$\mathsf{T}_1, \mathsf{T}_2 \in \mathbb{SE}\,(3)\,, \quad \mathsf{T} = \mathsf{T}_1\mathsf{T}_2 \quad \Rightarrow \quad \mathsf{T} \in \mathbb{SE}\,(3)\ . \tag{3.27}$$

Using transformation parametrized using $\mathfrak{se}\,(3)$, we can guarantee that the composition of transformations is in $\mathbb{SE}\,(3)$.
We define the homogeneous coordinates $\boldsymbol{\mathcal{M}}$ of the 3D point $\mathcal{M}$ as follow:

$$\boldsymbol{\mathcal{M}} \sim \begin{bmatrix} \mathbf{M} \\ 1 \end{bmatrix}\,, \tag{3.28}$$

with $\mathbf{M}$ the coordinates of $\mathcal{M}$ in $\mathbb{R}^3$ and $\sim$ the equality relation of the projective space $\mathbb{P}^n$, in this case $n = 3$, defined as follows:

$$\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{M}} \in \mathbb{P}^n, \quad \boldsymbol{\mathcal{L}} \sim \boldsymbol{\mathcal{M}} \quad \Leftrightarrow \quad \exists \alpha \in \mathbb{R}^*,\ \text{s.t. } \boldsymbol{\mathcal{L}} = \alpha\boldsymbol{\mathcal{M}}\,, \tag{3.29}$$

with $=$ the equality relation of the euclidean space $\mathbb{R}^{n+1}$.
Using the rigid transformation T_i, we can express the coordinate system change in homogeneous coordinates. Equation 3.25 is replaced by:

$$\boldsymbol{\mathcal{M}}_i \sim \mathsf{T}_i \boldsymbol{\mathcal{M}} \, . \tag{3.30}$$

The registration problems tackled in this thesis can be defined as the estimation of a rigid transformation T_i from the CAD coordinate system $\mathcal{C}$ to a camera coordinate system $\mathcal{C}_i$, as shown in Figure 3.3.

3.5.1 Camera Model

In this section, we describe the pin-hole camera model and its parametrization. We suppose that a projective camera, also called pinhole model, approximates properly our optical system [Hartley and Zisserman, 2003]. In this model, a 3D point $\mathcal{M}$ projects as a point $\mathbf{m}_i$ on the retina of camera $\mathcal{C}_i$. This projection is at the intersection of the retina and the optical ray passing through $\mathcal{M}$ and the camera center $\mathbf{O}_i = -\mathsf{R}_i^\top \mathbf{t}_i$. The retina is henceforth called image plane. It is a projection from $\mathbb{P}^3$ to $\mathbb{P}^2$, which can be presented by the matrix operation P_i:

$$\mathbf{m} \sim \mathsf{P}_i \boldsymbol{\mathcal{M}} \, . \tag{3.31}$$

By decomposing P_i we can separate the motion of the camera or extrinsic parameters R_i and $\mathbf{t}_i$; and the intrinsic parameters K_i:

$$\mathsf{P}_i \sim \mathsf{K}_i \left[\begin{array}{cc} \mathsf{R}_i & \mathbf{t}_i \end{array} \right] \, , \tag{3.32}$$

with K_i is a 3×3 matrix.

The intrinsic parameters are the internal properties of the optical system, which are the focal length (f_x, f_y) in both x and y directions, the skew θ_s (the angle between the image axes) and the principal point $\mathbf{u}_0 = [u_x \, u_y \, 1]^\top$. The matrix K_i is parametrized as follows [Faugeras and Luong, 2001]:

$$\mathsf{K}_i = \begin{bmatrix} f_x & -f_x \operatorname{cotan}(\theta_s) & u_x \\ 0 & \dfrac{f_y}{\sin(\theta_s)} & u_y \\ 0 & 0 & 1 \end{bmatrix} \, . \tag{3.33}$$

Nowadays with CCD cameras, a no-skew hypothesis is realistic [Hartley and Zisserman, 2003]. Therefore we rewrite Equation 3.33 as follows:

$$\mathsf{K}_i = \begin{bmatrix} f_x & 0 & u_x \\ 0 & f_y & u_y \\ 0 & 0 & 1 \end{bmatrix} \, . \tag{3.34}$$

It is possible to estimate these parameters during off-line process. This process is called calibration. Cameras are usually calibrated using a known pattern [Tsai, 1987, Zhang, 1999]. This calibration object for detection simplicity is often a checker board. We use

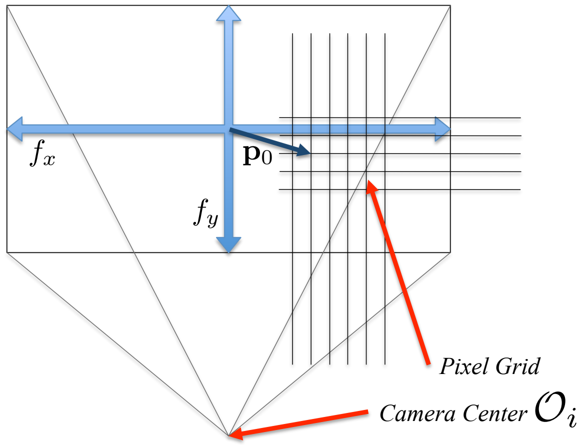


Figure 3.2: **Pinhole Camera Model** is parametrized by the matrix K_i of internal parameters. This matrix transform points from camera coordinate system $\mathcal{C}_i$ to the pixel grid. The focal length defines the camera field of view and the principal point $\mathbf{u}_0$ the shift of the camera center on the pixel grid.

the implementation provided by OpenCV to perform this task, which is a transposition of the Caltech calibration toolbox available for Matlab [Bouguet]. We say of a camera is calibrated when its intrinsic parameters are known.

In reality, the pin-hole model does not hold because of distortions. The distortions (radial and tangential) are due to the optical system (e.g. the lens), which deforms straight lines present in the scene to curves in the image. This phenomenon can be compensated for by a warping function. This function is parametrized by non-linear coefficients. These coefficients are estimated during the calibration along with the camera's intrinsic parameters.

3.5.2 Projection Formulation

Using the rigid motionT_i, Equation 3.31 is rewritten as follows:

$$\mathbf{m}_i = \mathsf{K}_i \mathbf{w}\left(\mathsf{T}_i \boldsymbol{\mathcal{M}}\right) \ , \tag{3.35}$$

with $\mathbf{w}$ the warping function defined as follows:

$$\begin{array}{rlcl} \mathbf{w}: & \mathbb{P}^3 & \longrightarrow & \mathbb{P}^2 \\ & \boldsymbol{\mathcal{M}}_i \sim \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} & \mapsto & \mathbf{p}_i \sim \begin{bmatrix} \frac{x_i}{z_i} \\ \frac{y_i}{z_i} \\ 1 \end{bmatrix} \end{array} , \tag{3.36}$$

which projects the 3D point $\mathcal{M}$ defined in the current camera coordinate system $\mathcal{C}_i$ by $\boldsymbol{\mathcal{M}}_i$ to the point $\mathbf{p}_i$ on the camera plane $\pi = [0\,0\,1]^\top$.

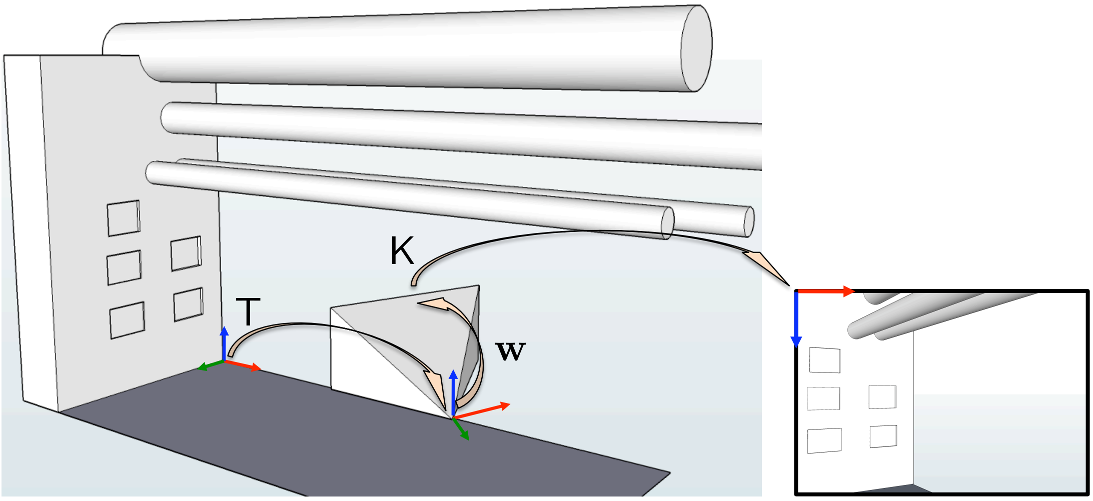


Figure 3.3: **Registration Schematic**. The rigid transformation T changes the coordinate system from CAD to camera. After projection on the camera plane using $\mathbf{w}$, the image coordinate system is obtained by using K.

.

3.6 Image Representation

In this section, we describe the image representation used in this thesis. Images are represented by computers as grids G_i of dimensions $r \times c$. Elements of this matrix are called pixels. They store an intensity for gray scale images or a tuple of intensities for color images. For simplicity, we study gray scale images or color images converted in gray scale, but the methods should easily be extensible to color images.

We suppose that there exists a function $\mathcal{I}_i$ that is discretely sampled by G_i, defined as follows:

$$\begin{aligned} \mathcal{I}_i : \quad \Omega \subset \mathbb{P}^2 & \quad \rightarrow \quad \mathbb{R} \\ \mathbf{m}_i \sim \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} & \quad \mapsto \quad \mathcal{I}_i\left(u, v\right) \ , \end{aligned} \tag{3.37}$$

with Ω the underlying continuous space sampled by the image grid of G_i. We suppose that the function $\mathcal{I}_i$ is differentiable for all degrees of differentiation (i.e. $\mathcal{C}^\infty$). For simplicity, we define $\mathcal{I}_i^{'}$ as the first derivatives of $\mathcal{I}_i$.

Since observations of the image function $\mathcal{I}_i$ are only available on a discrete grid $\{1, \cdots, r_{\mathsf{G}_i}\} \times \{1, \cdots, c_{\mathsf{G}_i}\}$, interpolation of the intensity is required when a point $\mathbf{m}_i$ is not one of the grid points. We use a bi-linear interpolation, which is defined as follows:

$$\mathcal{I}_i(u,v) = \frac{1}{4} \left(\begin{array}{cl} & (1+\underline{u}-u)(1+\underline{v}-v)\mathsf{G}_i(\underline{u},\underline{v}) \\ + & (u-\underline{u})(1+\underline{v}-v)\mathsf{G}_i(\underline{u}+1,\underline{v}) \\ + & (1+\underline{u}-u)(v-\underline{v})\mathsf{G}_i(\underline{u},\underline{v}+1) \\ + & (u-\underline{u})(v-\underline{v})\mathsf{G}_i(\underline{u}+1,\underline{v}+1) \end{array} \right) , \tag{3.38}$$

with $\underline{x}$ the integer component of x.

Finally, we suppose that the direct projections of a the same 3D point $\mathcal{M}$ in two different images $\mathcal{I}_i$ and $\mathcal{I}_j$ result in the same intensity response. This *image consistency assumption* can be summarized as follows:

$$\forall \mathcal{M}, \quad \mathcal{I}_i \left(\mathsf{K}_i \mathbf{w} \left(\mathsf{T}_i \boldsymbol{\mathcal{M}} \right) \right) = \mathcal{I}_j \left(\mathsf{K}_j \mathbf{w} \left(\mathsf{T}_j \boldsymbol{\mathcal{M}} \right) \right) \ . \tag{3.39}$$

In reality, this assumption does not hold because of image noise. Therefore, we suppose that G_i is a perturbed observation of true intensities defined by the image $\overline{\mathsf{G}}_i$. Furthermore, we suppose that the image noise is independent and has a Gaussian distribution with zero mean:

$$\mathsf{G}_i \left(u, v \right) = \mathcal{N} \left(\overline{\mathsf{G}}_i \left(u, v \right), \sigma_{\mathsf{G}_i} \right) \ , \tag{3.40}$$

with σ_{G_i} the noise's standard deviation.

3.7 Point and Region detectors

In this section, we review the state of the art for local feature detectors and matching algorithms. Point and region detectors are the base of most Computer Vision algorithms as they provide the 2D measurements necessary for pose estimation (c.f. Section 3.8), 3D point triangulation (c.f. Section 3.9.2), etc. Here we describe some background knowledge related to local features (points and regions), localization and matching along with some standard methods used in this thesis.

First, spacial detection is performed using a mathematical operator to localize a characteristic feature. A local feature is found, where the operator output attains a local extremum. The detection process often includes a further processing step to find stable features, since local features are the starting step for many algorithms. The result of these algorithms can only be as precise as the detector itself. Thus, a desirable property for a local feature detector is stability, which can be measured in terms of repeatability [Schmid et al., 2000]. Repeatability signifies that the detection is independent of changes in the imaging conditions, like camera parameters, viewpoint changes, and illumination conditions; i.e. a feature detected in one image, should also be detected at exactly the same position in the other image. Unfortunately, the detection process can be affected by noisy measurements that can result in some localization error. [Schmid et al., 2000, Mikolajczyk et al., 2005] present comprehensive evaluations of local feature detectors.

Second, mechanisms to match features across images have to be put in place. Matching of local features in two images is often done by correlation or with the help of a descriptor. While a correlation approach directly compares the intensities around the interest points, a descriptor characterizes the feature via the image structure in the neighborhood. Using the gradient field around an interest point has proven to be very successful in many computer vision applications [Lowe, 2004, Bay et al., 2008]. In the majority of cases the descriptor itself is a vector of predefined length containing weighted samples from this gradient field. While distinctiveness and occlusion robustness are reached by an appropriate size and sampling of the gradient field, invariance to image transformations is guaranteed by the detection of invariant property. These characteristics are covered in the remaining of this section.

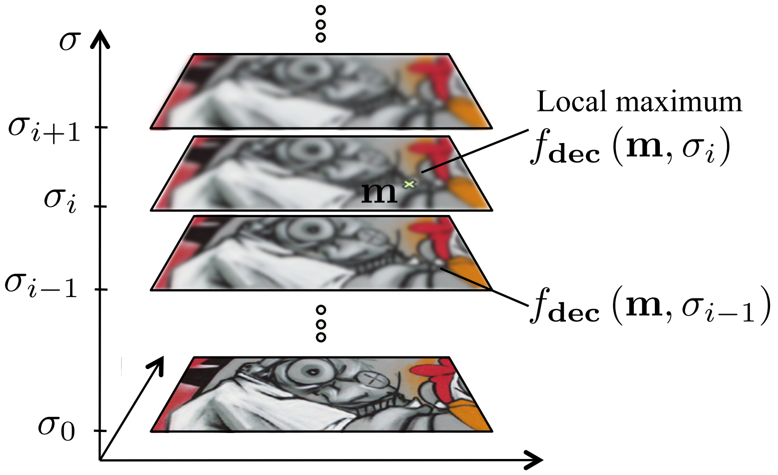

σ
σ_{i+1}
σ_i
σ_{i-1}
σ_0
$\mathbf{m}$
Local maximum
$f_{\mathbf{dec}}(\mathbf{m}, \sigma_i)$
$f_{\mathbf{dec}}(\mathbf{m}, \sigma_{i-1})$

Figure 3.4: Scale Space Representation of an Image. The different layers are created from the initial image by convolution with an increasing Gaussian blur kernel.

It is not in the scope of this thesis to discuss descriptors. An introduction and a comparative evaluation of descriptors can be found in [Mikolajczyk and Schmid, 2005].

3.7.1 Invariant Detection and Description

In this section, we describe how scale invariance is achieved. We say of a detector that it is scale invariant, when the same physical location can be detected in images captured from different distances (or with various focal lengths), as shown in Figure 3.4. In the last years, research efforts have been focused in developing detectors and descriptors which are invariant to changes of scale, rotation, and affine distortions.

3.7.1.1 Scale Invariance

Scale invariance is achieved by searching for interest points at different resolution in the image and finding a characteristic scale to describe the point. Thereby an image stack of increasingly smoothed layers is created to represent this differrent resolution in a “continuous manner” (c.f. Figure 3.4). This generates a *scale space*. [Lindeberg, 1994] has shown that the Gaussian kernel is the only valid smoothing operator for the creation of the scale space, because it does not introduce artificial structures in the image. Each layer is then processed with the particular detection operator, resulting in a stack of detector responses. Some implementation represents the stack as a pyramid because decreasing the image size by two is equivalent to doubling the σ of the blurring kernel, and is cheaper to compute. The created scale-space ensures that each feature will be represented at different scales. Thus, if two images contain the same features but observed at different sizes, there should be for each a layer on which they are equivalent.

The remaining task is to find this characteristic scale. Figure 3.5 illustrates the problem statement in the scale selection process. The term characteristic originally referred to the fact that the selected scale estimates the characteristic length of the corresponding

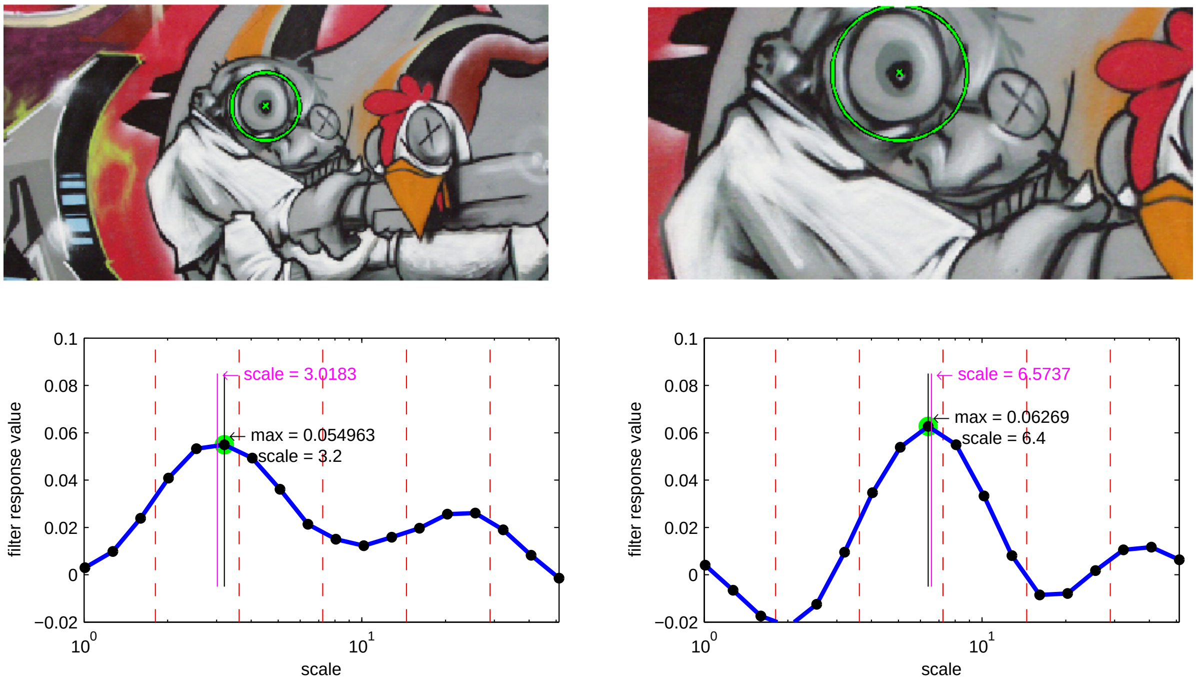
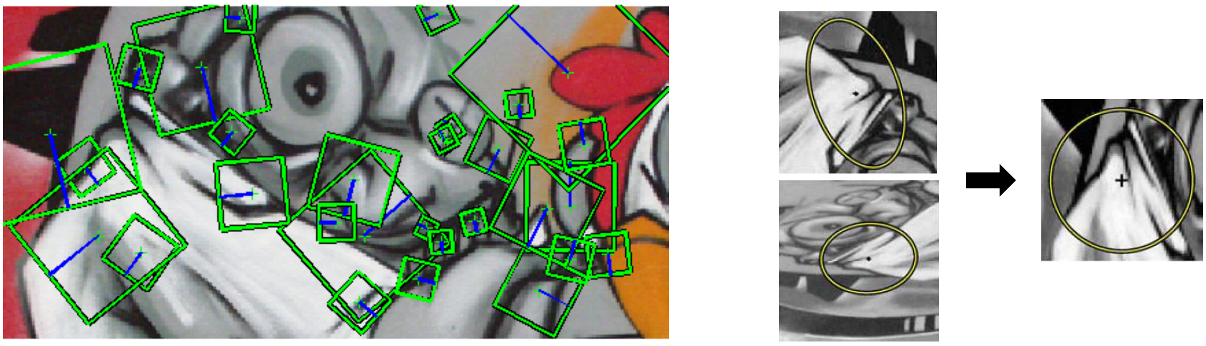


Figure 3.5: **Detection of Characteristic Scale for local features**: The top row shows two images of the same scene but captured with different focal length. The circle sizes are indicating the characteristic scales for the detected SIFT points and are according to the size of the features in the images. The bottom row shows the scale selection operator responses at the feature points over scale. Here the maximum of the filter response indicates the characteristic scale and the ratio of 6.4 to 3.2 reveals that the right image exhibits a zooming factor of 2.

image structures. [Lindeberg, 1998] studied automatic scale selection and the properties of the selected scales extensively. The idea is to select the characteristic scale of a local structure, for which a given function attains an extremum across scales. The selected scale measures the scale at which there is maximum similarity between the feature detection operator and the local image structures. This scale estimate will (for a given image operator) obey scale invariance under rescaling of the image pattern.

3.7.1.2 Rotation Invariance

Rotation invariance is achieved by an appropriate design of the descriptor, because the detector itself does not account for it. A majority of interest point descriptors are built from the local image structure around the detected point; more precisely it samples and quantized the gradient information into bins. When the descriptor is not intrinsically invariant to rotation as a histogram would, an ad-hoc solution is used. The method of choice is to estimate the direction of the strongest gradient. By rotating the gradient field in such a way that the major orientation is facing upwards, descriptors are then independent from the current feature orientation, thus accounting for rotation invariance. An illustration of the concept is given in Figure 3.6(a).



(a)Descriptor regions

(b)Affine deformation

Figure 3.6: **Descriptor Invariance**. 3.6(a): Interest point neighborhood used for creating the SIFT descriptor. The detection scale controls the size of the influencing area. Rotation invariance can be achieved by rotating the gradient field, such that the strongest gradient direction (overlaid as blue line) is facing upwards. 3.6(b): Invariance to affine transformations is achieved by determining the descriptor from a normalized point neighborhood that is influenced by the particular present affine shape.

3.7.1.3 Affine Invariance

In the case of affine transformations the scale change is, in general, different in each direction. Then automatically selected scales do not reflect the real transformation of a point. A shift-free estimation is possible if the image patch around an interest point is normalized according to the underlying affine transformation. Estimation of the underlying affine transformation can be done in different ways. Neighboring points can be used as supporting points for a homography estimation, as well as the shape of the second moment matrix. The estimated transformation is then used to normalize the descriptor. Figure 3.6(b) displays this idea. For more information on affine invariant features, the interested reader is referred to [Mikolajczyk et al., 2005].

In this thesis we focus on the quality of the 2D measurements given by local features, therefore we do not focus on affine and rotation invariance. Affine and rotation invariance have no consequence with respect to the precision of detection as opposed to the detection scale since it modifies the original image use to detect the features.

3.7.2 Corner versus Blob Detection

In this section, we describe the difference between corner and blob detectors and then focus on particular approaches. For an illustration of typical points and regions found by a corner and blob detectors, see Figure 3.7.

3.7.2.1 Corner Detectors

In this section, we review the state of the art on algorithm to detect corner points. A corner is defined to be the location where at least two dominant directions in an image

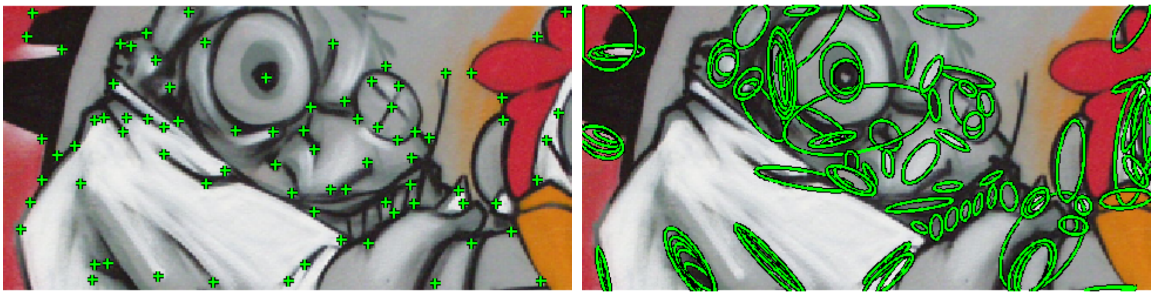


Figure 3.7: **Interest points** detected by the Harris corner detector (left) and the MSER blob detector (right). The ellipses in the right image specify the interest regions found by the detector. Note how the different methods detect different properties of the image.

intersect. These kinds of well defined positions will be found by such detectors. However, most corner detectors are sensitive not specifically to corners, but to local image regions which have a high degree of variation in all directions. Thus, they will also detect isolated points of local intensity maxima and minima.

The first automatic algorithm to detect corners are those of [Hannah, 1974] based on the gradient of the auto-correlation function and [Moravec, 1977] that searches for points with large variance. Then in photogrammetry, [Förstner and Gülch, 1987] proposes a two step algorithm first finding optimal windows where good features lie and then find where the feature exactly is in each windows. This discovery was followed by the one of [Harris and Stephens, 1988] in the machine vision field that looks at the structure tensor. The "Harris" corner is described in details in 3.7.3.1. More recently, features based on small circles have taken some momentum with feature like SUSAN [Smith and J.M.Brady, 1995], FAST [Rosten and Drummond, 2005] and FAST-ER [Rosten et al., 2010].

All these approach are scale dependent. A method to upgrade them to scale invariant methods can be found in 3.7.3.2.

3.7.2.2 Blob Detectors

In this section, we describe the principles behind some of algorithms to detect blobs. In comparison to corner detectors, a blob detector searches for areas that are brighter or darker than their neighborhoods. Each blob is, in general, localized by a well-defined point: its center of mass. Often the neighborhood size is dependent on the size of the blob itself. Unfortunately, algorithms that detect blobs in scale space often require massive blurring which worsens the precision of the detector. In this thesis, we only focused on region detectors based on differential filters. For a majority of this detection algorithm can be split up into 3 steps. First, the algorithm builds a set of detector responses based on a detector function $f_{\mathbf{dec}}$ to represent the scale space. Second, a non-maximum suppression approach is used to locate interest points spatially using $f_{\mathbf{dec}}$ and in scale using a function for scale selection $f_{\mathbf{sel}}$ ($f_{\mathbf{dec}}$ might be different from $f_{\mathbf{sel}}$). Finally detected feature points are interpolated in scale space to get a more accurate estimate. Additionally some algorithms require post processing (e.g edge response removal).

Other type of regions detector have been developed based on iterative thresholding or contours [Mikolajczyk et al., 2005].

The most prominent multi-scale features detector (SIFT) is based on the difference of Gaussians (DoG) and will be described in detail in Section 3.7.3.3. The DoG is an approximation of the Laplacian operator. The Laplacian operator was first used by Lindeberg [1998] method to detect local features in scale space. More recentely, a fast approximation of SIFT was introduced by Bay et al. [2008] that use implementation tricks to offer similar type of blobs but for a fraction of the computational cost. SURF features will be introduced in Section 3.7.3.4.

There is no distinctive frontier between the usage of corner points and interest regions. In many cases, it depends on the particular image structure to know which detector will deliver the best results. In general, corners are used for calibration (preciseness) and tracking (fast and stable across short baseline) and blobs for large scale problems (repeatable and stable for wide baseline). Detection performance (stability and repeatability) will increase significantly for both types of detectors if they are invariant to changes in the image.

3.7.3 Examplary Detectors

In this section, we describe some standard methods to detect features in an image.

3.7.3.1 Harris Corners

The original "Harris" corner detector [Harris and Stephens, 1988] is based on the second moment matrix, also referred to as the auto-correlation, computed from pixel intensity values. The local auto-correlation function measures local changes of the image $\mathcal{I}$ by calculating the correlation between a patch centered on the current point $\mathbf{m}$ and patches from the neighborhood of m. This is correlation is weighted by a Gaussian weight w, this leads to:

$$C\left(\mathbf{m}\right) = \sum_{\boldsymbol{\xi} \in \mathcal{N}_\mathbf{0}} w(\boldsymbol{\xi}) \cdot \left(\mathcal{I}\left(\mathbf{m}\right) - \mathcal{I}\left(\mathbf{m} + \boldsymbol{\xi}\right)\right)^2 \; . \tag{3.41}$$

The shifted image can be approximated by a first order Taylor expansion:

$$\mathcal{I}(\mathbf{m} + \boldsymbol{\xi}) \approx \mathcal{I}(\mathbf{m}) + \mathcal{I}'(\mathbf{m})\boldsymbol{\xi} \; , \tag{3.42}$$

which, when introduced in Equation 3.41 is resulting in

$$\begin{aligned} C(\mathbf{m}) &= \sum_{\boldsymbol{\xi} \in \mathcal{N}_\mathbf{0}} w(\boldsymbol{\xi}) \cdot \left(\mathcal{I}'\left(\mathbf{m}\right)\boldsymbol{\xi}\right)^2 \\ &= \boldsymbol{\xi}^\top \underbrace{\left(\sum_{\boldsymbol{\xi} \in \mathcal{N}_\mathbf{0}} w\left(\boldsymbol{\xi}\right) \begin{bmatrix} \mathcal{I}'_x(\mathbf{m})^2 & \mathcal{I}'_x(\mathbf{m})\mathcal{I}'_y(\mathbf{m}) \\ \mathcal{I}'_x(\mathbf{m})\mathcal{I}'_y(\mathbf{m}) & \mathcal{I}'_y(\mathbf{m})^2 \end{bmatrix} \right)}_{\mathsf{A}(\mathbf{m})} \boldsymbol{\xi} \; , \end{aligned} \tag{3.43}$$

with $\mathcal{I}'_x = \dfrac{\partial \mathcal{I}}{\partial x}$ and $\mathcal{I}'_y = \dfrac{\partial \mathcal{I}}{\partial y}$.

The matrix A captures the intensity structure around the point $\mathbf{m}$. It is often referred to as the structure tensor. It describes the gradient distribution in the local neighborhood of $\mathbf{m}$. A corner is characterized by a large variation of $C(\mathbf{m})$ in all directions. This characterization can be expressed by the eigenvalues λ_1, λ_2 of $\mathsf{A}(\mathbf{m})$; if both eigenvalues are large a corner is found, if just one eigenvalue is large, the point lies on an edge. As the exact computation of eigenvalues is expensive, Harris and Stephens define their detector function $f_{\mathbf{dec}}$ as a function of the trace and the determinant of A:

$$f_{\mathbf{dec}}(\mathbf{m}) = \lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2 = \det(\mathsf{A}(\mathbf{m})) - \kappa \cdot \operatorname{trace}(\mathsf{A}(\mathbf{m}))^2, \tag{3.44}$$

where κ is chosen to lie within 0.04 and 0.15.

Another similar detector function widely used [Kovesi] is:

$$f_{\mathbf{dec}}(\mathbf{m}) = \frac{\det(\mathsf{A}(\mathbf{m}))}{\operatorname{trace}(\mathsf{A}(\mathbf{m})) + \epsilon}, \tag{3.45}$$

as it is does not require to set a κ parameter.

Finally a non maximum suppression is used to keep only local maxima. Though the "Harris" corners have no scale invariance, it is possible to extend it to a scale invariant feature detector as demonstrated in the next section.

3.7.3.2 Scale Adapted Harris and Harris-Laplace Detector

Mikolajczyk and Schmid [2004] proposed a new interest point detector that combines the Harris detector with automatic scale selection according as explained in Section 3.7.1.1. To obtain scale invariance, they adapt the structure tensor A to scale changes. The detector is then independent of the image resolution and the scale-adapted structure tensor is defined by:

$$\mathsf{A}(\mathbf{m}, \sigma_{\mathcal{I}}, \sigma_{\mathcal{D}}) = \sum_{\boldsymbol{\xi} \in \mathcal{N}_0} w(\boldsymbol{\xi}, \sigma_{\mathcal{I}}) \begin{bmatrix} \mathcal{I}_x(\mathbf{m}, \sigma_{\mathcal{D}})^2 & \mathcal{I}_x(\mathbf{m}, \sigma_{\mathcal{D}})\mathcal{I}_y(\mathbf{m}, \sigma_{\mathcal{D}}) \\ \mathcal{I}_x(\mathbf{m}, \sigma_{\mathcal{D}})\mathcal{I}_y(\mathbf{m}, \sigma_{\mathcal{D}}) & \mathcal{I}_y(\mathbf{m}, \sigma_{\mathcal{D}})^2 \end{bmatrix} \tag{3.46}$$

First, local derivatives $\mathcal{I}_x, \mathcal{I}_y$ are computed using Gaussian kernels of a size determined by the differentiation scale $\sigma_{\mathcal{D}}$. Derivatives are then averaged in the neighborhood of the point by smoothing with a Gaussian weight $w(\boldsymbol{\xi}, \sigma_{\mathcal{I}})$. The detector function stays the same as in Equation 3.44. Scale-adapted Harris detector function localizes points stably, unfortunately it rarely attains maxima over scales [Mikolajczyk and Schmid, 2004].

This problem can be resolved by using the Laplacian-of-Gaussians (LoG) operator for scale selection. LoG operator selects correct characteristic scales in a scale-space representation:

$$f_{\mathbf{sel}}(\mathbf{m}, \sigma_i) := |\mathrm{LoG}(\mathbf{m}, \sigma_n)| = \sigma_n^2 \left|\mathcal{I}_{xx}(\mathbf{m}, \sigma_n) + \mathcal{I}_{yy}(\mathbf{m}, \sigma_n)\right|, \tag{3.47}$$

with $\mathcal{I}_{xx} = \dfrac{\partial^2 \mathcal{I}}{\partial x^2}$ and $\mathcal{I}_{yy} = \dfrac{\partial^2 \mathcal{I}}{\partial y^2}$. When the size of the LoG kernel matches with the size of local structures (e.g. corners), the response is maximal.

The *Harris-Laplace* detector combines the scale adapted Harris detector with the scale selection describe in Equation (3.47).

3.7.3.3 SIFT - Scale Invariant Feature Transform

In [Lowe, 2004], the author does not only describe a method to detect points, but also includes a novel descriptor design and an efficient matching step.

For detection, SIFT uses difference of Gaussians (DoG) filter for both location detection and scale selection. This approach reduces complexity significantly in comparison to the Laplace operator, since the detector response can be computed from the difference of two layers of a Gaussian image pyramid. This approach allows creating the detection stack D from the difference of neighboring layers of a Gaussian pyramid:

$$\begin{aligned} \underbrace{f_{\mathbf{dec}}(\mathbf{m}, \sigma_i)}_{:= f_{\mathbf{sel}}(\mathbf{m}, \sigma_i)} &= \underbrace{(G(\mathbf{m}, \sigma_{i+1}) - G(\mathbf{m}, \sigma_i))}_{\approx \nabla^2 G(\mathbf{m}, \sigma_i)} * I(\mathbf{m}) \\ &= G(\mathbf{m}, \sigma_{i+1}) * I(\mathbf{m}) - G(\mathbf{m}, \sigma_i) * I(\mathbf{m}) \,, \end{aligned} \tag{3.48}$$

with G a Gaussian kernel centered in $\mathbf{m}$ of standard deviation σ_i.

The strength of smoothing is controlled via σ_i, where $\sigma_0 = 1.6$ is defined for the original image and thus also valid for the very bottom pyramid layer. To achieve lower memory usage the image stack introduced before is now represented by an image pyramid grouped in octaves. An octave is a set of blurred images divided in intervals that starts with a kernel of σ_i for the first interval and finishes with $2\sigma_i$ for the last one. Between subsequent octaves down-sampling by a factor of 2 is performed, which retains the same information as smoothing the image with doubled standard deviation and then just considering every second pixel. While an octave contains images of equal resolution, it is divided into intervals created by increasing detector size or increasing blur, respectively. Intervals are uniformly positioned in scale space separated from each other by a constant scaling factor $k = 2^{1/N_{intervals}}$, where $N_{intervals}$ is the defined number of intervals per octave. Thus allowing to compute the relation from the detected feature scale to the original image scale:

$$\sigma_i = \sigma_0 \cdot 2^{octave + \dfrac{interval}{N_{intervals}}} \,. \tag{3.49}$$

Interest points $\langle \mathbf{m}, \sigma \rangle$ are located spatially and in scale via non-maximum suppression for each pyramid location $(\mathbf{x}, \sigma_i)$ according to location detector $f_{\mathbf{dec}}$ and scale selector $f_{\mathbf{sel}}$. A maximum is found by the investigation of a $3\times3\times3$ neighborhood when

$$\forall (\boldsymbol{\xi}, \nu) \in \mathcal{N}_{3\times3\times3}(\mathbf{x}, \sigma_i) \quad \text{s.t. } f_{\mathbf{dec}}(\mathbf{x}, \sigma_i) > f_{\mathbf{dec}}(\boldsymbol{\xi}, \nu) \,. \tag{3.50}$$

Then the location $(\mathbf{x}, \sigma_i)$ is defined to be an interest point $\langle \mathbf{m}, \sigma \rangle$.

Pyramid layers are only present at specific sampled scales σ_i. For a more accurate interest point localization than the one obtained from the sampled scales, detected feature points are interpolated in scale space leading to a second order estimate [Brown and Lowe, 2002]:

$$\widehat{\mathbf{u}} = \begin{pmatrix} \hat{\mathbf{m}} \\ \hat{\sigma} \end{pmatrix} = \arg\max_{\mathbf{u}} \left(f_{\mathbf{dec}}(\mathbf{u}_0) + \frac{\partial f_{\mathbf{dec}}}{\partial \mathbf{u}} (\mathbf{u} - \mathbf{u}_0) + \frac{1}{2} (\mathbf{u} - \mathbf{u}_0) \frac{\partial^2 f_{\mathbf{dec}}}{\partial \mathbf{u}^2} (\mathbf{u} - \mathbf{u}_0) \right) \tag{3.51}$$

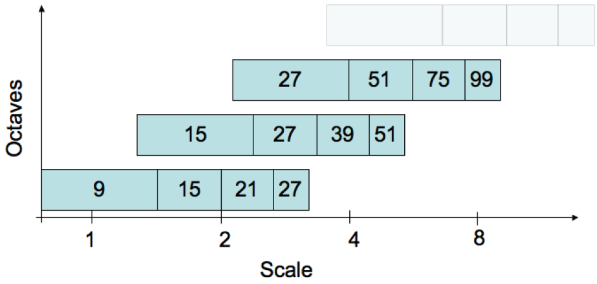


Figure 3.8: **SURF Box filter sizes**: the different rows define the filter sizes used in the SURF algorithm for each octaves. *Courtesy of [Bay et al., 2008].*

The DoG operator performs well for scale selection, yet detects less meaningful points or regions (e.g. on edges). This issue needs to be handled in a post processing step. SIFT offers repeatable features that can be matched across wide baseline. It has two main drawbacks: its processing time and its localization precision. Both are due to the scale space representation. For more information about SIFT, the reader is referred to Lowe [2004].

3.7.3.4 SURF - Speeded-Up Robust Features

Speeded up robust features [Bay et al., 2008] build on the strengths of SIFT. The SURF algorithm especially focuses on lowering computational complexity resulting in a much faster algorithm. SURF also approximates or even outperforms other detectors in terms of repeatability, distinctiveness, and robustness.

Compared to SIFT, SURF relies on the usage of integral images, which accounts for most of the reduction in computation time. It employs the determinant of the Hessian matrix adapted for scale-invariance as spatial feature detection *and* scale selection operator. The entries of the Hessian are calculated by convolving the appropriate Gaussian second order derivatives with the image at the analyzed position. SURF approximates derivatives with box filters of different sizes according to the current scale. The Hessian can then be evaluated at constant, low computational cost using integral images for arbitrary filter size:

$$\underbrace{f_{\mathbf{dec}}(\mathbf{m},\sigma_i)}_{:=f_{\mathbf{sel}}(\mathbf{m},\sigma_i)} = \det \begin{bmatrix} L_{xx}(\mathbf{m},\sigma_i) & L_{xy}(\mathbf{m},\sigma_i) \\ L_{xy}(\mathbf{m},\sigma_i) & L_{yy}(\mathbf{m},\sigma_i) \end{bmatrix}, \tag{3.52}$$

where L_{xx}, L_{xy}, L_{yy} are the responses of the image convolved with the according box filter.

The scale space is analyzed by up-scaling the filter size rather than iteratively reducing the image size. The smallest box filter has size 9×9 and the output is considered as the initial scale layer with scale $\sigma_0 = 1.2$. Following layers are obtained by filtering the image with gradually bigger masks. Sampled scales thus directly relate to the filter size s via

$$\sigma_i = \sigma_0 \cdot \frac{s}{9}\,. \tag{3.53}$$

The scale space is grouped into octaves as well. The assignment of different filter sizes to each octaves is illustrated in Figure 3.8. An octave includes a series of filter responses of equal size. In total an octave encompasses a scaling factor of 2, thus filter responses in the following octave are subsampled and are half the size.

In order to localize interest points $\langle \mathbf{m}, \sigma \rangle$ in the image and across scales, a non-maximum suppression in a $3 \times 3 \times 3$ neighborhood followed by an interpolation step is applied in the same fashion as for SIFT.

3.8 Single View Geometry

In this section, we introduce standard methods to estimate the rigid body motion T_i of a camera. It is based on constraints between 3D points $\mathcal{M}$ and their 2D observations $\widetilde{\mathbf{m}}_i$. Here we suppose that available 3D coordinates are noise free. Unfortunately, such an assumption for 2D projections does not hold. The methods to obtain the 2D coordinates can be imprecise. The image noise might lead to a numerical inaccuracy in localization process. As in [Appel, 2005], we suppose that the 2D observation $\widetilde{\mathbf{m}}$ is a stochastic process represented by a bi-variate Gaussian distribution:

$$\widetilde{\mathbf{m}} \sim \mathcal{N}\left(\overline{\mathbf{m}}, \Sigma_{\widetilde{\mathbf{m}}}\right), \text{ with } \Sigma_{\widetilde{\mathbf{m}}} = \mathsf{U}^\top \begin{bmatrix} \sigma_{\tilde{u}}^2 & 0 & 0 \\ 0 & \sigma_{\tilde{v}}^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathsf{U} \tag{3.54}$$

the associated covariance matrix and $\overline{\mathbf{m}}$ the point corresponding to the noise free projection of $\mathcal{M}$:

$$\overline{\mathbf{m}} \sim \mathsf{K}\mathsf{w}(\overline{\mathsf{T}}\boldsymbol{\mathcal{M}}), \tag{3.55}$$

Therefore, the estimate $\widehat{\mathsf{T}}$ based on $\boldsymbol{\mathcal{M}}$ and $\widetilde{\mathbf{m}}$ can only be an approximation of the correct transformation $\overline{\mathsf{T}}$. For the moment, we suppose that all measurements have the same order of imprecision in both directions:

$$\forall \widetilde{\mathbf{m}}, \quad \Sigma_{\widetilde{\mathbf{m}}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.56}$$

with n the number of measurements. We will relax this constraint in chapter 7, but within the remaining of this chapter we suppose this assumption true.

Many methods exist to estimate $\widehat{\mathsf{T}}$ [Lepetit and Fua, 2005, Lepetit et al., 2009]. The most basic method is based on the creation of a linear system. This method is referred to as the Direct Linear Transform (DLT) [Hartley and Zisserman, 2003]. These type of methods are often followed by a non-linear minimization of the cost function defined in Equation 3.55.

3.8.1 Non-linear Registration Cost Function

In this section, we describe the cost function based on the re-projection error that can be used for registration. We can express the alignment error of a given transformation

T with respect to a set of 2D-3D correspondences $\left\{\left\langle\widetilde{\mathbf{m}}^k, \boldsymbol{\mathcal{M}}^k\right\rangle\right\}$ by the following cost, related to equation 3.55:

$$\mathcal{G}_{\left\{\left\langle\widetilde{\mathbf{m}}^k, \boldsymbol{\mathcal{M}}^k\right\rangle\right\}}\left(\mathsf{T}\right) = \frac{1}{2}\sum_{k=1}^{n}\left\|\mathsf{Kw}\left(\mathsf{T}\boldsymbol{\mathcal{M}}^k\right) - \widetilde{\mathbf{m}}^k\right\|^2 . \tag{3.57}$$

This function is called registration error as it measures a distance between projected 3D points and observed 2D points. The units, in which Equation 3.57 is expressed, are pixel-squares. This cost is a sum of squares and can be expressed as Equation 3.8 by the norm of a $2n$-vector:

$$\mathcal{G}_{\left\{\left\langle\widetilde{\mathbf{m}}^k, \boldsymbol{\mathcal{M}}^k\right\rangle\right\}}\left(\mathsf{T}\right) = \frac{1}{2}\mathbf{y}_{\mathcal{G}_{\left\{\left\langle\widetilde{\mathbf{m}}^k, \boldsymbol{\mathcal{M}}^k\right\rangle\right\}}}\left(\mathsf{T}\right)^\top \mathbf{y}_{\mathcal{G}_{\left\{\left\langle\widetilde{\mathbf{m}}^k, \boldsymbol{\mathcal{M}}^k\right\rangle\right\}}}\left(\mathsf{T}\right) , \tag{3.58}$$

with $\mathbf{y}_{\mathcal{G}_{\left\{\left\langle\widetilde{\mathbf{m}}^k, \boldsymbol{\mathcal{M}}^k\right\rangle\right\}}}\left(\mathsf{T}\right) = \left[u^1 - \tilde{u}^1,\, v^1 - \tilde{v}^1,\, \cdots\, u^n - \tilde{u}^n,\, v^n - \tilde{v}^n\right]^\top$, $\mathsf{Kw}\left(\mathsf{T}\boldsymbol{\mathcal{M}}^k\right) = \mathbf{m}^k \sim \left[u^k\, v^k\, 1\right]^\top$ and $\widetilde{\mathbf{m}}^k \sim \left[\tilde{u}^k\, \tilde{v}^k\, 1\right]^\top$.
The registration error problem can then be expressed as a minimization problem as follow:

$$\widehat{\mathsf{T}} = \arg\min_{\mathsf{T}} \mathcal{G}_{\left\{\left\langle\widetilde{\mathbf{m}}^k, \boldsymbol{\mathcal{M}}^k\right\rangle\right\}}\left(\mathsf{T}\right) . \tag{3.59}$$

This minimum can estimation using a local optimizer as presented in section 3.3 with a starting point provided by a linear least-square solution such as the one provided by the DLT algorithm. We parametrize the estimated rigid transformation update using the Lie algebra $\Delta\mathsf{T} = \mathsf{T}\left(\Delta\mathbf{x}\right)$ (*c.f.* section 3.4) and the update used for the optimizer is compositional:

$$\widehat{\mathsf{T}} \leftarrow \mathsf{T}\left(\Delta\mathbf{x}\right)\widehat{\mathsf{T}} , \tag{3.60}$$

which is valid because $\mathsf{T}\left(\Delta\mathbf{x}\right) = \exp\left(\mathsf{A}\left(\Delta\mathbf{x}\right)\right)$ is a parametrization of $\mathbb{SE}\left(3\right)$ since $\Delta\mathbf{x}$ is small.

A camera is said *registered* when we know the absolute transformation T between coordinate systems that defines the 3D points and the camera coordinate systems. This camera pose is call a *full* pose.

3.9 Two and More Views Geometry

In this section, we briefly discuss epipolar geometry, reconstruction and then bundle adjustment. All these notions are based on 2D point correspondences. We say that a set of image points $\{\mathbf{m}_s\}$ correspond or are in correspondences when they are projections of the same 3D point $\mathcal{M}$.

3.9.1 Epipolar Geometry

Between two un-calibrated view i and j, with parallax, it exists a fundamental matrix F_{ij} [Faugeras, 1992, Hartley, 1992] that governs the underlying projective geometry from the source j to the target i:

$$\mathbf{m}_i^\top \mathsf{F}_{ij}\mathbf{m}_j = 0. \tag{3.61}$$

This epipolar constraint is weak as F_{ij} transforms points $\mathbf{m}_j$ into a line $\mathbf{l}_i = \mathsf{F}_{ij}\mathbf{m}_j$ in image i. For its estimation, there are different linear methods. The most notorious one certainly is the 8-points algorithm [Longuet-Higgins, 1981], which leads to a unique solution for F_{ij}. Others algorithms, which use less correspondences and therefore lead to multiple solutions exist, for example based on 7 points leading to 3 distinct solutions [Zhang, 1998].

These linear methods do not always provide the required precision because they only try to minimize an algebraic distance, based on equation 3.61. Therefore it is often followed by a non-linear minimization based on the point to line Euclidean distance between $\mathbf{m}_i$ and $\mathbf{l}_i = \mathsf{F}_{ij}\mathbf{m}_j$:

$$d_{\mathcal{L}}(\mathbf{m}, \mathbf{l}) = \frac{|\mathbf{m}^\top \mathbf{l}|}{|w|\ \|\underline{\mathbf{l}}\|}\ , \tag{3.62}$$

with $\mathbf{m} = [u\, v\, w]^\top$ and $\mathbf{l} = \left[\underline{\mathbf{l}}^\top\ l\right]^\top$.

When calibrated, we can factorize F_{ij} in the essential matrix E_{ij}, which incorporates the motion between the two cameras [Huang and Faugeras, 1989, Horn, 1990] as follows:

$$\mathsf{E}_{ij} = \mathsf{K}_i^\top \mathsf{F}_{ij} \mathsf{K}_j\ , \tag{3.63}$$

with K_i (respectively K_j) the intrinsic parameter matrix that captured image i (resp. j).

As shown in [Huang and Faugeras, 1989], E_{ij} is an essential matrix if and only if it is rank 2 and 2 of its singular values are equals. It can be decomposed as follows:

$$\mathsf{E}_{ij} = [\mathbf{t}_{ij}]_\times \mathsf{R}_{ij}\ , \tag{3.64}$$

where $\mathbf{t}_{ij}$ counts for the translation direction between the cameras and R_{ij} for the rotation.

Each matrix E leads to four possible decompositions, as shown in Figure 3.9. This ambiguity is solved using the points correspondences, because only the physically correct set of rotation and translation triangulates the image points in front of the cameras. When a correct decomposition is available, the pose is said to be oriented. It is interesting to see that $\mathbf{t}$ can only be determined up to unknown scale, as Equation 3.64 constrains only 5 degrees of freedom.

There exists two main decomposition methods: one based on the Singular Value Decomposition (SVD) [Hartley and Zisserman, 2003] and one using linear methods [Horn, 1990]. They both lead to four solutions. This ambiguity can be solved using chirality constraints [Hartley, 1998]. This constraints are based on the idea that image points can be produced only by 3D points in-front of the images. Therefore the rays between the camera centers and the respective image points should intersect in-front of the cameras.

A pose between cameras estimated from the essential matrix is called a *relative* pose.

3.9.2 Triangulation

In this section, we discuss available methods to triangulate a 3D point observed from different registered cameras. When image points are in correspondence in a set of registered cameras, their 3D coordinates can be estimated. This process is called *triangulation*. As for fundamental matrices and poses, it can be estimated via a linear system [Slabaugh et al., 2001, Hartley and Zisserman, 2003]. The linear estimation may lack precision and

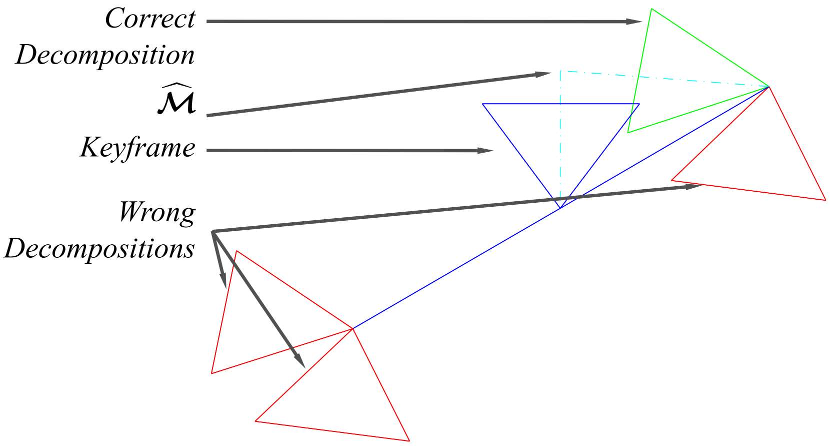


Figure 3.9: **Essential Matrix Decomposition** leads to 4 different cameras but only one can triangulate a 3D point $\widehat{\boldsymbol{\mathcal{M}}}$ in front of both cameras of the rig. In blue is displayed the canonical camera, in green the correct decomposition and in red the incorrect ones.

a meaningful geometric interpretation as it tries to minimize a distance between 3D rays, which is not directly related to a distance in the image.

The triangulation error can be expressed using a re-projection error:

$$\mathcal{G}_{\{\mathsf{T}_i\}}\left(\boldsymbol{\mathcal{M}}\right) = \frac{1}{2}\sum_{i=1}^{m} \left\|\mathsf{K}_i \mathbf{w}\left(\mathsf{T}_i \boldsymbol{\mathcal{M}}\right) - \widetilde{\mathbf{m}}_i\right\|^2 \ , \tag{3.65}$$

with m the number of available observations.

The triangulation problem can be expressed as a minimization of Equation 3.65 as follow:

$$\widehat{\boldsymbol{\mathcal{M}}} = \arg\min_{\boldsymbol{\mathcal{M}}} \mathcal{G}_{\{\mathsf{T}_i\}}\left(\boldsymbol{\mathcal{M}}\right) \ . \tag{3.66}$$

This minimization can be solved optimally: for two-views in closed-form by finding roots of a polynomial function of degree 6 [Hartley and Sturm, 1997, Hartley and Zisserman, 2003] or iteratively [Kanatani et al., 2008]; and for three-views by finding eigenvectors of 47 × 47 matrix [Stewenius et al., 2005] or using Gröbner basis [Byrod et al., 2007].

There is to date no solver to globally minimize the re-projection error based on the L_2-norm for the general n-views case. Therefore it requires a two-step approach. First, 3D coordinates are estimated by solving a linear least squares problem. Then this estimate is refined by a least-square iterative local solver. [Stewenius et al., 2005] observed that such a process often finds the global minimum.

In order to express this minimization problem as in Section 3.3, we rewrite the cost function defined in Equation 3.65 as a sum of $2m$ squares:

$$\mathcal{G}_{\{\mathsf{T}_i\}}\left(\widehat{\boldsymbol{\mathcal{M}}}\right) = \frac{1}{2}\mathbf{y}_{\mathcal{G}_{\{\mathsf{T}_i\}}}\left(\boldsymbol{\mathcal{M}}\right)^\top \mathbf{y}_{\mathcal{G}_{\{\mathsf{T}_i\}}}\left(\boldsymbol{\mathcal{M}}\right) \ , \tag{3.67}$$

with $\mathbf{y}_{\mathcal{G}_{\{\mathsf{T}_i\}}}(\boldsymbol{\mathcal{M}}) = [u_1 - \tilde{u}_1,\, v_1 - \tilde{v}_1,\, \cdots\, u_m - \tilde{u}_m,\, v_m - \tilde{v}_m]^\top$, $\mathsf{K}\mathbf{w}\left(\mathsf{T}_i\boldsymbol{\mathcal{M}}\right) \sim [u_i\, v_i\, 1]^\top$ and $\widetilde{\mathbf{m}}_i \sim [\tilde{u}_i\, \tilde{v}_i\, 1]^\top$.

Therefore, the minimization problem defined in Equation 3.66 using an iterative least-square optimizer. We parametrize the 3D point as $\widehat{\mathbf{M}} = [x\, y\, z]^\top$ and the update is additive:

$$\widehat{\mathbf{M}} \leftarrow \widehat{\mathbf{M}} + \Delta\mathbf{M}\,, \tag{3.68}$$

which is the natural formulation used in standard implementation [Lourakis and Argyros, 2009].

3.9.3 Bundle Adjustment

In this section, we describe the notion of bundle adjustment. Bundle Adjustment is a non-linear process to solve the structure from motion problem [Triggs et al., 1999]. Given a set of cameras and set of correspondences, bundle adjustment simultaneously optimizes the cameras' geometry (motion) and scene geometry (structure). It can estimates both intrinsic and extrinsic parameter of the cameras. The quality of an estimation of the structure and motion of a scene can be evaluated using a re-projection error, defined as follow:

$$\mathcal{G}\left(\{\mathsf{T}_i\}, \{\boldsymbol{\mathcal{M}}^k\}\right) = \frac{1}{2}\sum_{i=1}^{m}\sum_{k=1}^{n}\delta_i^k \left\|\mathsf{K}_i\mathbf{w}\left(\mathsf{T}_i\boldsymbol{\mathcal{M}}^k\right) - \widetilde{\mathbf{m}}_i^k\right\|^2\,, \tag{3.69}$$

with m the number of cameras, n the number of 3D points and δ_i^k a visibility Dirac function defined as follow:

$$\delta_i^k = \begin{cases} 1 & \text{if} \quad \widetilde{\mathbf{m}}_i^k \text{ exists} \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.70}$$

δ_i^k literally translates when it is equal to 1 "the 3D point $\boldsymbol{\mathcal{M}}^k$ is observed in image i".

This cost function can be expressed as a sum of squares:

$$\mathcal{G}\left(\{\mathsf{T}_i\}, \{\boldsymbol{\mathcal{M}}^k\}\right) = \frac{1}{2}\mathbf{y}_{\mathcal{G}}\left(\{\mathsf{T}_i\}, \{\boldsymbol{\mathcal{M}}^k\}\right)^\top \mathbf{y}_{\mathcal{G}}\left(\{\mathsf{T}_i\}, \{\boldsymbol{\mathcal{M}}^k\}\right)\,, \tag{3.71}$$

with

$$\mathbf{y}_{\mathcal{G}}\left(\{\mathsf{T}_i\}, \{\boldsymbol{\mathcal{M}}^k\}\right) = \begin{bmatrix} \left[\ \delta_1^1\,[u_1^1 - \tilde{u}_1^1\ \ v_1^1 - \tilde{v}_1^1]\ \ \delta_2^1 \cdots\ \ \delta_m^1\,[u_m^1 - \tilde{u}_m^1\ \ v_m^1 - \tilde{v}_m^1]\ \right]^\top \\ \left[\ \delta_1^2\,[u_1^2 - \tilde{u}_1^2\ \ v_1^2 - \tilde{v}_1^2]\ \ \delta_2^2 \cdots\ \ \delta_m^2\,[u_m^2 - \tilde{u}_m^2\ \ v_m^2 - \tilde{v}_m^2]\ \right]^\top \\ \vdots \\ \left[\ \delta_1^n\,[u_1^n - \tilde{u}_1^n\ \ v_1^n - \tilde{v}_1^n]\ \ \delta_2^n \cdots\ \ \delta_m^n\,[u_m^n - \tilde{u}_m^n\ \ v_m^n - \tilde{v}_m^n]\ \right]^\top \end{bmatrix}. \tag{3.72}$$

The bundle adjustment can be expressed as a least-square minimization problem:

$$\left\{\widehat{\mathsf{T}}_i, \left\{\widehat{\boldsymbol{\mathcal{M}}}^k\right\}\right\} \arg \min_{\{\mathsf{T}_i, \{\boldsymbol{\mathcal{M}}^k\}\}} \mathcal{G}\left(\{\mathsf{T}_i\}, \{\boldsymbol{\mathcal{M}}^k\}\right)\,. \tag{3.73}$$

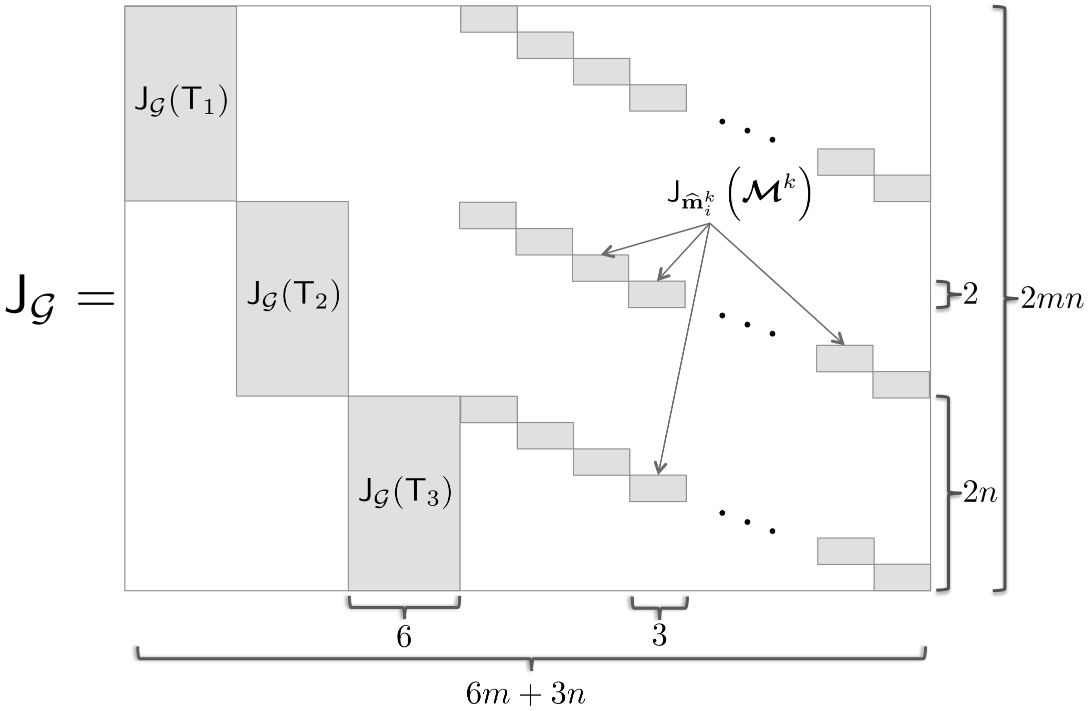

$\mathsf{J}_{\mathcal{G}} =$
$\mathsf{J}_{\mathcal{G}}(\mathsf{T}_1)$
$\mathsf{J}_{\mathcal{G}}(\mathsf{T}_2)$
$\mathsf{J}_{\mathcal{G}}(\mathsf{T}_3)$
$\mathsf{J}_{\widehat{\mathbf{m}}_i^k}\left(\boldsymbol{\mathcal{M}}^k\right)$
2
$2mn$
$2n$
6
3
$6m + 3n$

Figure 3.10: **Jacobian Matrix for Bundle Adjustment**: On the left hand side of the matrix are derivatives with respect to motion parameters and on the right hand side with respect to structure parameters. This matrix is extremely sparse because most of the parameters are independent.

It can be minimized using a local least-square optimizer. We use the same parametrization and update function as introduced in Section 3.8.1 for the rigid transformation and in Section 3.9.2 for the structure. This leads to the following updates:

$$\left\{ \begin{array}{lll} \forall i \;\; \widehat{\mathsf{T}}_i & \leftarrow & \Delta\mathsf{T}_i\,\widehat{\mathsf{T}}_i \\ \forall k \;\; \widehat{\mathbf{M}}^k & \leftarrow & \widehat{\mathbf{M}}^k + \Delta\mathbf{M}^k \end{array} \right. \tag{3.74}$$

The number of parameters to estimate is $6m + 3n$: $6m$ for m cameras and $3n$ for n 3D points. This leads to a large jacobian $\mathsf{J}_{\mathcal{G}}$ matrix $2nm \times (6m + 3n)$. Fortunately, $\mathsf{J}_{\mathcal{G}}$ is sparse, as illustrated in Figure 3.10. Therefore the full matrix does not have to be stored in memory but only non-zero values. The maximum required memory to store $\mathsf{J}_{\mathcal{G}}$ is $18mn$ depending on the points visibility (δ_i^k). Details for implementing a sparse bundle adjustment method can be found in [Lourakis and Argyros, 2009].

Part II

Application

In this part, we describe the application framework. In the next chapter 4, we present the proposed workflow for discrepancy check using augmented reality visualization. Then in chapter 5, we focus on the navigation methods developed to deal with such software that includes CAD data and aligned images. Finally in chapter 6, we focus on the registration approach developed for this application based on industrial components present in many civil applications: *anchor-plates*.

CHAPTER

FOUR

AR BASED DISCREPANCY CHECK - A NEW WORKFLOW

We integrate our approach for *Discrepancy Check* into the day to day inspection and documentation process of plant erection. This task is mandatory to know the status of the construction and to estimate present and future clashes[1]. If a clash is detected at an early stage, its impact can be mitigated in terms of construction delay and cost. This is done by a visit of the building site and it is performed by civil engineers that are responsible for the correctness of an erection. Usually, each engineer is limited to study one type of system for example pipes. He verifies that wall pass-throughs are located at the right place, that the supporting structure is conformed and finally that the pipes are placed at the correct position. In general, the inspection leads to quality reports and if necessary redlines on technical drawings [Clayton et al., 1998]. This documentation often includes pictures: this help the communication with the different departments of the company. We extended the current process by offering means to register images with the virtual mock-up to improve visualization, discrepancy detection, documentation and reporting.

A single user usually manages the inspection, though different users can inspect different systems at the same time. The inspection is a two-step process which is separated between on-site pictures acquisitions and the interaction with VID to obtain augmentation and to perform the checks. This is described in Section 4.1. The related processes supported by the proposed system are described in Section 4.2.

4.1 Use Cases

The engineer, also denominated here as the user, first decides what component or set of components needs to be inspected. Using his favorite CAD viewer or VID, which is an enhanced CAD viewer, he determines within the complex where are located the items to inspect. Then the user goes on-site in the designated area with a calibrated[2] digital camera. These inspections can occur at different stages of the plant life cycle :

[1] Clashes: when a virtual model colides with a built item, which makes it imposible to install the components represented by the virtual model.

[2] All the camera, we use, are calibrated offline. VID includes a GUI to calibrate cameras.

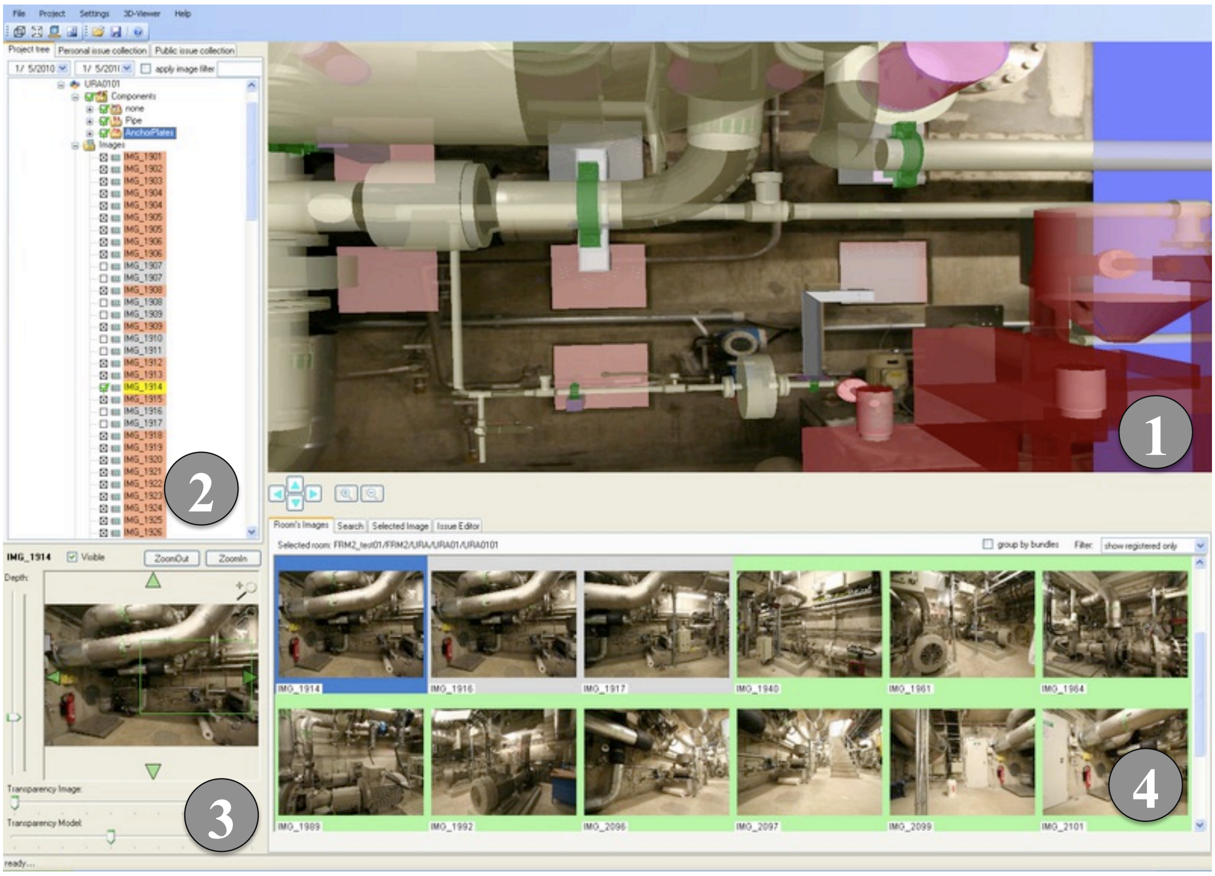


Figure 4.1: **VID Main Graphical User Interface**: (1) the 3D renderer that displays models and image, (2) the custom tree-view that gives access to the hierarchy of the project (3) the zoom and pan interface that command the virtual camera motion in a mixed view (4) the thumbnail browser that can display set of images based on a user-specified criteria.

construction, commissioning, maintenance, or decommissioning. The user acquires images of a room or system under study. The images should cover the scene from different angles, so that when merged with the virtual model they capture the scenes from different viewpoints. Additionally, for registration purposes anchor-plates should be visible in some of the shots. Back at the office, the user transfers the images on his workstation. In case the user is located in a remote site, the image acquisition can be performed by someone else trained to handle the camera. The images are then transferred by email or by more secure means. Once the images are available on the workstation, the user can attach them to the virtual model using VID. Figure 4.2 summarizes the acquisition scenario.

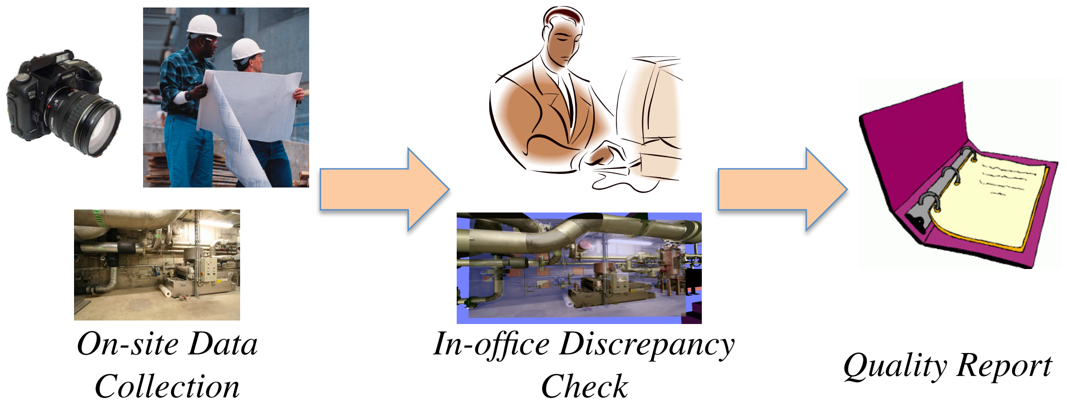


Figure 4.2: **Use case Diagram**: The user acquires images from the item under study, that are registered to the model using VID back at the office to obtain an augmentation that can be used for discrepancy check, that can be used to compile a construction quality report.

4.2 Basic Visual Inspection and Documentation (VID) Workflow

The software VID is a client that connects to a database where all the necessary information is stored; its internal organization is discussed in appendix B.2. When starting VID, one connects to a database server and selects the project one wants to work on. A project corresponds to one instance of a plant design. Each plant is unique but often corresponds to an instance of the same original design.

When the project is loaded one can access the hierarchy of the project using a tree-view sorted from buildings to rooms. Implementation details of this GUI are given in Appendix B.4. Most of the action is available at the room level which has three children branch: CAD components, Images and Issues. VID supports engineers in three different processes that are discussed in the following sections: first and foremost CAD visualization (c.f. Section 4.2.1), then guided image registration (c.f. Section 4.2.2) that allows the user to align an image with CAD model, and finally inspection of discrepancy using mixed views (c.f. Section 4.2.3).

Other functionalities are available in VID to support these processes. For example, newly acquired images, that picture the structure, can be attached to database. Guided through a wizard, the user selects the camera used to capture these images. The images are automatically un-distorted[3] and stored in the database. All this extra functionality are briefly discussed in Appendix B.

[3] Compensation for camera lens radial and tangential distortion that deforms straight lines into curves.

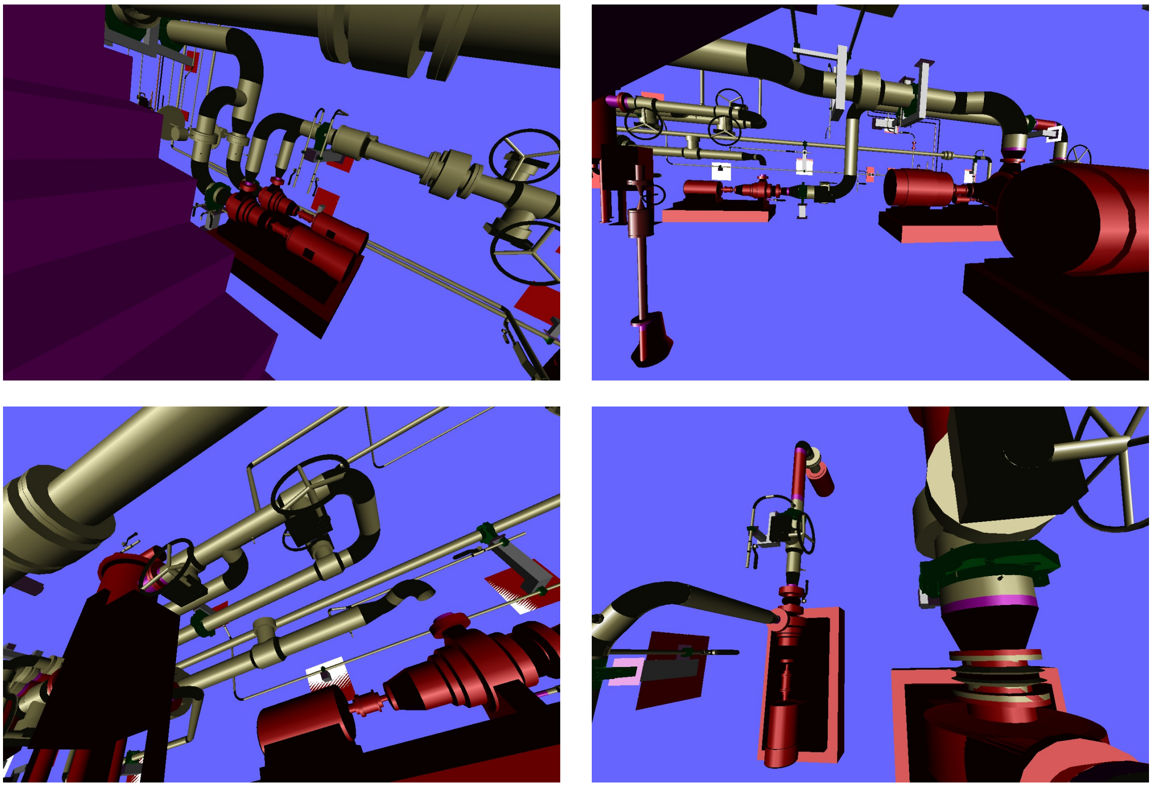


Figure 4.3: **VID a CAD viewer**: VID's primary task is to inspect CAD model. It can display in real-time complex 3D models.

4.2.1 Augmented CAD Viewing

CAD viewing and inspection is one of VID first aim. Therefore the renderer is at the center of the graphical user interface. CAD components can be selected in the tree-view, to be displayed in the 3D renderer. To improve usability, component can be removed from the rendering queue by double clicking directly in the 3D viewer. It avoids the need to navigate through hundreds of name in a tree-view to disable one component. Interaction can be mouse-based or button-based to manipulate the virtual camera (pan, zoom and rotate). These buttons are used to improve the usability on tablet PC. Rendering property (transparency) are changed using the sliders in the zoom and pan GUI, as shown in Figure 5.6. Examples of 3D models displayed with VID can be seen in Figure 4.3.

If registered images are available, they can be displayed in the renderer, example of such view can be seen in Figure 4.4. Images can be selected from the tree-view with thumbnail display when hovering over the image name or via a thumbnail browser at the bottom of the 3D viewer. When a mixed view is activated, the user can change the zoom factor and center of view using the zoom and pan mixed view interface. The commands' translation from a 2D thumbnail to a 3D camera are discussed in detail in Section 5.2. To facilitate the navigation between mixed views the user can request a particular view, for example a view from the left of the current image or a close-up. This should simplify the load for the user by having direct access to a new viewpoint, instead of browsing the

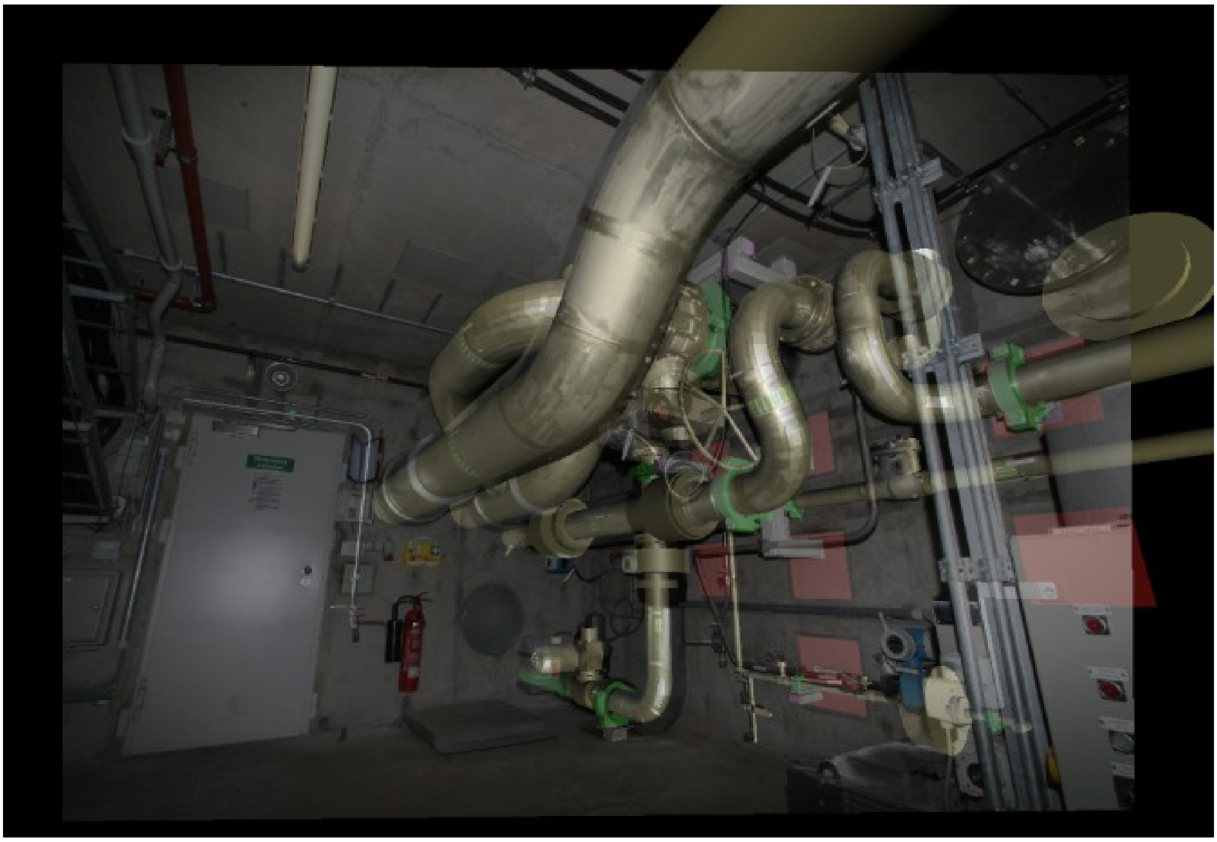


Figure 4.4: **VID an augmented CAD viewer**: VID supports the inspection of CAD Model via augmented reality. It offers to align and display an image within a CAD Model.

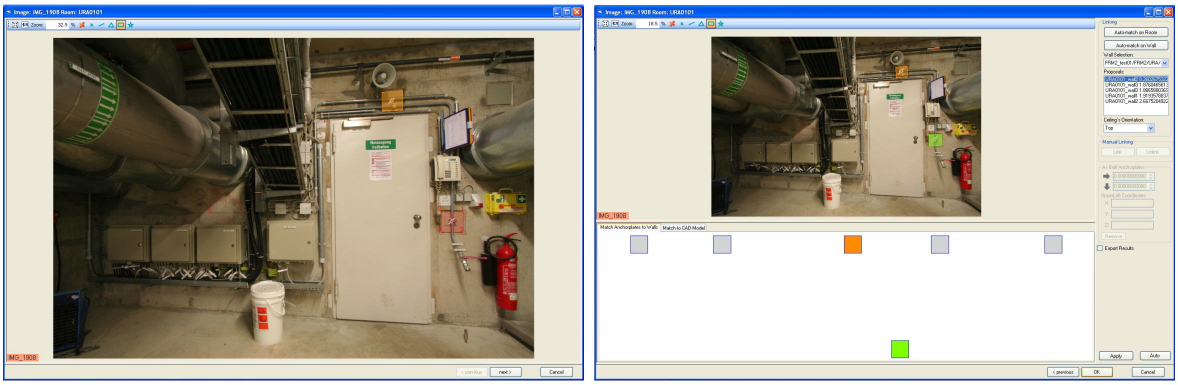
complete list of images. These techniques are discussed in Section 5.3.

4.2.2 Image Registration based on Anchor-plates

To register an image, one has to select at least one anchor-plate and needs to match[4] it with its corresponding 3D model, see Figure 4.5. Once this is done, the pose of the camera is known, its location in 3D and its orientation with respect to the CAD model coordinate system. This allows the users to properly display the image in the 3D renderer.

A user interface (UI) for anchor-plates selection and matching and computer vision algorithms supports the user during registration. This UI includes algorithm to ease and speed up the registration. For selection, a segmentation algorithm was developed where the user is only required to select a region of interest in the image where a anchor-plates lays. The algorithm then tries to find a structure that could be a plate and propose candidates to users. Though theoretically one anchor-plates is enough to obtain the pose, to offer an automatic matching two anchor-plates are required. Therefore user usually selects two anchor-plates, to be supported during matching that is cumbersome to do manually because a wall has often tens sometimes hundreds of anchor-plates (see

[4] put in relation.



(a)Step 1. Segmentation

(b)Step 2. Matching

Figure 4.5: **Anchor-Plates Based Registration GUI**. In order to register an images, first (a) the user segment the Anchor-plates in the image to perform this task he is supported by a segmentation tool; then (b) he matches the extracted image anchor-plates to their model.

Figures 6.1 and 6.3 for examples of anchor-plates' layouts). For matching, when two or more anchor-plates are available, the software finds a similar layout in the model. This algorithm that can consider all walls as input which greatly simplifies the user work when the difference between walls is hard to visualize. These techniques are explained in some length in Chapter 6.

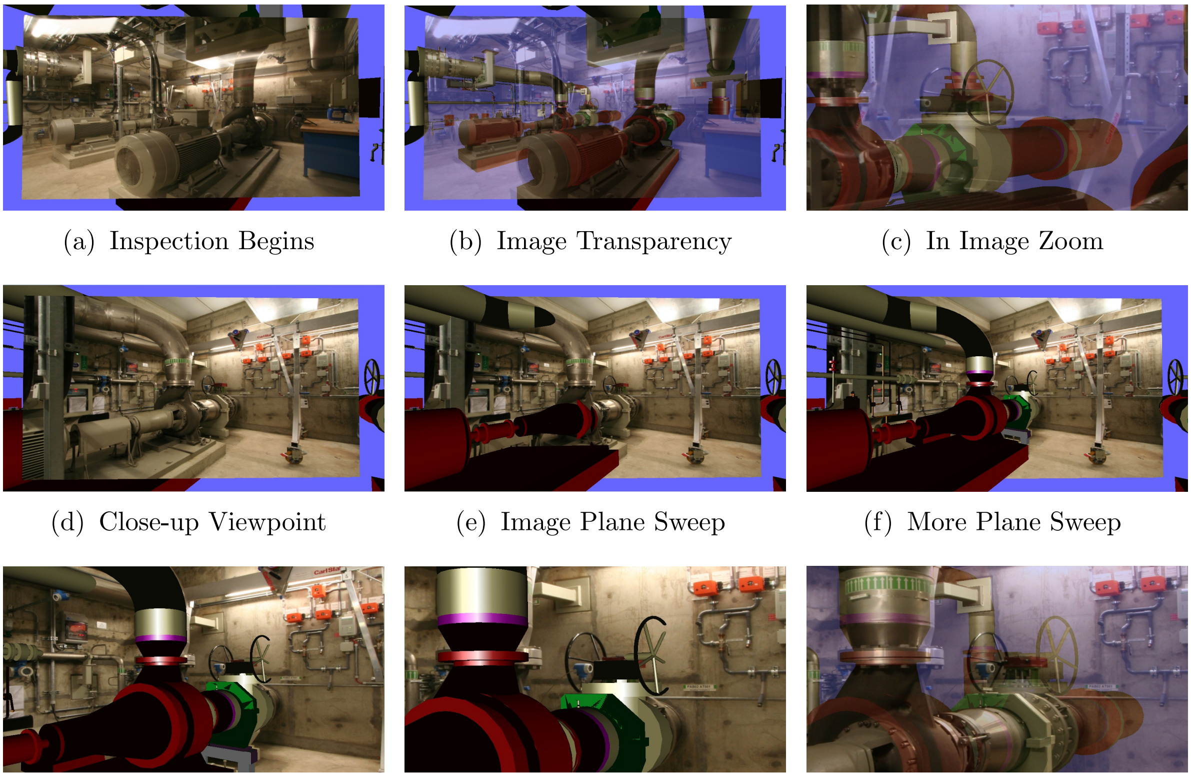
4.2.3 Discrepancy Check and Reporting

If available registered images can be directly displayed in the 3D renderer. Then the virtual camera location is locked to the image projection center and the image is rendered in front of the virtual camera. This gives what is called a *mixed view*. The user can reveal discrepancy by changing the transparency of rendered objects (c.f. Figure 4.8) and 3D Model (c.f. Figure 4.9), or the depth of the virtual image plane or VIP[5] (c.f. Figure 4.10) and by focusing the view (c.f. Figure 5.2). Figure 4.6 demonstrates how a user reveals a discrepancy. The system include all the necessary interaction to modify the scene appearance to detect discrepancy.

Issue can be documented and reported using a dedicated GUI that allows users to annotate snapshots of mixed view. Issue are usually defined as a set of mixed views that picture the problem along with a set of textual comments that detail it. If necessary engineers can perform metric measurements in the image, using triangulated points, to estimate more precisely discrepancy. Thus allowing the user to infer on the impact of a discrepancy. For example to make sure that it will not create a clash with a design update because even though the virtual model of the update does not collide with the discrepancy it has to be within a safety range[6]. Screenshots of the GUI and annotated

[5] The virtual image plane (VIP) is the base of the pyramid formed by the camera center and the back-projected image corners in 3D, c.f. Figure 5.2.

[6] Regulations require free space around potential dangerous machines in order to be safely operated



(a) Inspection Begins

(b) Image Transparency

(c) In Image Zoom

(d) Close-up Viewpoint

(e) Image Plane Sweep

(f) More Plane Sweep

(g) In Image Zoom

(h) Further Image Zoom

(i) Final Augmentation

Figure 4.6: **User interaction for discrepancy check**. The user can change the scene rendering property in order to reveal discrepancy between the model and the images. The user can change the transparency of the model or of the image plane. He can translate the image plane. He can zoom and pan in a image and request new view point in designated direction. All these interactions are supported by VID.

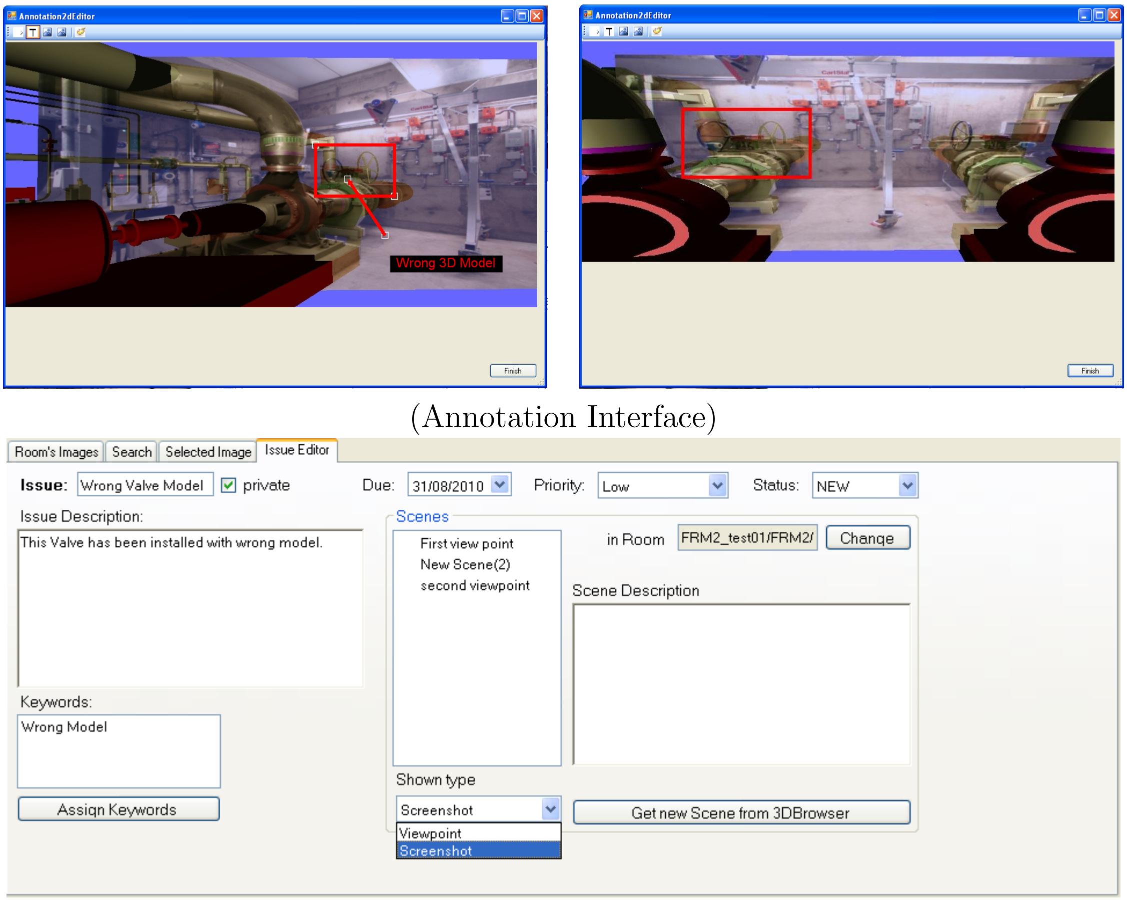
mixed views can be viewed in figure 4.7.

Once a discrepancy has been documented using VID, several options are available to the engineer depending on its severity. The discrepancy can be minor and therefore be documented as a redline on the augmented CAD model; a user, which accesses the model a posteriori has access to this annotations. Or the discrepancy can be major and lead to two different conclusions. Either, a formal complaint is sent to the sub-contractor for misconstruction. In this case, the mixed view is used to document the complaint. Or the discrepancy is integrated in the model by a new design incorporating for the discrepancy. This new design can be based on measurement taken with the help of VID.

Conclusion

In this Chapter, we described the process developed during this thesis to support civil engineers detect discrepancy using AR visualization. It is applied to follow the correctness

[Gausemeier et al., 2002].



(Annotation Interface)

Issue: Wrong Valve Model ☑ private

Due: 31/08/2010

Priority: Low

Status: NEW

Issue Description:

This Valve has been installed with wrong model.

Keywords:

Wrong Model

Scenes

First view point
New Scene(2)
second viewpoint

in Room FRM2_test01/FRM2/

Scene Description

Shown type

Screenshot

Viewpoint
Screenshot

(Documentation Interface)

Figure 4.7: **Discrepancy Documentation GUI**. When a discrepancy or a problem is found using VID the user can document it using this GUI. He can attached different 3D scenes (image, rendering properties and displayed models) and annotated views along side a description, its status and its importance. All the data is then stored in our database.

of a power-plant construction but it could also be used for other type of manufactured goods such as ships or printed controller boards. In the next Chapter, we present two navigation techniques to operate in the CAD software that includes high resolution registered images (c.f. Chapter 5) and a registration method adapted for large structure such as power-plant (c.f. Chapter 6).

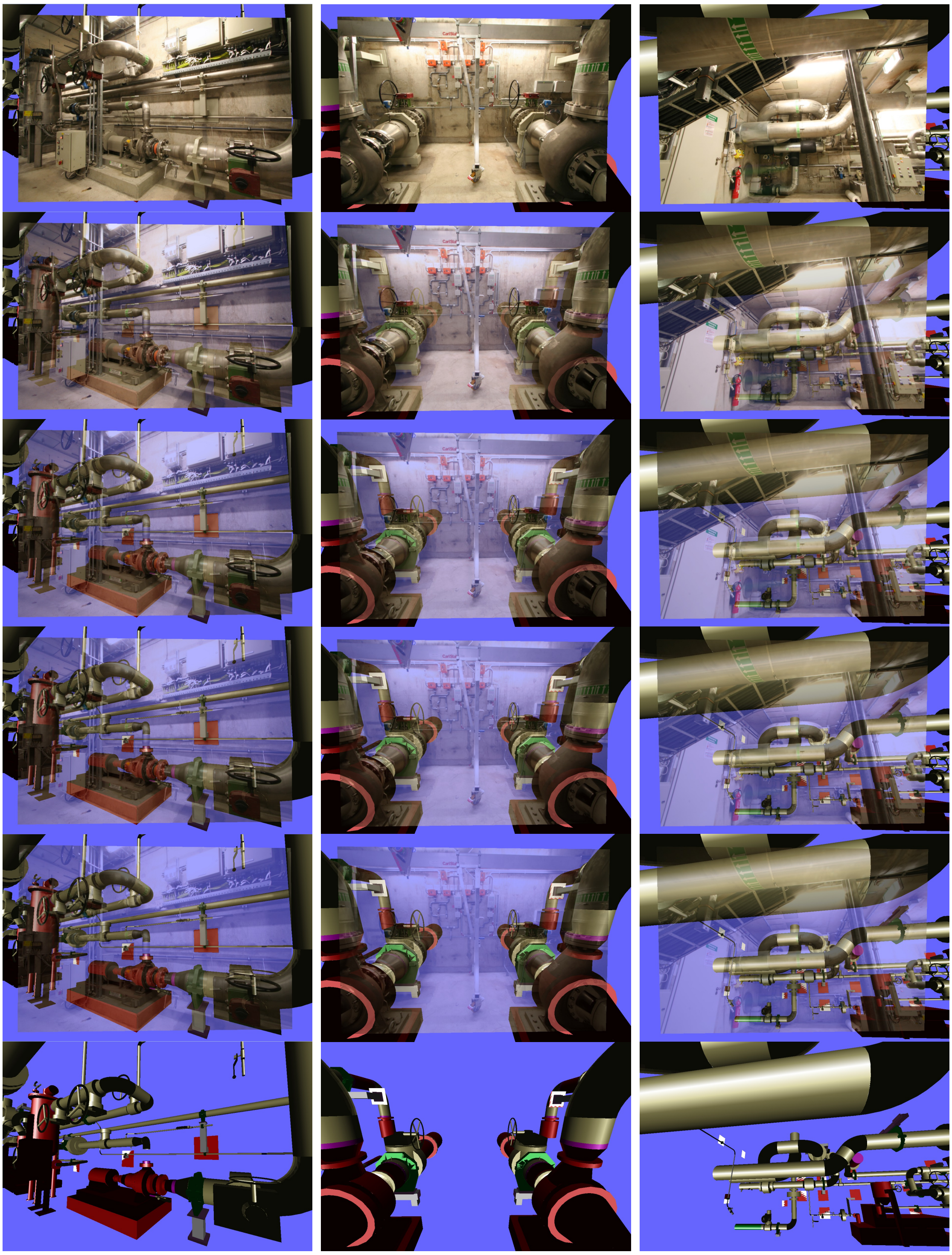


Figure 4.8: **Impact of modifying the Transparency of the Virtual Image Plane (VIP) on Mixed Views**: The change of transparency of the plane on which the image is textured, allowing to switch from the Picture (reality) at the top to CAD Model (virtually) ot the bottom. This interaction helps to detect differences between the mock-up and the built state.

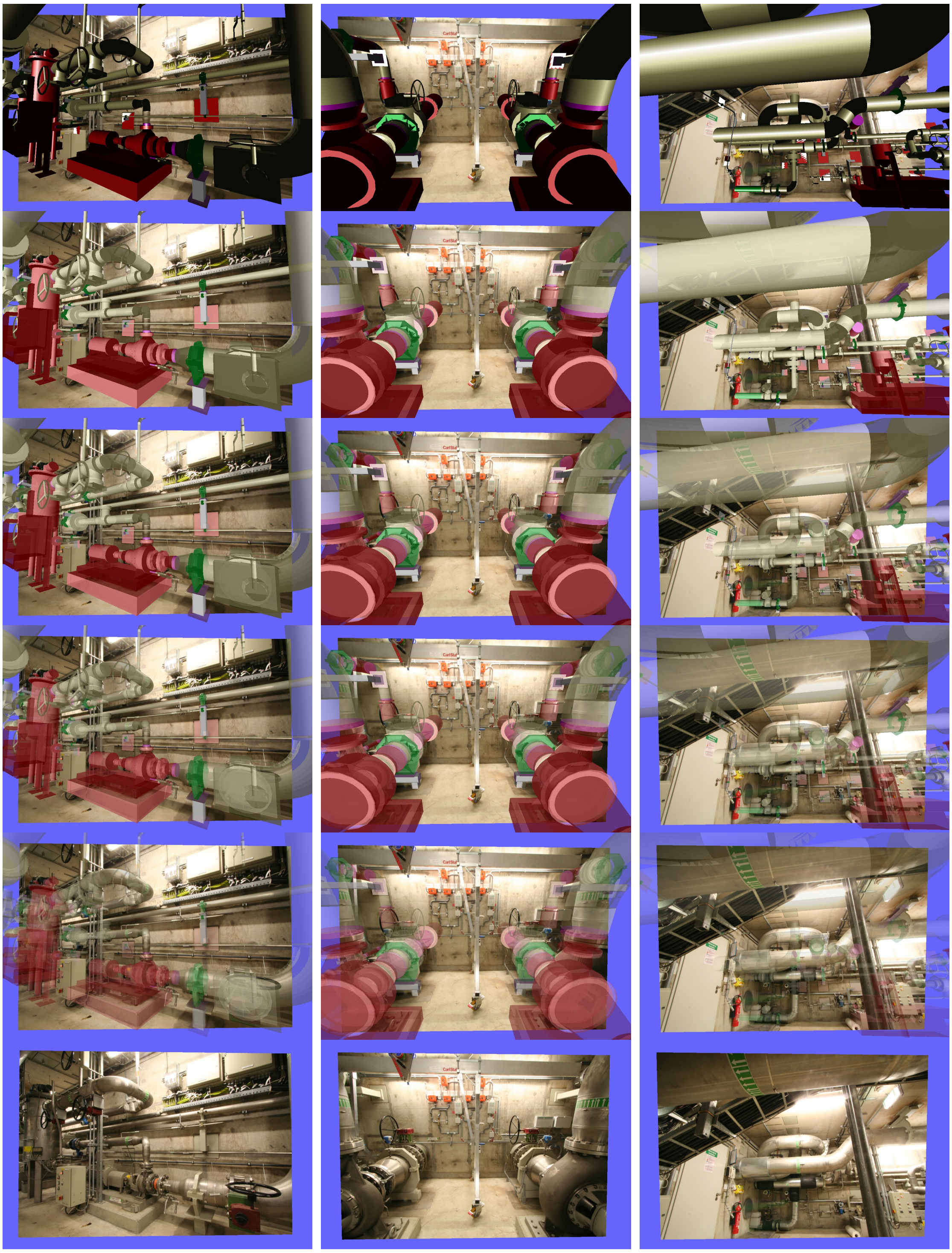


Figure 4.9: **Impact of Changing the Transparency Property of the CAD Model on Mixed Views**: The change of transparency of the *3D model* (e.g. pipes) allows the user to switch from a regular mixed view (Augmented Reality) at the top to image (Reality) at the bottom. This interaction helps to detect differences between the mock-up and the built state.

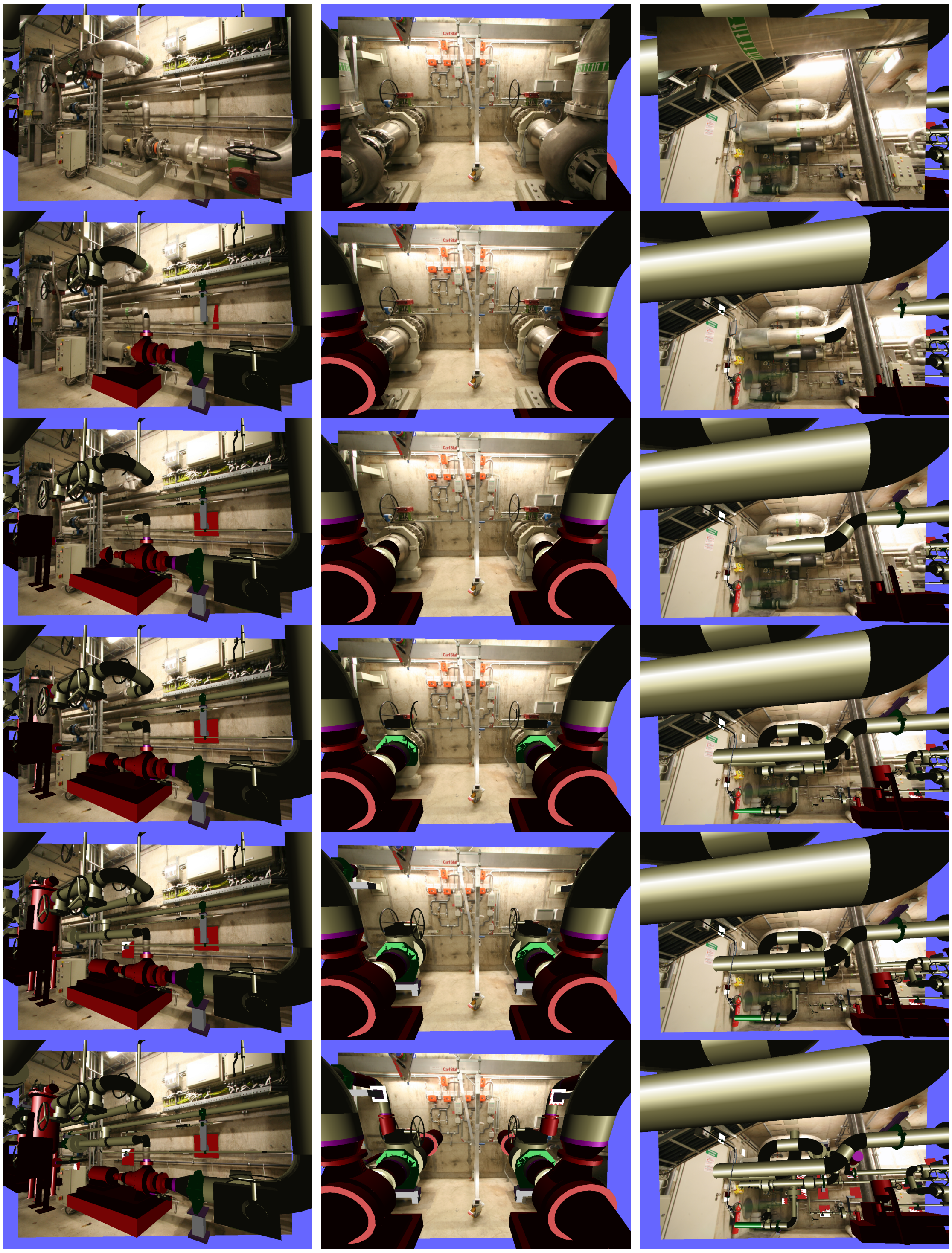


Figure 4.10: **Impact when Changing the Virtual Image Plane's Depth on Mixed Views**: The change of depth of the *VIP* with respect to the model allows the user to switch from image (Reality) at the top to regular Mixed View (Augmented Reality) at the bottom. This interaction helps to detect differences between the mock-up and the built state, especially since it allows to follow pipelines easily.

CHAPTER

FIVE

NEW USER INTERACTIONS OF AN AUGMENTED CAD SOFTWARE

The proposed augmented CAD viewer requires additional navigation tools in order to be usable. A lot of research has been performed to evaluate the best way to operate a virtual camera [Ware and Osborne, 1990]. Unfortunately it is not possible to change freely the viewpoint in a mixed view and keeping a correct registration at the same time, as the modification of the virtual camera should not create any parallax. The plane, where the picture is textured on, can only be translated in the $z - axis$ (c.f. Figure 4.10) and the viewpoint (e.g. position and orientation of the virtual camera) cannot be changed. In order to stay consistent in this mixed world, methods for browsing through images need to be introduced. One should avoid going through a list of images sorted by names or dates when navigation can be made easier by using geometric information about the viewpoints. In order to address these limitations, we first develop a new "zoom and pan" user interface for navigation within a mixed view. We then investigate the problem of navigation within a set of registered images using 'virtual' 3D points and thus allowing users to access other mixed views intuitively.

In this Chapter, we first review related work in image navigation using 3D information in Section 5.1. Then in Section 5.2, we describe our "zoom and pan" interaction and in Section 5.3 a new method to browse images using 3D information. Finally in Section 5.4, we present results delivered by these methods.

5.1 Prior Art in AR-CAD Interactions

An immense amount of work has been done to mix CAD models and real images, and to interact with them. We summarize some of these approaches, which use still images in the next section and then we discuss different systems to navigate within registered images.

5.1.1 Augmented Reality and CAD

Augmented reality is generally applied to video-streams or live-streams but many industrial projects use photo-based Augmented Reality[1], because it is easier to integrate in existing workflows. Navab et al. [1999b] present a platform named *cylicon* that uses AR in order to re-engineer the CAD model and create an as-built model. Interaction with *cylicon* is developed to facilitate reconstruction by minimizing the number of interactions with the 2D CAD and images. The user does not interact with the mixed world but the software uses information from the registration to reconstruct a 3D model of the plant. Pentenrieder et al. [2007] use AR for factory planning. Their solution supports factory engineer to decide whether a virtualy planned factory upgrade is feasible in reality. They however do not describe the interactions used apart from measurement capabilities.

Augmented 2D models have been investigated in [Appel and Navab, 2002]. They align technical drawings (2D models) and images to create *co-registered orthographic perspective views*. These views mix floor maps and pictures thus adding information from the map to the reality. Most of their work focuses on creating automatically a good mixed view, for example by properly merging the blueprints with an image by detecting the floor. Their main goal is to help civil worker to use floor maps, which are complex documents. However the use of 2D models is fading away.

All these approaches only offer static augmentation and no interactive navigation. Therefore the user suffers from a certain loss of 3D perception, when observing a mixed view.

5.1.2 3D Navigation

Photo-tourism[2] [Snavely et al., 2006] offers a good review of image based 3D navigation. In this paper, the authors collect a large amount of images gathered from the Internet from a unique natural scene. They register the image set. This is performed using image features and structure from motion. This provides a sparse reconstruction and the camera viewpoints (3D position and orientation). Photo-tourism allows one to display the image frustum (representing the camera) and the sparse structure. The user can select a given frustum and navigate through the images. Photo-tourism proposes three methods to navigate. First, object-oriented, which gives access to all images that visualize a given objects represented abstractly by its sparse reconstruction. Second, by selecting another frustum, the virtual cameras move to the selected viewpoint. Third, from a source image it offers six directions: zoom in, zoom out, left, right, up and down. The next image is automatically selected using the sparse structure visible in the source image. But sparse reconstruction might not always be available or feasible, therefore limiting the use of this approach. Furthermore, photo-tourism does not contain any tool to navigate (zoom and pan) in an image thus forcing the user to interact directly in the 3D world, which is not always intuitive. They introduce in [Snavely et al., 2008] alternative navigation tools: "scene specific control" such as orbit motion and panoramas. They used reconstructed feature points and angle of views to determine different controls.

[1] Augmentation based on a single still image.

[2] Also called photo-synth.

Another popular approach is the moviemap [Mohl, 1981, Uyttendaele et al., 2004] where a multi-camera system moves through an environment. In most moviemap systems, the cameras are mounted on a car that travels along the street. These methods offer 3D navigation between images. From any view, the user can either go to a following position (where an image was acquired) or do a left-right turn (around its current position). Since the image sequences are recorded using a fix setup, the geometry between the views are perfectly known and stay constant over time. This allows the system to always have the same behavior. Such a system is being used, in the commercial software *Google map*[3] as a feature named *Google street view*, to provide a real view of the urban environment from a given position on the map.

5.2 Interaction Within a Mixed View

A mixed view is different from a ‘regular’ augmentation. While a regular augmentation displays the image and a 3D model that has been moved to the camera coordinate system. The user view can be changed by directely manipulating the position of the camera that acquires the image. The camera can be fixed to an HMD [Webster et al., 1996] or to a display (e.g. cellphone) [Goose et al., 2004]. In a mixed view, the image is textured on a 3D rectangle that is fixed in the CAD coordinate system. VID proposes different interaction in this mixed world. 3D components can be added or removed. The image plane can be translated in z in order to visualize the model in front or from the back of the image, as shown Figure 4.10. The transparency of all objects can be modified: frustum or model, thus allowing one to obtain a good mixture between virtuality and reality. We will first explain how to position the frustum in the model and then how to perform the zoom and pan.

5.2.1 Mixed View Positioning

All the images that are stored in the model have been registered to the coordinate system of the model. By registered, we mean that we know the internal (focal, skew and principal point) and external (position and orientation) parameters of the camera. We denote by K the matrix of the internal parameters of a camera and for the external parameters, by R its rotational part and by $\mathbf{t}$ its translational one. For more details on the the notation please refer to Chapter 3. A 3D point $\mathbf{M}$ relates to an image point $\mathbf{m}$ of the camera as follows $\mathbf{m} \sim \mathsf{K}\left(\mathsf{R}\mathbf{M} + \mathbf{t}\right)$.

A virtual viewpoint has 7 degrees of freedom that need to be set: a field of view, a rotation and a *Center of Projection*. The *Center of Projection* of the camera is defined in the CAD coordinate system as $\mathbf{O} = -\mathsf{R}^\top\mathbf{t}$.

The image frustum (c.f. Figure 5.2) is a quadrangle formed by four 3D corners $\mathbf{C}^j$ with $j \in \{1 \cdots 4\}$. The 3D corners are defined as follow:

$$\mathbf{C}^j = \alpha\mathsf{R}^\top\mathsf{K}^{-1}\mathbf{c}^j + \mathbf{O} \tag{5.1}$$

[3] http://maps.google.com

with $\alpha \in \mathbb{R}_*^+$ that defines the frustum translation in z and $\mathbf{c}^j$ the corners of the image in pixel coordinates.
Computing the field of view ϕ_{vc} of the virtual camera is straightforward from K, this opening angle is represented in Figure 5.2.
In order to obtain a view centered on the image, the virtual camera needs to be rotated by a matrix R_{vc} from the standard neutral camera position point in the $-z$ direction with the y direction being up-right. This matrix is formed using the image corners. We define the three unit vectors $\mathbf{V}_x \sim \mathbf{C}^1\mathbf{C}^2$ (the red/horizontal vector in Figure 5.2), $\mathbf{V}_z \sim \mathbf{O} - \frac{1}{4}\sum_{i=1}^{4} \mathbf{C}^i$ (resp. green/out of the plane in Figure 5.2) and $\mathbf{V}_y = \mathbf{V}_z \times \mathbf{V}_x$(resp. blue/vertical in Figure 5.2) and form then the orthogonal matrix:

$$\mathsf{R}_{\mathsf{vc}} = [\mathbf{V}_x\mathbf{V}_y\mathbf{V}_z] \tag{5.2}$$

We have summarized all the information necessary to setup the virtual image plane ($\mathbf{C}^i$) and the virtual camera (ϕ_{vc}, R_{vc} and $\mathbf{O}$) in a mixed view. Now we focus our attention on the proper way to interact with the virtual camera.

5.2.2 Zoom and Pan Interaction

In order to make full use of modern high-resolution cameras in photo-based augmented reality, the idea of zoom and pan from 2D user interfaces has to be transposed. Regular desktop screens offer a resolution of around 2 mega-pixels whereas professional cameras offer 15 mega-pixels. Further restrictions have to be considered, as we cannot expect to have the complete screen available for the augmentation because standard GUIs of CAD software are cluttered with different tools.

Typical 2D UIs for the zoom and pan allow one to set a zoom factor and a center of interest or *Focus* materialized by an image point $\mathbf{f}$. This describes the *Area of Interest* that has to be displayed and it is defined by its four corners $\mathbf{a}^j$. Using Equation 5.1, we can obtain $\mathbf{A}^j$ (resp. $\mathbf{F}$) from image points a^j (resp. f).

We now redefine virtual camera rotation R_{vc} from equation 5.2 using the 3D points $\mathbf{A}^j$ and $\mathbf{F}$. $\mathbf{V}_x$ and $\mathbf{V}_y$ are defined in the same manner. Only the last vector $\mathbf{V}_z$ is changed as follow $\mathbf{V}_z = \mathbf{FO}$. The field of view ϕ_{vc} is set the same way as before using the $\mathbf{A}^j$ instead of the $\mathbf{C}^i$.
To focus the mixed view on a particular area of interest the user will set the area of interest in a 2D thumbnail, making it intuitive to navigate in a mixed view. Additionally it should be pointed out that we could zoom out of the image to gain access to a more contextual view of the model surrounding the image. Some results are visible in figure 5.3.

5.3 3D Navigation

In order to ease the use of augmented CAD, new navigation methods have to be introduced. These methods have to be intuitive (i.e. made of simple interactions) for the acceptance of the solution. The proposed tools make full use of the available geometric

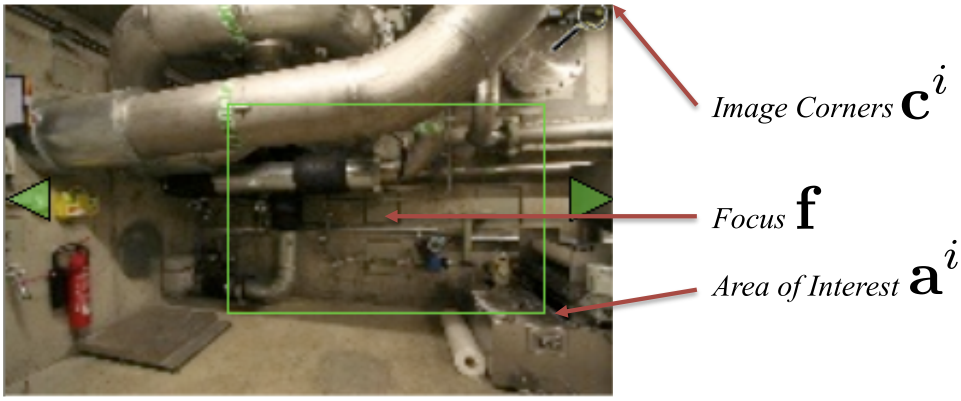


Image Corners $\mathbf{c}^i$
Focus $\mathbf{f}$
Area of Interest $\mathbf{a}^i$

Figure 5.1: **Formalization of a 2D Zoom and Pan interface**: the user sets a focus **f** and a scale factor, which defines an *Area of Interest* $\mathbf{a}^i$ in an image describe by its corners $\mathbf{c}^i$. As a convention $\mathbf{c}_1$ (respectively $\mathbf{a}_1$) is the upper left corner of the image (resp. of the area of interest) and the number goes clock-wise.

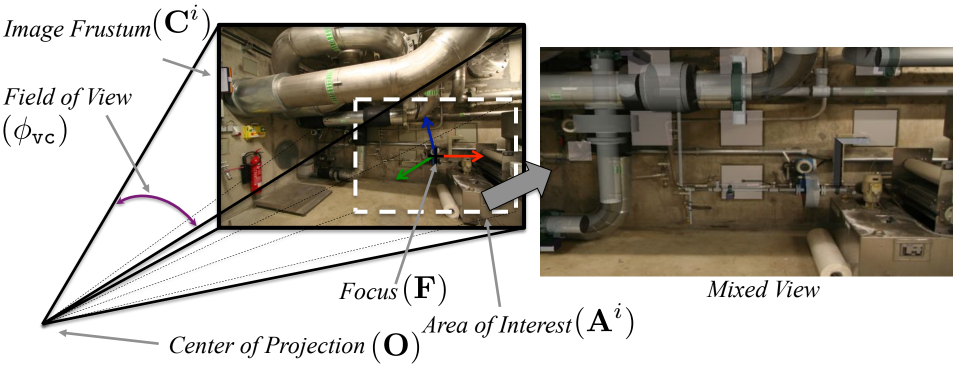


Image Frustum $(\mathbf{C}^i)$
Field of View $(\phi_{\mathtt{vc}})$
Focus $(\mathbf{F})$
Area of Interest $(\mathbf{A}^i)$
Center of Projection $(\mathbf{O})$
Mixed View

Figure 5.2: **Virtual camera under a zoom and pan motion**: The virtual camera rotates to be centered on the *Focus* **F**, the *Field of View* $\phi_{\mathtt{vc}}$ changes to fit the *Area of interest* $\{\mathbf{A}^j\}$, note that the *Center of Projection* **O** stays unchanged to avoid parallax; on the right the resulting *Mixed View* with a gray transparent model focused on the white dashed area.

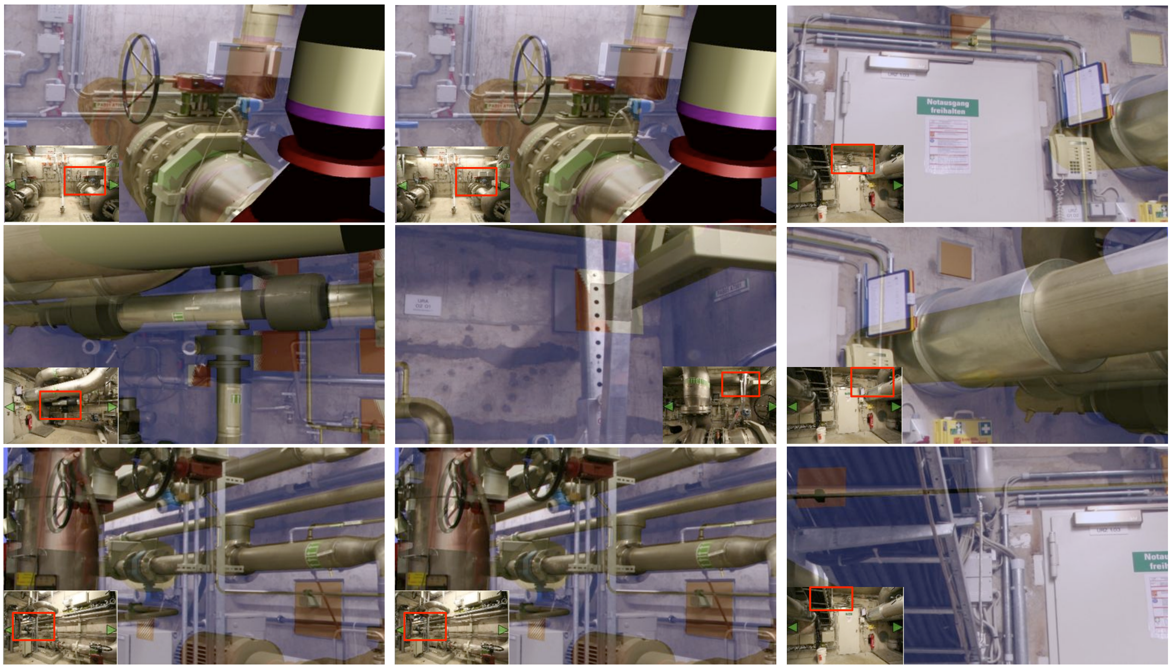


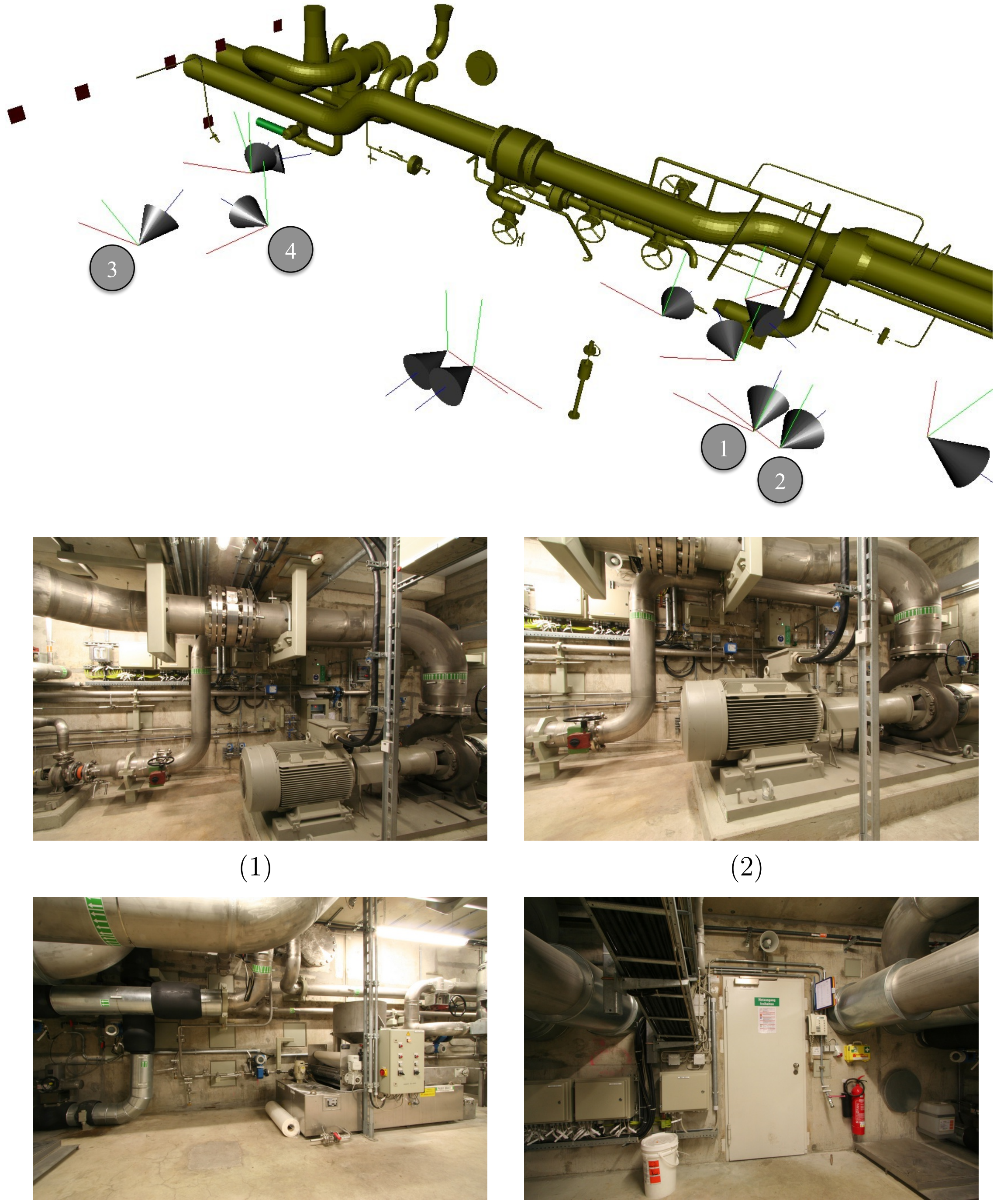
Figure 5.3: **Results of Zoom and Pan motion**: the virtual camera focuses on the area represented by the marked rectangle in the thumbnail image. This interface allows a detailed view of the area of interest but still with access to information about the context from the thumbnail. The virtual model is pictured in red and gold, the light blue is the background of the rendered scenes.

information offered by a set of mixed views. In this section, we introduce a new method to navigate in a set of registered images.

It might sound trivial to choose the next viewpoint to visit in a given direction. Unfortunately it is much more complex if we want to provide natural navigation to the user. From a mathematical point of view we can clearly say whether a viewpoint is at the left or right hand side of another viewpoint because we know where the camera is in 3D. As shown in Figure 5.4 this would work perfectly for viewpoint 1 and 2. But, this would not take into account the direction in which the camera is looking at. In Figure 5.4, imagine what happens if the user is positioned at viewpoint 3 and chooses to move to the right based only on viewpoint positions. This would lead him to viewpoint 4. This is not satisfactory because a right hand side view of the scene was requested and the proposed view 4 shows a left hand side. The problem lies in the fact that we do not want the next viewpoint to be in a specific direction but that the selected viewpoint pictures the scene in this specific direction. In order to consider both position and orientation, we use the local position of *virtual points*.

5.3.1 Virtual 3D-Points

First, we determine in which direction each camera is pointing. Using this line of sight, we approximate the focus depth (maximum distance to all visible objects). This limit



(1)

(2)

(3)

(4)

Figure 5.4: **Exemplary virtual scene used in VID**: the gold pipe is part of the CAD model that was augmented with pictures; the cones represent the locations of a subset of registered images; Notice the geometric ambiguity: (1) is to the left of (2) which can be extrapolated from the centers of projection, but this method does not extend to the pair (3) and (4) where the relation is inverted; the relative position between the scene and the views has to be used.

is represented by a point that we call a *virtual 3D point*. It is virtual because it is not directly linked to the real structure. For each camera i, they are computed as the intersection of the (half-)line of sight (from image i starting from $\mathbf{m}$) $\mathbf{L}_i(\mathbf{m}) = \{\mathbf{M} | \mathbf{M} = \alpha \mathsf{R}_i{}^\top \mathsf{K}_i^{-1} \mathbf{m} + \mathbf{O}_i, \alpha \in \mathbb{R}^+\}$ and the 3D structure noted $\mathbf{S}$. Virtual 3D points are defined as:

$$\check{\mathbf{M}}_i(\mathbf{m}) = \mathbf{L}_i(\mathbf{m}) \cap \mathbf{S} \,. \tag{5.3}$$

Often very little information is available about the scene (a coarse reconstruction, some models, points used for calibration, camera positions...), thus we cannot guarantee that $\mathbf{L}_i$ intersects the available 3D information. Therefore $\mathbf{S}$ has to be a continuous and abstract representation of the scene. We decided to use some outer boundary of the scene that should be easily computed from the available data (this will be covered in more detail in Section 5.4.2), named $\mathbf{B}$.

Figure 5.5 illustrates the idea of the virtual points lying on a bounding surface. Now that we have information about the focus depth of each cameras we can classify the relative position between views.

5.3.2 Direction Classifier

We want to classify neighboring views in 6 different clusters: left, right, up, down, zoom-in and zoom-out. These clusters is the directions available in the image interaction GUI. The user can then decide in which direction the view should be moved. The classification will be based on the relative position in the image of the *virtual 3D points*' projection.
We consider the source image $\mathcal{I}_s$ and its neighbors $\mathcal{I}_{n_s}$. We project all virtual points $\check{\mathbf{M}}_{n_s}(\mathbf{m})$ onto the image $\mathcal{I}_s$. This returns a set of points $\check{\mathbf{m}}_s\left(\check{\mathbf{M}}_{n_s}(\mathbf{m})\right)$ in the image s, for simplicity we define $\check{\mathbf{m}}_s\left(\check{\mathbf{M}}_{n_s}(\mathbf{m})\right) = \check{\mathbf{m}}_s^{n_s}(\mathbf{m})$. To summarize $\check{\mathbf{m}}_s^{n_s}(\mathbf{m})$ is the projection on the camera s of the virtual point issued from the point $\mathbf{m}$ defined from image n_s. Now to classify the relation between the camera s and n_s we only have to analyze the position of virtual points $\check{\mathbf{m}}_s^{n_s}(\mathbf{m})$ in relation to the area of interest $\mathbf{a}_s^j$ and the focus $\mathbf{f}_s$ by applying the following test:

- *left*: $\check{\mathbf{m}}_s^{n_s}(\mathbf{f}_{n_s}) \in \triangle(\mathbf{f}_s, \mathbf{a}_s^1, \mathbf{a}_s^4)$
- *right*: $\check{\mathbf{m}}_s^{n_s}(\mathbf{f}_{n_s}) \in \triangle(\mathbf{f}_s, \mathbf{a}_s^2, \mathbf{a}_s^3)$
- *up*: $\check{\mathbf{m}}_s^{n_s}(\mathbf{f}_{n_s}) \in \triangle(\mathbf{f}_s, \mathbf{a}_s^1, \mathbf{a}_s^2)$
- *down*: $\check{\mathbf{m}}_s^{n_s}(\mathbf{f}_{n_s}) \in \triangle(\mathbf{f}_s, \mathbf{a}_s^3, \mathbf{a}_s^4)$
- *zoom in*: $\forall j$, $\check{\mathbf{m}}_s^{n_s}\left(\mathbf{a}_{n_s}^j\right) \in \square(\mathbf{a}_s^1, \mathbf{a}_s^2, \mathbf{a}_s^3, \mathbf{a}_s^4)$
- *zoom out*: $\forall j$, $\check{\mathbf{m}}_{n_s}^s(\mathbf{a}_s^j) \in \square\left(\mathbf{a}_{n_s}^1, \mathbf{a}_{n_s}^2, \mathbf{a}_{n_s}^3, \mathbf{a}_{n_s}^4\right)$

$\triangle(a, b, c)$ is the triangle formed by the points abc and $\square(a, b, c, d)$ is the square formed by the points $abcd$. $\mathbf{a}_{n_s}^j$ (resp. $\mathbf{f}_{n_s}$) are set to the corners (resp. the center) of the image n_s. Whereas $\mathbf{a}_s^j$ (resp. $\mathbf{f}_s$) can be set to the corners (resp. the focus) of the area of interest of s by using the zoom and pan GUI. Thus allowing one to specify more precisely

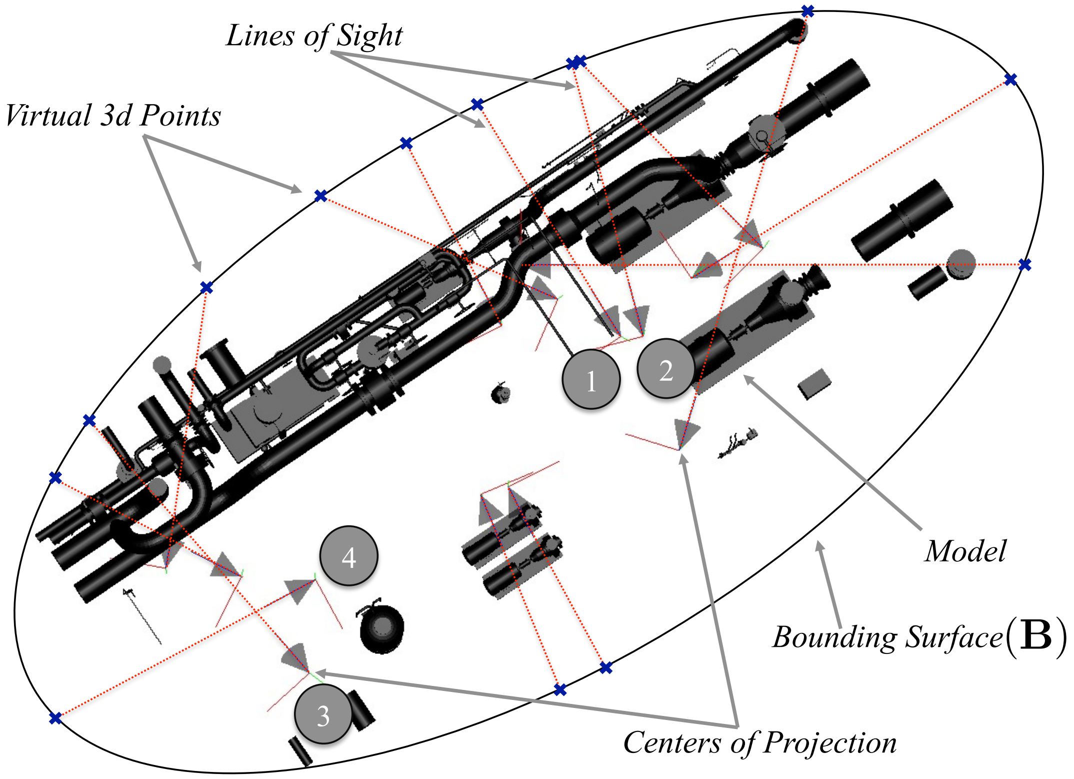


Figure 5.5: **Top View of an Augmented CAD Model**: the bounding surface encapsulates the model and the centers of projections. The Virtual 3D points are the intersections between the lines of sight and the bounding surface, they inform about the relation between the scene and the viewpoints. The images taken by the label image (1) to (4) are visible in Figure 5.4.

the request for the next view. In order to sort all the neighboring frames we first verify if they are classified as zoom in/out then, if they were not, we verify if they are left, right, up or down. To get the next image in a specific direction, we pick for the direction (left,right,etc) the image with its $\check{\mathbf{m}}_s^{n_s}(\mathbf{f}_{n_s})$ closest to $\mathbf{f}_s$ and for the zoom the image with the biggest projected area. The zoom in and zoom out motions are seperated from the directional motions to prevent an undesirable change of scale when requesting an image in a specific direction. It should be noted that we always verify that $\check{\mathbf{M}}_{n_s}(\mathbf{m})$ is in front of the camera s, so the proposed image is always looking in the same direction as the source.

5.4 Empirical Results

In this section, we discuss practical issues. First, we detail the implementation and then we present some experimental results for the 3D navigation technique.

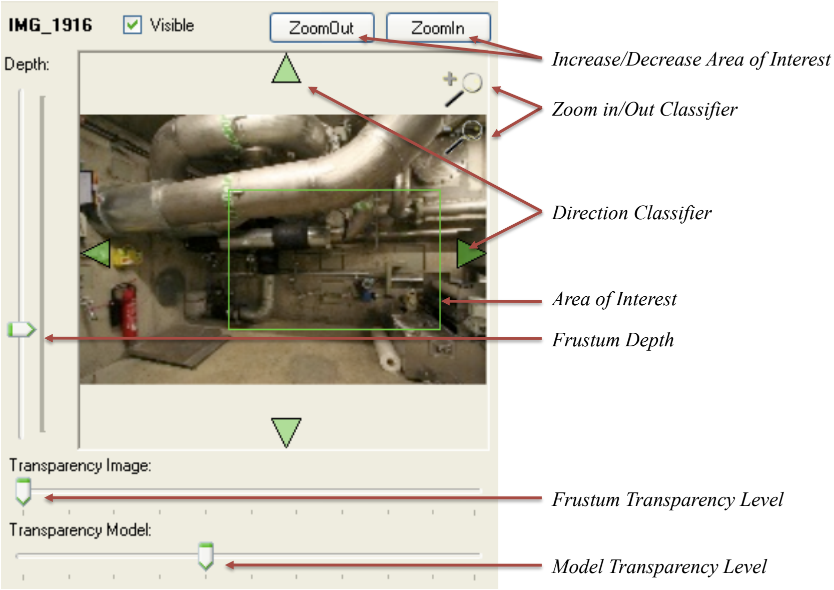


Figure 5.6: **Image Manipulator Graphical Interface**: This graphical component allows the user to manipulate the mixed view; he can change the focus of the scene, the transparency of the frustum and of the model, or the frustum depth; This to reveal discrepancy between the image and the model; He can also request other viewpoints using the classifiers.

5.4.1 Mixed View Manipulation - Implementation Details

We tested this approach within VID. The zoom and pan interaction and the 3D navigation, which we introduced, are merged in one reusable UI component (c.f. Figure 5.6). It retrieves information about the local geometry of the scene and image data from the database. It sends updates for the virtual camera and the image to be displayed to the rendering engine. All interactions are mouse controlled and are transmitted in real-time.

The outer-bound of the room is represented by an ellipsoid. An ellipsoid is a simple geometric structure, which allows fast collision detection but approximates the 3D scene's boundaries well. We use the Minimum Volume Enclosing Ellipsoid (MVEE) algorithm provided by [Moshtagh, 2005] to compute it. This ellipsoid is the smallest ellipsoid that contains the Centers of Projections and the 3D points used for registration. It can also incorporate models or sparse reconstruction. This representation has the advantage of dealing naturally with open environments. Results of computed ellipsoids are visible in figure 5.7. The ellipse model performs well for the model we used because it is clustered room-wise. Since the rooms are composed as a set of cuboids, they are well represented by ellipses.

5.4.2 User based Experiments

Even though this classifier does not have uncertainty, we need to verify that the proposed navigation is natural to a human. In order to validate the algorithm developed for the

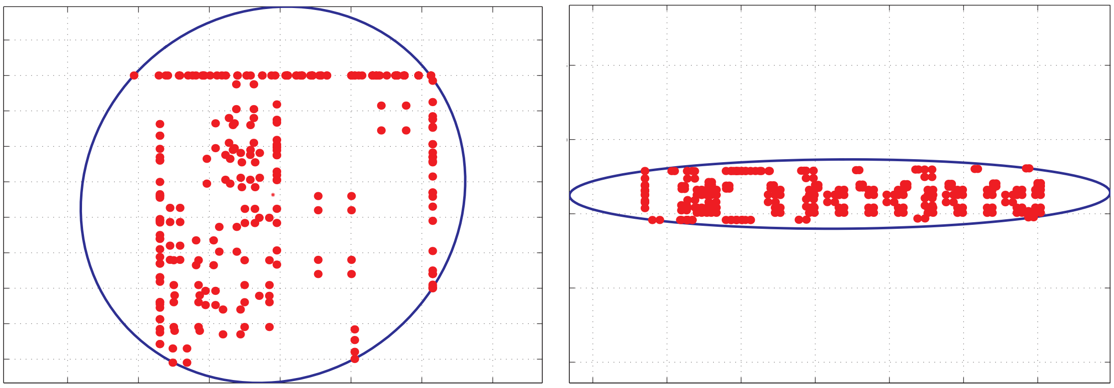


Figure 5.7: **Result of Minimum Volume Enclosing Ellipsoid on Power-Plant Room *Reality Model*.** The red dots represents the reality model composed of anchor-plates corners.

3D navigation we conducted a user study. It includes ten participants (two women, eight men), from the age of 24 to 32, and 2 sets of registered images. The task was to select two images they felt were to the left (resp. to the right) of a designated image called the source. The test users never visited the scene where the images have been captured. The sets were presented randomly to the user. We measured the time spent by each user on each set. The goal of these tests was to verify that the presented algorithm was offering a natural navigation.

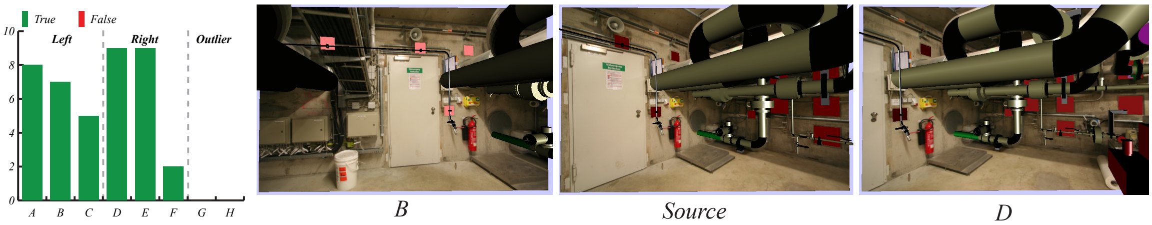
The first set was composed of 9 images including the source, 3 images considered by the direction classifier to be to the left, 3 images to the right and 2 outliers (images from the same scene but totally unrelated to any of the other image). We considered this set to be the simplest since it was acquired from a tripod which was only rotating. The average time for the users to perform the classification was 1 minute and 20 seconds.

Figure 5.8 shows the results of this experiments. It demonstrates that the algorithm handles such a scenario perfectly; the algorithm classified the images in agreement with the participants (0 percent misclassification).

The second set was also composed of 9 images including the source, 4 images to the left, 3 images to the right and 1 outlier. This set was more complex than the first one since the motions were not only composed of rotations. It also included translations. Additionally some views were extremely close to one another sometimes making the decision more complex. The average time for the users to perform the classification was 3 minutes and 29 seconds.

Figure 5.9 summarizes the results. The algorithm still agreed with 88 percent of the participants' decisions. Additionally, 50 percent of our algorithm errors involved image D, which was extremely close to the source. However, the position selected by the algorithm for this image still agreed with 66 percent of the participants, which shows that our algorithm works well even with ambiguous views.

Finally, some results for the zoom-in/zoom-out displacement are shown in figure 5.10. This mode of motion offers a good way to reach a specific part of the model.

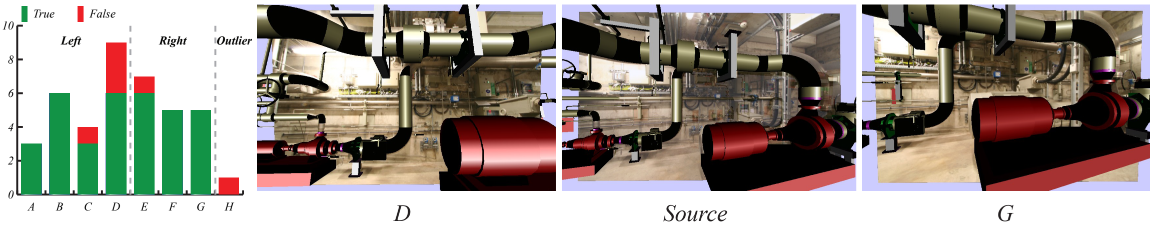


B

Source

D

Figure 5.8: **View Classifier - Experimental results for set 1 (Rotation)**. Test users had to select 2 images from A to H that they felt were on the left of the source and 2 to its right. If the decision of a user was in accordance to the algorithm we labeled it as true, false otherwise. All the images were correctly classified (see Chart). B is clearly to the left since the door is to the extreme left of the image Source and centered in B. D is clearly to the right since the door almost disappeared.



D

Source

G

Figure 5.9: **View Classifier - Experimental Results for Set 2 (Free Motion)**. The users' decisions were mainly in accordance with the algorithm result. The most misclassifications happened with image D which is extremely close to the Source but still to the left; see the pump that appeared on the left.

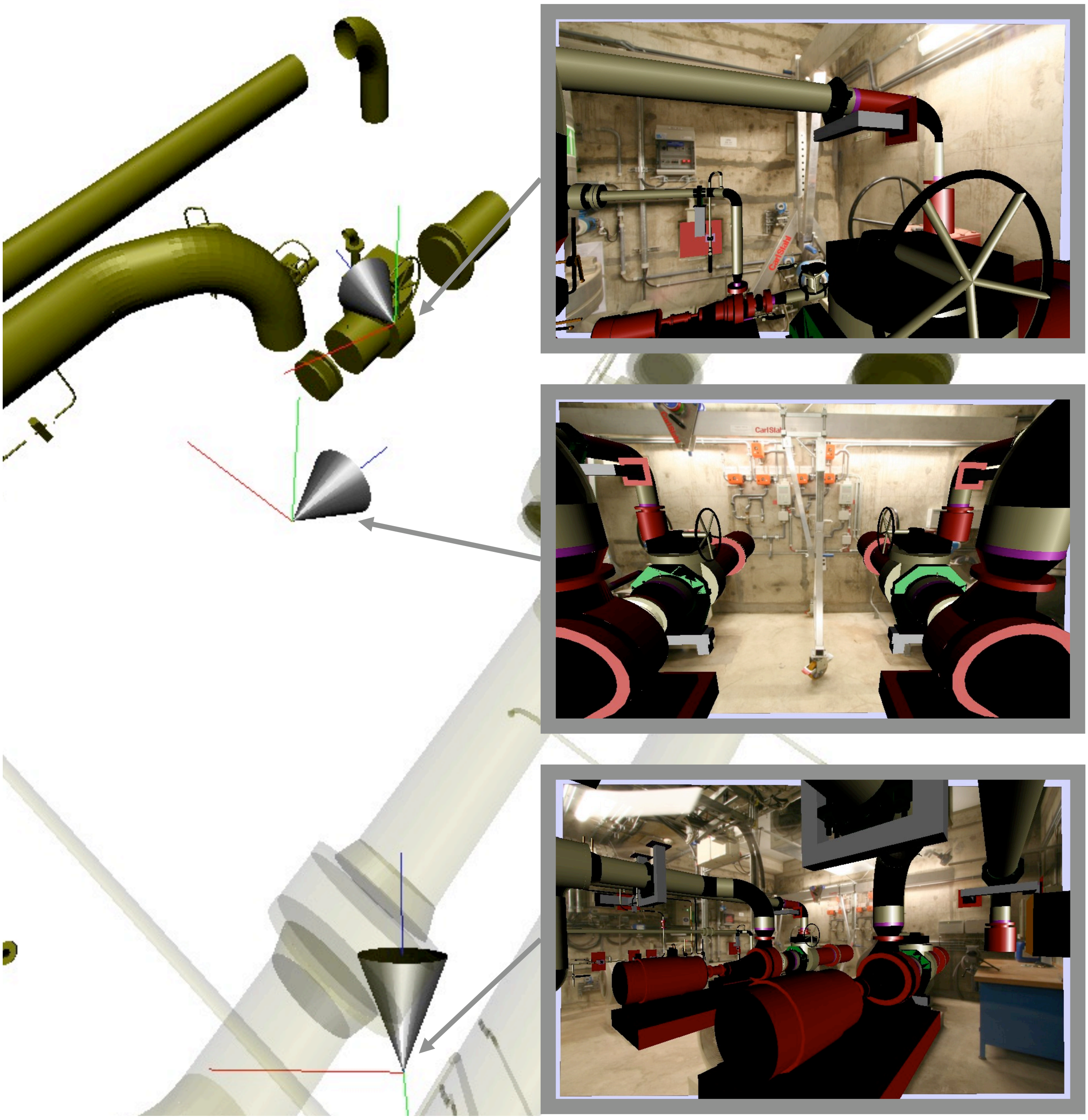


Figure 5.10: **Example of a zoom in/out motion**; on the bottom with the most outer view and on the top the most inner view, the local geometrical relation is visible in the 3D scene; see that the viewpoints' directions do not need to be aligned.

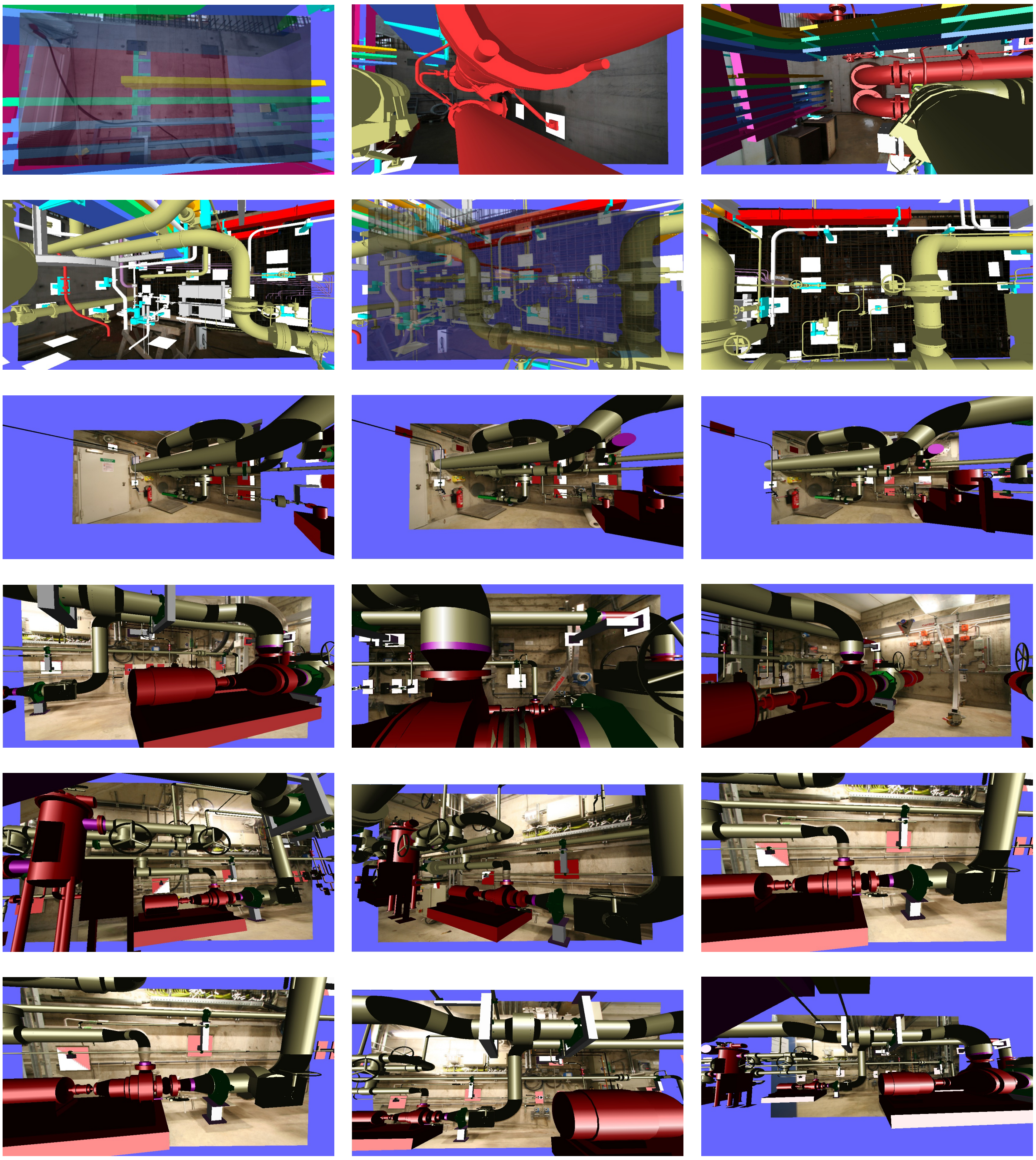


Figure 5.11: **Example of left/right navigation** The user can request a new view point that the presented algorithm estimates based on the focus of the current view. The left (resp. right) images correspondents to a requested view point from the center images to the left (resp. right).

Conclusion

In order to provide an intuitive augmented CAD viewer, we developed two new 3D user interaction to intuitively navigate through mixed views and interact with them. We extended the zoom and pan functionality to such mixed viewing environment. We also proposed a new method for navigating in a set of calibrated views requiring no reconstruction. These two techniques have combined into a standard GUI component that could be integrated in any CAD viewer, turning it into an augmented CAD viewer software.

CHAPTER

SIX

A COMPONENT-BASED REGISTRATION METHOD FOR INDUSTRIAL AUGMENTED REALITY

In this chapter, we focus our attention on the estimation of the rigid transformation between the CAD coordinate system and a camera. This is the corner stone of every AR system to offer to the user a correct alignment (without scene engineering) of an image and a virtual object. In the last two decades many approaches have been proposed to perform this task. Unfortunately, very few of these methods are scalable to register thousands of photographs from a structure of the size of a power plant. Here, we focus on the primary method for alignment used in VID. It is based on *anchor-plates* because these industrial components are broadly used in plant engineering. These plates are embedded in concrete structure and are used to fix various components, see Figure 6.1. They are described in more details later in Section 6.1.

As stated by [Klinker et al., 1998] augmented reality needs reality models. Models that correctly represent the reality and that are usable by an AR System for registration. For example, in the car industry it is quite popular to use drill holes [Echtler et al., 2003, Pentenrieder et al., 2007] because they are verified, precisely localized and well identifiable. Following this philosophy we use an available *Reality Model*, which for civil construction is based on anchor-plates. We define what these components are in the following Section 6.1. Then in Section 6.2, we introduce an algorithm to extract them from plant images and in Section 6.3 we explain the developed method to automatically match them to its 3D model. Finally in Section 6.4, we present registration results obtained with the proposed registration method.

For notation and definition please refer to Chapter 3.

6.1 Anchor-Plates

In AR, the use of reliable components is necessary. Otherwise the registration results are not trustworthy. If a part of the model is misplaced and used for registration, the estimated transformation T would only be correct with respect to the used part and completely wrong for the rest of the model. These kinds of behavior are not desirable. The used components have to build as planned or need to be surveyed after construction. In addition, they should consist of simple primitives to ease their detection. After discussions with plant designers, they proposed to use anchor-plates[1] because they are positioned with lots of care and when installed by subcontractors they are even surveyed. In fact they are the most reliable components in the factory, as they are often used as reference coordinate system by builders.

Anchor-plates are metallic structures embedded into the concrete walls used in the majority of industrial edifices. They are mounting points to fix other components: pipes, supports, cable-racks, control boxes, etc. For real industrial applications, they are the most suitable solutions in terms of general applicability. Unfortunately, they have not been designed with computer vision applications in mind. On the contrary, they are often made or painted in such way that they are not easily popping out to the eye. They are therefore difficult targets to segment and to track because they are often partially occluded after site construction. And since they are not designed to be distinguishable (e.g. they are painted like the walls), their detection is complex. Furthermore, the images acquired on-site are often noisy, which does not ease the segmentation process. Figure 6.1 demonstrates typical anchor-plates' layout at different stages of the construction. These are the typical images we want to augment.

We developed an interactive method to segment them combining Canny edge detector [Canny, 1986] and Hough transform [Duda and Hart, 1972]. The advantage of anchor-plates is that their positions are almost unique within a room. This leads to an automatic 2D-3D matching procedure, performed with a homography estimation method.

6.2 Anchor-plates Segmentation

Despite their simple geometry, performing an automatic extraction of anchor-plates is far from being easy. In fact, anchor-plates are used as mounting points for many kinds of plant components. Consequently they are, in most of the cases, partially occluded. Therefore, in addition to their wide variety, their appearance is highly dependent on the attached object, on the viewpoint and on the illumination conditions. The techniques that work well for 2D marker detection, such as adaptive thresholding followed by an exhaustive search of closed quadrangles, do not permit to extract the Anchor-plates since their borders are often incomplete and not well contrasted.

We evaluated different approaches to automatically segment the anchor-plates and determine their borders but few methods provided acceptable results. The best results were obtained using a semi-assisted approach, where the user selects an area around an Anchor-Plate and then validates a choice among a set of segmentation candidates suggested by the system. The candidates are obtained in four steps:

1. A Canny edge detection [Canny, 1986] is performed in a user-specified area;
2. Using the obtained edges, a line detection based on a Hough transform [Duda and Hart, 1972] makes it possible to reconstruct incomplete borders;

[1] In some facilities, the so-called Dowel-Plates are used instead. They are similar to Anchor-plates in shape and function.

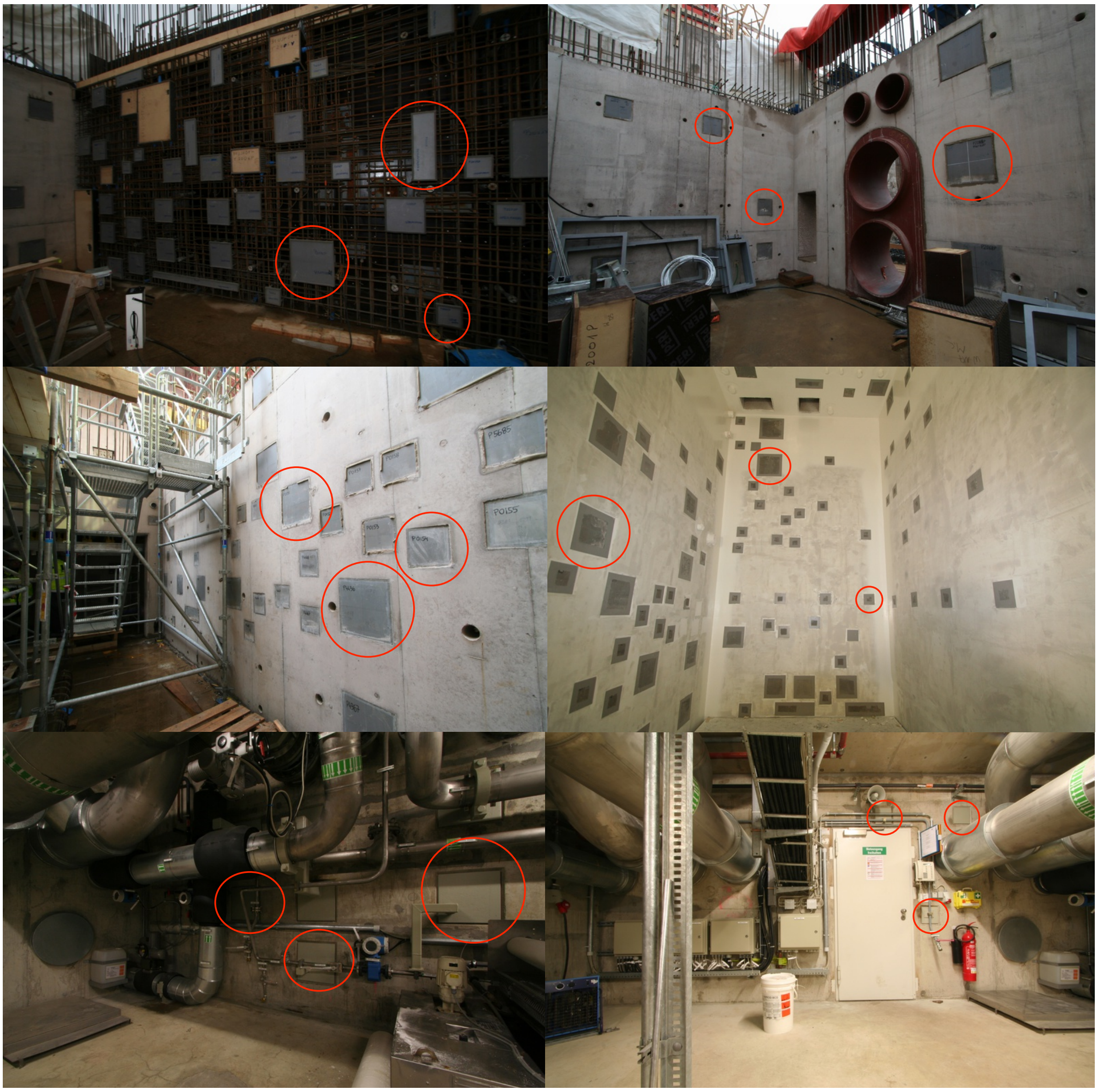


Figure 6.1: **Example of Anchor-Plates.** Anchor-plates are installed and welded to wall's steel armatures before the concrete is poured. After walls are often built the contour of the plates are not well distinguishable and when the plant is finished the plates are occluded by support and pipes. All this makes their segmentation hard.

3. All the quadrangles that can be obtained using the detected lines are formed;

4. All the formed quadrangle are ranked based on their appearance.

The score, used for ranking, is composed of constraints on the angles, the size, the relative length of the sides and the orientation of the quadrangles. It allows to reduce the number candidates to a small set that should contain the correct Anchor-Plate borders. Additionally to these geometric constraints, we also define a signature to each line based on the edge profile. Each opposite line is expected to have a similar signature. The choice of parameters to weight each criterion is estimated off-line using a set of manually segmented anchor-plates. This allows the system to be power-plant or system specific because all the rooms in such a facility are not finalized the same way. For example some rooms have painted walls when others are let untouched with the concrete directly visible.

This algorithm gives very good results due to the combination of the Canny edge detector and the Hough transform that permit to be robust to partial occlusions and to various illumination conditions. Once the sides are reconstructed, the constraints based on the general geometry and intensity information of the anchor-plates allow us to obtain the desired extraction as it can be seen in the Figure 6.2. The proposed algorithm is a good compromise between precision and speed because it is limited to a region of interest. This satisfies the targeted objective of a user-friendly software since the users only need to validate a choice among a small set of possible extractions.

6.3 Anchor-plates Matching

Once the anchor-plates are detected, they have to be matched with the 3D data in order to compute a pose. The layout of the anchor-plates on a wall is mostly unique, as shown in Figure 6.3, there are rarely two similar organizations in the same room. So it is possible to find the correspondences between the anchor-plates extracted from the image and multiple 3D candidates automatically. Since the detected anchor-plates all belong to the same plane in 3D, one can estimate a homography to validate the matching. Supposing that the correspondences between the anchor-plates in the image $\{\mathbf{c}^{ij}\}$ and the 3D ones $\{\mathbf{C}^{ij}\}$ are known, where i is the index of the anchor-plates and j the index of the corner from 1 to 4, there is a homography that defines this geometrical relationship. In both the image and the 3D world, their four corners coordinates represent fully an anchor-plate.

Given one 3D and one 2D anchor-plate, the correspondence between corners cannot be found. For a rectangular shape, there are 4 solutions (mirroring not included). This local orientation is given by the EXIF (Exchangeable Image File Format) rotation tag. This is provided by most of modern cameras to determine whether an image is a portrait or a landscape. In most images, there are few detectable anchor-plates as most of them are not in the field of view or too cluttered to be extracted. For each putative correspondences between the elements of $\{\mathbf{c}^{ij}\}$ and $\{\mathbf{C}^{ij}\}$, we compute a homography H_{ij}. This is done using the Direct Linear Transform (DLT) [Hartley and Zisserman, 2003] with all four correspondences.

In order for the results to be stable, we normalized the points (2D and 3D), as suggested in [Hartley, 1995]. Empirically, we found out that using a non-isotropic normalization

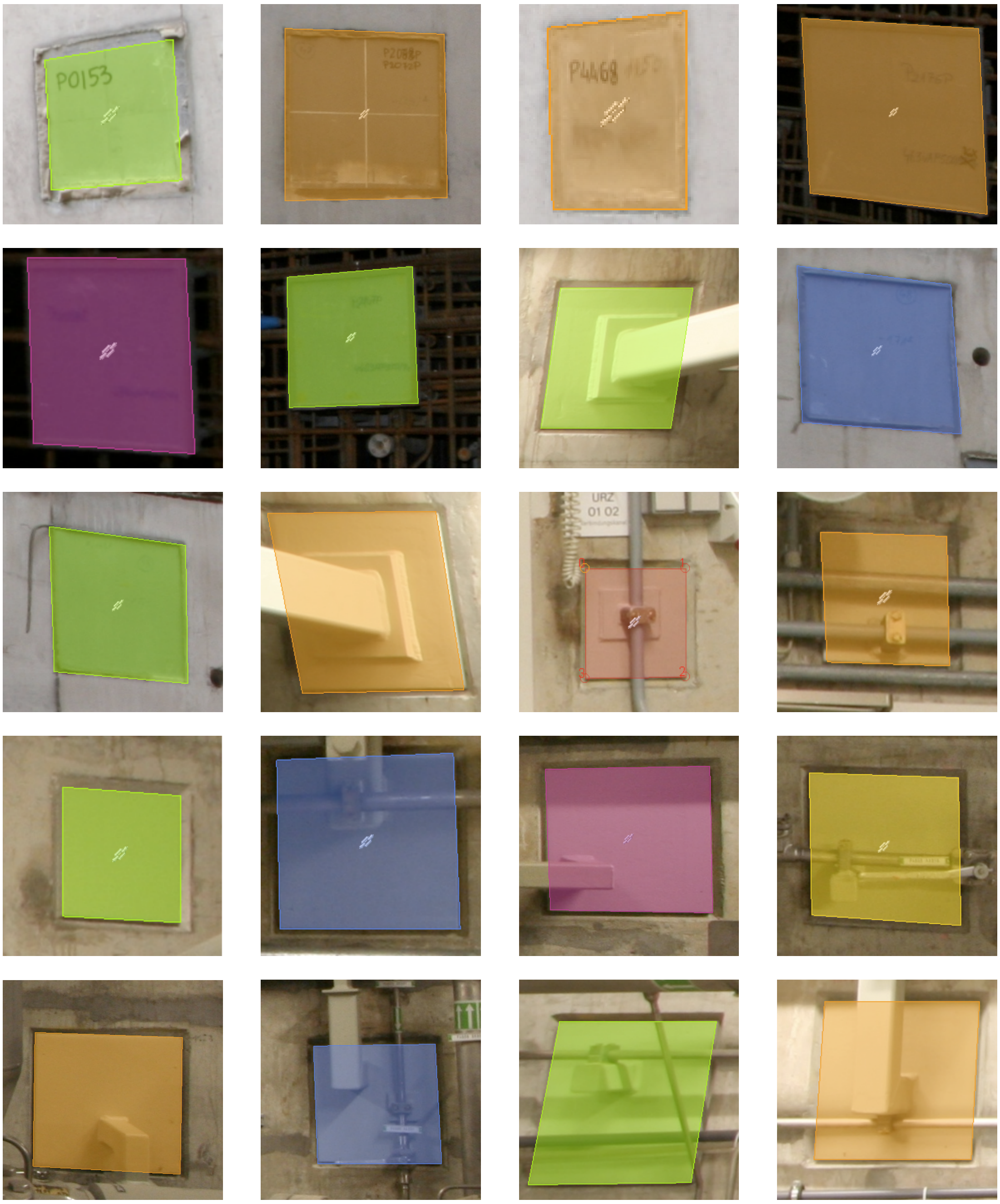
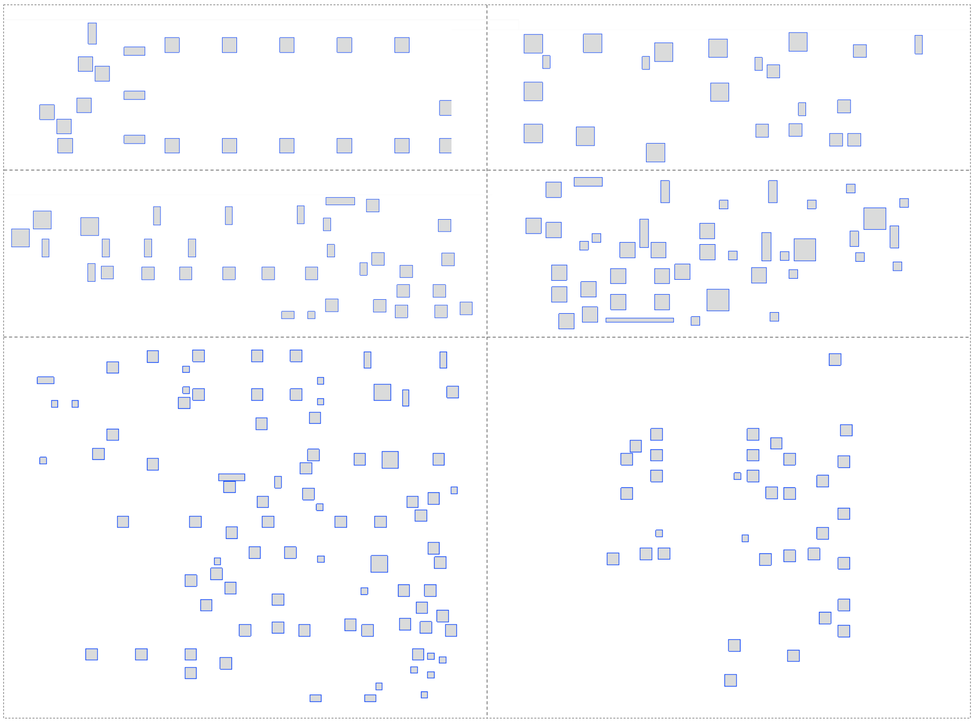
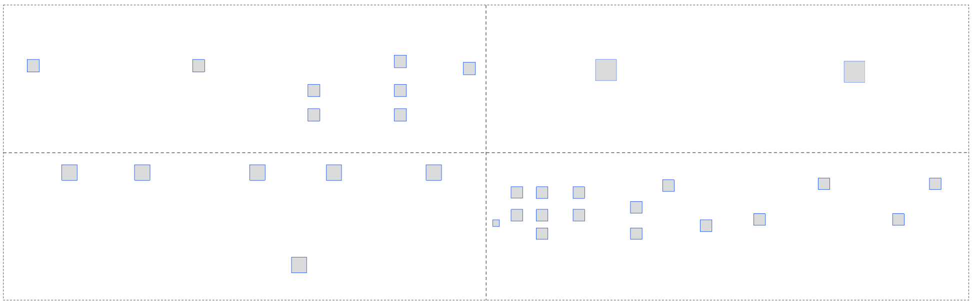


Figure 6.2: **Results of Anchor-Plate Segmentation**. These segmentations were obtained using a region of interest around the anchor-plates and were always within the ten first candidates proposed by the algorithm.



(a) Room from a plant during construction



(b) Room from a built plant

Figure 6.3: **Exemplary Blueprints of Anchor-Plates Layout**: (a) shows data from a power plant under construction including information for the floor and the ceiling (data-set 1); (b) represents the anchor-plates geometric organization within 4 walls of a room from a plant in use (data-set 2). Note how the geometric relation of two anchor-plates is often unique within a room.

was more robust. That is maybe because our points are often spread in one dimension (horizontally) as shown in Figure 6.1. During the normalization the dimensionality of the 3D points is reduced to 2. Projecting the 3D points onto the wall coordinate system.

For each putative correspondence between 3D candidates and 2D segmentation we have an estimate H. This can be estimated efficiently as it requires only one DLT for each 2D segmentation and the actual homography is composed using the 3D candidates location with respect to a unit reference square. We use a mean re-projection error in order to grade the quality of each homography. The right matching between the 2D set and the 3D should minimize the following formula:

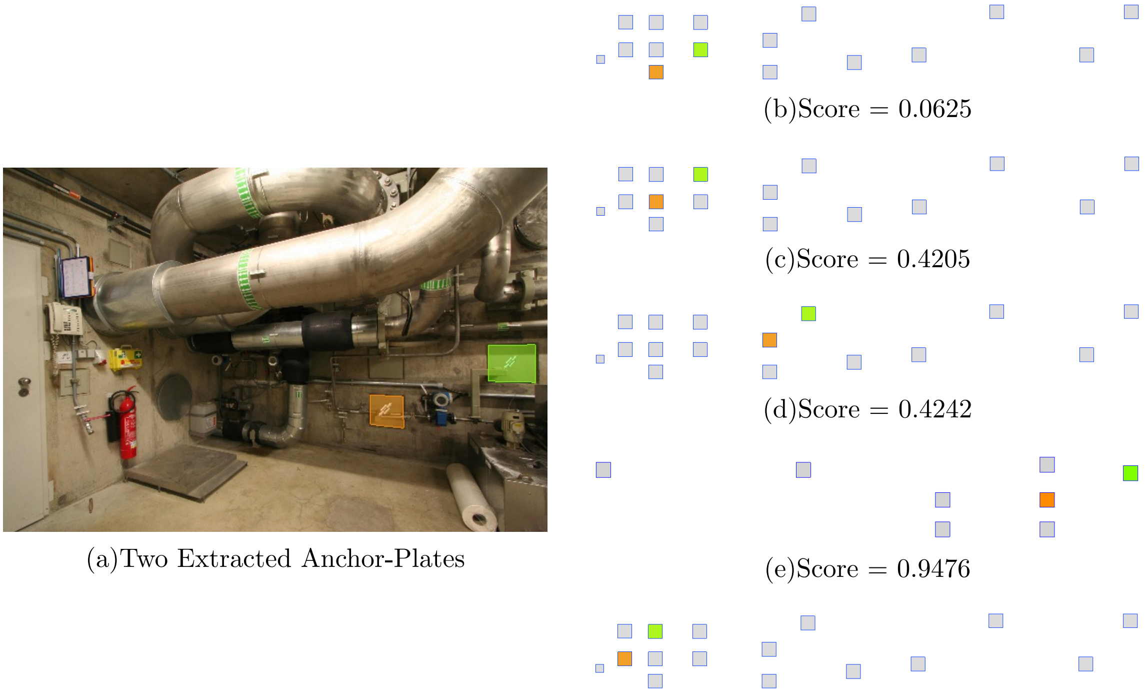
$$e\left(\mathsf{H}\right) = \frac{1}{4m} \sum_{i=1}^{m} \sum_{j=1}^{4} \left\| \mathbf{c}^{ij} - \mathbf{w}\left(\mathsf{H}\mathbf{C}^{ij}\right) \right\|^{2} \ , \tag{6.1}$$

where $\mathbf{c}^{ij}$ is the j^{th} corner of the i^{th} 2D anchor-plate, $\mathbf{C}^{ij}$ is the 3D point related to the j^{th} 2D corner of the corresponding i^{th} anchor-plate and m is the number of anchor-plates segmented. The 2D segmentations that were not used to estimate H are matched such that they minimized 6.1.

We then refine the result by re-normalizing the 3D points once matching have been obtained. We normalize these points only using the matched points. This gives a more comparable re-projection error. In case of multiple answers (i.e. several putative matches which have an error $e\left(\mathsf{H}\right)$ under a threshold) the user has to decide which one is correct within a subset of 5 possibilities. Since the same layout of the anchor-plates on a wall is rarely reproduced, anchor-plates from all walls are used as input. This eases the workflow since no walls have to be selected. For pictures of the ceiling where the EXIF rotation tag can not be used to determine the local orientation (landscape or portrait), we supply additional inputs to the matching algorithm. These additional inputs are generated by rotation of the order within each Anchor-Plate leading to 4 additional inputs for the algorithm. Typically, there are 4 walls, 4 floors and 4 ceilings as input for the matching algorithm for a 4-corners room; the floor and ceiling are duplicated with 4 different local orientations.

Figure 6.4 shows the behavior of the ranking procedure on a image acquired in the room represented from data-set 1 described in Figure 6.3(b). For this example, two anchor-plates were extracted in the image and they were matched to the 3D model. This room's reality model is composed of 32 anchor-plates distributed over 4 walls.

When all the points are matched, the camera motion T is estimated by minimizing the re-projection error using a Levenberg-Marquardt optimizer. This process returns the camera pose in the coordinate system of the 3D model. This allows us to create an augmented CAD by positioning the image into the 3D view. These transformations are stored in the database and linked to the image.


(a)Two Extracted Anchor-Plates

(b)Score = 0.0625

(c)Score = 0.4205

(d)Score = 0.4242

(e)Score = 0.9476

(f)Score = 0.9579

Figure 6.4: **Ranking of Anchor-Plates Matches**: Matches are ranked using a normalized registration error; We see that the five best candidates have similar geometric organization, but the best candidate has a score, which is clearly separated from the others.

6.4 Registration Results

We present now the results of the presented approach using the two data-sets presented in Figure 6.3. For each image we extracted the anchor-plates using our segmentation tools. Then we apply the matching procedure for the first data-set (Fig. 6.3(a)), which includes more than two hundred anchor-plates in the model, and for the second data-set (Fig. 6.3(b)) which includes thirty-two anchor-plates. For both experiments the matching procedure found the correct correspondences between the 2D information and the 3D model even though the 2D segmentation might not always be perfect. Matching results are presented in figures 6.5, 6.7, 6.9 and 6.11. Using the estimated 2D-3D correspondences we compute the pose of the camera and then create a mixed view that composes the image with the CAD model. Augmentation results are visible in Figures 6.6, 6.8, 6.10 and 6.12. The obtained augmentations are visually satisfying and offer a sufficient quality to perform discrepancy check.

Conclusion

In this chapter, we presented a tool for the registering of an image to a CAD model. The presented method follows the philosophy of *Reality Models*, that are 3D model for AR. We present anchor-plates, which have the desired property of reality models in civil engineering. These industrial components are used largely in civil works and their geometry are verified, which make them trustworthy for registration. Because their detection is difficult we presented a segmentation method that is able to propose candidate extractions to the user. And we introduced an automatic method for 2D-3D matching of this industrial object leading to a reliable pose estimate. These registered images can be used by civil engineers to check for discrepancies but they also create a better documentation of the CAD model that include discrepancies, un-documented components, modification of the design. They can be, in addition, considered as *keyframes*.

These keyframes can help to create new applications such as on-site maintenance using visual tracking technologies. It can also provide new methods to register images from the plant, which could overcome some of the limitation of component-based registration. The presented method requires for at least two anchor-plates to be visible to obtain a pose. This constraint is hard to enforce. But, using local images features and pose estimation from features, one can use the keyframe created with the anchor-plate-based procedure to register additional images. Local features, relative pose estimation and keyframe-based registration will be the focus of the rest of this thesis.

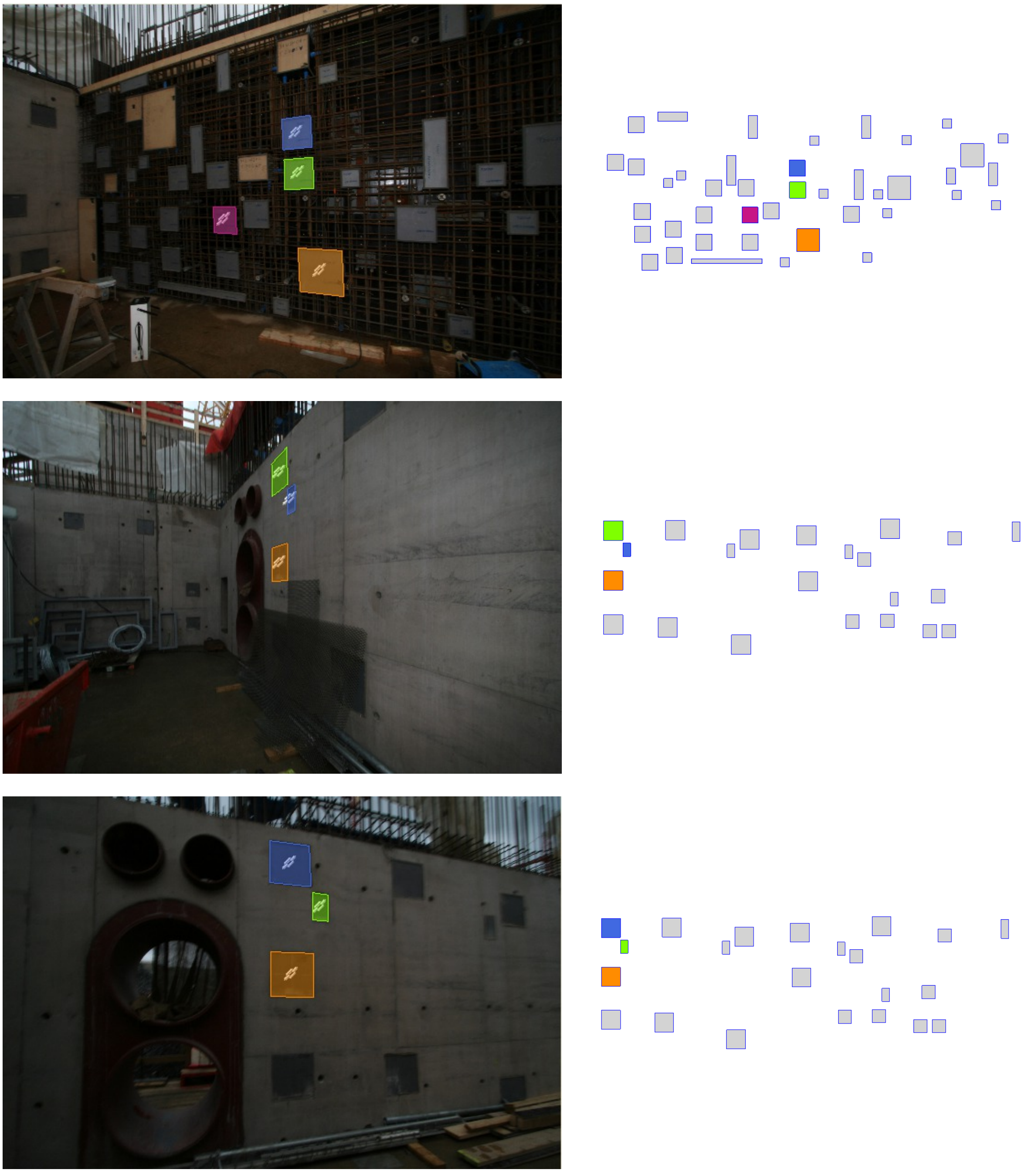


Figure 6.5: **Anchor-Plates Matching Results 1** using images of a plant during construction. The algorithm used 3D information from the 4 walls, the ceiling and the floor (Data-set 1 - Figure 6.3(a)) and segmentation provided from 6.2. It automatically detected the correct wall and matched to the correct structure.

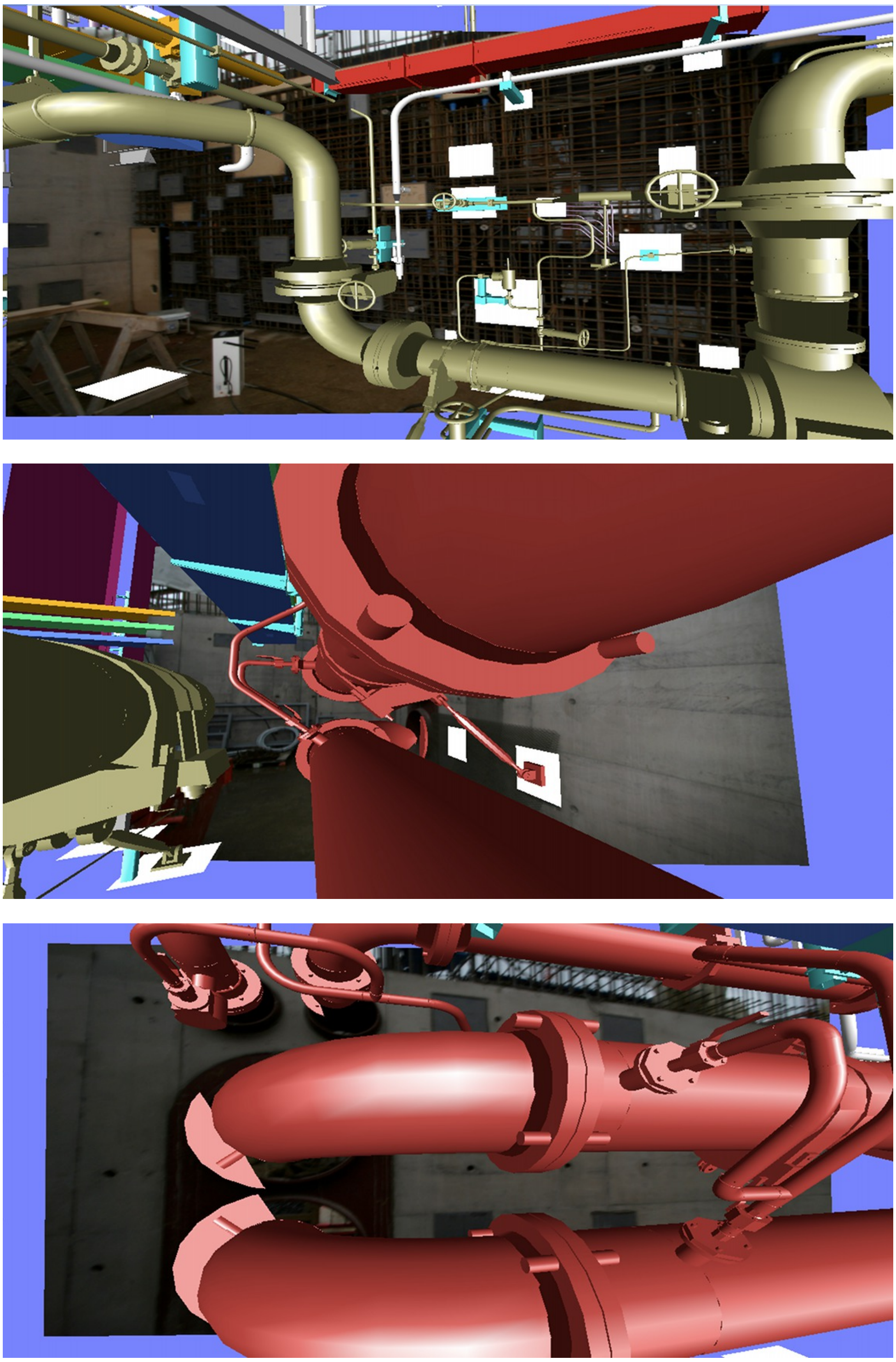


Figure 6.6: **Augmentation of Power-plant Images 1**, the pose was obtained using the 2D-3D correspondences displayed in figure 6.5. Snapshots obtained directly from VID.

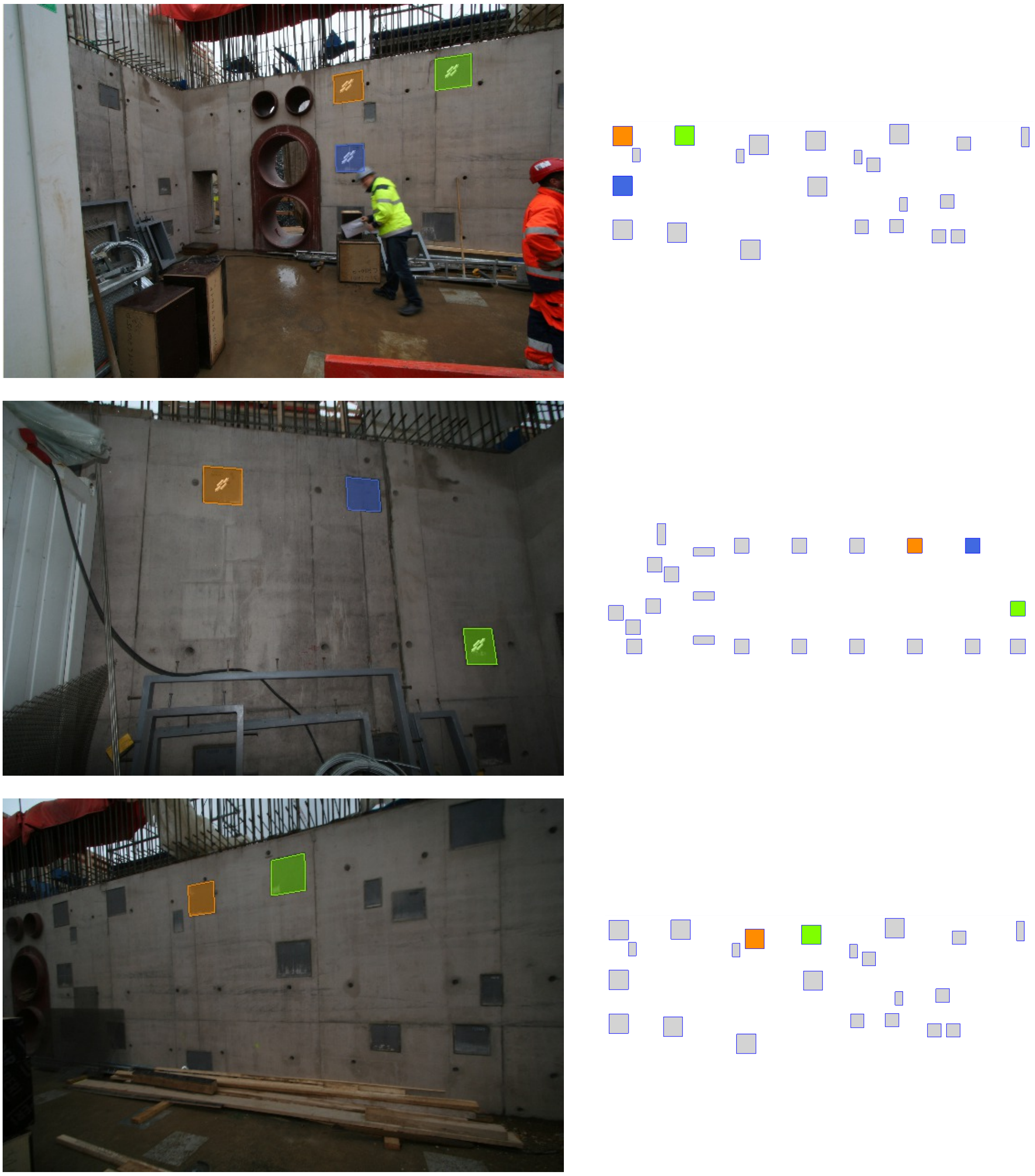


Figure 6.7: **Anchor-Plates Matching Results 2** using images of a plant during construction. The algorithm used 3D information from the 4 walls, the ceiling and the floor (Data-set 1 - Figure 6.3(a)) and segmentation provided from 6.2. It automatically detected the correct wall and matched to the correct structure.

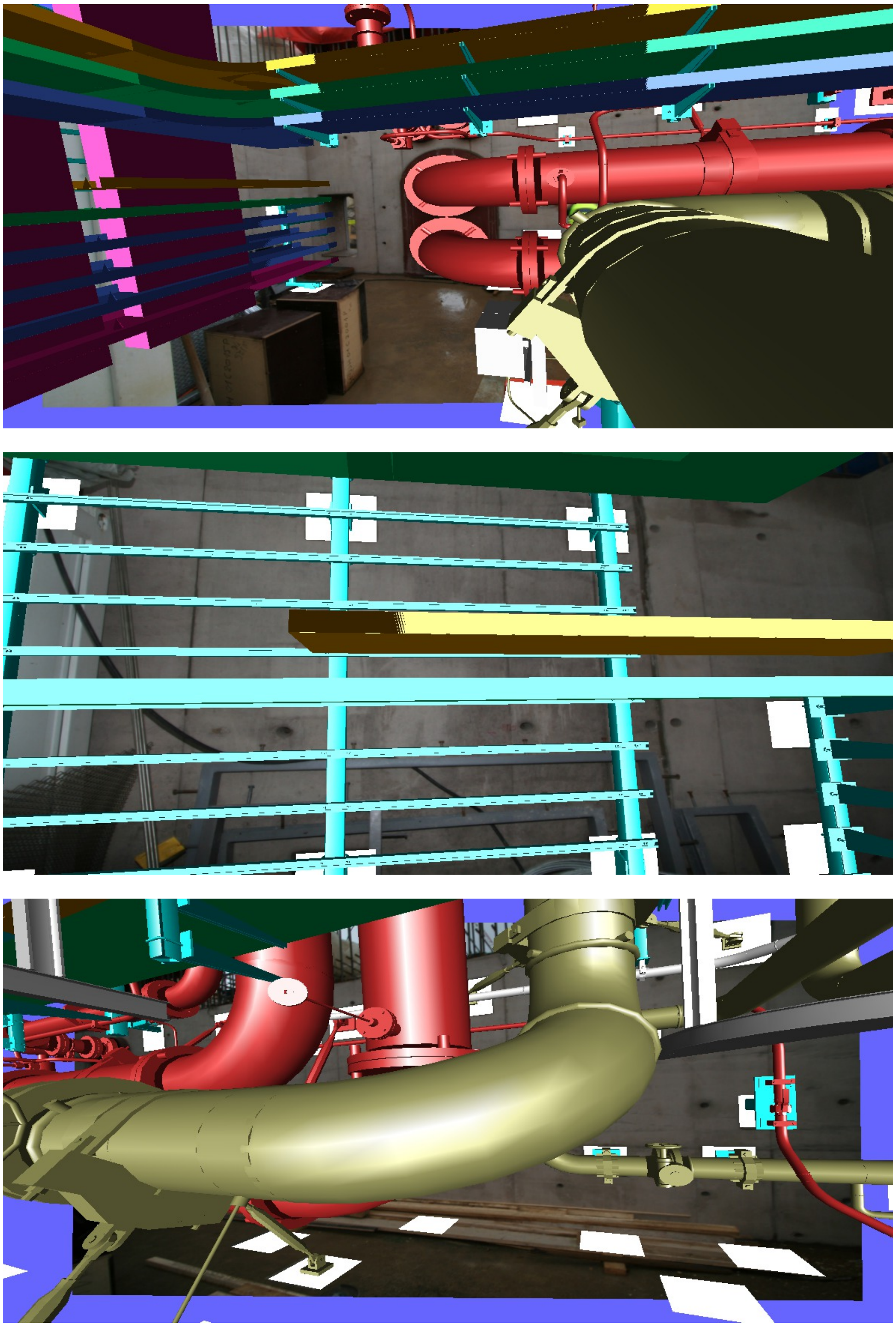


Figure 6.8: **Augmentation of Power-plant Images 2**, the pose was obtained using the 2D-3D correspondences displayed in figure 6.7. Snapshots obtained directly from VID.

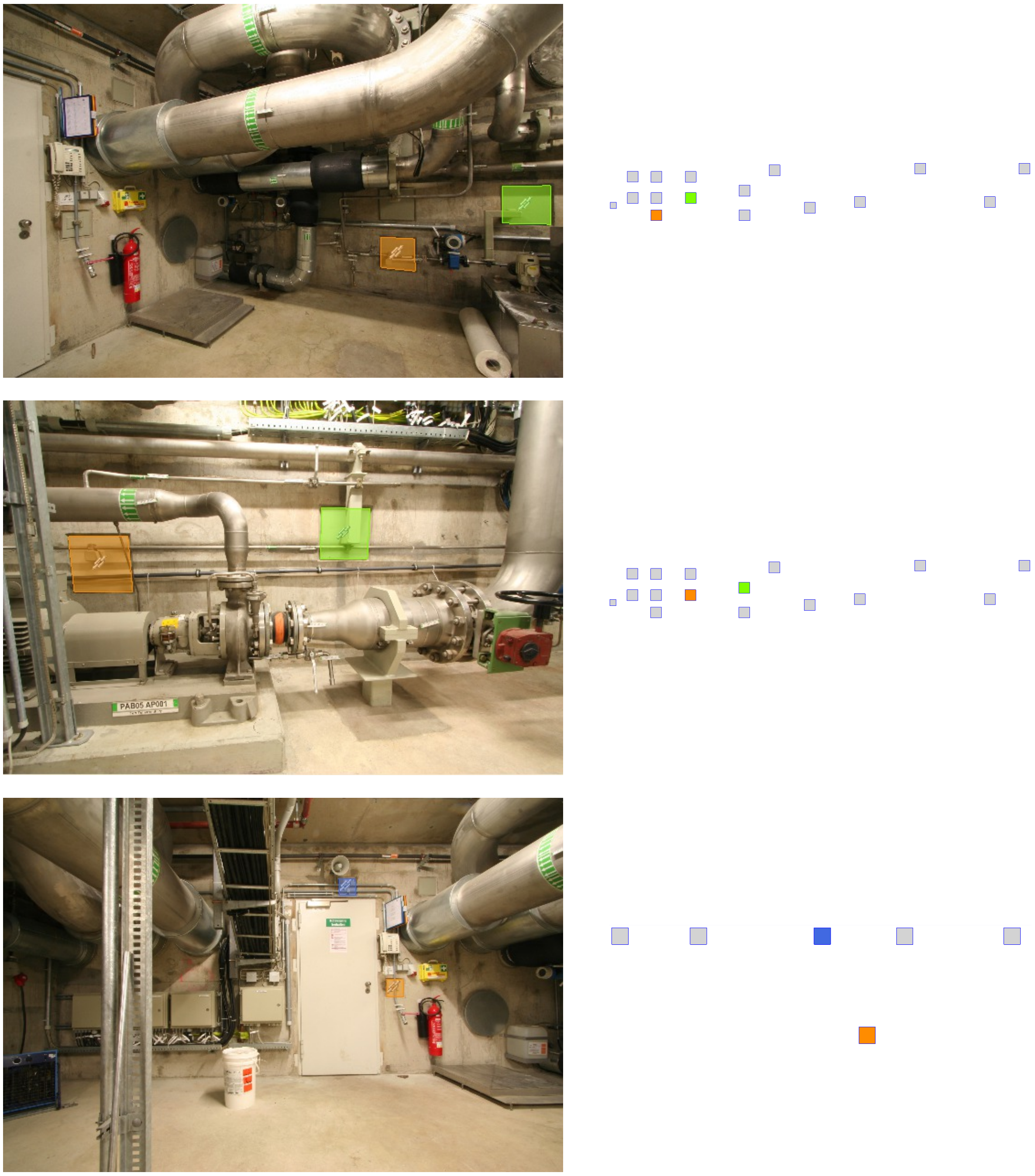


Figure 6.9: **Anchor-Plates Matching Results 3** using images of a plant in operation. The algorithm used 3D information from the 4 walls (Data-set 2 - Figure 6.3(b)) and segmentation provided from 6.2. It automatically detected the correct wall and matched to the correct structure.

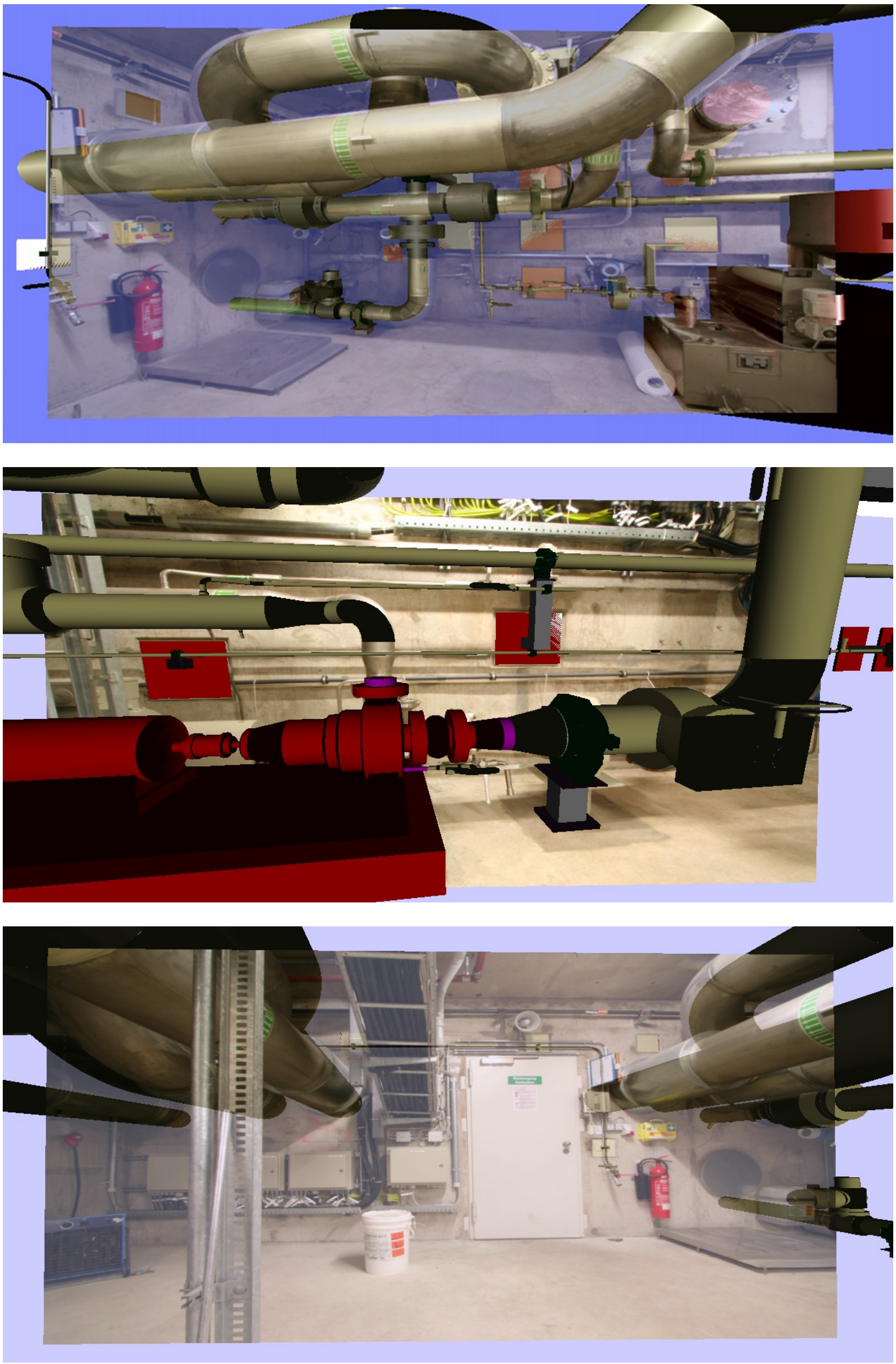


Figure 6.10: **Augmentation of Power-plant Images 3**, the pose was obtained using the 2D-3D correspondences displayed in figure 6.9. Snapshots obtained directly from VID.

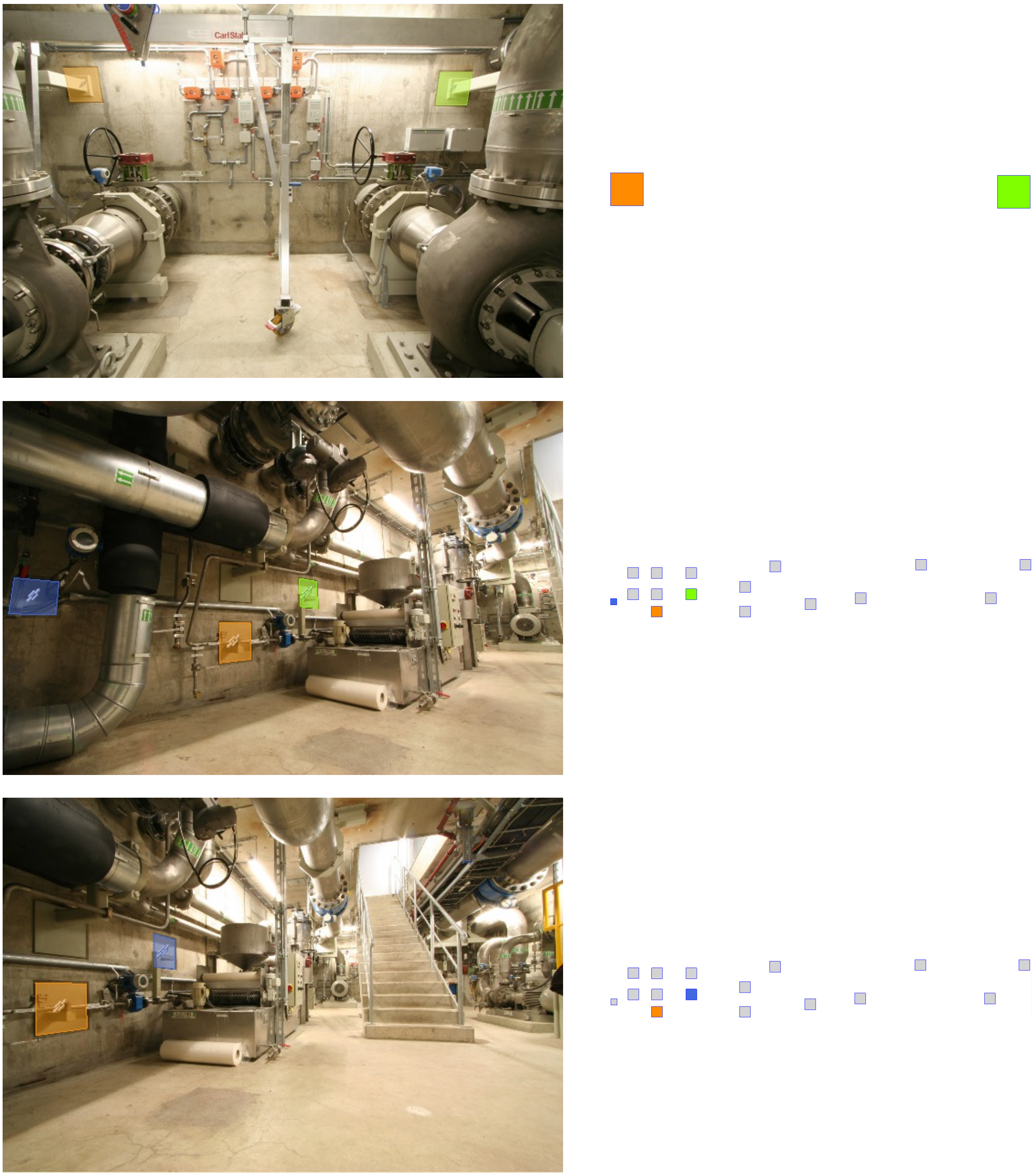


Figure 6.11: **Anchor-Plates Matching Results 4** using images of a plant in operation. The algorithm used 3D information from the 4 walls (Data-set 2 - Figure 6.3(b)) and segmentation provided from 6.2. It automatically detected the correct wall and matched to the correct structure.

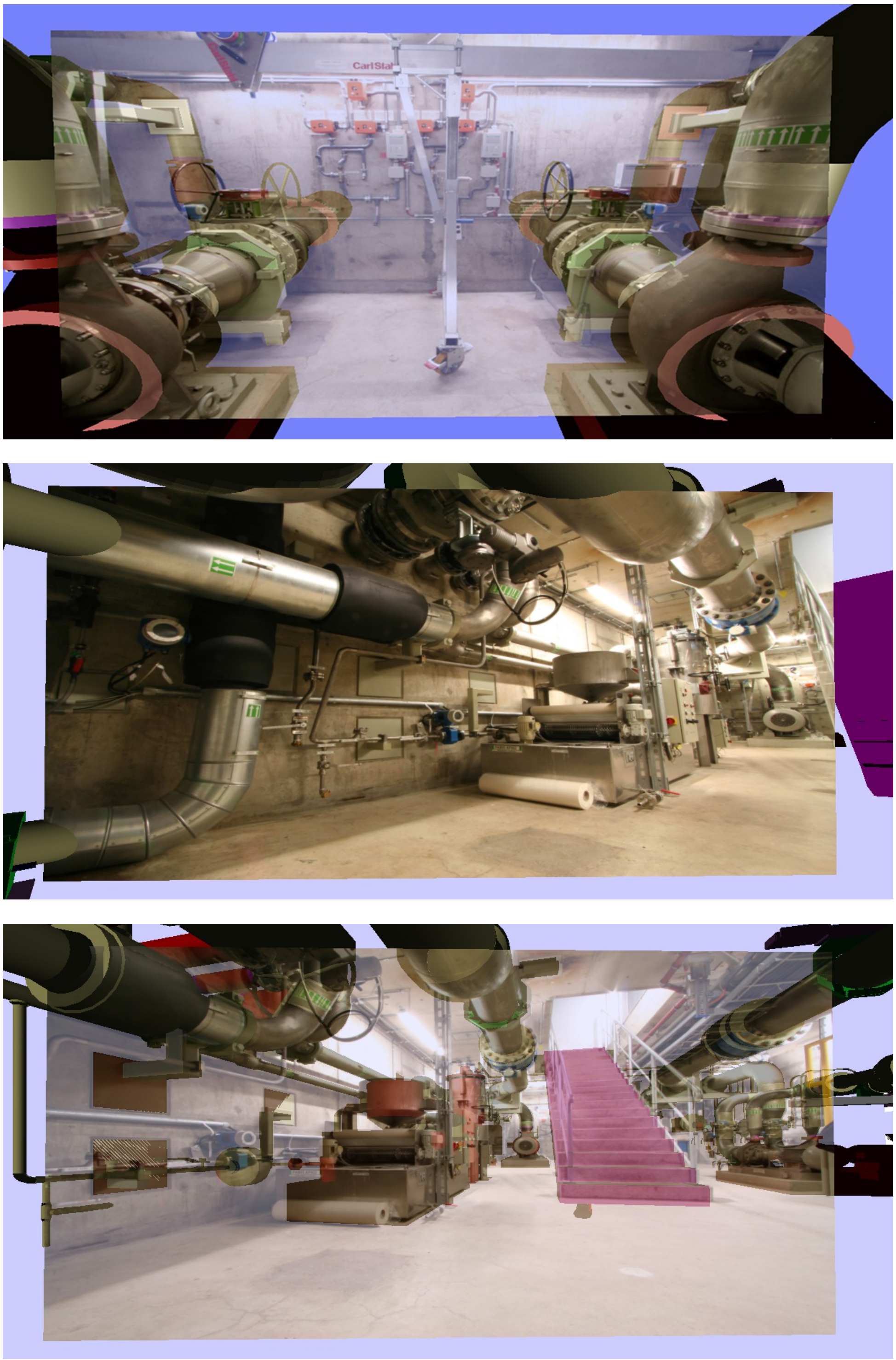


Figure 6.12: **Augmentation of Power-plant Images 4**, the pose was obtained using the 2D-3D correspondences displayed in figure 6.11. Snapshots obtained directly from VID.

Part III

Advances in Feature based Registration for Augmented Reality

In the previous part of this thesis, we introduced an Augmented Reality Application for *discrepancy check*, which includes a method to register images to a CAD Model. In this part we develop a pipeline to register new images based on these registered images. Since we want this registration method to be applicable in general settings, it has to work with wide-baseline. Therefore it relies on multi-scale features to estimate an initial relative pose. We first introduce a method to measure the localization quality for multi-scale features in Chapter 7. Then in Chapter 8 we present a method to estimate relative pose, which is able to cope with bad measurements by evaluating the quality of an alignment on both 2D coordinates and images information. Finally in Chapter 9, we present a general method to extend a relative pose to a full pose necessary to obtain an mixed view.

CHAPTER

SEVEN

LOCATION UNCERTAINTY FOR SCALE INVARIANT FEATURE POINTS

Image feature points are the basis for numerous computer vision tasks. These points usually are representatives of some images characteristics such as corners or blobs. When put in correspondence across images, they can be used for pose estimation or object detection. State of the art algorithms detect features that are invariant to scale and orientation changes. While feature detectors and descriptors have been widely studied in terms of stability and repeatability, their localization error has often been assumed to be uniform and insignificant.

We argue in this chapter that this assumption does not hold for scale-invariant feature detectors and demonstrate that the detection of features at different image scales actually has an influence on the localization accuracy. We introduce a general framework to determine the uncertainty of multi-scale image features. This uncertainty is represented via an anisotropic covariance with varying orientation and magnitude. We apply our framework to the well-known SIFT and SURF algorithms. Finally, after demonstrating in synthetic experiments that the presented framework behaves appropriately, we show the usefulness of such covariance estimates for bundle adjustment.

7.1 Problematic

One of the core tasks in computer vision is to discover the movement undertaken by a camera between two acquisitions. Applications where the knowledge of relative transformations between images is useful range from movie special effects to medical imaging. They use techniques such as structure from motion [Fitzgibbon, 2001], tracking [Atasoy et al., 2009] or image enhancement [Capel and Zisserman, 1998]. In the context of this thesis, a relative pose between images can be used to extend it to a full pose in order to obtain a mixed view (see Chapter 9). Solutions for the relative registration problem can be cast in two categories, as explained bellow.

In the first category, we find direct or template based approaches where pixel intensity values are directly compared to each other [Baker and Matthews, 2004, Bartoli, 2006, Benhimane, 2007]. Direct methods offer good performances. But because of their limited basins of convergence, they require to be properly initialized and therefore are mainly used for tracking.

In the second category, we find methods that are based on image primitives. They are called feature based methods. In this thesis, we focus on points, but primitives such as lines [Hanek et al., 1999] or edges [Klein and Drummond, 2003] are also suitable. These kinds of algorithms first need to extract meaningful and stable points based on mathematical operators and then describe them in a distinctive way in order to offer a method to match the local features across views. Local features points should have the following properties : Distinctness, Invariance, Stability, Seldomness and Interpretability [Förstner, 1987].

In particular, a lot of attention has been given to detecting points that could be detected and matched across wide baseline [Lindeberg, 1998, Matas et al., 2002, Lowe, 2004, Triggs, 2004, Bay et al., 2008]. These features are often detected in scale-space and therefore are referred to as multi-scale local features. Methods based on features have been successfully applied in numerous fields. A few of these application are: scene modeling [Snavely et al., 2008], 3D tracking [Vacchetti et al., 2004a] and image retrieval [Lazebnik et al., 2006].

Features extraction is an estimation process and like every numerical algorithm it makes approximations. These approximations or errors need to be evaluated as they are used for further calculus (e.g. pose estimation) and might contaminate the following estimates.

The use of multi-scale features has exploded since the publication of SIFT [Lowe, 1999, 2004]. SIFT have been used in countless applications. To give an idea of the impact it had on the community, this paper has been cited at least 7114 times[1] almost twice as much as the seminal RANSAC paper [Fischler and Bolles, 1981]. Unfortunately, few are those [Haja et al., 2008] who studied the precision of multi-scale features or to be more exact their un-precision. As of today there is still no theoretical framework to estimate it. Furthermore, countless methods provide the methodology to use or require uncertainty measurements: for linear and non-linear estimation of vision parameters [Förstner, 1987, Kanatani, 2000], for robust parameter estimation [Sur et al., 2008, Raguram et al., 2009], for guided matching [Ochoa and Belongie, 2006], for 3D tracking [Park et al., 2008]. Since they do not have access to such measurements they either suppose the error to be uniform across measurements or are forced to use uncertainty based on the image auto-correlation, suitable for mono-scale corners such as Harris'. Finally, when available, covariance information will allow merging heterogeneous data, for example points and lines [Morris and Kanade, 1998]

The general framework presented in this chapter can be seen as a meticulous application of the guideline proposed for error estimation of computer vision algorithm in [Haralick, 1994]. For an overview of multi-scale features and notation used in this chapter the reader is reffered to section 3.7. First in Section 7.2, we describe prior works in uncertainty estimation, followed by a description of the approximation made by scale invariant feature detection methods. Then we introduce in Section 7.3 our framework to express

[1] Search performed using google scholar June 2010.

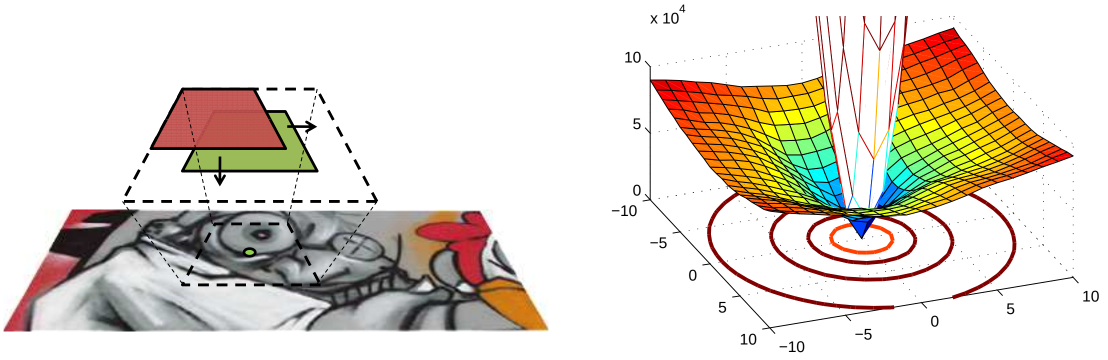


Figure 7.1: **Self Matching Function** of an image patch around a found feature point (left). Residual of the self-matching and its 2$^{\text{nd}}$ order approximation (right). The ellipse indicates the covariance describing the localization uncertainty.

the uncertainty related to the curvature of detector function for multi-scale features. In Section 7.5, we apply this framework to SIFT and SURF and evaluate our framework in Section 7.6. Finally in Section 7.7, we demonstrate the usefulness of the estimated uncertainty in model fitting.

7.2 Related Work in Uncertainty Estimation

Multi-scale local features have been studied extensively in term of repeatability and stability [Schmid et al., 2000, Mikolajczyk et al., 2005]. But we are still missing a measure of their localization precision. This problem has been tackled for *interest points* (e.g. corners). And with this section, we try to give a complete overview of different methods used in the past.

Common to all approaches is the assumption of a Gaussian error model and hence the characterization of the location uncertainty using a 2D covariance matrix often visualized as an ellipse. To be able to give an accurate covariance estimate, it is important to know where the localization error originates from. It is identified either to be pixel intensity noise or to arise from the detection algorithm itself.

On the one hand, pixel intensity noise is derived from the capturing process and influenced by image sensor noise and the inherent sampling. On the other hand, while there are a number of different algorithms for feature point extraction, all of them have in common that they sample in image- and (if applicable) in scale-space and that they approximate the particular detector response. This can lead to a localization shift arising from the respective detection method.

One of the first detection algorithm to offer a precision measure was proposed in [Förstner and Gülch, 1987]. Their detection is a two step process: first we find optimal windows where good features lie and we then search where the feature is in that window, such that it maximizes a linear operator. An uncertainty measurement is attached to each feature based on the inverse of the normal equation system that the coordinates of the feature satisfy.

Most of the prior work in features localization uncertainty assumed that the error can be estimated from the inverse of the Hessian of the auto-correlation function. It supposes that a corner-like feature (e.g. Harris) maximizes the auto-correlation function (self similarity) [Morris and Kanade, 1998, Kanatani, 2000, Kanazawa and Kanatani, 2003], see also figure 7.1. The use of the inverse of the Hessian is justified in [Kanazawa and Kanatani, 2003] as it achieves the *Cramer-Roa lower bound*. Brooks et al. [2001] use a similar approach but propose to flip by 90° the obtained covariance and show that they obtained better results using such a trick for model estimation. This would imply that the orientation estimated from the auto-correlation is not reliable or of little impact. This measure is also used for KLT features [Lucas and Kanade, 1981, Tomasi and Kanade, 1991] by Dorini and Goldenstein [2006].

The auto-correlation based uncertainty is also used for intensity based alignment [Anandan and Irani, 2002, Koeser and Koch, 2008]. It computes covariances for a dense set of pixels. This “reliability” idea was sketched in [Shi and Tomasi, 1994] where “good feature to track” are points that have a small residual which enforces to keep good tracks. Covariances were employed in the same framework to infer the precision of the tracks in [Nickels and Hutchinson, 2002]. This time the covariance is not based on the auto-correlation but on the correlation between images. This was similarly used in [Skoglund and Felsberg, 2007] for the sum of absolute differences (SAD).

Steele and Jaynes [2005] on the other hand focus on the detector function and address the problem of its response inaccuracy based on pixel noise. For each pixel in the image they define a noise model. The evaluated noise models are: identically distributed noise, independent variably distributed noise where each pixel can have its own variance, and correlated variably distributed noise where the noise at connected neighbors for a single pixel is correlated. Instead of estimating covariances directly from pixel intensity values, they propagate the initial noise covariances through the detection process of the Förstner-corner detector [Förstner and Gülch, 1987] to come up with a covariance estimate for each feature point. The simplest noise model delivers the worse results in comparison to more complex models, but is still outperforming the traditional method, where covariances are estimated directly from pixel intensity values not considering any noise model. They summarize that, in general, more sophisticated noise models lead to improvements in the covariance estimation. While covariance directionality is estimated correctly, it tends to underestimate feature uncertainty.

For image flow, Singh and Allen [1992] estimate of the velocity of the image point along with a covariance measure. Compared to previous body of work they based they confidence on the density of the response distribution. It should be able to handle multiple local minima but the hypothesis that the response of the detector is independent does not match a common image.

Orguner and Gustafsson [2007] also study the accuracy for Harris corners. The measure is based on the probability that a pixel is a true corner in the region around the corner estimate. They have found that the accuracy for a corner point can vary depending on the different image color channels (RGB). The justification for the extra-computational burden of such a method is unclear in comparison to the inverse of the Hessian.

Kanatani has been working on uncertainty modeling intensively [Kanatani and Morris, 2001, Kanatani, 2004]. Thus [Kanazawa and Kanatani, 2003] raise the question of the usefulness of covariance matrices for image features. They showed that when using corner-like feature, the advantage of using covariance estimated from the auto-correlation function for homography and fundamental matrix estimation was limited. They state that the estimated covariances seem to be isotropic and of similar size across the image. In comparison [Brooks et al., 2001] demonstrate an error reduction for fundamental matrix estimation. They concentrate on various tests to analyze the value of covariance information in a parameter estimation process. Given that the covariance information is itself subject to estimation errors, they measure the impact of imprecise covariances parameter estimates. They conclude that not only covariance information itself can be valuable, but also the extent to which this information may be inaccurate until they corrupt the model computed using them gets degraded.

It is important to note that all the previously cited work based their argumentation on corner detectors which are *not* scale-invariant. Only [Haja et al., 2008] provide a comparison of region detectors with respect to localization accuracy. They argue that localization accuracy is dependent on the particular detection scale. Localization accuracy is evaluated in terms of matching precision of regions by examining the feature point location and region overlap; however, they do not parametrize the localization error of an interest point itself. They state that significant differences between detectors exist, depending on the type of images used. The presented results serve as an additional evaluation to existing studies, and can be used to choose the appropriate detector for a desired target application.

This brief review of the state of the art demonstrates the limit of the proposed methods, which are either too specific or not backed up by convincing evaluations. In this chapter, we will try to show that the covariance estimate follow the same error as the detector *and* prove that it offers real benefits. In order for the presented method to be useful we need a general framework that can be applied generally to detectors. Finally, as scale-invariant region detectors extract image regions complementary to the corner-like features, we claim two things:

1. Due to the focus on interest regions instead of points the shape of covariances will be in general anisotropic.

2. The magnitude of the covariances will vary significantly due to detection in scale space.

7.3 Formalization of the Error and Scale Space Detection

The following analysis is based on the underlying assumption that a detection process locates a feature and that this process generates a measurement error that conforms to a

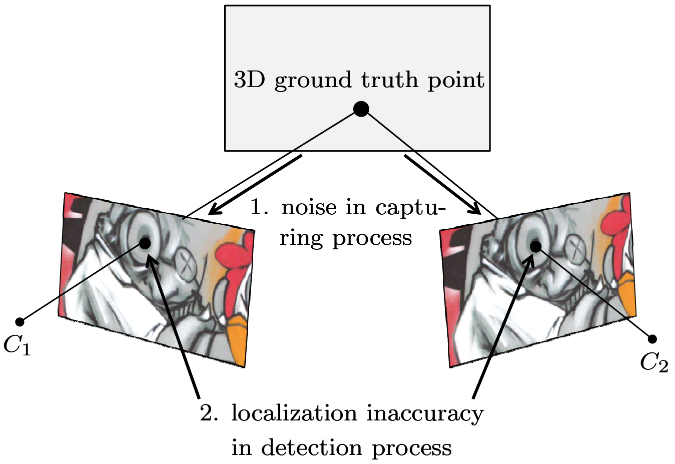

3D ground truth point
1. noise in capturing process
C_1
C_2
2. localization inaccuracy in detection process

Figure 7.2: **Error sources for a wrong localization of interest points**. The ground truth point is assumed to be known in the 3D scene. The capturing process, mapping the 3D point into a 2D image coordinate system, introduces noise in the pixel intensity values. In addition, the interest point detection process itself has an inherent error depending on the particular algorithm. Both sources of error account for the interest point localization error in an image.

bivariate normal distribution. We present the error assumption and a general representation of scale space feature detection. In this section the general uncertainty evaluation framework is explained, which was developed to estimate the covariance matrix describing this distribution.

7.3.1 Source of Uncertainty and Error Model

In the following explanation we assume that the ground truth location of a 3D point in the scene related to an interest point in the image is known. The error when localizing an interest point is then identified to occur due to two reasons (see also Figure 7.2):

First, the capturing process maps a 3D point into an image coordinate system. The accuracy of this mapping is dependent on the resolution and color depth of the camera. Discretization of pixel locations and a quantization of intensity values modify the true point in this first step. In addition the camera sensor will introduce pixel intensity noise. Thus, even if we have a perfect detection algorithm it is impossible to estimate a feature point location exactly.

Second, the detection algorithm itself introduces some localization error. The representation of the operator response within an image stack is done at predefined scales. Hence, the scale space is not represented continuously. Most algorithms limit the effects of this problem by interpolating in scale space to get a more accurate interest point estimate both in spatial location and scale. For an approximation of order n, the error introduced by this means will correspond to the error residual of the interpolation and be of order $n + 1$. Additionally, detection operator functions are approximated for faster computation, although this has the drawback that an interest point will not be found at its ground truth position but with a small offset.

For some detection algorithms it is possible to propagate individual pixel intensity error covariances through the detection process to come up with a covariance matrix for a found interest point [Haralick, 1994]. Steele and Jaynes [2005] have demonstrated this approach for the Förstner-corner detector. For the propagation of the error, the Jacobian of the detector with respect to pixel intensity values is needed. For more complicated and especially non-linear interest point detectors, the Jacobian can only be approximated. Thus this approach has not been considered for detectors searching in scale space.

In this work, the statistical model for the observable localization error is set to be multivariate Gaussian distributed.

7.3.2 General Formulation of Detection in Scale Space

The uncertainty estimation framework we developed is not only applicable to one particular feature detection algorithm, but is valid for all detectors building upon a representation in scale-space. Common to all these scale invariant feature detectors is a two step approach to find feature points. The following mathematical representation is novel and allows for an easy adaption of our estimation framework to other detectors.
First, a scale-space representation in form of an image stack D (see Figure 7.3-left) is created with the mathematical *feature detection operator* ($f_{\mathbf{dec}}$) at preselected scales $\sigma_i \in \{\sigma_j\}_{j=1\ldots N}$ from the image $\mathcal{I}$. N is the number of layer present in the stack and σ_i refers to the actual scale the layer. The detection operator $f_{\mathbf{dec}}$ depends on the particular algorithm and is not necessarily a linear function. For the calculation of the operator response $D(\mathbf{m}, \sigma_i)$ at a specific location in scale space $(\mathbf{m}, \sigma_i)$, the image neighborhood $\mathcal{N}_{\mathbf{m}}$ is taken into consideration. For each layer $D(\bullet, \sigma_i)$ of the stack, local maxima are then detected at positions $\mathbf{m}$ via a non-maximum suppression approach, leading to a first set $\mathbb{P}_1$ of feature point candidates:

$$D(\mathbf{m}, \sigma_i) := f_{\mathbf{dec}}\left(\mathcal{I}(\mathcal{N}_{\mathbf{m}}), \sigma_i\right) \tag{7.1}$$

$$\mathbb{P}_1 := \bigcup_{i=1}^{N} \left\{ \langle \mathbf{m}, \sigma_i \rangle \text{ s.t. } \forall \mathbf{x} \in \mathcal{N}_{\mathbf{m}} \setminus \{\mathbf{m}\}, \, D(\mathbf{m}, \sigma_i) > D(\mathbf{x}, \sigma_i) \right\} \tag{7.2}$$

Second, the algorithm selects interest points from $\mathbb{P}_1$ for which the response S to the *scale-selection operator* $f_{\mathbf{sel}}$ attains a local maximum over scale (see Figure 7.3-left). Points for which the scale-selection operator attains no extremum or for which the response is below a threshold τ are rejected:

$$S(\mathbf{m}, \sigma_i) := f_{\mathbf{sel}}\left(\mathcal{I}(\mathcal{N}_{\mathbf{m}}), \sigma_i\right) \tag{7.3}$$

$$\mathbb{P}_2 := \left\{ \langle \mathbf{m}, \sigma \rangle \text{ s.t. } \langle \mathbf{m}, \sigma \rangle \in \mathbb{P}_1, \sigma = \arg\max_{\sigma_i} S(\mathbf{m}, \sigma_i), S(\mathbf{m}, \sigma) > \tau \right\} \tag{7.4}$$

The selected scale indicates the scale at which a maximum detector response to the local image structure is observed. It is relatively independent of the image resolution and is related to the structure and not to the resolution at which the structure is represented.

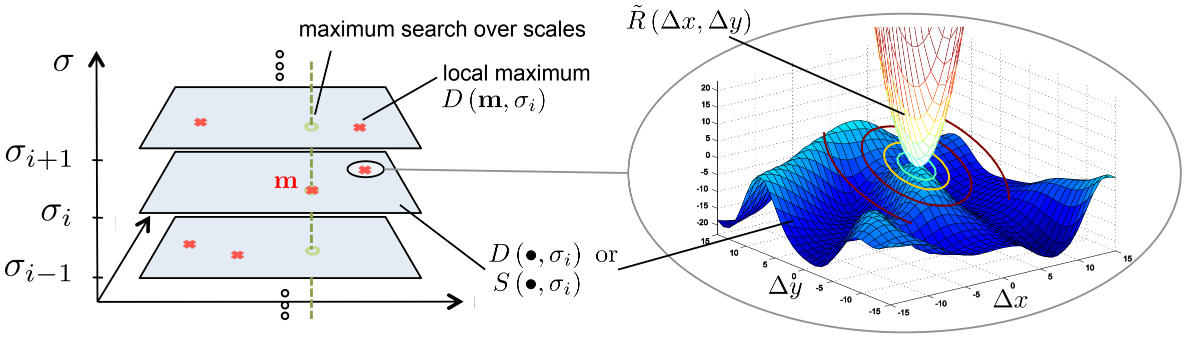

maximum search over scales
local maximum
$D(\mathbf{m}, \sigma_i)$
σ
σ_{i+1}
σ_i
σ_{i-1}
m
$D(\bullet, \sigma_i)$ or
$S(\bullet, \sigma_i)$
$\tilde{R}(\Delta x, \Delta y)$
Δy
Δx

Figure 7.3: **Detection Function defined in Scale Space and its related Hessian** (left) Features are detected in scale-space via a local maximum search in each stack layers to find the pixel location (u, v) of the features, followed by a maximum search over all scales; (right) Detector function response D at a given feature location and the corresponding residual function R (bell curve) that provides the covariance matrix via its Hessian.

Furthermore, the scale estimate will permit stable feature point localization under image resizing, in plane rotations, and similarity transformations.

The outcome of the detection process is a set $\mathbb{P}_2$ of feature points $\langle \mathbf{m}, \sigma \rangle$. Post processing steps may be present to reject those points from $\mathbb{P}_2$, which are not stable to detect. For example points on lines or in ridges can be rejected, as their position is hard to localize.

Local features detector presented in section 3.7.3 can directly be integrated in such a framework using their detector function $f_{\mathbf{dec}}$ and scale selection function $f_{\mathbf{sel}}$.

7.3.3 Error in Scale Detection

In our approach, we do not integrate the uncertainty for the scale selection, because it does not impact the precision of the localization. If the scale uncertainty, based on $f_{\mathbf{sel}}$, would be used, the resulting 3D ellipsoid $(x,\ y,\ \sigma)$ defined in scale space would collapse to a 2D ellipsoid in the original image equivalent to the 2D covariance $(x,\ y)$. Therefore our error measurements are only based on 2D localization uncertainty.

Though, knowledge of the error in scale detection can be beneficial in other computer vision methods. For example, it could be inserted in descriptor algorithms such as [Lowe, 2004, Bay et al., 2008] where the Gaussian smoothing strength over the gradient image could be computed from it. Additionally, information such as the descriptor orientation uncertainty could also be used in such a way to create a more robust descriptor. Unfortunately, the propagation of these error measurements into a covariance for the descriptor estimation and their use might be far from trivial.

7.4 Uncertainty Estimation Framework

Based on the general feature detection process presented before, we now want to explain how to estimate an error covariance for a feature point. One can see that for interest point localization only the feature detection operator, represented by the scale-space stack D, is of importance. The detection process is accomplished by a local maximum search within this detection stack. It is followed by a search for the characteristic scale in S; however, the scale selection process does not influence the interest point location. Thus, for the evaluation of a detection error the particular layer $D(\bullet, \sigma)$ of the detection pyramid is the determining factor.

Interest points relate to extrema in the operator output. Therefore, the search for a local maximum in D will lead to the same feature point location $\mathbf{m}$ as minimizing the "residual" cost function $R(\mathbf{\Delta m})$:

$$R(\mathbf{\Delta m}) = D(\mathbf{m}, \sigma) - D(\mathbf{m} + \mathbf{\Delta m}, \sigma) \tag{7.5}$$

$$\mathbf{m} = \arg \max_{\mathbf{x} \in \mathcal{N}_{\mathbf{m}}} D(\mathbf{x}, \sigma) = \arg \min_{\mathbf{\Delta m} \in \mathcal{N}_{\mathbf{0}}} R(\mathbf{\Delta m}) \,. \tag{7.6}$$

A graphical representation of R is given in Figure 7.3-right.

During the rest of this chapter, we only search for maxima, but this formulation can be also applied when searching for a local minimizer $\mathbf{m}$ of D. We just have to redefine R as $-R$ but the rest of the formulation stays untouched. We decide to separate the maximum search case from the minimum one over using the absolute value of R, since derivatives of R could not be continuous when using $\|.\|$.

R and its related coordinates $\mathbf{\Delta m}$ have their origin at point $\mathbf{m}$. For a small neighborhood $\mathbf{\Delta m} = (\Delta u, \Delta v) \in \mathcal{N}_{\mathbf{0}}$ we can approximate $R(\mathbf{\Delta m})$ via a Taylor expansion to the second order for feature point $\langle \mathbf{m}, \sigma \rangle$ (see also Figure 7.3):

$$\begin{aligned} R(\mathbf{\Delta m}) \approx \tilde{R}(\mathbf{\Delta m}) &= R(\mathbf{0}) + \frac{\partial R(\mathbf{\Delta m})}{\partial \mathbf{\Delta m}} \mathbf{\Delta m} + \frac{1}{2} (\mathbf{\Delta m})^\top \frac{\partial^2 R(\mathbf{\Delta m})}{\partial \mathbf{\Delta m}^2} \mathbf{\Delta m} \\ &= \frac{1}{2} (\mathbf{\Delta m})^\top \mathsf{H} \mathbf{\Delta m} \,. \end{aligned} \tag{7.7}$$

R and its first derivative vanish at position $\mathbf{0}$, because it is a minimum of R. The second order term remains, where the Hessian H characterizes the curvature at the interest point $\mathbf{m}$. Simply speaking, for a low curvature, the detection process will imply an error due to the missing discriminative behavior of $D(\bullet, \sigma)$ in the neighborhood $\mathcal{N}_{\mathbf{m}}$, whereas for a high curvature the spatial detection process will be more accurate. The curvature of a function around a point $\mathbf{m}$ is given by the Hessian H. Therefore it is natural to regard the inverse of the Hessian as a measure for the feature localization uncertainty. A more theoretic argumentation was made by Kanazawa and Kanatani [2003].

The estimation process then happens in two steps:

1. Estimate the covariance for each interest point $\langle \mathbf{m}, \sigma \rangle$ from the Hessian related to the point, according to

$$\begin{aligned} \Sigma = \mathsf{H}^{-1} &= \begin{bmatrix} R_{xx}(\mathbf{\Delta m}) & R_{xy}(\mathbf{\Delta m}) \\ R_{xy}(\mathbf{\Delta m}) & R_{yy}(\mathbf{\Delta m}) \end{bmatrix}^{-1} \\ &= - \begin{bmatrix} D_{xx}(\mathbf{m}, \sigma_i) & D_{xy}(\mathbf{m}, \sigma_i) \\ D_{xy}(\mathbf{m}, \sigma_i) & D_{yy}(\mathbf{m}, \sigma_i) \end{bmatrix}^{-1} , \end{aligned} \tag{7.8}$$

where R_{xx}, R_{xy}, R_{yy} and D_{xx}, D_{xy}, D_{yy} respectively, are the second order derivatives of R and D respectively. Most algorithms compute the Hessian already, for example to reject edge responses. In this case the Hessian does not need to be recalculated. The inverse in Equation (7.8) always exists, as the Hessian at an interest point is always positive or negative definite. Moreover the inverse can be computed with minimal computational cost, since there exists a closed from solution for a 2×2 matrix. Nevertheless it is often unnecessary to compute this inverse. For example the inverse of the covariance is used for a weighted least square for bundle adjustment, see section 7.7.

2. Depending on the particular creation process of the detector stack D, it may be required to propagate the covariance matrix back to the initial scale σ_0; σ_0 here is according to blurring present in the initial image. By doing so it is ensured that covariances retain their proportional relationship. In particular rescaling is needed, if layer $D(\bullet, \sigma_i)$ does not have the same resolution as $D(\bullet, \sigma_0)$; this often is the case for computational reasons. A back projection is done via the ratio of layer resolutions (obtained by res()):

$$\Sigma^{(0)} = \Sigma \cdot \left(\frac{\operatorname{res}(D(\bullet, \sigma_0))}{\operatorname{res}(D(\bullet, \sigma_i))} \right)^2 . \tag{7.9}$$

$\Sigma^{(0)}$ here refers to the covariance associated with a feature point $\langle \mathbf{m}, \sigma \rangle$ at position $\mathbf{m}$ in the initial image, describing its localization precision.

The proposed method is applicable to feature detection algorithms detecting points in scale space. In the following section we demonstrate how to implement the framework for SIFT and SURF.

7.5 Applications of the Uncertainty Estimation Framework

The covariance estimation framework is easily applied to SIFT and SURF. The covariance is calculated according to Equation (7.8) as the inverse of the Hessian at the interest point location $\langle \mathbf{m}, \sigma \rangle$ in scale-space from the detector response map D. To get a more robust estimate it is useful to increase the influence region from 3×3 to a 5×5 neighborhood and calculate the Hessian as a Gaussian weighted sum:

$$\Sigma\left(\mathbf{m}\right) = \left(- \sum_{\widetilde{\mathbf{m}} \in \mathcal{N}_{\mathbf{n}}} w(\widetilde{\mathbf{m}}) \cdot \begin{bmatrix} D_{xx}(\widetilde{\mathbf{m}}, \sigma) & D_{xy}(\widetilde{\mathbf{m}}, \sigma) \\ D_{xy}(\widetilde{\mathbf{m}}, \sigma) & D_{yy}(\widetilde{\mathbf{m}}, \sigma) \end{bmatrix} \right)^{-1} , \tag{7.10}$$

with $w(.)$ a gaussian weight centered on $\mathbf{n}$.

The second order derivatives D_{xx}, D_{xy}, D_{yy} are calculated by taking differences of neighboring sample points. The according filters are

$$d_{xx} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}, \quad d_{yy} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \quad d_{xy} = \frac{1}{4} \cdot \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} . \tag{7.11}$$

In details, derivatives are computed by point-wise multiplication of the filters with the detector response map values resulting in

$$D_{**} = d_{**} \cdot D(\mathcal{N}_{\mathbf{m}}, \sigma_i). \tag{7.12}$$

Note that the interpolation step shown in Equation (3.51) will lead to a detection scale $\hat{\sigma}$ which is not represented by pyramid scales σ_i. By interpolation between pyramid layers one can calculate the detector response map $D(\widehat{\mathbf{m}}, \hat{\sigma})$ at the particular characteristic scale $\hat{\sigma}$. Covariance estimation is performed at this characteristic scale; so at a given octave and (sub)interval. As pyramid layers of different octaves possesses a different size, this requires back propagation of covariances to the original image size according to

$$\Sigma^{(0)} = \Sigma \cdot (2^{octave})^2 \,, \tag{7.13}$$

which is equivalent to equation 7.9 using the "octave" idea used in SIFT and SURF. To lower complexity, covariances can be estimated at the detection scale σ rather than at $\hat{\sigma}$. The initial detection scale σ relates to a scale σ_i at which a layer is already represented in the pyramid. This simplification avoids a costly interpolation; however, is not degrading the result significantly. Using $D(\widehat{\mathbf{m}}, \sigma)$ as the reference layer for the Hessian calculation requires a back projection of

$$\Sigma^{(0)} = \Sigma \cdot (2^{octave} + (\hat{\sigma} - \sigma))^2. \tag{7.14}$$

Covariances estimated in this manner can only be determined up to an unknown global scale: the noise level. It could be approximated by using a method like patch match [Ji et al., 2010] but since it would require an additional processing for each image we suppose that the noise is constant across different images. We do not use the normalization suggested in [Kanazawa and Kanatani, 2003, Kanatani, 2004] either as we want to preserve the proportions between covariances.

Therefore we force all covariance matrices to be scaled such that a circular feature detected in the very bottom pyramid layer will approximately have Frobenius norm 1. This constant factor has been determined experimentally for SIFT and for SURF. Note, that scaling is only performed for numerical reasons and does not change the influence of covariance in any way.

7.6 Covariance Evaluation

The following contains a description of the experiments which were carried out to ensure that the uncertainty estimates are related to the real underlying location error distribution. First, the proposed covariance estimates are compared to the computer-generated error distribution and the accuracy of the estimates itself is evaluated. Second, the behavior of covariances over scale is investigated.

7.6.1 Statistical Error Modeling

The idea behind the statistical error sampling employed, is to create synthetic images with which it is possible to control the ground truth location of feature points. For detected

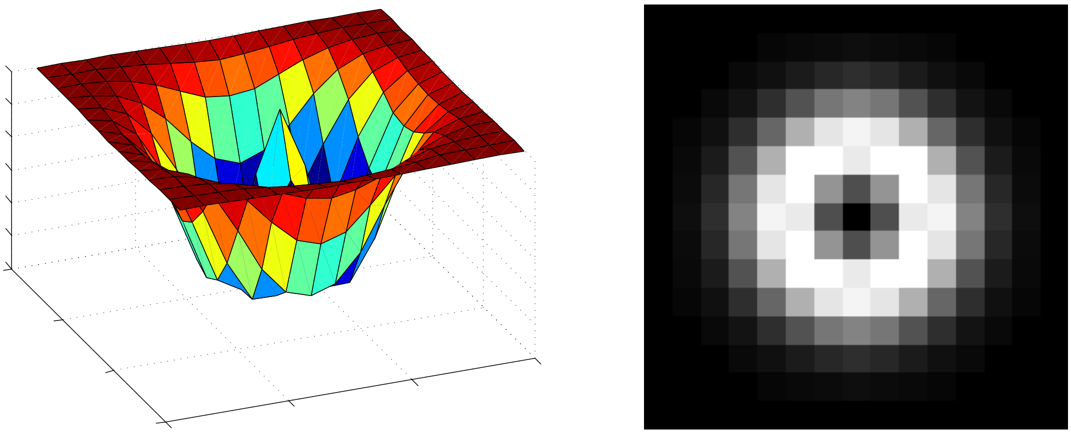


Figure 7.4: **Image generating maximal response for the SIFT detector**. An interest point will be found exactly in the middle of the feature. Note, that for displaying this feature in an image the pixel intensities have to be scaled such that they are fitting to the value range present in the image.

feature points in these images it is then possible to measure the localization error. By perturbing the feature image we can capture the statistical behavior of the detector.

7.6.1.1 Optimal Images for SIFT and SURF

SIFT uses a linear filter - the difference of Gaussians (DoG) - as detection operator. For controlling the feature point location, the output of the operator convolution has to be maximum at the specified ground truth location. It is achieved via a matched filter approach by placing a DoG itself at the desired feature location $\mathbf{m}_0$:

$$\mathcal{I}(\mathbf{m}) = G(\mathbf{m} - \mathbf{m}_0, \sigma_{i+1}) - G(\mathbf{m} - \mathbf{m}_0, \sigma_i) \; . \tag{7.15}$$

An exemplary Optimal SIFT feature is visible in Figure 7.4.

For SURF, a matched filter approach is not directly feasible, because the determinant of the Hessian is a nonlinear detection operator. Yet, it is possible to create an optimal image, where the SURF detector will have maximum response at a predefined position. This is solved by finding an optimal image that maximizes the SURF filter under some constraints as demonstrated in [Zeisl, 2009, Schweiger et al., 2009]. An exemplary optimal SURF feature is visible in Figure 7.5

For more details on optimal SIFT and SURF features, the reader is referred to [Schweiger et al., 2009]

7.6.1.2 Covariance from MLE versus Presented Methods

To build up a localization error distribution, repeating the detection several times does not give the desired result, since the detection process is deterministic [Kanatani, 2004]. By adding pixel noise in the original image, the localization error is expected to change as well. We then compute a maximum likelihood estimate for mean and covariance of the

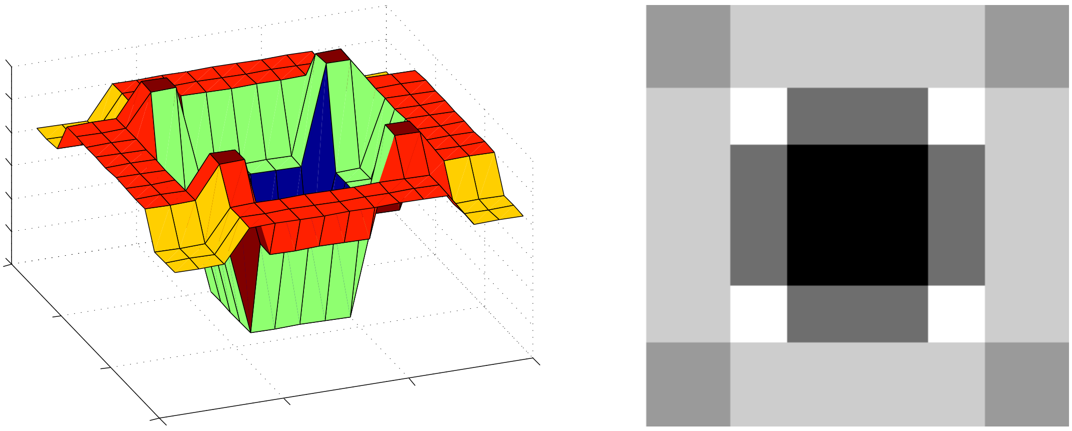


Figure 7.5: **Image generating maximal response for the SURF detector**. An interest point will be found exactly in the middle of the feature. Note, that for displaying this feature in an image the pixel intensities have to be scaled such that they are fitting to the value range present in the image.

sampled localizations of the feature. This error distribution from the detection process is then compared to our Hessian based covariance. In order to test the influence of viewpoint changes, the initial synthetic image is additionally warped with a perspective transformation. This will in general generate a synthetic image shape not representing the optimal image feature any more. Still the ground truth feature point can be warped exactly. This should explain in a more realistic manner the error made by a detector. We suppose the MLE to be our ground truth and we expect it to be close to our estimate. Results for the error modeling are shown in Figure 7.6.

The maximum likelihood estimate of the sampled error distribution and our covariance estimates are compared to each other via the Bhattacharyya distance [Bhattacharyya, 1943], which is measuring the similarity of two discrete probability distributions. For multivariate Gaussian distributions $N(\boldsymbol{\mu_1}, \Sigma_1), N(\boldsymbol{\mu_2}, \Sigma_2)$ the distance is defined as:

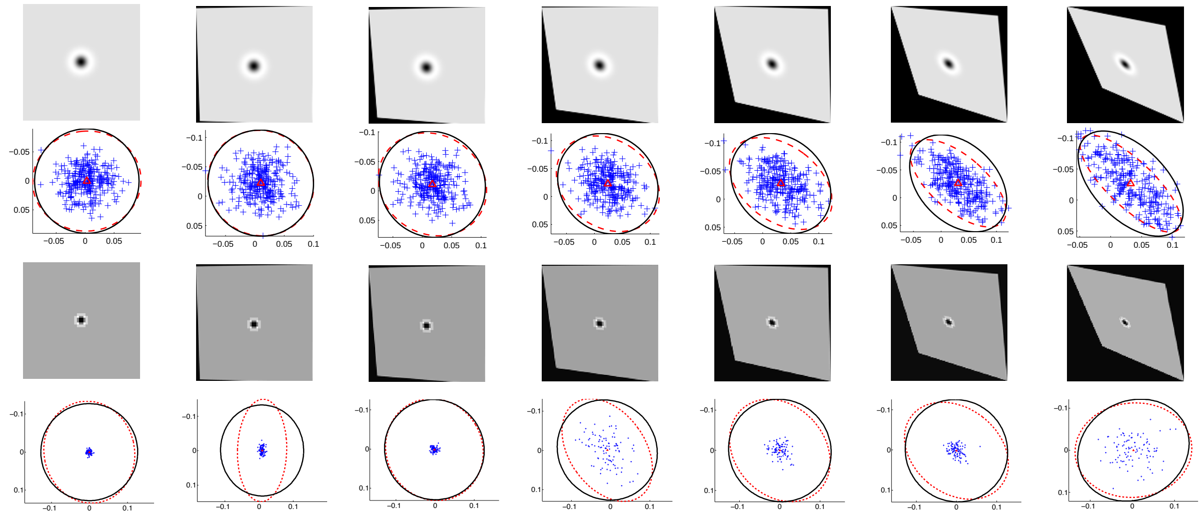
$$d_B = \frac{1}{8}\left(\mu_1 - \mu_2\right)^\top \Sigma^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\ln\left(\frac{\det\Sigma}{\sqrt{\det\Sigma_1 \cdot \det\Sigma_2}}\right) \tag{7.16}$$

$$\text{with } \Sigma = \frac{\Sigma_1 + \Sigma_2}{2}. \tag{7.17}$$

A normalization before comparison is necessary as the estimated covariances can only be determined up to scale. In addition $\mu_1 = \mu_2$ holds true, as both covariance are located at the interest point location. Table 7.1 lists the results for varying viewpoint changes.

The error distribution for SIFT is following the warping in the image and so does the covariance estimate. For SURF the error does not depend on the feature shape; still the covariance estimate fits to the modeled error distribution as well.

Notable is that the estimate is circular in most cases. This coincides with [Kanazawa and Kanatani, 2003], which state that for the Harris corner detector covariances are of circular shape. Comparable to the Harris detector, SURF is also applying the determinant of the Hessian as detection operator, and thus the shape of covariances is coherent. Still



(a) 0° (b) 10° (c) 20° (d) 30° (e) 40° (f) 50° (g) 60°

Figure 7.6: **Covariance Evaluation Using Synthetic Features**. Distribution of location error (blue cross) and comparison of maximum likelihood estimate (dashed line) to our Hessian based covariance estimate (solid line) with respect to varying perspective distortion (a) - (g) for SIFT (top) and SURF (bottom).

Viewpoint change	0°	10°	20°	30°	40°	50°	60°
	Bhattacharyya distance($\cdot 10^3$)						
SIFT	0.181	0.850	0.955	2.72	7.94	32.9	50.2
SURF	1.92	69.54	0.359	17.98	4.82	9.11	3.11

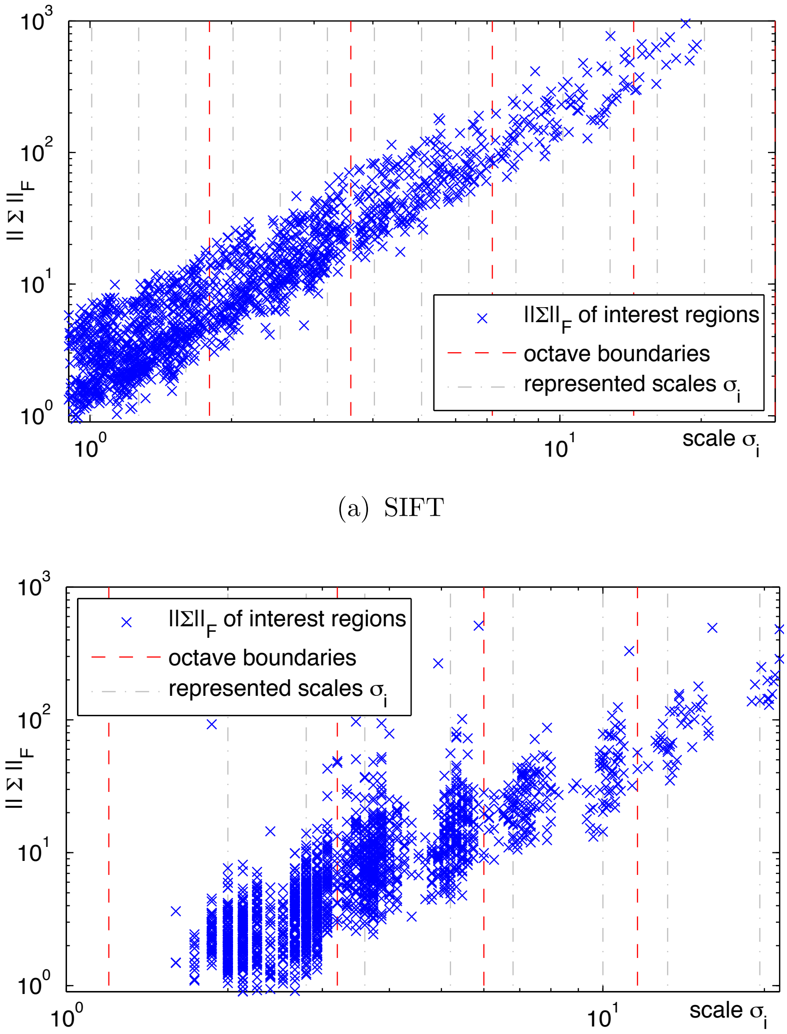
Table 7.1: **Numerical Evaluation Using Synthetic Features**. Covariance compared between Maximum Likelihood and our estimate

the covariance estimate and error distribution are following the scale, which is demonstrated in the next section. For more elliptic shaped error distributions, we can observe a tendency to underestimate the error. The effect can also be seen in the Table 7.1 where we observe a greater Bhattacharyya distance. From the evaluation it can be concluded that the covariance estimate does approximately represent the underlying localization error distribution.

7.6.1.3 Covariance Dependence on Scale

We now investigate the dependence of covariances on the particular detection scale. Interest points within several real images are detected and the Frobenius norm for each covariance matrix is calculated. Figure 7.7 displays the change of the covariance norm over the related detection scale.

The curves show that features detected at higher scales are declared less accurate compared to features detected at lower scales. This is intuitive as layers in the detection pyramid corresponding to higher scales, are built from more blurred or sub-sampled



(a) SIFT

(b) SURF

Figure 7.7: **Covariance Evolution with Respect to Scale Change**. Frobenius norm of estimated covariances for interest regions detected in real images. The covariance norm and thus also the localization accuracy is clearly dependent on the detection scale for both SIFT and SURF features..

versions of the original image. This loss of information is the reason for the increasing localization error.

A second experiment was undertaken to demonstrate the dependency of the localization precision on the particular detection scale. It includes two images which have been recorded with different cameras. The first was shot with a high quality camera and high resolution; the second was recorded with a standard webcam exhibiting low resolution. We tried at best for both the camera to have the same pose for them to capture the same features from the same point of view. The expectation is that the latter camera generates an image with more blur due to the low quality. In the high resolution image, a multiple of feature points are detected; particularly at scales not visible in the webcam image, because the webcam is missing these image details.

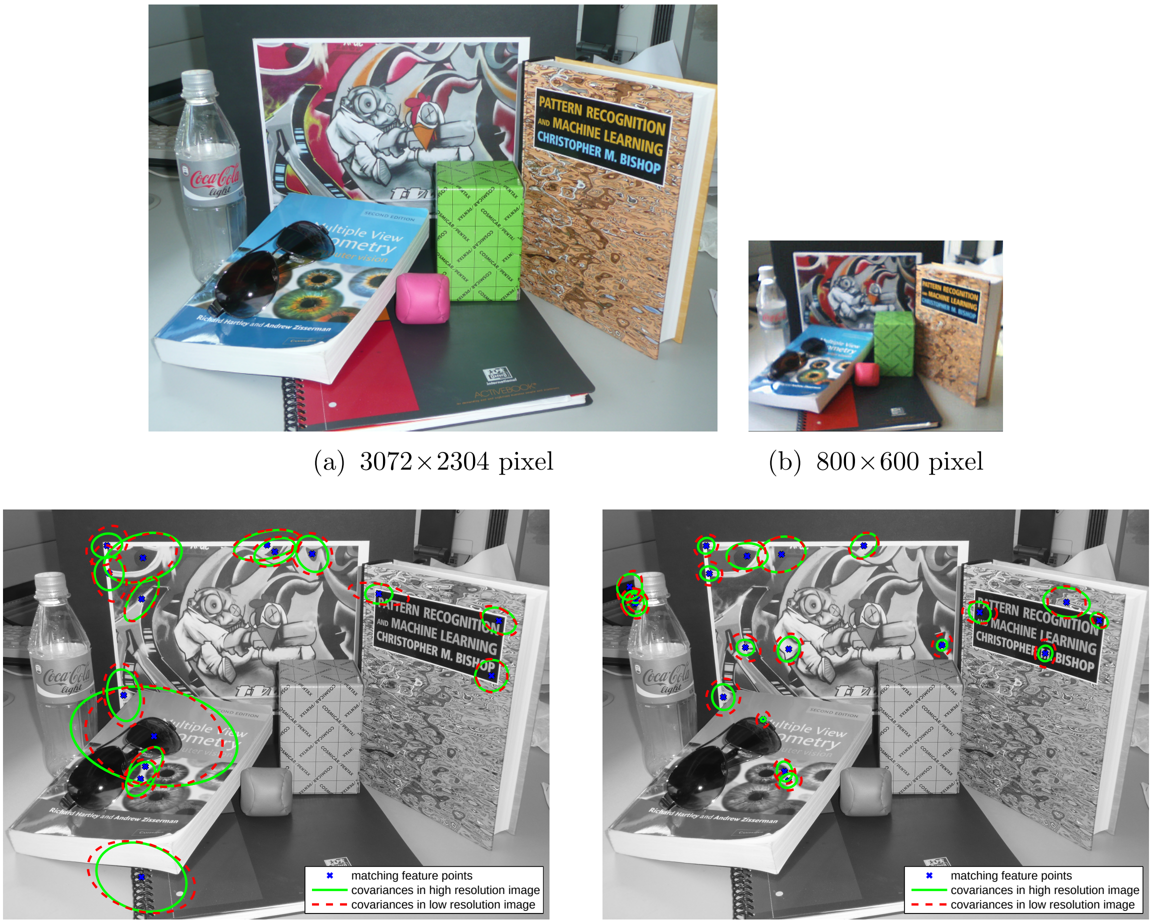
One could expect that feature points in the low quality image will be localized with greater un-precision; however corresponding feature points found in both images are detected at pyramid layers where the underlying image feature are of equal size. Thus the errors for localizing these particular features are the same in both images in relation to the image size. If one would compare just the covariances to each other the absolute size is of course smaller in the low resolution image. Using covariance information from multi-scale image offers a natural normalization for images captured at different resolution. Figure 7.8 illustrates covariances for matching feature points in the two images. Here covariances in the low resolution image are projected to the high resolution image by means of the underlying homography. One can see that the produced covariance are of comparable shape and size for both sensors.

In a third experiment a video of a moving pattern is recorded using an off-the-shelf camcorder. SIFT features within the pattern are detected in each frame and their covariances are calculated; subsequent corresponding points are tracked over the sequence. By this means it is possible to demonstrate the change of covariances under perspective transformations, rotation changes and resizing of the pattern. In Figure 7.11 selected frames are displayed from the sequence and the covariance change can be observed. First, we can see that when the pattern gets smaller (Zoom and Tilt) the covariances follow this behavior. Because the features get detected on layers of stack that have been less blurred. Second, we can observe the stability of the covariance orientation as they turn on par with the pattern rotation.

All these experiments verified that computing the uncertainty of the multi-scale local features based on the inverse of the Hessian detector function followed the error made by these detectors. Furthermore, we demonstrated the error behavior of multiscale features: a feature detected on higher (coarse resolution) scale is worth located than a feature detect lower (fine resolution) scale. We will now prove the usefulness of a covariance information for model fitting.

7.7 Results for Model Fitting

While the correctness of covariance estimates was investigated in the previous section, we still need to verify the usefulness of this information in order to justify their needs. Bundle adjustment as described in Section 3.9.3 is used to demonstrate this point. We decide to



✖ matching feature points
covariances in high resolution image
covariances in low resolution image
✖ matching feature points
covariances in high resolution image
covariances in low resolution image

(a) 3072×2304 pixel

(b) 800×600 pixel

(c) SIFT

(d) SURF

Figure 7.8: **Covariance Evolution for different Sensors** The top row shows a high and low resolution image from a scene captured with a standard photo camera and a webcam, respectively. The figures below illustrate covariances for matching feature points in the two images. As matching feature points are detected at corresponding characteristic scales, i.e. feature shapes in the images are of equal size, the difference in resolution and quality does not lead to a less accurate localization. Projecting covariances from the low resolution image to the high resolution image, shows that the localization precision is almost identical.

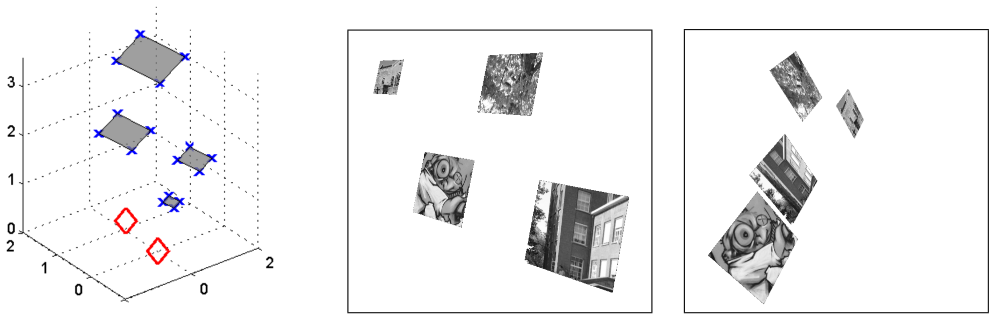


Figure 7.9: **Synthetic Scene used for Bundle Adjustment**: (left) Images are generated from virtual cameras (red diamonds) capturing a scene made of four textured planes, (middle and right) Rendered scenes for two camera positions

use bundle adjustment because it is the gold standard for relative pose estimation, which is used in the VID to register images based on keyframes.

When bundle adjustment is defined as a least square problem, the observation $\mathbf{y}_{ik}$ is directly used to compute the cost as described in Equation 3.71. Each observation is defined as the difference between the estimated re-projection from the current estimate and the 2D measurement:

$$\mathbf{y}_{ik} = \mathsf{K}_i \mathbf{w}\left(\widehat{\mathsf{T}}_i \widehat{\boldsymbol{\mathcal{M}}}^k\right) - \widetilde{\mathbf{m}}_i^k \,. \tag{7.18}$$

Covariance information can easily be introduced in this minimization problem by transforming the least square problem to a weighted least square. This is done by incorporating the covariance in each observation $\mathbf{y}_{ik}$ as follow:

$$\mathbf{y}_{ik} = \Sigma_{ik}^{-\frac{1}{2}} \left(\mathsf{K}_i \mathbf{w}\left(\widehat{\mathsf{T}}_i \widehat{\boldsymbol{\mathcal{M}}}^k\right) - \widetilde{\mathbf{m}}_i^k\right) \,, \tag{7.19}$$

with Σ_{ik} the covariance measured for the 2D point $\widetilde{\mathbf{m}}_i^k$ in image $\mathcal{I}_i$.

The scene used for bundle adjustment is created synthetically, so its geometry is known beforehand. It consists of four parallel quadratic image patches located at different depths from the camera center (Figure 7.9). The scene is captured by 2 cameras from varying viewpoints. Therefore, we known camera calibration matrix K and transformation $\overline{\mathsf{T}}$. The feature points including their covariance estimates are computed in each of the images.

An initial estimate of the 3D structure and camera poses is created from matched interest points and the corresponding relative pose estimated linearly, as described in Chapter 3. In this setup, it is ensured that no outliers are present, by checking that correspondences resulting from a matching step are consistent with the known homographies between the two images. The present problem setup exhibits a gauge freedom [Morris et al., 2001, Kanatani and Morris, 2001] leaving the first camera undetermined. Therefore its 3D pose is fixed at position $[\mathbf{I}\ \mathbf{0}]$ and not changed during the optimization. The gauge freedom also accounts for the change of the global scale during the optimization, which is normalized afterwards to represent the known translation between cameras of each patch.

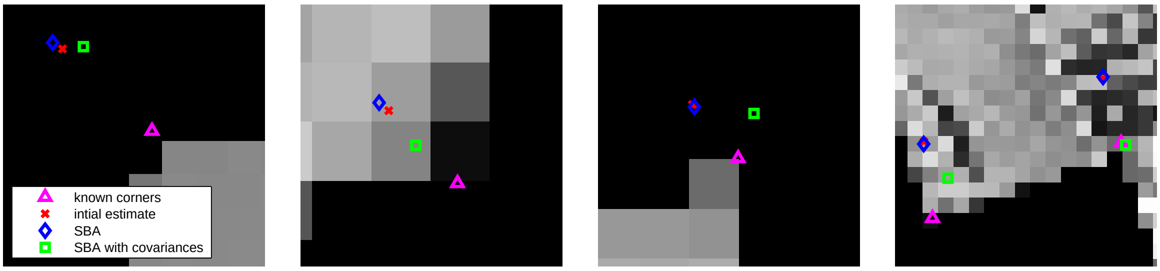


Figure 7.10: **Visual Evaluation Results of Bundle Adjustment**. Corner point locations of patches for different pose estimates obtained from bundle adjustment. It can be seen that the alignment of the corners using the covariance are closer to the true solution.

Finally, for every patch the target reprojection error between the known corner points $\bar{\mathbf{c}}$ and the projections of 3D corner points $\bar{\mathbf{C}}$ is computed by means of the estimated mapping parameters:

$$e = \frac{1}{4}\sum_{i=1}^{4}\left\|\bar{\mathbf{c}}_i - w\left(\widehat{\mathsf{T}}\bar{\mathbf{C}}_i\right)\right\|. \tag{7.20}$$

For these simulations the sparse bundle adjustment framework provided by [Lourakis and Argyros, 2009] is used. Table 7.2 summarizes the performance improvement of bundle adjustment with covariance estimates employed. It is interesting to note that the reprojection error is smaller for smaller patches. This effect is due to the fact that more distinctive feature points with smaller covariances were detected in smaller patches. Estimated corner points calculated with the initial pose estimate and bundle adjustment are displayed in Figure 7.10 and show more coherent pose estimates when using covariance information.

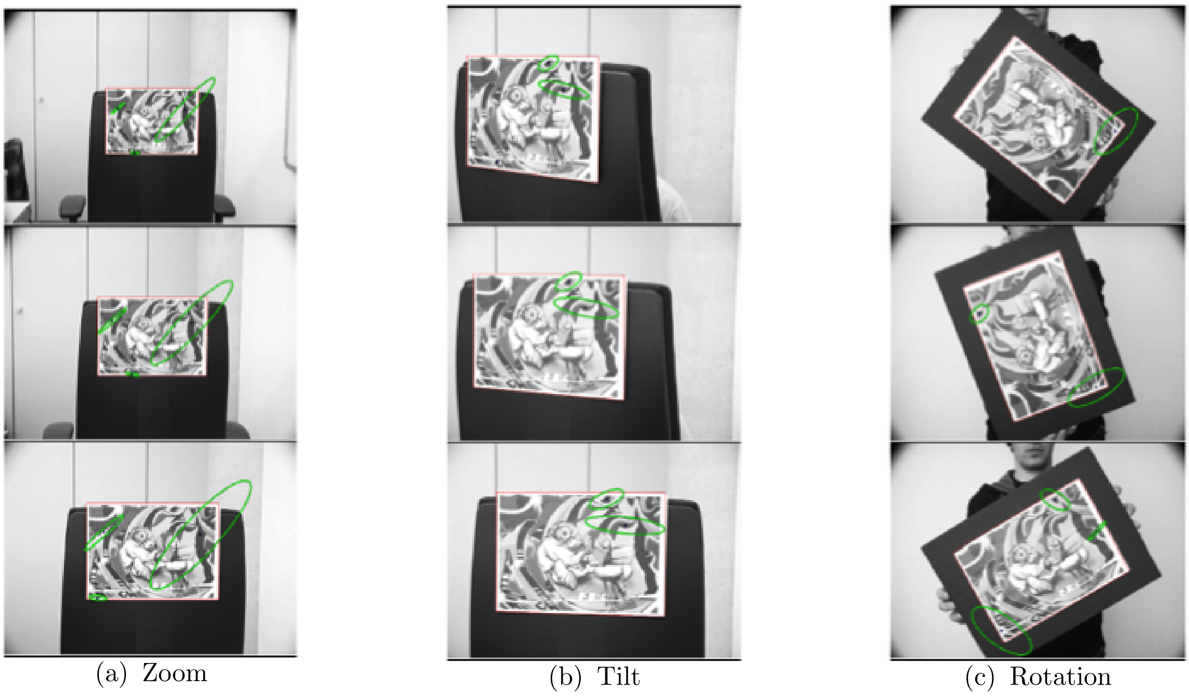
	mean all patches		smallest patch		largest patch	
covariance	W/O	W/	W/O	W/	W/O	W/
SIFT	2.031	1.759	1.941	1.672	2.088	1.828
SURF	2.554	2.363	2.518	2.292	2.631	2.464

Table 7.2: **Re-projection error after bundle adjustment** with and without covariance estimates used. The values indicate the mean performance as pixel offset for 100 different image pairs. Smallest and largest patch refer to the patch size seen in each of the images (and not to their actual size in 3D).

Conclusion

Evaluating the localization uncertainty of local features offers the possibility to remove hard (unexplainable) threshold in computer vision algorithms and to replace them by confidence measures. It also allows better estimations and self diagnostics [Meidow, 2008]. In this chapter, a novel framework for estimating location uncertainty for scale invariant feature points has been presented. It was shown that the covariance of the localization error can be calculated from the detector response map in the neighborhood of a feature point without significant computational overhead. As expected, covariances differ according to the particular detection scale and interest region shape. We showed first that our estimated covariances followed the error made by multi-scale detector and that using covariance for multi-scale features was indeed useful to obtain better registration. This could contradict the conclusion made by Kanazawa and Kanatani [2003], where they claim that covariance computation does not offer any significant advantage. We conjecture that the justification for this is two-fold: First computing covariance based on the auto-correlation property of the image might not be characteristic of the detector precision. This would imply that for example that a covariance computed for an Harris corner should be computed from its cornerness function and not from the direct image intensities. Second corners and regions have different properties. Corners are precise and offer a uniform quality measurement, but can only be matched across short baselines. Blobs can be matched across larger baselines at the cost of some precision loss and in general present heterogeneous precision across measurements. Therefore we think that a conclusion drawn based on corners might not be generalizable to regions.

Finally, in a recent publication, Sur [2010] used the method presented in this chapter to improve the robust estimation of fundamental matrices based on features points within an *a contrario* RANSAC, demonstrating in another scenario the usefulness of the covariance information for multi-scale features.



(a) Zoom

(b) Tilt

(c) Rotation

Figure 7.11: **Impact of Motion on Covariance Orientation** Selected frames from a video sequence showing a pattern under scale change, perspective transformations, and rotation. The covariances for tracked feature points are changing accordingly: 7.11(a) covariances are bigger the larger the feature is in the image; 7.11(b) covariances are compressed due to the perspective distortion; 7.11(c) covariances follow the rotation of the pattern.

CHAPTER

EIGHT

COMBINING PHOTOMETRIC AND GEOMETRIC INFORMATION FOR POSE ESTIMATION

In this chapter, we present a novel approach for the relative pose estimation problem from image point correspondences. Unlike classical algorithms, such as the Gold Standard algorithm (with or without covariance information), the proposed approach ensures that the matched points are photo-consistent throughout the pose estimation process. In fact, common algorithms use the photometric information to extract the feature points and to establish the 2D point correspondences. Then, they focus on minimizing, in a non-linear scheme, geometric distances between the projection of reconstructed 3D points and the coordinates of the extracted image points without taking the photometric information into account. This might not be optimal as we saw in the Chapter 7 that the precision of feature point is limited. The approach we propose merges geometric and photometric information in a unified cost function for the non-linear minimization. This allows us to achieve results with higher precision and also with higher convergence frequency compared to optimizing only a geometric re-projection error, which is defined to be the gold-standard [Hartley and Zisserman, 2003].

8.1 Introduction

In Computer Vision, relative pose estimation corresponds to the task of finding the geometric transformation between two cameras. Using two images each acquired by a calibrated camera, it is possible to use a set of corresponding feature points to estimate the pose parameters, i.e. the relative rotation and translation between the two cameras. This task is a core problem of several computer vision applications especially in Augmented Reality for tracking or when using pose from keyframes. Since the seminal work of [Longuet-Higgins, 1981] where an 8-point algorithm was proposed to compute the pose via the essential matrix, many works have been published either to generalize to the non-calibrated case [Hartley, 1992, Faugeras, 1992], or to improve its robustness [Hartley, 1995, Chojnacki et al., 2003] or as proposed recently, to solve it efficiently in a closed-form algorithm with the minimal set of seven points [Zhang, 1998] or when calibrated with five

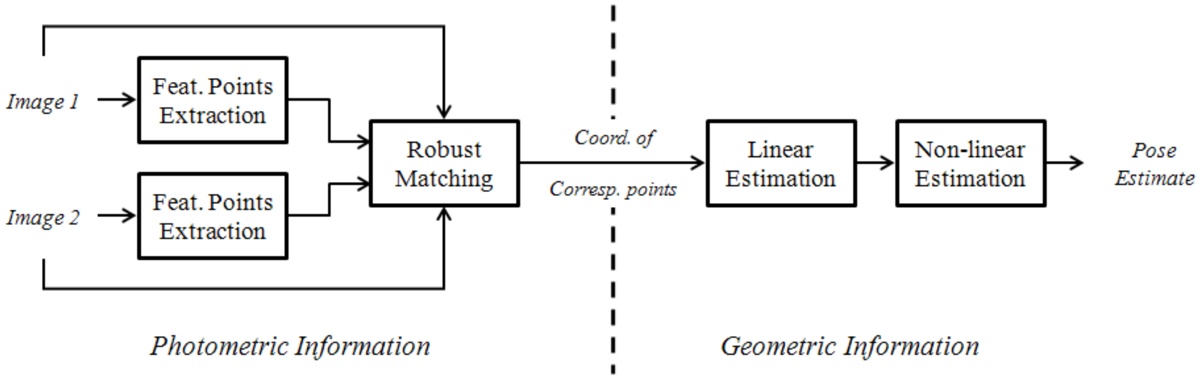


Figure 8.1: **Classical Approaches** set aside the image photometric information once the matching has been established. Only geometric information is used in the pose estimation.

points [Nister, 2004]. The standard scheme that one can find in reference computer vision books [Faugeras and Luong, 2001, Hartley and Zisserman, 2003] or in earlier work (e.g. [Horn, 1989]) is to actually use several corresponding points (from some tens to few hundreds) and to minimize a least-squares cost function making use of all available correspondences. In practice, this allows for the pose estimate as well as the 3D points to be precise. Data normalization precedes linear estimation or a closed-form solution, which generally serves as initialization to a non-linear minimization. The most commonly used algorithm to perform this task is the well known Gold Standard algorithm [Hartley and Zisserman, 2003]. It is based on minimizing the reprojection error (Eq. 3.69). This algorithm allows to obtain the Maximum Likelihood estimate of the transformation matrices and can be easily adapted to the Fundamental matrix using projection matrices. As depicted by figure 8.1, this algorithm makes only use of the coordinates of the image points. No photometric information (color, texture, image gradients,...) is used once the matching of the feature points has been established.

The Gold Standard algorithm works well in practice and gives satisfactory results in most cases. Recent improvements have been proposed such that the linear estimation has a better conditioned measurement matrix [Wu et al., 2005] or such that the optimizer converges toward the global minimum [Hartley and Kahl, 2007] using a L_∞ norm. In the presence of small number of noisy feature points, even if a robust estimation [Fischler and Bolles, 1981] is applied to remove outliers in the matching process, it is hard to recover a precise pose. It is possible to get a geometrically correct and locally optimal pose, but it can be, in some cases, far from the real one because 2D measurements used to compute the relative pose are not perfect (see Chapter 7). This unprecision is mainly due to the fact that such approaches do not guarantee a photometric consistency with respect to the images.

In this chapter, we introduce an additional constraint to the traditional reprojection error for the final non-linear optimization. We enforce the reconstructed 3D points to have the same appearance in both images and not only to be close to their 2D measurements.

We first discuss related work in merging heterogeneous information for parameter estimation in Section 8.2. Then we describe the gold standard method for the relative pose estimation in Section 8.3, followed by the proposed method to improve this estimation in Section 8.4. Finally we evaluate our method on simulated data as well as industrial site pictures in Section 8.5.

8.2 Previous Work in non only geometric for Structure for Motion

In Chapter 3 and the previous section, we introduce pose estimation methods that were solely based on geometric measurements obtained from features points. But researchers tried over the years to create better methods that use different type of measurements.

There was a large trend in Factorization methods to use constraint from points and lines [Triggs, 1996, Morris and Kanade, 1998]. Oliensis and Werman [2000] add intensity from the image to points and lines measurements to a factorization algorithm. Unfortunately because of the linearization of factorization approach, the intensities information does not constrain the structure estimated for the 3D points but only the motion of the cameras.

Hanek et al. [1999] add measurements obtained from cylinders to more classic points and lines measurements. They propose a probabilistic method to merge this information into a Maximum likelihood framework to obtain a camera pose with respect to a CAD model. In [Vacchetti et al., 2004b], full pose are estimated using a CAD model based on edges and features points. Edges are integrated using a multiple hypothesis scheme into a bundle adjustment based on harris corners. No gauge freedom is presented has the Harris corners are linked to the CAD model. Similarly [Comport et al., 2003] integrates distance to 3D lines and Ellipses to a visual servoing framework.

Hybrid methods have been extensively studied improve intensity based tracking [Baker and Matthews, 2004, Bartoli, 2006, Benhimane, 2007] where a homography is estimated between every frame. [Masson et al., 2003] replace non continuous information from the template (e.g. edges) that could make the tracking fail [Benhimane et al., 2007] by distance to closest edge. This “new template” is fed to a hyperplane tracker. Pressigout and Marchand [2005, 2007, 2008] add to a regular first order intensity based tracker, where error are estimated in difference of pixel intensities, distance between edges. These distances are measured in the direction of the gradient *à la* [Klein and Drummond, 2003, Vacchetti et al., 2004b]. They performed their minimization using a re-weighted least squares. Masson et al. [2003] use a similar idea but limit intensity differences to points from the template that correspond Harris corners.

Switching methods [Clark and Green, 2005] were also concidered in order to merge heterogeneous data. These methods minimize a cost function based on one information and then switch to another one. This process would then iterate until convergence. For example in deformable registration [Johnson and Christensen, 2002] propose to switch between a landmark based “rigid” registration and intensities based “deformable” registration until convergence. [Fakih and Zelek, 2008] also show that using more information improve the estimation process. They merge an optical flow method (dense) with essential matrix estimation approach (sparse). This method computes computes optical flows given a set of motion computed from 5 point RANSAC [Fischler and Bolles, 1981]. These optical flows are then fed back to a robust estimator in a classic Expectation Maximization (EM) framework.

All these methods demonstrate the usefulness of additional information for model fitting in general. Unfortunately they also show some limits. Particularly they do not tackle the hard problem of structure from motion via bundle adjustment, which is considered to be the Gold Standard for relative pose estimation. Furthermore, they either work on specific motion (e.g. homographies) or the intensity information do not constrain the estimation of the structure. In order to improve the Gold Standard, we add constraints to the optimization. We enforce that the estimated 3D structure is photo-consistent across input images.

In spirit, it is similar to [Shi and Tomasi, 1994] where they attach a patch to each of their features and estimate an affine deformation between frame for each patches. This idea was build on to create a patch based structure for motion [Chang, 2002]. Instead of minimizing a cost function based on the geometry during the bundle adjustment they minimize solely intensity differences using homographic model. Their model is based on an additive update that linearizes the group of homographies. Therefore they only estimate an approximation of the motions. On the other hand we introduce a correct compositional framework that combines geometric and intensity information to create a robust and precise method to solve the general problem of bundle adjustment.

8.3 Gold Standard for Relative Pose Estimation

In this section, we re-introduce the non-linear cost function used for the estimation of the bundle adjustment. This is the gold standard for the refinement of the relative pose [Hartley and Zisserman, 2003]. This is only a remainder, for a more detail introduction to multi-view geometry the reader is referred to Chapter 3. For simplicity of notations we only consider scene captured by two calibrated cameras but the method extends to more views in a straight forward manner [Acero, 2009]. In order to get the equation as readable as possible covariance information are also not included. It should be noted that the proposed method goes beyond handling bad measurement by trying to correct them. First we recall the general bundle adjustment cost function defined in Equation 3.69:

$$\mathcal{G}\left(\{\mathsf{T}_i\},\{\boldsymbol{\mathcal{M}}^k\}\right)=\frac{1}{2}\sum_{i=1}^{m}\sum_{k=1}^{n}\delta_i^k\left\|\mathsf{K}_i\mathbf{w}\left(\mathsf{T}_i\boldsymbol{\mathcal{M}}^k\right)-\widetilde{\mathbf{m}}_i^k\right\|^2 ,$$

which we rewrite for the two cameras case as follow:

$$\mathcal{G}\left(\{\mathsf{T}_1,\mathsf{T}_2\},\{\boldsymbol{\mathcal{M}}^k\}\right)=\frac{1}{2}\sum_{k=1}^{n}\left(\left\|\mathsf{K}_1\mathbf{w}\left(\mathsf{T}_1\boldsymbol{\mathcal{M}}^k\right)-\widetilde{\mathbf{m}}_1^k\right\|^2+\left\|\mathsf{K}_2\mathbf{w}\left(\mathsf{T}_2\boldsymbol{\mathcal{M}}^k\right)-\widetilde{\mathbf{m}}_2^k\right\|^2\right) . \tag{8.1}$$

Unfortunately this equation has some gauge freedom. First, T_1 cannot be constrained using only correspondences a set of 2D correspondences $\left\langle\widetilde{\mathbf{m}}_1^k,\widetilde{\mathbf{m}}_1^k\right\rangle$ between an image source $\mathcal{I}_1$ and an image target $\mathcal{I}_2$. Therefore we fix $\mathsf{T}_1=\mathsf{I}$ and we rewrite Equation 8.1

as follow:

$$\mathcal{G}\left(\mathsf{T},\{\boldsymbol{\mathcal{M}}^k\}\right)=\frac{1}{2}\sum_{k=1}^{n}\left(\left\|\mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{M}}^k\right)-\widetilde{\mathbf{m}}_1^k\right\|^2+\left\|\mathsf{K}_2\mathbf{w}\left(\mathsf{T}\boldsymbol{\mathcal{M}}^k\right)-\widetilde{\mathbf{m}}_2^k\right\|^2\right)\ , \tag{8.2}$$

with

$$\mathsf{T}:=\begin{bmatrix}\mathsf{R} & \mathbf{t}\\ \mathbf{0}^\top & 1\end{bmatrix}\ . \tag{8.3}$$

Starting from an estimate of the relative motion $\widehat{\mathsf{T}}$ and of the 3D structure $\left\{\widehat{\boldsymbol{\mathcal{M}}}^k\right\}$ we use a least square optimizer to minimize Equation 8.2. Methods to obtain an initial motion and structure estimate can be found in Section 3.9. The Gold Standard algorithm is often summarized as:

$$\left\{\widehat{\mathsf{T}},\left\{\widehat{\boldsymbol{\mathcal{M}}}^k\right\}\right\}=\arg\min_{\mathsf{T},\{\boldsymbol{\mathcal{M}}^k\}}\ \frac{1}{2}\sum_{k=1}^{n}\left(\left\|\mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{M}}^k\right)-\widetilde{\mathbf{m}}_1^k\right\|^2+\left\|\mathsf{K}_2\mathbf{w}\left(\mathsf{T}\boldsymbol{\mathcal{M}}^k\right)-\widetilde{\mathbf{m}}_2^k\right\|^2\right)\ . \tag{8.4}$$

The non-linear optimizers iteratively updates the motion parameters and the reconstructed 3D points by computing increments that reduce the distance between projection of the 3D points and their corresponding image points. In most case the updates are computed from Gauss-Newton or Levenberg-Marquardt (c.f. Section 3.3).

Unfortunately, this error minimization only ensures geometric validity of the structure consistency of the points even when using covariance information. It is important to note that in this framework the result of the feature points detection is never corrected based on photometric information.

8.4 Proposed Hybrid Method for Non-linear Pose Estimation

Due to some factors that generally provide inaccurate feature points localization (such as motion blur or noise in the images and detection bias), the standard way that we described above lacks precision. This is due to the fact that the feature point positions are only corrected during the non-linear minimization via the optimization of the 3D point positions. This adjustment is only based on geometric constraints related exclusively on 2D coordinates. Therefore, a large amount of information is set aside since no photometric information is used once the matching has been established.

To make use of all available information throughout the pose estimation process, we propose to alter the cost function used in the non-linear minimization (Eq. 8.2) such that photometric information is also taken into account. Next we describe how to incorporate this function in the final pose refinement step. Figure 8.2 gives an overview of our method.

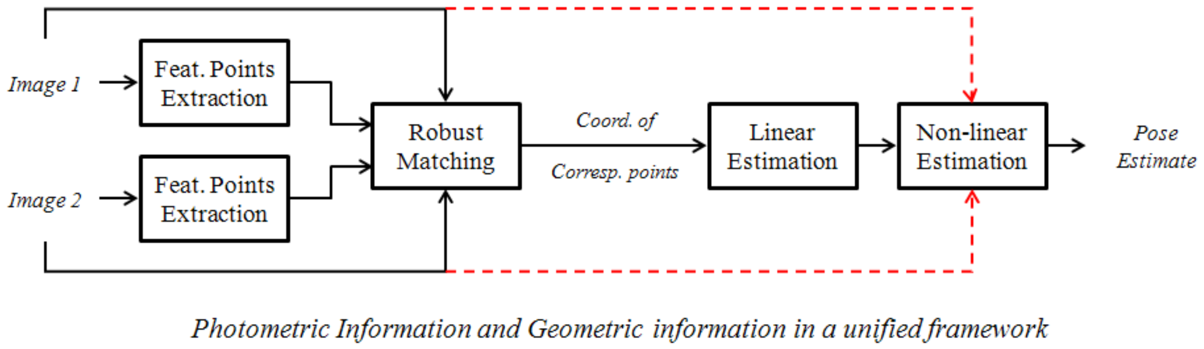


Figure 8.2: **Our Hybrid Approach** uses the photometric information not only for the matching but also for the non-linear estimation where photometric and geometric cues are combined.

8.4.1 Combining Geometric Distances with Intensity Differences

In order to incorporate photometric information, we consider the fact that the neighborhood of the 2D points $\widehat{\mathbf{m}}_1^k$ and $\widehat{\mathbf{m}}_2^k$ should be photometrically consistent. To enforce this consistency, we optimize the pose parameters and the 3D points coordinates such that, both the cost function defined in equation (8.2) and the sum-of-squared differences of the intensities of the projected neighboring points are minimized.

Combining intensities with other of type measurements offer challenges as they do not have the same scale and influence. Here, some issues should be carefully taken into account:

1. How can we define the neighborhood of the 2D points in the both images to sample intensities?

2. At which stage should the photometric term be used?

3. How should the geometric and the photometric terms be weighted with respect to one another?

8.4.1.1 Intensity Sampling

Concerning the first issue, we define the neighborhood of the feature points in the two images using samples on the tangent plane of the reconstructed 3D points since any surface can be locally approximated as planar, as shown in Figure (8.3). We show later that even a fronto-parallel approximation of the tangent planes is enough to obtain better result than the gold standard. As a consequence the neighborhood of 2D feature points is adapted when the pose and the 3D point locations are refined during the non-linear minimization. We denote by $\boldsymbol{\mathcal{Y}}^{kl}$ a point in the neighborhood $\mathcal{N}_{\boldsymbol{\mathcal{M}}^k}$ of a 3D point $\boldsymbol{\mathcal{M}}^k$. The set $\mathcal{N}_{\boldsymbol{\mathcal{M}}^k}$ is in the tangent plane $\boldsymbol{\pi}_{\boldsymbol{\mathcal{M}}^k}$ to $\boldsymbol{\mathcal{M}}^k$ defined as:

$$\boldsymbol{\pi}_{\boldsymbol{\mathcal{M}}^k}{}^\top \boldsymbol{\mathcal{M}}^k = 1\,, \tag{8.5}$$

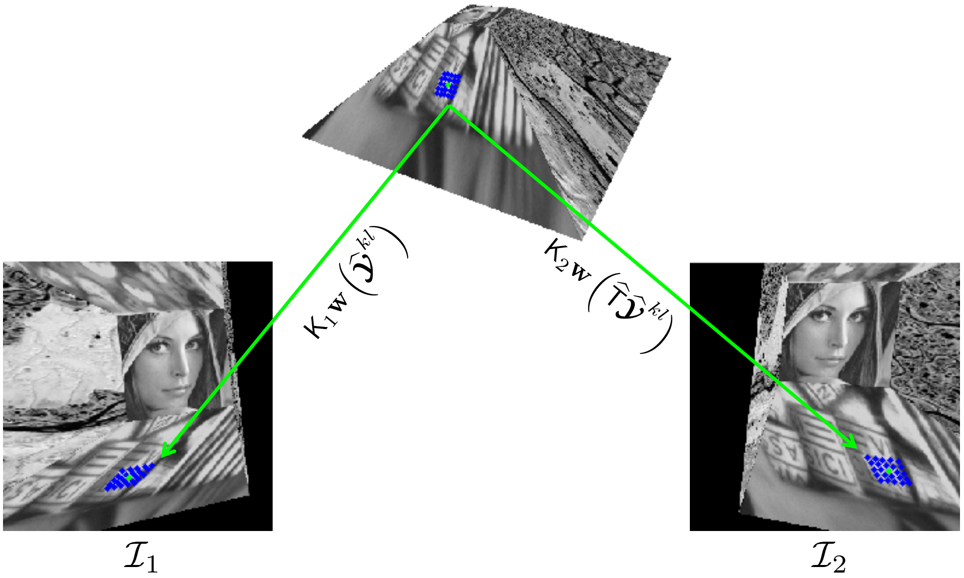

$\mathsf{K}_1\mathbf{w}\left(\widehat{\boldsymbol{\mathcal{Y}}}^{kl}\right)$
$\mathsf{K}_2\mathbf{w}\left(\mathsf{T}\widehat{\boldsymbol{\mathcal{Y}}}^{kl}\right)$
$\mathcal{I}_1$
$\mathcal{I}_2$

Figure 8.3: **Intensity Sampling Strategy**: the Neighbors $\widehat{\boldsymbol{\mathcal{Y}}}^{kl}$ (in blue) defined in 3D around triangulated point $\widehat{\boldsymbol{\mathcal{M}}}^{k}$ (in green), they are projected in the image to create $\mathtt{Patch}_1^{\mathtt{k}}$ and $\mathtt{Patch}_2^{\mathtt{k}}$ as shown in left and right pictures.

with $\boldsymbol{\pi}_{\boldsymbol{\mathcal{M}}^k}{}^\top = \left[\mathbf{n}^{k\top}\ d_k\right]$ its normal.

Now that we can sample intensities for a given 3D point $\boldsymbol{\mathcal{M}}^k$, we define a sum of squares differences between pixel intensities as follow:

$$\mathcal{P}\left(\mathsf{T}, \boldsymbol{\mathcal{M}}^k\right) := \sum_{\boldsymbol{\mathcal{Y}}^{kl} \in \mathcal{N}_{\boldsymbol{\mathcal{M}}^k}} \left(\mathcal{I}_1\left(\mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{Y}}^{kl}\right)\right) - \mathcal{I}_2\left(\mathsf{K}_2\mathbf{w}\left(\mathsf{T}\boldsymbol{\mathcal{Y}}^{kl}\right)\right)\right)^2 . \tag{8.6}$$

8.4.1.2 Selective Activation of Intensity based Cost Function

In general, the minimization of such a cost function is based on a first-order Taylor expansion of the cost function (such as in Gauss-Newton or Levenberg-Marquardt) and converges in a small basin of convergence. This is because the convexity assumption (needed by such minimizers in order to succeed) is very local for Equation 8.6.

The convexity cannot be guaranteed if the image patches obtained by projecting the current estimates of the 3D neighborhood $\widehat{\boldsymbol{\mathcal{Y}}}^{kl}$ in the two images do not project on to overlapping physical area. In this case the optimizer will most probably diverge as the gradient's direction is meaningless. Therefore the initialization has to be precise.

Unfortunately, when using a pose resulting from a linear estimation and to perform triangulation from the image points $\left(\widetilde{\mathbf{m}}_1^k, \widetilde{\mathbf{m}}_2^k\right)$, the distances between measured points and their reprojected 3D points can be large (depending on the inaccuracy of the point extraction and the initial pose). The table 8.1, which was obtained using simulated data, confirms this argument. In presence of a bad initialization, the projection of the neighbors

#Points/Noise	0.01	0.1	1.0	2.0
8	0.3619	3.9054	96.3384	179.4612
10	0.2066	1.9684	17.0605	24.6839
25	0.1002	1.0511	10.4050	18.5587
50	0.0660	0.6916	6.0568	14.0087

Table 8.1: **Gross Triangulation Error from a linear relative pose**: Evolution of the mean residual error with respect to the ground truth (in pixels) over 200 runs after applying the 8-points algorithm and the optimal triangulation, with respect to the number of points and the Gaussian noise.

in each image might give two patches that will not overlap the same area of the observed scene.

Consequently, we activated the intensity cost function based on a geometric distance and based on a similarity measure between the patches in order to ensure that the projected patches are close enough. Let us denote by $\texttt{Patch}_1^{\texttt{k}} = \left\{\mathcal{I}_1\left(\mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{Y}}^{kl}\right)\right)\right\}$ and by $\texttt{Patch}_2^{\texttt{k}} = \left\{\mathcal{I}_2\left(\mathsf{K}_2\mathbf{w}\left(\mathsf{T}\boldsymbol{\mathcal{Y}}^{kl}\right)\right)\right\}$, the ordered sets of intensities obtained by projecting the 3D point $\boldsymbol{\mathcal{M}}^k$ in both images. The similarity measure used is the Normalized Cross Correlation (NCC).

The image information related to the current estimates of $\widehat{\boldsymbol{\mathcal{M}}}^k$ are considered when $\rho^k = 1$ and are not considered when $\rho^k = 0$ where:

$$\rho^k = \begin{cases} 1 \quad \text{if} & \& \begin{cases} d\left(\widetilde{\mathbf{m}}_1^k, \mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{M}}^k\right)\right) < \tau_1 \;\&\; d\left(\widetilde{\mathbf{m}}_2^k, \mathsf{K}_2\mathbf{w}\left(\widehat{\mathsf{T}}\boldsymbol{\mathcal{M}}^k\right)\right) < \tau_1 \\ \text{NCC}\left(\texttt{Patch}_1^{\texttt{k}}, \texttt{Patch}_2^{\texttt{k}}\right) > \tau_2 \end{cases} \\ 0 \quad \text{otherwise} \end{cases} \tag{8.7}$$

8.4.1.3 Adaptive Weighting Factor for Mixing Heterogeneous Data

Finally, for the last issue, since the geometric and photometric data are heterogeneous, we mix them in an unified cost function. If we would simply stack them together, we would have massive scale differences. In fact, intensity differences could vary between $[-255, 255]$ (when the pixel intensity is coded in 8 bits) while geometric distances are expressed in pixels. In order to obtain a more uniform observation, we scale the geometric distance by the inverse of the variance of the vector $\left[\left(\mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{M}}^k\right) - \widetilde{\mathbf{m}}_1^k\right)^\top \left(\mathsf{K}_2\mathbf{w}\left(\widehat{\mathsf{T}}\boldsymbol{\mathcal{M}}^k\right) - \widehat{\mathbf{m}}_2^k\right)^\top\right]^\top$. We do the same for the photometric distance with the vector $\left[\left(\mathcal{I}_1\left(\mathsf{K}\mathbf{w}\left(\widehat{\mathcal{Y}}^{kl}\right)\right) - \mathcal{I}_2\left(\mathsf{K}'\mathbf{w}\left(\widehat{\mathsf{T}}\widehat{\mathcal{Y}}^{kl}\right)\right)\right)^\top\right]^\top$. These scales are computed at initialization. This compensate for the difference of scale and gross outlier data.This idea was also develloped in photogrammetry and is called “variance component analisys” [Förstner, 1979]. It is usually used as a *a-posteriori* measure to evaluate outlier data. In our case we use it as global *a-priori* measure on the data quality. This scaling is slightly “smoother” than the one proposed in [Pressigout and Marchand, 2007] where they scale each their measurements by the maximum absolute value. The scaling factors α_g for the geometric term and α_p for the photometric term are then included in the unified cost function.

8.4.2 Unified Cost Function

We take into account the issues explained above and we modify the original minimization defined in equation (8.4) to create a new hybrid minimization for bundle adjustment:

$$\left\{\widehat{\mathsf{T}}, \left\{\widehat{\boldsymbol{\mathcal{M}}}^k\right\}\right\} = \arg\min_{\mathsf{T},\{\boldsymbol{\mathcal{M}}^k\}} \quad \alpha_g \mathcal{G}\left(\mathsf{T},\{\boldsymbol{\mathcal{M}}^k\}\right) + \alpha_p \mathcal{P}\left(\mathsf{T},\{\boldsymbol{\mathcal{M}}^k\}\right) . \tag{8.8}$$

In addition to the traditional geometric error term of the gold standard, a photometric error term is added:

$$\mathcal{P}\left(\mathsf{T},\{\boldsymbol{\mathcal{M}}^k\}\right) = \sum_k \rho^k \sum_{\boldsymbol{\mathcal{Y}}^{kl}\in\mathcal{N}_{\boldsymbol{\mathcal{M}}^k}} \left(\mathcal{I}_1\left(\mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{Y}}^{kl}\right)\right) - \mathcal{I}_2\left(\mathsf{K}_2\mathbf{w}\left(\mathsf{T}\boldsymbol{\mathcal{Y}}^{kl}\right)\right)\right)^2 . \tag{8.9}$$

In the next section, we provide implementation details.

8.4.3 Implementation Details

The neighborhoods are defined as a regular grid around each 3D point $\boldsymbol{\mathcal{M}}$ oriented according to the given normal $\mathbf{n}$. Each of the neighborhoods has a specific size defined by an edge length s. s is constrained upon an upper bound b of the edge length of a patch in the image. The length s and the bound b are iteratively estimated using Algorithm 8.1. The number of elements in the neighborhood can be adjusted depending on the desired sampling.

Algorithm 8.1: Compute the wanted edge length s of a 3D neighborhood, given a 3D points $\boldsymbol{\mathcal{M}}$, its normal $\mathbf{n}$, a pose T and the maximum desired size of the patch d in the image in pixels

```
Require: $\boldsymbol{\mathcal{M}}$ ,$\mathbf{n}$, $\mathsf{T}$, $d$
  $s \leftarrow 1$; $max_{border} \leftarrow d$
  repeat
    $s \leftarrow s * d/max_{border}$
    Create a neighborhood of size $s$ in 3D of $\boldsymbol{\mathcal{M}}$ and $\mathbf{n}$
    Create Patch1, Patch2 by projecting the neighborhood in both images
    Compute $size_1^{border}$ the distance between each corners of Patch1
    Compute $size_2^{border}$ the distance between each corners of Patch2
    $max_{border} \leftarrow \max\left(size_1^{border}, size_2^{border}\right)$
  until $|max_{border} - d| > \epsilon$
```

Since the area of the neighborhood is not updated during the minimization, the variation of scale within the 3D structure must be limited in order to insure that a patch does not become too small or too large within the image (aperture problem). This could happen if the structure's scale shrinks which would lead to an increase in the patch's relative size in the image. In order to prevent such behaviors, we force the norm of the estimated translation $\widehat{\mathbf{t}}$ to be always equal to 1. During the optimization $\widehat{\mathsf{T}}$ and the 3D points $\widehat{\mathbf{M}}^k$ are modified with the increments $\Delta\widehat{\mathsf{T}}$ and $\Delta\widehat{\mathbf{M}}^k$ given by a Gauss-Newton algorithm as follow:

$$\begin{cases} \widehat{\mathsf{T}} & \leftarrow & \Delta\widehat{\mathsf{T}}\,\widehat{\mathsf{T}} \\ \forall i, \;\; \widehat{\mathbf{M}}^k & \leftarrow & \frac{\widehat{\mathbf{M}}^k + \Delta\widehat{\mathbf{M}}^k}{\|\widehat{\mathbf{t}}\|} \\ \widehat{\mathbf{t}} & \leftarrow & \frac{\widehat{\mathbf{t}}}{\|\widehat{\mathbf{t}}\|} \end{cases} \tag{8.10}$$

This update does not modify the value of geometric constraint (Eq. 8.2) because it is independent of the scale.

One important thing to note, when implementing such a method, is that the sparsity one can find in bundle adjustment is still present. The block structure of the Jacobian (see Figure 3.10) stays unchanged. A new set of blocks is added corresponding to the jacobian $\mathsf{J}_{\mathcal{P}}$. The storage requirement of the jacobian (for 2 cameras) switch from $4n * (12 + 3n)$ to $(4n + a^2n) * (12 + 3n)$ with a the resolution of the 3D neighborhoods.

8.5 Experiments and Results

The implementation of our method was performed using Matlab. The optimizer is a classic Gauss-Newton, which stops after 50 iterations or when the norm of the update step is less than 10^{-9}. For initialization we used the normalized 8-point to get the pose and we obtain the 3D points using the optimal triangulation [Hartley and Sturm, 1997]. For all the experiments, we used a maximum patch edge length of 35 pixels and a neighborhood resolution (a) of 35×35. For the threshold introduced in Equation 8.7 we used $\tau_1 = 15$, such a threshold will not make our method dependent on the bad behavior of the Gold Standard since we can easily assume that it brings the points $\widehat{\mathbf{m}}_i^k$ in such a range and $\tau_2 = 0.3$, which is a very loose barrier to guarantee that the two patches are close enough to each other. We first discuss the experiments on synthetic data and then present results using real image pairs.

8.5.1 Synthetic Experiments

In order to generate the synthetic images we used a 3D pyramid with the top node cut out to create a flat area, the object used is pictured in Figure 8.3. All the faces were textured using real images. Harris corners were selected in the first image and transferred to the second image using the correct transformation $\overline{\mathsf{T}}$. This creates true correspondences of properly textured points. For our experiments we then perturbed these 2D localization by a zero mean Gaussian noise of variance σ. For each of the experiments, we tested our method using the exact normals and also using a fronto-parallel (FT) assumption (i.e. $\mathbf{n} = [0, 0, 1]^\top$). We compare our result to the one obtained with the Gold Standard.

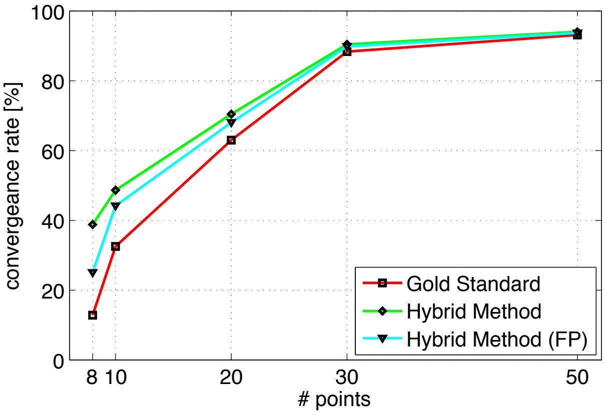


Figure 8.4: **Convergence rate with varying number of correspondences**: It shows results of the experiments with increasing number of points and a Gaussian noise of 0.5. It demonstrate a clear benefit for low number of matches even with a fronto-parallel assumption (FP).

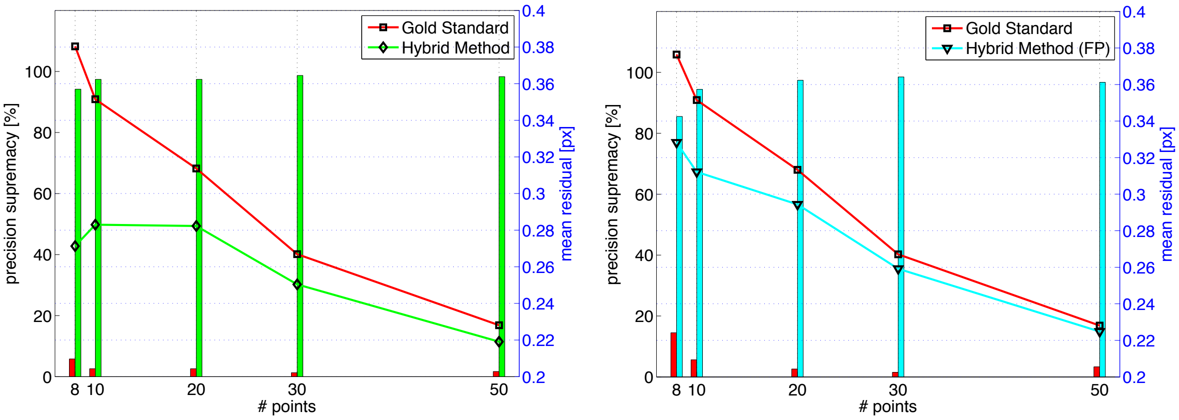
We consider that an approach has converged when the resulting pose $\widehat{\mathsf{T}}$ reprojects the exact 3D Points $\overline{\mathcal{M}}^k$ in a range inferior to σ (the variance of the Gaussian noise) of correct image points $\overline{\mathbf{m}}_2^k$. For example with a gaussian noise of 0.5, if the residual is inferior to 0.5, it has converged. When numerical precision based on measurement obtained with our Hybrid Method (respectively Hybrid Method with FP assumption) is compared to the Gold Standard, it is solely calculated when both methods converged. This gives an idea on precision repeatability.

The first experiment measures the impact of the number of available correspondences on our hybrid method compared to the gold standard. We perform bundle adjustments for 625 different poses, using a Gaussian perturbed 2D measurements ($\sigma = 0.5$). Figure 8.4 sustains that the additional information represented by the difference of intensity avails a better convergence rate, especially with a low number of points. Figure 8.5 demonstrates that our approach performs better and improves the precision with or without known normals.

We then test the behavior of our algorithm against increasing noise in localization of extracted features. For each level of noise, we use 625 different poses and 50 points correspondences. The result are summarized for the convergence rate in Figure 8.6 and for the precision in Figure 8.7. When the noise level is small our approach has the same convergence rate and precision as the Gold Standard . When the noise increases the Gold Standard convergence rate degrades faster than with our method (with and without known normals).

The next batch of experiments targeted the stability with respect to the distance to the scene. We used 50 points, a Gaussian noise of 0.5, and 625 poses for each depths. The result in Figure 8.8 shows again that our method performs better than the Gold Standard even with the fronto-parallel assumption. The precision obtained with our method follows the behavior shown previous experiments.

Finally, experiments with presence of blur and noise in the image were conducted. It



(a) Using correct normals (b) Using fronto-parallel (FP) assumption

Figure 8.5: **Impact of Number of Measurements**: we compare the Gold Standard against our Hybrid Method, in presence of a Gaussian noise (0.5) with increasing numbers of point pairs. The plots display the obtained mean registration error after convergence. The Bars represent proportion percentage by which a method out performs the other. It shows precision improvements and repeatable performances with or without a fronto-parallel assumption.

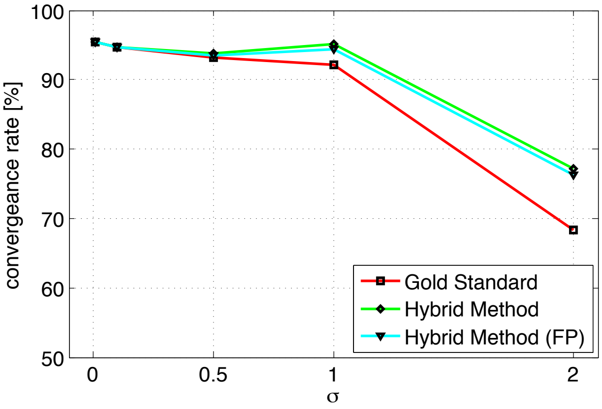
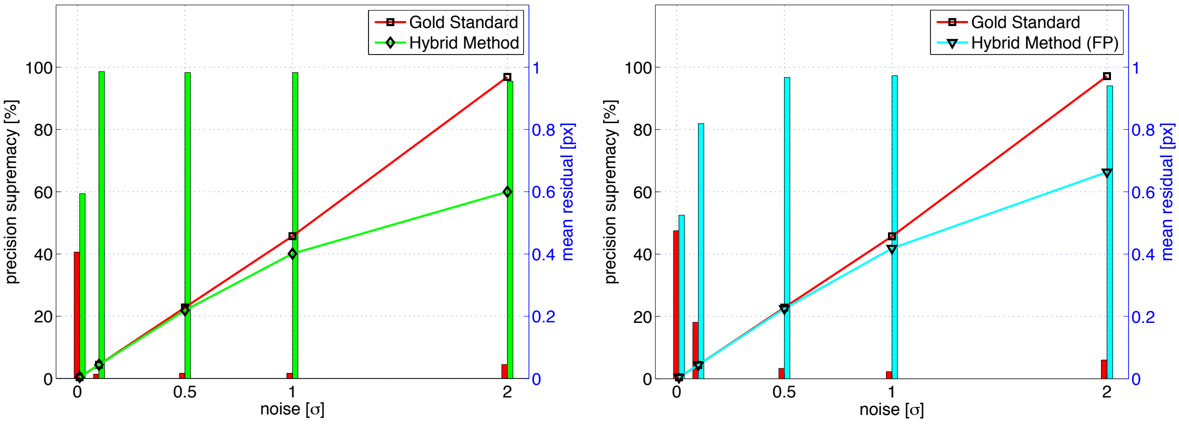


Figure 8.6: **Convergence rate with respect to increasing noise**. We show the results of an experiment using 50 points and varying Gaussian noise add to the 2D localization. It shows that the additional intensity information enables our approach to be more robust to bad measurements.



(a) Using correct normals

(b) Using fronto-parallel (FP) assumption

Figure 8.7: **Impact of Noisy of Measurements**: we compare the Gold Standard against our hybrid method with 50 points in presence of an increasing Gaussian perturbation of the 2D point locations. The plots display the obtained mean registration error after convergence. The Bars represent proportion percentage by which a method out performs the other. It shows precision improvements and repeatable performances with or without a fronto-parallel assumption.

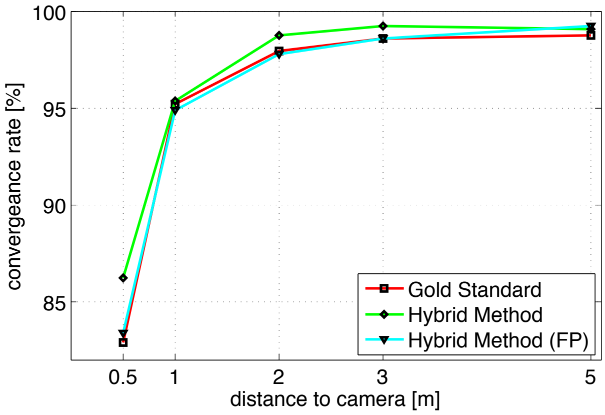


Figure 8.8: **Convergence Rate with changing scale**. We show the effect of an increasing distance from the scene with 50 points and a Gaussian noise of 0.5. It de monstrates the stability of the proposed approach even with scale differences between the image.

has shown that the convergence rate of our approach was only affected when there were massive perturbations in the image intensities. The accuracy was naturally degraded by the noise and blur. It should be noted that in the fronto-parallel approximation's case the performances were less affected than in the case where normal were known. This can be explained by the fact that the influence of the approximations of the normal is larger than noisy or blurry measure within the image.

8.5.2 Real Scenes Experiments

We tested our approach to obtain augmented images for VID. In order to recover the 3D pose of the image, we used an image manually registered to the 3D model. Using SIFT and a robust estimator we obtained a set of correct correspondences as shown in Figure 8.9. Then we applied our algorithm to these correspondences. We used the fronto-parallel assumption during the non-linear estimation. Once the rotation, and the translation are recovered the length of baseline is recovered manually. This shows one of the possible application of our algorithm. It should be pointed out that the extractor of the original SIFT (difference of Gaussian) might not give the best features for our algorithm since it does not guarantee that the extracted point is locally well textured. One way to maybe improve the resulting pose would be to use the response of the difference of Gaussian in scale space instead of the bare intensity.

Conclusion

In this chapter, we showed that merging geometric and photometric information in a unified cost function at the final non-linear minimization for the relative pose estimation process constraints better the results. As shown through the experiments on synthetic data, the proposed method is valuable for anyone using the classical Gold Standard algorithm for relative pose estimation since by simply modifying the cost function based on re-projection error by the one we propose, the results will be more robust. Unfortunately the additional computational burden might limit its use for real-time application but the extra precision could prove to be valuable. For the general application of this method we would need to integrate the normal estimation as in [Murray and Little, 2005, Furukawa and Ponce, 2007]. Additionally a minimal parametrization of the relative pose [Skarbek and Tomaszewski, 2009], which fixes the gauge, should improve the behavior of the optimizer.

This method supposes that the intensity measurements were not correlated with the point locations, which is generally not true and this should be investigated in the future. Nevertheless, the assumption leads to better pose estimates that only using the geometric information. This assumption was used because the geometry-based cost function helps constraining the intensity-based one that tends to diverge if not well initialized.

Additional studies (not shown in this thesis) have demonstrate that this method was particularly valuable when calibrating cameras [Acero, 2009]. Camera calibration using a known pattern offers the perfect setup for this hybrid approach because the structure of the pattern is known. This gives access to perfect normals and the known colors of

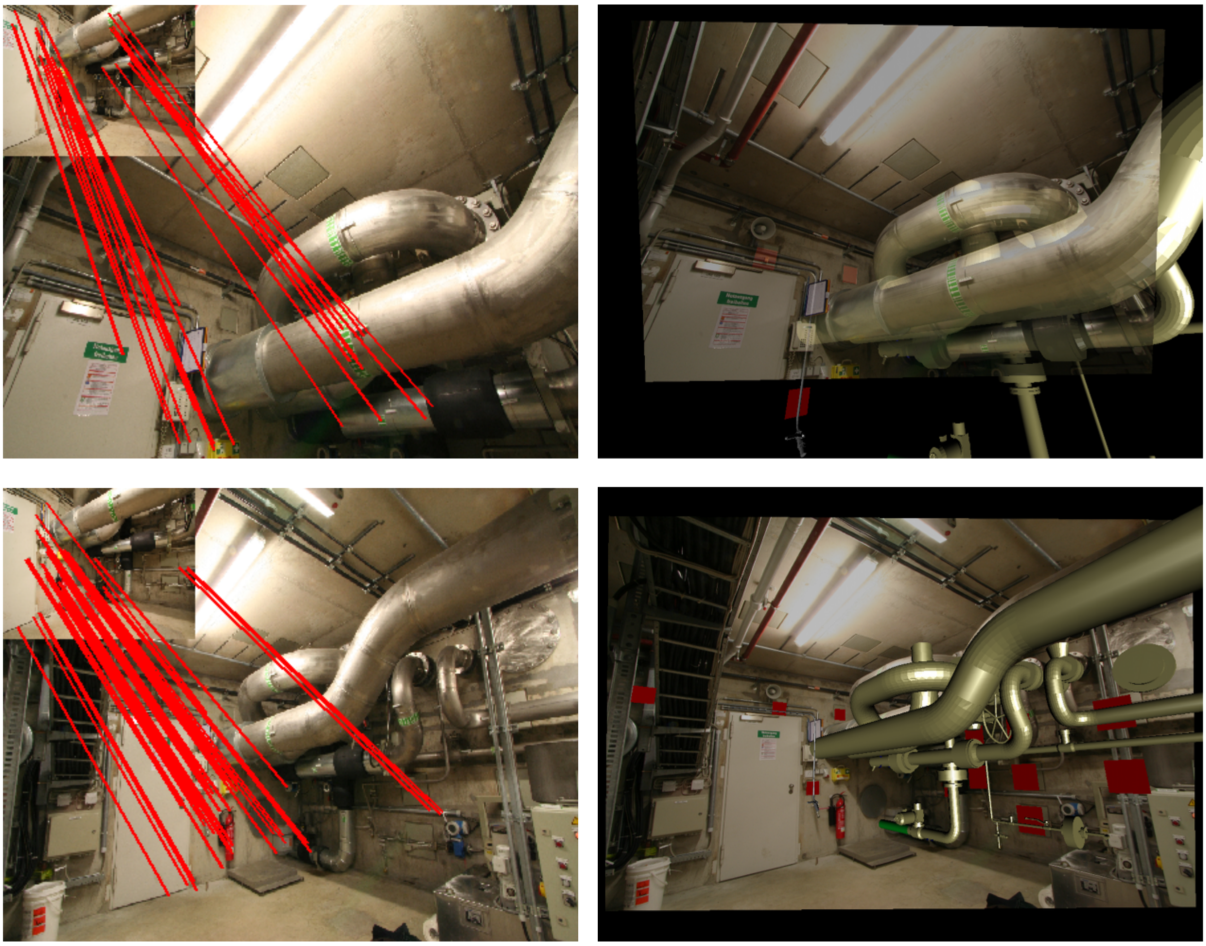


Figure 8.9: **Hybrid Methods Applied to Images form an Industrial Compound**: the left hand graphics represents the corresponding pair of points corrected using our method, the upper snapshot is the original keyframe; the right hand images display the resulting augmentation displayed in VID.

the pattern allows to directly calculate differences between image intensities and expected pattern colors (white or black) and not between each pair of images.

In the next chapter, we present another way to employ a hybrid approach when constraints between intensity is the only available source of measurement. This allows one to compute accurately a full pose from a single keyframe.

CHAPTER

NINE

FULL POSE FROM A SINGLE KEYFRAME

Photo-based Augmentation is a growing field in particular for Industrial Augmented Reality (IAR) applications. Registration is at the core of every photo-based AR software. When a single keyframe is used, the unknown length of the baseline between the two cameras has to be estimated in order to superimpose virtual models onto the image. In this Chapter, we present an automatic algorithm to augment the relative pose, estimated using a single keyframe, into a full pose that will permit superimposition. We estimate the length of the baseline by propagating known 2D-3D correspondences to the target image using perspectively corrected template matching and followed by a refinement of the estimated full pose that combines geometric and photometric information. We show that the hybrid method introduced in Chapter 8 can be applied for aligning image to CAD model, for example with images registered using the interactive registration method presented based on Anchor-plates in Chapter 6.

9.1 Difference between a Relative and a Full Pose

For Notations and a more general introduction to 3D computer vision the reader is referred to Chapter 3.

We designated a keyframe as an image registered to a 3D coordinate system (e.g. the CAD coordinate system). We suppose, without lose of generality, that the transformation T_1 between the scene and the keyframe is the identity. We want to estimate the pose of second camera using the available keyframe. For simplicity, we write the transformation to the second camera as T.

When estimating a relative pose between the keyframe and the second camera, we can only determine the relative translation up to an unknown scale factor. Therefore, we suppose that $\mathbf{t}$ is a unit vector for which we have to find the correct scaling. We define the full pose parametrized by the scale s as:

$$\mathsf{T}(s) = \begin{bmatrix} \mathsf{R} & s\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} . \tag{9.1}$$

The bundle adjustment cost function, defined in Equation 3.69, has a gauge freedom. A method that minimizes such this cost function cannot estimate the true scale of the observed structure because this cost is invariant to changes in scales:

$$\begin{aligned}
\forall s \neq 0, \quad \mathcal{G}\left(\mathsf{R}, s\mathbf{t}, \left\{s\boldsymbol{\mathcal{M}}^k\right\}\right) &= \mathcal{G}\left(\mathsf{T}\left(s\right), \left\{s\boldsymbol{\mathcal{M}}^k\right\}\right) && (9.2)\\
&= \frac{1}{2}\sum_{k=1}^{n}\left(\begin{array}{r} \left\|\mathsf{K}_1\mathbf{w}\left(s\boldsymbol{\mathcal{M}}^k\right) - \widetilde{\mathbf{m}}_1^k\right\|^2 \\ + \left\|\mathsf{K}_2\mathbf{w}\left(\left[\mathsf{R}s\boldsymbol{\mathcal{M}}^k + s\mathbf{t}\right) - \widetilde{\mathbf{m}}_2^k\right\|^2 \end{array}\right) && (9.3)\\
&= \frac{1}{2}\sum_{k=1}^{n}\left(\begin{array}{r} \left\|\mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{M}}^k\right) - \widetilde{\mathbf{m}}_1^k\right\|^2 \\ + \left\|\mathsf{K}_2\mathbf{w}\left(s\mathsf{T}(1)\boldsymbol{\mathcal{M}}^k\right) - \widetilde{\mathbf{m}}_2^k\right\|^2 \end{array}\right) && (9.4)\\
&= \frac{1}{2}\sum_{k=1}^{n}\left(\begin{array}{r} \left\|\mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{M}}^k\right) - \widetilde{\mathbf{m}}_1^k\right\|^2 \\ + \left\|\mathsf{K}_2\mathbf{w}\left(\mathsf{T}(1)\boldsymbol{\mathcal{M}}^k\right) - \widetilde{\mathbf{m}}_2^k\right\|^2 \end{array}\right) && (9.5)\\
&= \mathcal{G}\left(\mathsf{T}\left(1\right), \left\{\boldsymbol{\mathcal{M}}^k\right\}\right) && (9.6)\\
&= \mathcal{G}\left(\mathsf{R}, \mathbf{t}, \left\{\boldsymbol{\mathcal{M}}^k\right\}\right)\ . && (9.7)
\end{aligned}$$

Note that we take some liberty with the notation such that $s\boldsymbol{\mathcal{M}}^k = \left[s\mathbf{M}^{k^\top}\ 1\right]^\top$.

In this chapter we focus on the estimation of the unknown translation length/scale s which is necessary to augment the target image.

A common method to recover the scale s is to manually define a correspondence between a known 3D point and an image point or using a known 3D distance in object space. These two methods will be briefly described in the next two subsections.

9.1.1 Scale from a 3D point

Using a given correspondence between a point $\mathbf{c}_2$ in the target image and a 3D point $\boldsymbol{\mathcal{C}}$ defined in the coordinate system of the first camera, which satisfies:

$$\mathsf{T}\left(s\right)\boldsymbol{\mathcal{C}} \propto \mathsf{R}\mathsf{C} + s\mathbf{t} \propto \mathbf{p}_2 \text{ with } \mathbf{p}_2 = \mathsf{K}_2^{-1}\mathbf{c}_2\ , \qquad (9.8)$$

the translation scale can be deduced as follows:

$$\text{if } \left\|\left[\mathbf{p}_2\right]_\times \mathbf{t}\right\| \neq 0 \quad \Rightarrow \quad s = -\frac{\left(\left[\mathbf{p}_2\right]_\times \mathbf{t}\right)^\top \left[\mathbf{p}_2\right]_\times \mathsf{R}\mathsf{C}}{\left\|\left[\mathbf{p}_2\right]_\times \mathbf{t}\right\|^2}\ . \qquad (9.9)$$

9.1.2 Scale from a 3D distance

The determination of the scale can also be performed using a known 3D distance. The norm of a 3D reconstructed (using the unit translation) segment visible in both image is estimated and then the ratio between the obtained norm and the known 3D distance gives the scale s. Since $\mathsf{R}\mathsf{M} + \mathbf{t} \propto \mathbf{p}_2$ we can deduce:

$$\left[\mathbf{p}_2\right]_\times \left(z\mathsf{R}\mathbf{p}_1 + s\mathbf{t}\right) = \mathbf{0} \Rightarrow z\left[\mathbf{p}_2\right]_\times \mathsf{R}\mathbf{p}_1 = -s\left[\mathbf{p}_2\right]_\times \mathbf{t}, \qquad (9.10)$$

with $\mathbf{M} = [x\, y\, z]^\top$ and $\mathbf{p}_1 = \mathbf{w}(\mathbf{M})$.
We define $\mathbf{a} := [\mathbf{p}_2]_\times \mathsf{R}\mathbf{m}$ and $\mathbf{b} := [\mathbf{p}_2]_\times \mathbf{t}$, we can express the depth of a 3D point with respect to the length of the baseline s, as:

$$z = -s\frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|^2} \tag{9.11}$$

Let $\mathbf{M}^1$ and $\mathbf{M}^2$ be the triangulated 3D points using the pairs of observations $(\mathbf{m}_1^1, \mathbf{m}_2^1)$ and $(\mathbf{m}_1^2, \mathbf{m}_2^2)$ using R and $\mathbf{t}$. Since $s > 0$ this leads to:

$$\left\|\mathbf{M}^1 - \mathbf{M}^2\right\| = \left\|z_1\mathbf{p}_1^1 - z_2\mathbf{p}_1^2\right\| = s\left\|\frac{\mathbf{a}_1^\top \mathbf{b}_1}{\|\mathbf{a}_1\|^2}\mathbf{p}_1^1 - \frac{\mathbf{a}_2^\top \mathbf{p}_1^2}{\|\mathbf{a}_2\|^2}\mathbf{m}_2\right\| . \tag{9.12}$$

Knowing the distance $\|\mathbf{M}^1 - \mathbf{M}^2\|$ one can find the scale s of the translation by:

$$s = \frac{\|\mathbf{M}^1 - \mathbf{M}^2\|}{\left\|\frac{\mathbf{a}_1^\top \mathbf{b}_1}{\|\mathbf{a}_1\|^2}\mathbf{p}_1^1 - \frac{\mathbf{a}_2^\top \mathbf{p}_1^2}{\|\mathbf{a}_2\|^2}\mathbf{m}_2\right\|} . \tag{9.13}$$

Both of these approaches are neither automatic nor make use of information linked to the original keyframe established during its registration. In the next section, we review other solutions existing in the literature to estimate a full pose using keyframes.

9.2 Prior Art in Translation Scale Estimation

The problem of determining the scale of a translation for a relative pose was studied intensively in the field of ego-motion estimation and visual odometry for fully calibrated multi-camera system. For this scenario, they assume known the full pose of each camera available in their system. The standard method [Nister et al., 2004] is based on 3D points tracks that are used directly to fix the scale of the relative pose. This is similar to the idea of extending the relative pose using one 3D point (Section 9.1.1). This requires consistent point tracks across the views of a stereo rig and over time. Therefore the stereo rig has to have some overlap. This assumption was lifted by Kim et al. [2007] for a general multi-camera system. They first compute the relative rotation of the multi-cameras system by averaging the different relative rotation. Then, the scale estimation problem is expressed as a triangulation problem where the location of the first camera in the multi-camera setup is triangulated from the motion of each single camera. Using a similar setup Clipp et al. [2008] propose a method that can estimate a 6 degrees of freedom motion between each acquisitions. They estimate an essential matrix for the left camera and estimates the scale of the translation using one temporal correspondences from the right camera. Unfortunately both of these methods suffer from critical motions. To overcome this problem, Clipp et al. [2009] propose to use a multi-camera system with a limited overlap. They compute the rotation using two temporal correspondences for the left image of the stereo rig and one for the right images and the translation from one "4-views" 3D point.

The idea to use registered images or keyframes are very popular in Augmented Reality where 3D models are often available. One of the first system based on keyframes was limited to 2D similarity between frames and therefore was not suffering from the unknown scale problem [Stricker, 2001, Stricker and Kettenbach, 2001]. Chia et al. [2002] get around the scale problem by using two keyframes simultaneously to fix the gauge. Their system matches points to the previous temporal frame and transfers them to the keyframes. Therefore, they directly have a full pose instead of just a relative pose that would need to be extended. Bleser et al. [2005] estimate full pose instead of relative pose by first obtaining the depth of tracked SIFT point based on a CAD model. Similarly Platonov et al. [2006] use good features to track [Shi and Tomasi, 1994] and estimate the scale by triangulating point on the CAD model. Following the same trend, Najafi [2007] use textured CAD model to render a frame on a spherical surface that are then used as keyframes. Again they can directly use full pose estimation as they have access to a complete and correct CAD model of the scene being captured. We presented in [Georgel et al., 2008] a method that could extend a relative pose using partial CAD information. It is based on a robust plane segmentation and matching approach and could handle scene clutter. Unfortunately, it requires some feature points to be detected on planar structure, which is a hard assumption in an industrial setup.

These methods requires either a multi-cameras system, several keyframes or some dense CAD model to be usable. Our method leverages these constraints by using only one keyframe and limited 3D information known in the keyframes. After estimating a relative pose, we perform a search for the the known 2D-3D correspondences to estimate a full pose, similar to a template matching.

Method based on template matching are not novel but are often limited or highly expensive. For example, Kameda et al. [2004] match the current frame to a CAD model using registered CCTV cameras. They create affine rectified templates of selected landmarks using registered cameras and the CAD models. They then search for these landmarks in the current view to be register via a template matching on the complete image. A match is validated if secondary landmark confirm the hypothesis. Here, no relative pose is computed between the images, the images are just used to support a 2D-3D registration. Vacchetti et al. [2003, 2004a] also try to match the current view to a keyframe in a tracking framework. They solve the wide baseline problem by transforming it to a narrow baseline problem by using the most recent pose estimate available from their tracking. The matching is then based on linearized homography (affine transformation) using the current Jacobian.

The disadvantage of the previous works is that it only uses an approximation of the warping, which limits their use for wide baseline estimation. The novelty of our approach lies in the fact that we parametrized the homographic warping used in the template matching based on the translation scale and the estimated relative pose, which allows for a candidate pose to self-verify based on image information. Additionally we require only one 2D-3D correspondence from the keyframe.

Stricker and Navab [1999] also use a previously registered image to estimate the pose and the change in focal length/zoom of the camera by propagating information from the keyframe. Unfortunately the estimation was only linear and the correspondences were given by hand. In comparison, our method refines the estimated non-linear parameters using both photometric and geometric information from the images in non linear least square manner using the method developed in Chapter 8.

9.3 Translation Scale Estimation from Perspective Template Matching

Here we introduce our new method to estimate the length of the baseline using template matching. We henceforth assume that we have access to a number of correspondences $(\mathbf{c}_1, \boldsymbol{\mathcal{C}})$ between the keyframe and the model, referred as the control points. These correspondences of a 2D point $\mathbf{c}_1$ with a 3D point $\boldsymbol{\mathcal{C}}$ are usually established during the registration of the keyframe. Let $\mathbf{l}_2$ be the epipolar line in the second image induced by the point $\mathbf{c}_1$ and the relative pose $\mathsf{T}(1)$ in the target image. All points $\mathbf{c}_2$ on $\mathbf{l}_2$ correspond to a unique scale s. Similar to equation (9.9), this bijective relation is deduced using $\mathsf{T}\left(s\right)\boldsymbol{\mathcal{C}} \propto \mathsf{K}_2^{-1}\mathbf{c}_2$ as follows:

$$\forall \mathbf{c}_2 \in \mathbf{l}\,,\ \left[\mathsf{K}_2^{-1}\mathbf{c}_2\right]_\times (\mathsf{RC} + s\mathbf{t}) = \mathbf{0} \Rightarrow s = -\frac{\left(\left[\mathsf{K}_2^{-1}\mathbf{c}_2\right]_\times \mathbf{t}\right)^\top \left[\mathsf{K}_2^{-1}\mathbf{c}_2\right]_\times \mathsf{RC}}{\left\|\left[\mathsf{K}_2^{-1}\mathbf{c}_2\right]_\times \mathbf{t}\right\|^2}, \tag{9.14}$$

this relation is true for all points that satisfy $\left[\mathsf{K}_2^{-1}\mathbf{c}_2\right]_\times \mathbf{t} \neq \mathbf{0}$. This particular case is discussed in Section 9.4.2.

Furthermore, if we suppose that $\boldsymbol{\mathcal{C}}$ is locally planar and that $\mathbf{n}$ ($\|\mathbf{n}\| = 1$) is a normal vector to this plane (which can be obtained from a CAD model), each of the points $\boldsymbol{\mathcal{C}}$ induces a set of homographies defined as:

$$\mathsf{H}\left(s, \pi_{\boldsymbol{\mathcal{C}}}\right) = \mathsf{R} - s\frac{\mathbf{t}\mathbf{n}^\top}{d}\ , \tag{9.15}$$

between the keyframe and the image to register, with $\boldsymbol{\pi}_{\boldsymbol{\mathcal{C}}} = \left[\mathbf{n}^\top, d\right]^\top$ the plane around $\boldsymbol{\mathcal{C}}$ and d being the distance between the point $\boldsymbol{\mathcal{C}}$ and camera center of the keyframe:

$$\boldsymbol{\pi}_{\boldsymbol{\mathcal{C}}}^\top \boldsymbol{\mathcal{C}} = 1\ . \tag{9.16}$$

For each of these homographies, we have a one to one mapping between neighbors of $\mathbf{c}_1$ and the neighbors of $\mathbf{c}_2$. Therefore, it is possible to define an intensity based criterion to match $\mathbf{c}_1$ to the correct $\mathbf{c}_2$. Our template matching score $f_{\boldsymbol{\mathcal{C}}}\left(s\right)$ is defined as follows:

$$f_{\boldsymbol{\mathcal{C}}}\left(s\right) = \mathrm{SM}\left(\mathcal{I}_1, \mathsf{H}^{-1}\left(s, \boldsymbol{\pi}_{\mathbf{C}}\right)\left(\mathcal{I}_2\right)\right)\,, \tag{9.17}$$

with $\mathcal{I}_1$ (respectively $\mathcal{I}_2$) being an image patch defined around $\mathbf{c}_1$ (respectively $\mathbf{c}_2$) and SM being any similarity measure. The template search can be then expressed as an extremum search on the one dimensional function $f_{\boldsymbol{\mathcal{C}}}$. This search is efficient because Equation 9.14 guarantees a unique s for each points of $\mathbf{l}_2$. So finding the scale s can be summarized as computing $f_{\boldsymbol{\mathcal{C}}}$ for each $\mathbf{c}_2 \in \mathbf{l}_2$ and looking for the extremum of the function. A schematic

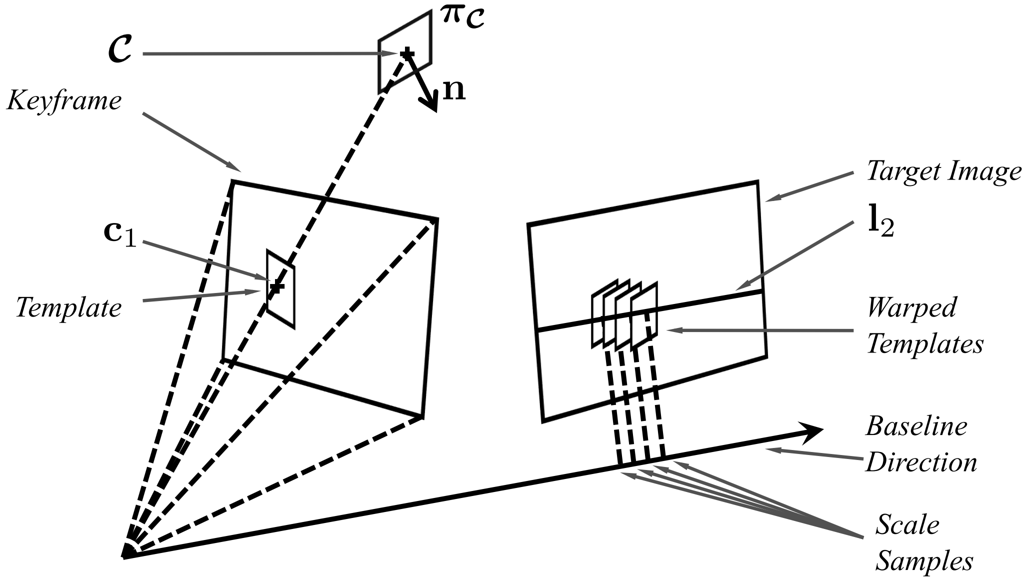


Figure 9.1: **Scale from one propagated 2D-3D correspondence**: the 3D point $\boldsymbol{\mathcal{C}}$ projects on $\mathbf{c}_1$ in the *keyframe* and $\mathbf{c}_1$ maps to the epipolar line $\mathbf{l}$ in the *target image*. The template matching is performed between the *template* around $\mathbf{c}_1$ and *warped templates* on $\mathbf{l}_2$. The warp is parametrized using the plan $\boldsymbol{\pi}_{\boldsymbol{\mathcal{C}}}$ and the *scale samples.*

of the search is shown in Figure 9.1. This discrete search is then refined by minimizing a nonlinear cost that combines both geometric and photometric information as expressed in the following section.

It is important to understand that this template search is different from a dense stereo depth estimation using parsing windows on rectified view [Scharstein et al., 2002] because we include change of scales.

9.3.1 Nonlinear Refinement

If the propagated 2D-3D correspondences $\left(\mathbf{c}_2^j, \boldsymbol{\mathcal{C}}^j\right)$ were added to the geometric cost (Eq. 8.2), it would stay optimal with respect to $\mathsf{T}(s)$ because it is optimal for $\left(\mathsf{T}(1), \left\{\boldsymbol{\mathcal{M}}^k\right\}\right)$ and the propagated points $\mathbf{c}_2^j$ have been selected to verify $\mathsf{K}_2\mathbf{w}\left(\mathsf{T}(s)\boldsymbol{\mathcal{C}}^j\right) = \mathbf{c}_2^j$. But the selected scale s is not optimal with respect to Equation 9.17 because it is discretely sampled over the epipolar lines. Therefore it needs to be refined. Using a similar approach as previously described in Chapter 8, which combines geometric and photometric information we create a hybrid cost function that estimates a full pose. First we define a least square photometric cost function based on the template matching results by:

$$\mathcal{P}_{\left\{\boldsymbol{\mathcal{C}}^j\right\}}\left(\mathsf{T}\right) = \sum_j \sum_{\boldsymbol{\mathcal{Y}}^{jl} \in \mathcal{N}_{\boldsymbol{\mathcal{C}}^j}} \left(\mathcal{I}_1\left(\mathsf{K}_1\mathbf{w}\left(\boldsymbol{\mathcal{Y}}^{jl}\right)\right) - \mathcal{I}_2\left(\mathsf{K}_2\mathbf{w}\left(\mathsf{T}\boldsymbol{\mathcal{Y}}^{jl}\right)\right)\right)^2 , \tag{9.18}$$

with $\mathcal{N}_{\boldsymbol{\mathcal{C}}^j}$ being the 3D neighborhood of $\boldsymbol{\mathcal{C}}^j$ defined by the plane $\boldsymbol{\pi}_{\boldsymbol{\mathcal{C}}^j}$.
This leads to the following least square minimization problem:

$$\left(\widehat{\mathsf{T}}, \left\{\widehat{\mathcal{M}^k}\right\}\right) = \arg\min_{\mathsf{T},\boldsymbol{\mathcal{M}}^k} \mathcal{G}\left(\mathsf{T}, \left\{\mathbf{M}^k\right\}\right) + \mathcal{P}_{\left\{\boldsymbol{\mathcal{C}}^j\right\}}\left(\mathsf{T}\right) \ . \tag{9.19}$$

We would like to emphasize two important facts. First that such a cost function is only locally convex around its minimum. Therefore, the non-linear minimization should be carefully initialized. This is done by finding an initial estimate of the scale using our template matching. And second that the problem's formulation does not have any gauge freedom [Morris et al., 2001] since we are estimating a full pose using real 3D data. This is one of the main difference to the method defined in the previous Chapter where we enforce the unit scale in order to obtain a stable minimization. In the next section implementation details is given and the performance of the approach is evaluated.

9.4 Experimental Results

All the experiments exposed in this section are designed to evaluate the precision and stability of the presented method for full pose from wide baseline keyframe. First we describe some detail of implementation (Section 9.4.1), then we discuss the behavior of the algorithm around the epipole (Section 9.4.2). Finally, we focus on synthetic simulations (Section 9.4.3) and demonstrate the usability of this method in the context of VID (Section 9.4.4).

9.4.1 Implementation Details

The experiments were all run using Matlab and then later integrated into VID. The initial parameter (relative rotation R, translation $\mathbf{t}$ (of unit norm) and structure $\boldsymbol{\mathcal{M}}^k$) are oriented (i.e. the points are triangulated in front of the camera) and are supposed to be optimal for Equation 8.2. Since we are oriented we only have to consider positive scale s. The method is divided into three steps. First, the template matching is performed (i.e. discrete search). The epipolar line is sampled every 5 pixel. The template size is 32×32 pixels. The similarity measure used for Equation 9.17 is the normalized cross correlation (we search for a maximum) which handles changes in illumination and contrast and the associated threshold $\tau = 0.8$. We tested using a sum of squares differences, which lead to slightly inferior results when used on real images. Test using the SSD gave similar results. Second, an initial scale is selected. We choose the best scale from the set of scales estimated from each of the template matchings. This is performed by applying the score (Equation 9.17) to all 2D-3D correspondences using all the obtained scales and by choosing the scale that maximized the sum of the score (2D-3D correspondences with score lower than τ are discarded). This provides an initial estimate for the refinement and a set of matched 2D-3D correspondences $\left\{\mathbf{c}_1^j, \boldsymbol{\mathcal{C}}^j\right\}$. Finally, the full pose is refined, and we estimate the full pose based on Equation 9.19. For the nonlinear minimization we normalize the gray scale intensity information between zero and one to give similar importance to $\mathcal{G}$ and $\mathcal{P}$. It seems that this minimization is better initialized than the

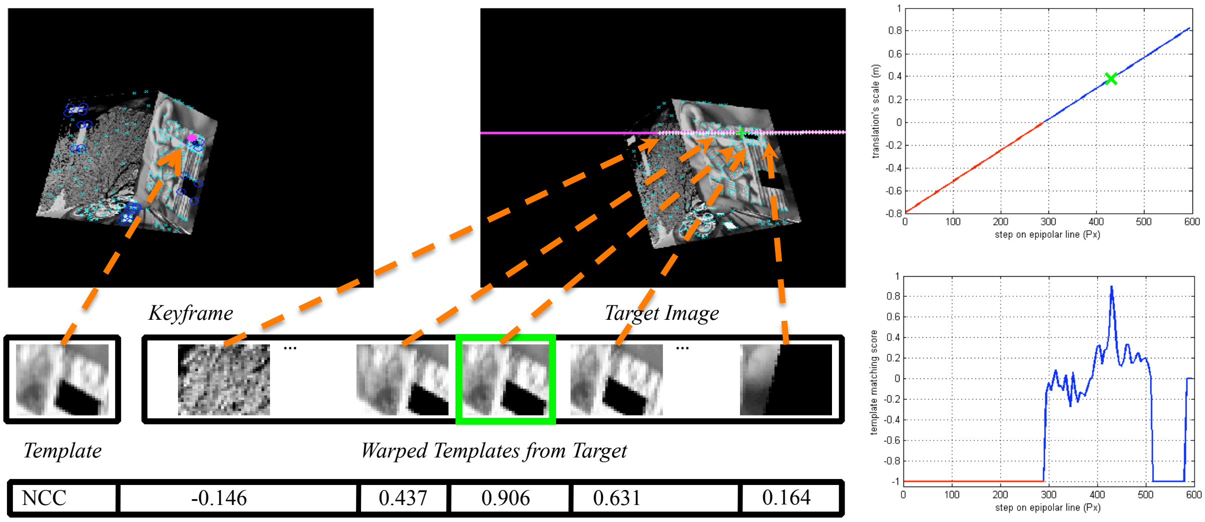


Figure 9.2: **Template Matching Scheme**: cyan crosses are Harris corner; blue circles 2D-3D correspondences; pink dot is the 2D-3D correspondences being currently evaluated, which maps to the pink epipolar line; and white crosses correspond to scale samples. Each of the samples relates to a warped template and a relative NCC score. The upper graph represents the evolution of the scale (negative scales in red are not considered) and the lower graph represents the NCC scores over all samples. Object highlighted in green correspond to the correct scale.

relative pose estimation from Chapter 8 therefore does not require a normalization by variance and selective activation. Additionally the fact that there is no gauge freedom offers more constraint on the optimization. The algorithm is summarized in 9.1 and a schematic of the process is described in figure 9.2.

9.4.2 Estimation around the Epipole

In this section, we describe the behavior of our method around the epipole. As mentioned in Section 9.3, the relation (Eq. 9.14) between a point on the epipolar line and the scale s is valid if and only if $\left[\mathsf{K}_2^{-1}\mathbf{c}_2\right]_\times \mathbf{t} \neq \mathbf{0}$. This occurs on the epipoles.

First, if $\boldsymbol{\mathcal{C}}$ projects on the first epipole $\mathbf{e}_1$ then $\mathbf{l}_2$ is reduced to a point in the second image $\mathbf{e}_2$ [Faugeras and Luong, 2001]. This does not allow to fix the scale.

Secondly, if $\mathbf{c}_2$ is selected to be the second epipole $\mathbf{e}_2$ then $\left[\mathsf{K}_2^{-1}\mathbf{c}_2\right]_\times \mathbf{t} = \mathbf{0}$. Because :

$$\begin{aligned}
\mathbf{e}_2 &= \mathsf{K}_2\mathbf{w}\left(\mathsf{T}(s)\boldsymbol{\mathcal{O}}_1\right) && (9.20)\\
&= \mathsf{K}_2\mathbf{w}\left(\mathsf{R}\mathbf{0} + s\mathbf{t}\right) && (9.21)\\
&= \mathsf{K}_2\mathbf{w}\left(s\mathbf{t}\right)\ , && (9.22)
\end{aligned}$$

Algorithm 9.1: Propagation of 2D-3D correspondences from a keyframe to a target image to obtain a full pose.

```
Input: A relative pose $\mathsf{T}$, the image point correspondences $\left\{\tilde{\mathbf{m}}_1^k, \tilde{\mathbf{m}}_2^k, \boldsymbol{\mathcal{M}}^k\right\}$, 2D-3D
       correspondences $\left\{\mathbf{c}_1^j, \boldsymbol{\mathcal{C}}^j\right\}$ and a threshold $\tau$
Output: A refined full pose $\widehat{\mathsf{T}}$ and structure $\widehat{\boldsymbol{\mathcal{M}}}_i$ to scale.
$\mathsf{F} := \mathsf{K}_2^{-\top}\,[\mathbf{t}]_\times\,\mathsf{R}\mathsf{K}_1^{-1}$;
$f_{total} := 0$;
$s_{best} := 0$;
foreach $\boldsymbol{\mathcal{C}}^j$ do
  $f_{max} := 0$;
  $f_{current} := 0$;
  foreach $\mathbf{c}_2^j \in \mathcal{I}_2 \cup \mathsf{F}\mathbf{c}_1^j$ do
    Compute $s$ from Eq. 9.14 ;
    if $(s > 0)\&(f_{max} > f_{\boldsymbol{\mathcal{C}}^j}(s))$ then
      $f_{max} = f(s)$;
      $s_{max} = s$;
    end if
  end foreach
  if $f_{max} > \tau$ then % correct scale found
    foreach $\boldsymbol{\mathcal{C}}^l$ do
      if $f_{\boldsymbol{\mathcal{C}}^l}(s_{max}) > \tau$ then % check if $\boldsymbol{\mathcal{C}}^l$ is matched by $s_{max}$
        $f_{current}+ = f_{\boldsymbol{\mathcal{C}}^l}(s_{max})$ ;
      end if
    end foreach
    if $f_{current} > f_{total}$ then
      $s_{best} = s_{max}$;
      $f_{total} = f_{current}$ ;
    end if
  end if
end foreach
if $s_{best} > 0$ then % Bring to scale
  $\mathbf{t} = s_{best} \times \mathbf{t}$ ;
  $\boldsymbol{\mathcal{M}}^k = s_{best} \times \boldsymbol{\mathcal{M}}^k$;
  Estimate $\left(\widehat{\mathsf{T}}, \left\{\widehat{\boldsymbol{\mathcal{M}}}^k\right\}\right)$ by minimizing Eq. 9.19 using $\left\{\boldsymbol{\mathcal{C}}^j\right\}$ ;
  return true;
else
  return false; % no candidate for full pose;
end if
```

which leads to

$$\begin{aligned}\left[\mathsf{K}_2^{-1}\mathbf{e}_2\right]_\times \mathbf{t} &= \left[\mathsf{K}_2^{-1}\mathsf{K}_2\mathbf{w}\left(s\mathbf{t}\right)\right]_\times \mathbf{t} && (9.23)\\ &= \left[s\mathbf{t}\right]_\times \mathbf{t} && (9.24)\\ &= \mathbf{0}\,. && (9.25)\end{aligned}$$

The phenomenon behind this singularity can be visualized in Figure 9.3. When $\mathbf{c}_2$ gets closer to the second epipole $\mathbf{e}_2$ the scale of the translation grows exponentially. It projects the camera towards infinity. A similar problem can be found in the optimal 2 view triangulation from Hartley and Sturm [1997] as pointed out by Kanatani et al. [2008].

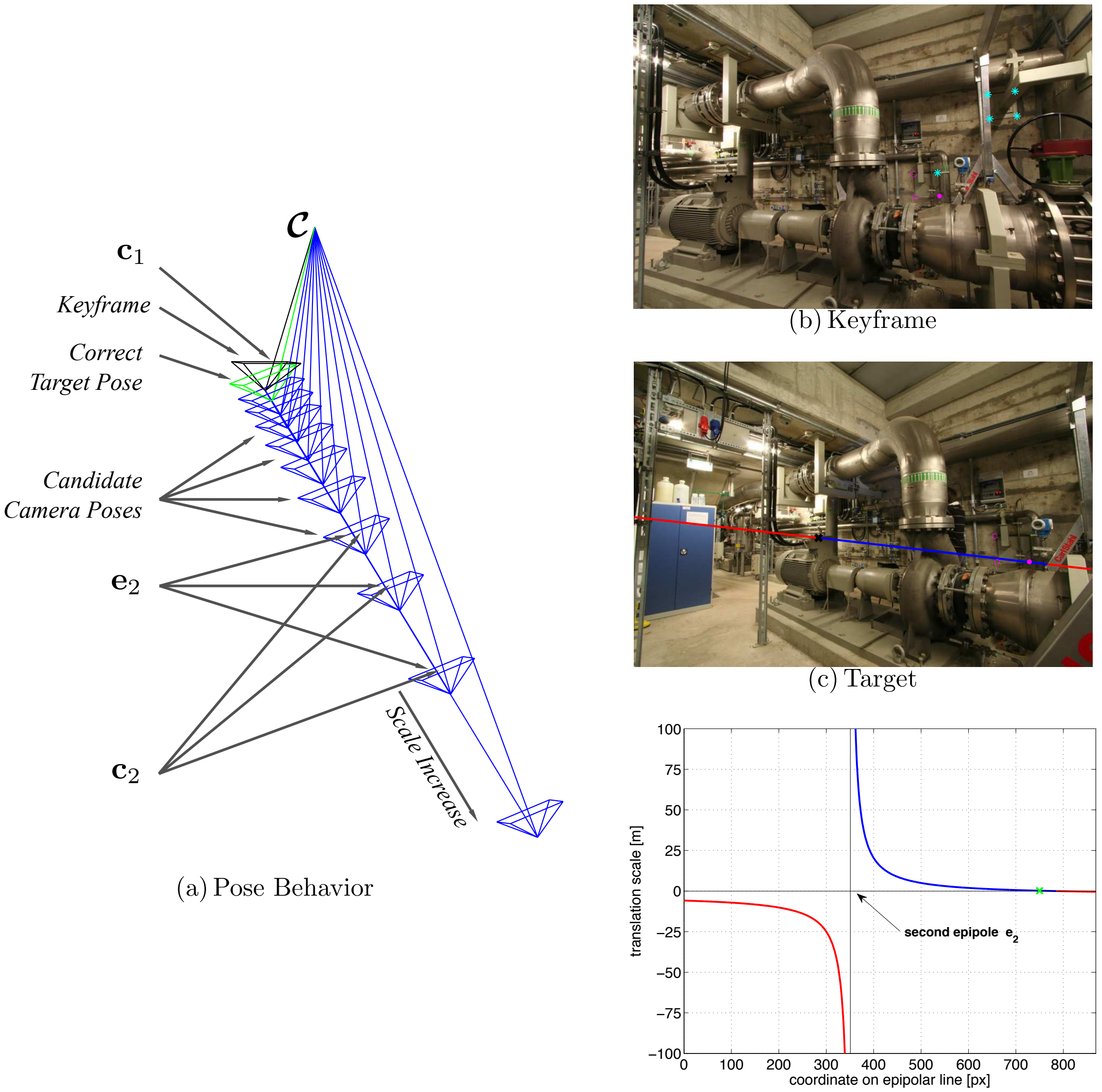
So the 3D point $\mathcal{C}$ used for the estimation of the scale cannot be on the line between the two cameras center $\boldsymbol{\mathcal{O}}_1$ and $\boldsymbol{\mathcal{O}}_2$. This can easily be checked by verifying that $\boldsymbol{\mathcal{C}}_1$ does not project on the epipole $\mathbf{e}_1$. And if $\boldsymbol{\mathcal{C}}$ does not project on $\mathbf{e}_1$ then it cannot project on the second epipole $\mathbf{e}_2$.

Nevertheless, if the second epipole is in the image domain it will need to be carefully dealt during the implementation of this approach as it create a discontinuity in the function which define s as shown in Figure 9.3(d). In our implementation, we remove $\mathbf{e}_2$ from the candidate of $\mathbf{c}_2$ and we verify that the transition of scale over the epipolar $\mathbf{l}$ is smooth. This is done to avoid that a shift of 1 or 2 pixel on the epipolar line multiply the scale hundred time. This would not be a realistic displacement of the cameras because it would mean that a slight registration offset would result in a massive change of the camera location.

9.4.3 Synthetic Simulations

In the following section we describe the experiments performed to study our method. The experiments are based on a synthetic model, which is formed of 3 textured planes. The motion between the images is perfectly known and is used as ground truth data. Harris corners [Harris and Stephens, 1988] are detected in the keyframe and then are propagated to the target image using the true motion. At most 154 Harris corners are used during the experiments. For visibility reasons this number might be smaller. On each plane of the synthetic model eight 2D-3D correspondences are marked, each of these correspondences is linked to a normal vector. The number of 2D-3D correspondences used, is at most 24 (if not specified otherwise) depending on their visibility. Examples of generated views are visible in figure 9.2. The changes in depth of the viewpoints is designed to be almost constant across all experiments (if not specified otherwise) in order to have comparable re-projection errors between the different experiments.

The error of an estimation is measured using the true (i.e. noiseless) 3D control points $\left\{\overline{\boldsymbol{\mathcal{C}}}^k\right\}$ and the estimated full pose $\widehat{\mathsf{T}}$. We project the 3D points with the estimated full pose and measure the distances to the 2D points obtained from the ground truth. The mean of this distance is our error measurement, this is often called the mean target residual error (RE). As in Chapter 8, whether the algorithm converged is decided based on the resulting RE. We declare that the method converged if its RE is bellow the noise level



(a) Pose Behavior

(b) Keyframe

(c) Target

(d) Scale Variation

Figure 9.3: **Epipole Singularity**: When the scale increases the projection $\mathbf{c}_2$ of the control point $\boldsymbol{\mathcal{C}}$ converges toward the epipole $\mathbf{e}_2$. At the limit, we have have singularity were the candidate pose is at infinity. Positive Scale are marked in blue. The pink dot represent the best candidates and pink circles are found matches. Cyan stars are unmatched control points. Black crosses correspond to the epipoles.

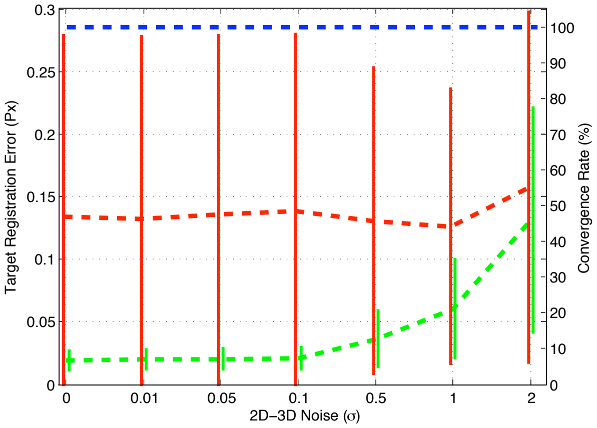


Figure 9.4: **Stability to Keyframe Registration Error**: This experiment was performed with increasing noise in the 2D points localization used to register the keyframe. The red line represents the mean target registration error (RE) after the template search, the green the RE after the nonlinear refinement, the blue the convergence rate and vertical bars standard deviations.

plus precision limit: 0.05 pixel (this value is explained in Section 9.4.3.1). All the mean RE plots are based RE that converged. Six experiments were conducted, which are now explained in detail.

9.4.3.1 Synthetic Experiment with Error in the Keyframe Registration

In order to verify the stability of the method towards misregistration of the keyframe, we simulate slightly wrong alignment between the 3D points and the 2D points. This is performed by adding Gaussian noise to the 2D points before computing the pose of the keyframe. This induces a small error in the alignment between the model and the image, which includes an error in the orientation of the normal. We ran these experiments with seven different levels of noise on hundred images. The results are summarized in figure 9.4.

The algorithm always converged to good solution with respect to the input noise. The refinement step always improves the result of the template matching. Furthermore, the resulting RE is small. These experiments also reveals the numerical limitation of the use of intensitiy information.A target registration error bellow 0.05 is rarely achieved. We speculate that this originates from the discrete method used to create the images and the loss of information after warping.

9.4.3.2 Synthetic Experiment with Error in the Relative Pose

In a second experiment, we test the stability of the method against an error in the relative pose estimate, because it is more than likely that feature points used to obtain the relative pose are localized with some error (c.f. Chapter 7). In order to simulate this error, a Gaussian noise is added to the Harris corner (2D points) in both keyframe and target. These perturbation has a direct impact on the quality of the initial relative pose, even

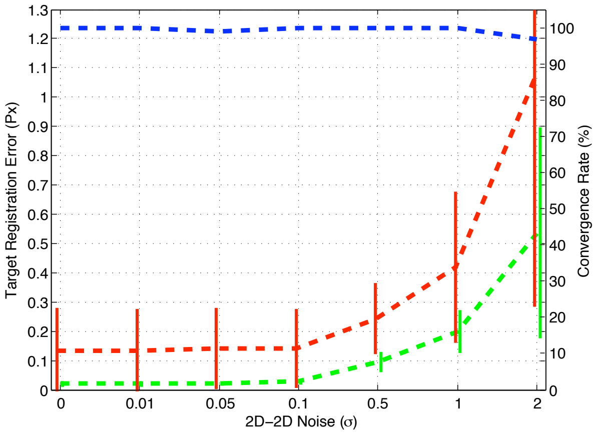


Figure 9.5: **Stability to Relative Pose Error**: This experiment was performed with increasing noise in the 2D points localization used for the relative pose estimation. The red line represents the mean target registration error (RE) after the template search, the green the RE after the nonlinear refinement, the blue the convergence rate and vertical bars standard deviations.

with the use of the Gold Standard algorithm (c.f. Chapter 8). We ran these experiments with seven different noise levels on hundred images. The obtained convergence rates and RE of the approach are summarized in figure 9.5.

The method rarely diverges (up to 4%). We assume that the algorithm only does not converge when the error in the relative pose is so large that the epipolar lines induced by the image points (of the 2D-3D correspondences) miss their true corresponding points by far. Again we see that the refinement step drastically improves the result obtained by the template matching. Furthermore, in comparison to the previous experiment 9.4.3.1 the RE obtained with the template matching grows at a faster pace than the one using additional refinement. This can be explained by the fact that we use additional information (photometric) during the nonlinear optimization which corrects also the relative pose estimated from the noisy measurements.

9.4.3.3 Stability with respect to Number of 2D-3D Correspondences

For this experiment we want to study the impact of the number of 2D-3D correspondences on our method. We randomly select a subset of the 2D-3D correspondences. The threshold for the convergence was set to 1 pixel error because the lower bound threshold (0.05) was selected using 24 2D-3D correspondences. Such a precision however cannot be expected when only using less control points. We used again hundred poses. The results of this experiment are presented in Figure 9.6.

The first comment, that can be drawn from this experiment, is that the number of 2D-3D correspondences has a direct impact on to the convergence rate. We suppose that the randomly selected 2D-3D correspondences are not always well visible (e.g. perspectively too distorted to be recognized in the target image). Such perspective distortions rarely happen with real images because the relative pose is often not computable in this case. For

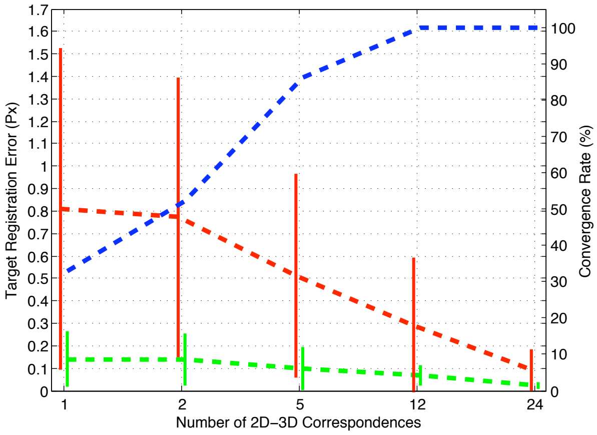


Figure 9.6: **Stability to the number of available 2D-3D Correspondences**: This experiment was performed with varying number of available 2D-3D correspondences in the Keyframe. The red line represents the mean target registration error (RE) after the template search, the green the RE after the nonlinear refinement, the blue the convergence rate and vertical bars standard deviations.

example SIFT is effective up to 30° of perspective distortion [Wu et al., 2008]. Secondly the precision of the method is satisfactory even when only one correspondence is available.

9.4.3.4 Stability to Scale Change

We then wanted to verify that our method was stable to scale changes between the two images. We sampled poses with five different zoom factors. Again, we use a threshold of 1 pixel error. The result are presented in Figure 9.7.

The variation of the RE magnitude is the consequence of the varying scale. When the object is closer to the camera the RE increase (automatically) even if the underlying error in pose is the same. This is because the distance are magnified by the zoom factor. Often the focal length is increased to compensate for this phenomena; we decide to not apply this idea in order to let the experiment describe the underlying problem. This experiment shows that the proposed method is stable even when the scale factor varies drastically.

9.4.3.5 Noise and Image Blur

In order to verify the performance of the approach in actual usage, one needs to know its stability with respect to noise image and blur. When images are acquired using a camera, noise is always present because of sensor limitation, and blur can occur when the camera is handled by a human and not fix on a tripod. To evaluate the characteristics of our scale estimation method towards this perturbation we perform two experiments. First, we add an independent Gaussian noise on both the intensity of the keyframe and the target image. Second we blur the keyframe and the target. The blur is performed using an increasing kernel size. Both experiments are performed on hundred images. The convergence threshold for both experiments id 0.5 pixel. The results for the noise is visible

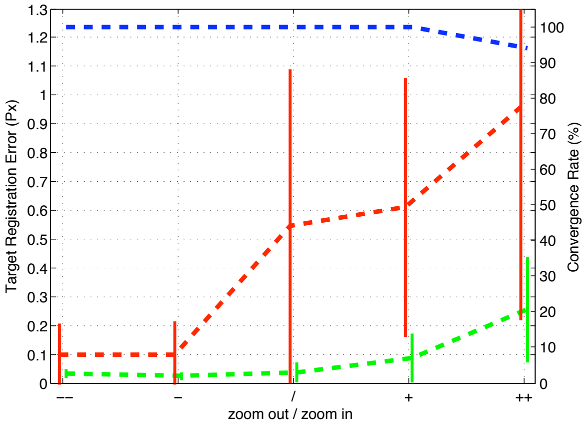
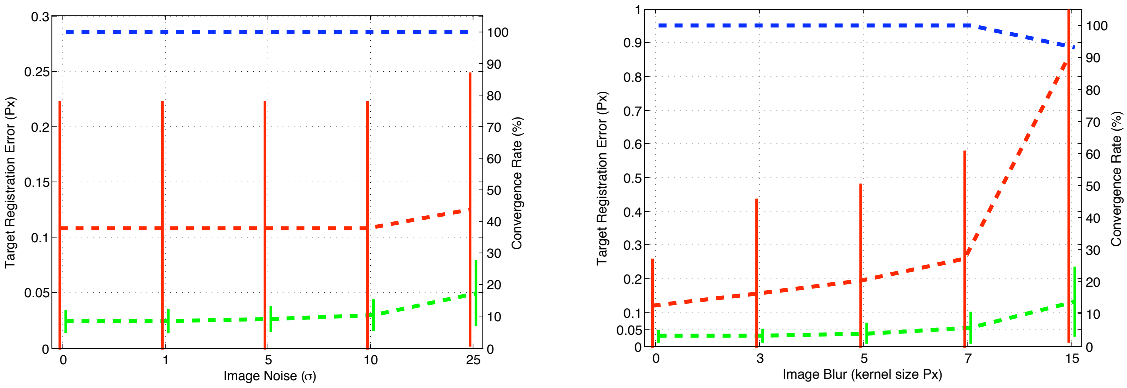


Figure 9.7: **Stability towards Change in Scene Scale**: The scene were rendered at different scale from strong zoom-out on the left to strong zoom-in. The red line represents the mean target registration error (RE) after the template search, the green the RE after the nonlinear refinement, the blue the convergence rate and vertical bars standard deviations.

in Figure 9.8(a) and for the blur in Figure 9.8(b).
The noise experiments demonstrate again than the refinement step is crucial to obtain a precise full pose. Furthermore, we can see that the standard deviation of the resulting error is small, which proves that we always reach a stable optimum even with massive disturbances in the images. This is because we simultaneously minimize the photometric and geometric cost over all the observations. The method does a good job of handling the loss of information due to the blurring effect. The obtained RE after the refinement, at maximum, doubles from the non-blurred image (smaller that 0.2 pixel).

9.4.4 Application to Automatic Pose Estimation from a Single Keyframe for Industrial Augmented Reality

We implemented the presented method within our Industrial Augmented Reality Software (Part II). The keyframes are registered using anchor-plates (Chapter 6). The corners of these anchor-plates and their corresponding image points are used as 2D-3D correspondences. The relative pose is estimated using SIFT points, RANSAC for the 8-point algorithm and the Gold Standard algorithm in order to obtain a relative pose. In order to register an image the user has to select a keyframe; it should have enough overlap to obtain a relative pose. The method is successfully used on power plant's images over the past years. It offers to the end user a fast and automatic method to register additional images. Some results are visible in Figure 9.9, 9.10, 9.11 and 9.12. This various conditions show the good behavior for our application. The method is mainly limited by the essential matrix estimation method, which heavily relies on SIFT matching. It has shown to be sufficient for our scenario. Unfortunately, images acquired under different atmospheric condition (e.g. with flash and without) rarely match. This would be definitely a direction of research for an even broader applicability.



(a) (b)

Figure 9.8: **Stability to Intensity Perturbation**: These experiments was performed with increasing image noise (a) and image blur (b). The red line represents the mean target registration error (RE) after the template search, the green the RE after the nonlinear refinement, the blue the convergence rate and vertical bars standard deviations.

Experiments demonstrate that the proposed method is stable during zooming, and is robust to noise as well as blur. The obtained results are satisfactory to be used as the standard method in VID to align images once some keyframe are available. The main reason for such accurate performances is the use of a well initialized hybrid nonlinear refinement that handles different types of data and therefore can deal with the noise existing in real measurements.

Conclusion

In this chapter, we presented an automatic method to extend a relative pose to a full pose. The relative pose is sufficient for many Computer Vision applications. However, in Augmented Reality the full pose is needed to correctly superimpose the virtual object onto the real view of the world. In such applications, relative pose is of limited use.

The method introduces a homographic warp that is parametrized by the translation length. It uses a hybrid 6 degrees of freedom pose estimation, which has no gauge freedom. This extends the method presented in Chapter 8. The hybrid pose estimation minimizes intensity differences and the re-projection error at the same time . We have demonstrated through extensive synthetic experiments the robustness and precision of the proposed method and shown its applicability in the context of our industrial photo-based augmented reality application. Not requiring multiple pre-registered images or multiple 2D-3D correspondences greatly broaden the application of keyframe for Photo-based Augmented Reality.

Though the method was used for calibrated cameras, the template matching can be extended using projection matrix when calibration is unknown. Additionally one could use the non-linear method in bundle adjustement when surveyed point are available.

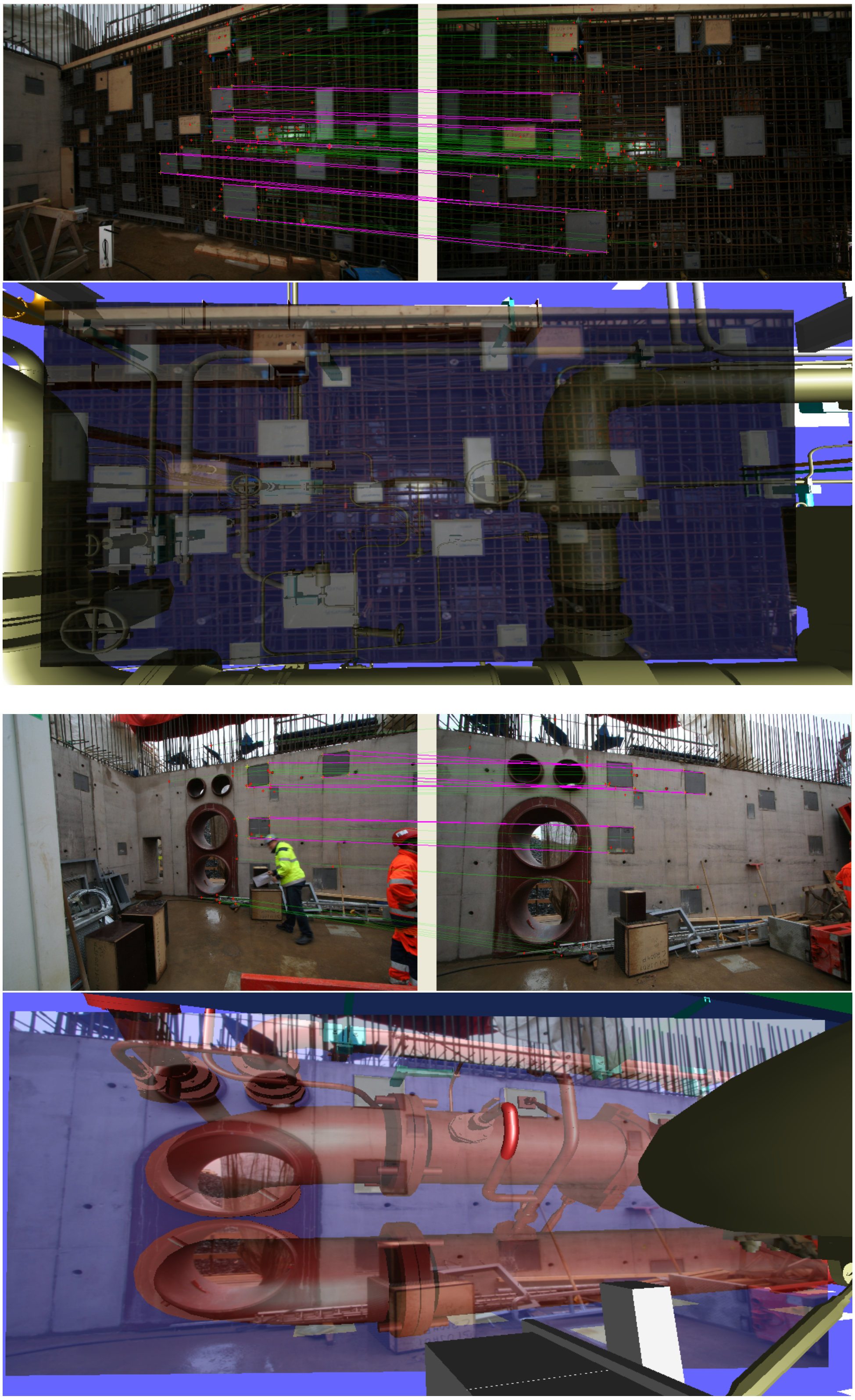


Figure 9.9: **Full Pose from Wide Baseline Keyframe from Powerplant Images (1)**: (top) The matching and propagation results: propagated 2D-3D correspondences in pink (left the keyframe, right the target), unmatched 3D points in yellow; matched features in green; (bottom) The resulting augmentation.

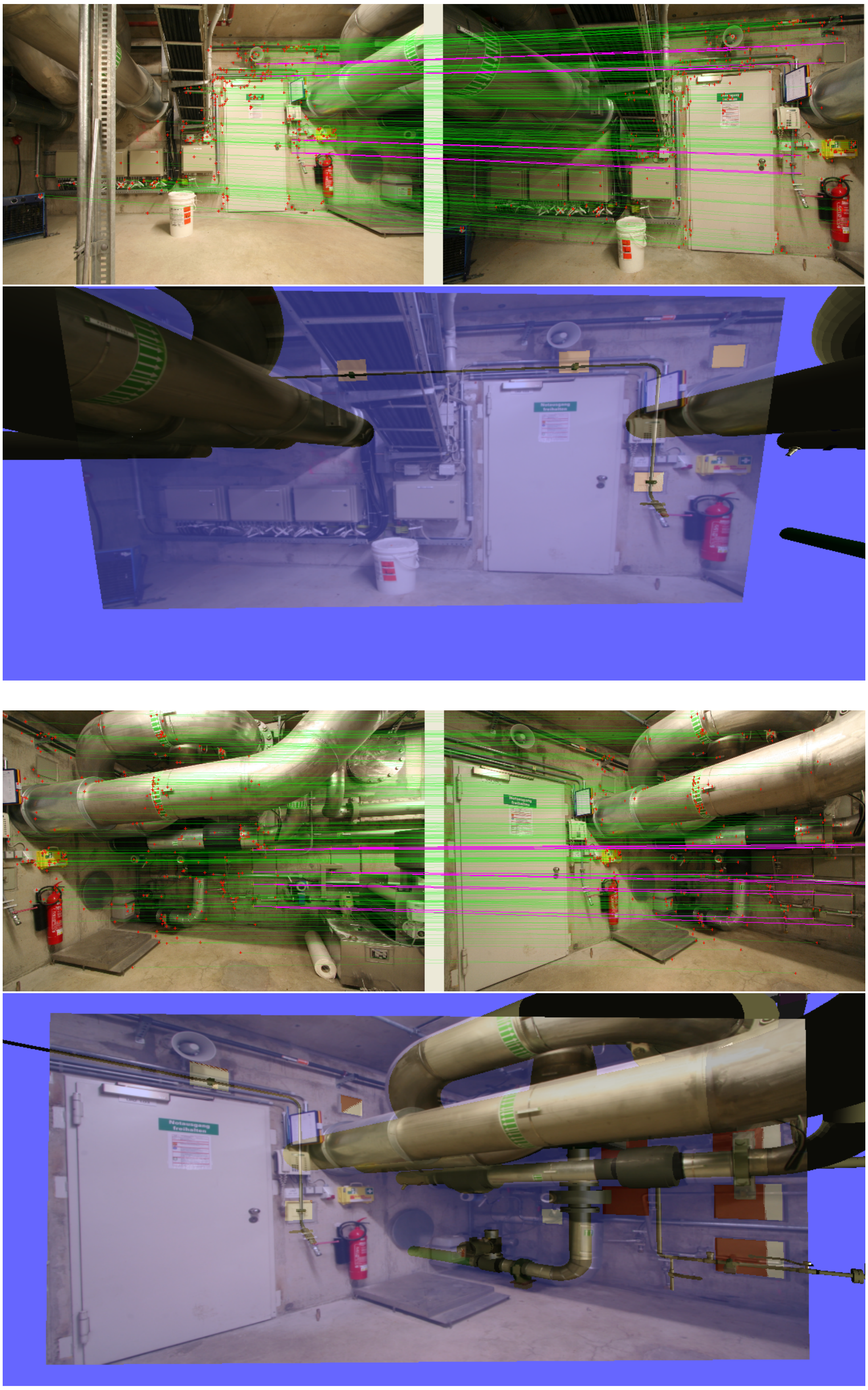


Figure 9.10: **Full Pose from Wide Baseline Keyframe from Powerplant Images (2)**: (top) The matching and propagation results: propagated 2D-3D correspondences in pink (left the keyframe, right the target), unmatched 3D points in yellow; matched features in green; (bottom) The resulting augmentation.

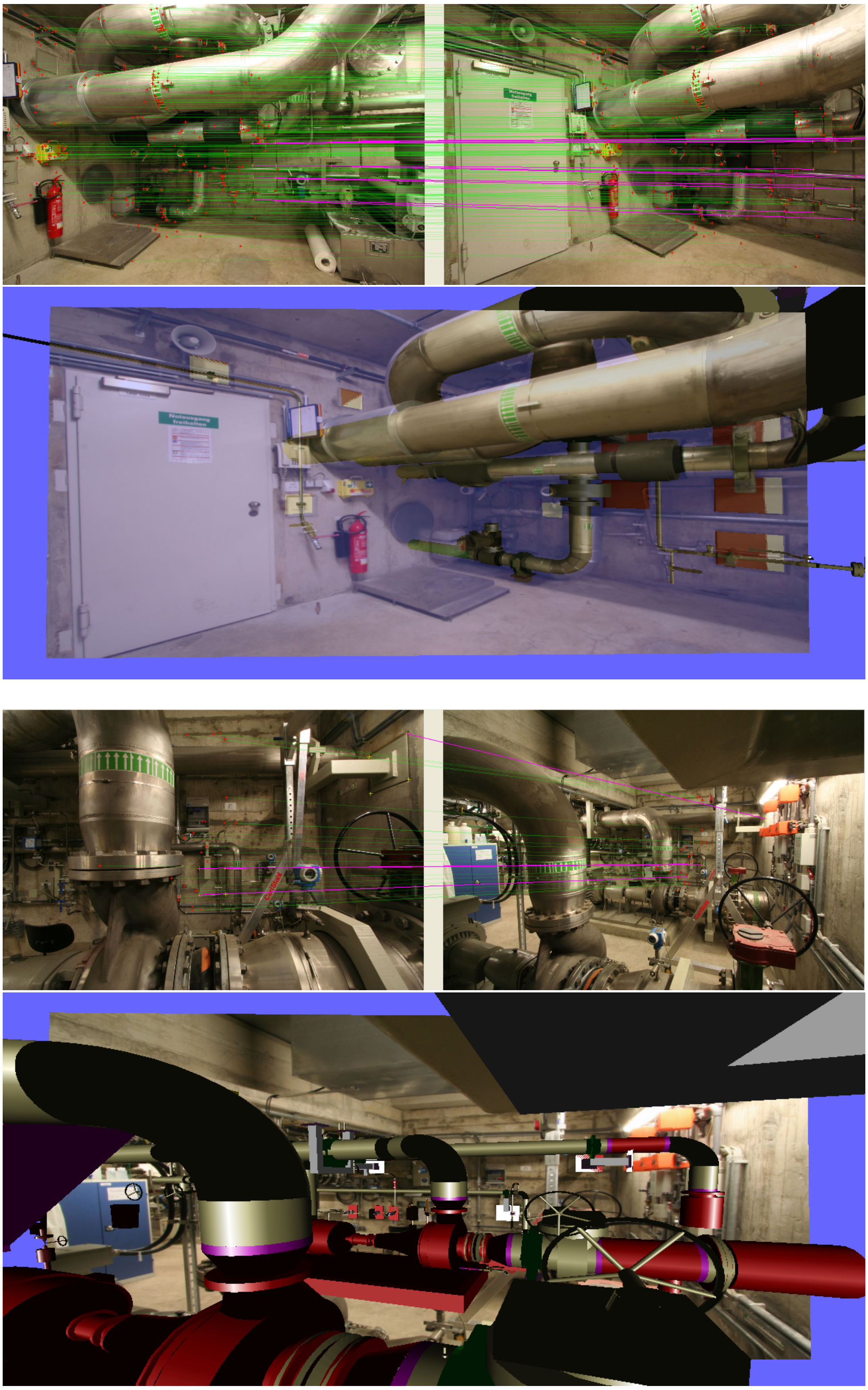


Figure 9.11: **Full Pose from Wide Baseline Keyframe from Powerplant Images (3)**: (top) The matching and propagation results: propagated 2D-3D correspondences in pink (left the keyframe, right the target), unmatched 3D points in yellow; matched features in green; (bottom) The resulting augmentation.

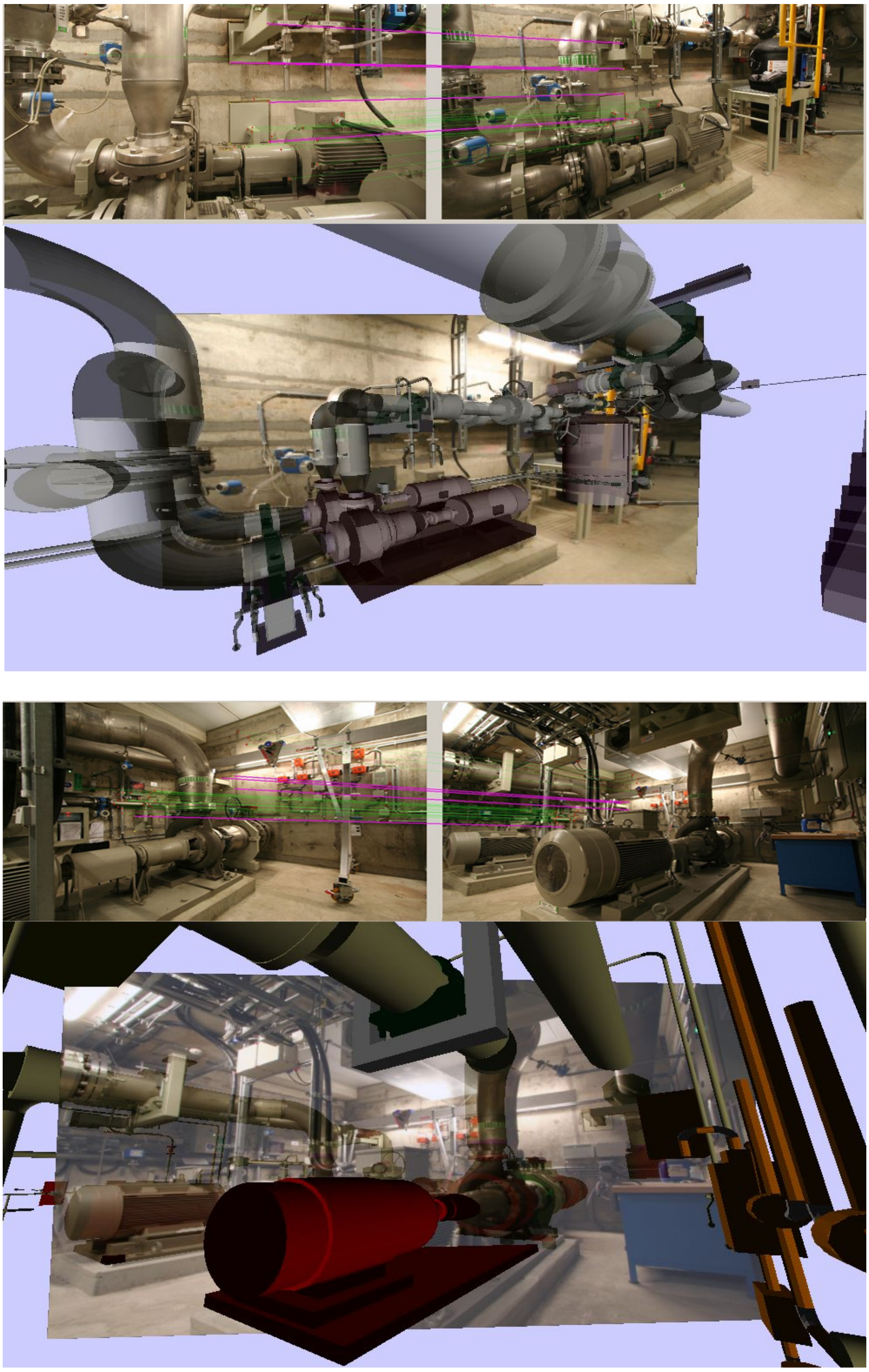


Figure 9.12: **Full Pose from Wide Baseline Keyframe from Powerplant Images (4)**: (top) The matching and propagation results: propagated 2D-3D correspondences in pink (left the keyframe, right the target), unmatched 3D points in yellow; matched features in green; (bottom) The resulting augmentation.

Part IV

Conclusion

CHAPTER

TEN

OPENING

"Le mieux est l'ennemi du bien." - *Voltaire*

In this chapter, we summarize the contributions presented in this thesis. We also draw guidelines for working on industrial research in academia. Finally we discuss current limitations and promising research directions for Augmented Reality and 3D Computer Vision.

10.1 Summary

In this thesis, we developed a complete system for an actual industrial problem: *discrepancy check*. We proposed to estimate these discrepancies using AR visualizations. By superimposing a CAD model on an image, a civil engineer could evaluate whether or not a component was built as planned. We introduced new methods to interact with such computer generated scenes. We allowed the user to zoom and pan in an AR view as if he was interacting with a regular image. We also introduced a new method to access near views using directional query. For example, the user could request a view that captures the object being inspected from the left and the software would search in its database and display the corresponding image based on its pose relative to current view.

Our system includes two new approaches to support the user in his task when aligning images to the CAD model. The first method is based on industrial components that are present in most industrial compounds: anchor-plates. The second uses images that are already registered to the model (i.e. keyframes) and automatically aligns new images to the CAD model. The usefulness of our system is currently being tested to monitor the construction of a large power-plant. The question of whether or not our solution will be adopted is still unclear. It will take time to answer this question but I believe that it offers a decent solution for an existing problem. The easy deployment and the quick verification capability are two strong arguments for using our solution to determine the correctness of a built object. If the construction is found to be incorrect, the plant engineer can decide if an as-built model is critical, and generate one for this specific location; for example using laser scanners.

In this thesis, we did not only focus on the usability of our system, but we also introduced solutions for 3D computer vision problems. We presented a new method to estimate the error localization of image multi-scale local features. We demonstrated usefulness of this information previously questioned for corner-like features [Kanazawa and Kanatani, 2003]. We also introduced a new method for the structure from motion problem based on a non-linear cost function that combines the usual geometry-based re-projection error with intensity differences from the images. This cost function compensates for errors made during the local feature localization by looking back at the pixel information. We showed that this method consistently leads to better registration results than the current gold standard that only uses a re-projection error. Finally we developed a new algorithm for the automatic estimation of the length of the baseline based on the available epipolar geometry between the cameras and a sparse set of 2D/3D correspondences. All the presented methods are practical and can be applied in standard vision systems. For example, the keyframe based pose estimation system was integrated in our discrepancy check system.

10.2 Academic Goals versus Industrial Requirements: Lessons Learned

I understood overtime that the idea found along the way of industrial sponsored research have to be evaluated right away. It does not only need to be a new method compared to the state of the art. The implementation and testing should be realizable in a short time period. I found that a good limit was around 3 months for a project that would be yearly reviewed. A longer period might delay the project and would certainly impact the relationship between the partners as they might not appreciate the complication that the implementation of a new idea could generate. Most importantly the new method should have an impact for the end users. It should simplify or render his work more effective. That could include the introduction of a new workflow. If the new idea fits these criteria, one has to find a scenario where there is a clear benefit compared to the current method for the partners. The proposal submitted to the partners should concentrate on the fact that the company should gain or save money by using your method. If the idea does not fulfill the requirements, it does not mean that it should be discarded but it will have little chance to be sponsored via an industrial project.

An important principle to remember when developing a new solution is to always start by developing a naive procedure even if it has to be manual. This usually gives useful information on what are the critical problems to tackle. First by testing the approach, one can see its limitations and then find solutions to the real underlying problems. Developing directly a complex solution should be avoided as it might hide unexpected problems that one can only discover when testing a complete solution. Sometimes a simple missing feature might discourage a user and might completely put a project in jeopardy. For example an object detection algorithm will certainly not work all the time. If no back-up solution is available to the user, some of the data will not be usable by the user. A simple user interface to manually select the object will give him the possibility to use all the available data. This type of semi-automatic or user supported methods rarely raise interests for the research community but they are of interest for a user. It is important for a user to know that even if the approach does not always work, he will still be able to achieve his task.

Nowadays, AR is in a hype period [Mark Billinghurst, 2007, Lens-Fitzgerald, 2010]. One should be careful to increase the expectations slowly, as we still cannot guarantee the experience depicted in Holywood movies. If the expectation would raise too fast, AR might follow the path of Virtual Reality and never become a mass market technology. In order to prove the usefulness of AR to solve existing or new problems, it needs to be better than current solutions. It should also be scalable because if it cannot be applied to the real size of the problem then it will stay a "niche" technology. And one should not try to solve every problem with AR, it is an unfortunate trend that is not always effective. During this thesis, we never intended to create a perfect solution but something simple that works and would be beneficial for the end-user.

10.3 Roads to be Paved

Our work like most AR systems is far from being mature, as Klinker et al. [2004] said "AR is still in its infancy". This thesis is one step forward and there are many others steps to be taken in order for it to become a reality. Not only should one perform additional user studies to ensure that the proposed interactions maximize the efficiency of the worker for the task at hand. New visualization should be developed to support him in his task. For example, the quality of the registration should be available in a meaningful way to the user. Currently, little information is available to the user to inform him whether the augmentation he is observing is correct. A general sense of correctness can be guessed by a trained operator, for example by comparing the rotation of the model with respect to the one of the image, but no finer grain information is available. A re-projection error even when coupled with covariance measures is complex to interpret even for experts and makes no sense for novices. We could propagate the estimated registration uncertainty to the 3D space and project it back to the image. This could give an estimate of the alignment error for each pixel for example via a color mapping. This should provide a better understanding to the user on whether the part of the augmentation used to evaluate a discrepancy could be trusted.

In this thesis, we focused on interactive techniques to support an engineer to detect discrepancy but the ultimate goal is to offer an automatic method to evaluate them. A finer process could be imagined to improve the classification between correctness and discrepancy, as discrepancies are not completely random. Discrepancies can be classified in four categories: shifts, when the component has the same shape but its position has some offset; deformations, for example pipe section are longer than planned; mismatches, when the component is not of the correct type and therefore has a different shape; and mistakes, when the component is just missing. Using a set of aligned images, we could develop a method that would label CAD components using these different types of discrepancies. For shifts, different poses could be used as labels. This type of approach would not only give information about the discrepancy but it would offer correction to apply to the 3D model.

Another interesting path to explore would be to use all the available photographs together to refine the registration. This would involve a bundle adjustment. We implemented the first steps of such an approach but the resulting quality never was acceptable: the obtained registrations were not always convincing and the variability of the results was too large. The major problem is the heterogeneous observations mixed in the non linear optimization. State of the art systems often use only one type of features points (e.g. SIFT) and the results are rarely investigated for registration error but more in terms of aesthetics. In our system, we have to use 3D points from the CAD model (anchor-plates corners) and 2D points (anchor-plates corners and images features). We discussed, in Chapter 7, how to properly use multi-scale type features in a bundle adjustment. Unfortunately mixing different cost functions and uncertainty requires a proper weighting, which is not trivial as shown in Chapter 8. This should be studied in a near future. It would also be interesting to integrate to the cost function propagated control points in the bundle adjustment as presented in Chapter 9. Furthermore an optimizer based on such a non linear cost function needs to be properly initialized. This is still an active area of research, which would need to be addressed to use bundle adjustment in our scenario. The problem of creating an efficient bundle adjustment that would run at an interactive rate to allow user interaction needs also to be researched. The use of GPU computational power might be an interesting direction to investigate to accelerate this iterative process.

In this thesis we focused our attention on calibrated cameras and it would be interesting to extend some of the approaches to uncalibrated cameras as it would allow the user to acquire image in-focus using an auto-focus and at a selected scale using a analogue zoom. We showed, in Chapter 9, that using intensity and geometric measures together could be easily applied to different pose estimation problems. Early work [Acero, 2009] has demonstrated that camera calibration could benefit greatly from such a system and therefore should be further investigated.

In Chapter 7, we discussed the impact of multi-scale feature detection algorithm on the localization precision. It would also be interesting to propagate the uncertainty of the detection to the descriptor. Surprisingly little work has been performed to study the statistics of descriptors [Ke and Sukthankar, 2004, Nister and Stewenius, 2006]. The propagation of the error to the descriptor might also help to compress the descriptor by keeping only valuable information, which would not require any offline training.

By using our solution for discrepancy check, a new reality CAD model is created that can be used for new applications. It contains registered images, as well as information about the quality of the CAD models. This can be directly used to create AR supported maintenance. The images registered could be used as keyframes in a feature-based tracking system. By equipping a maintenance worker with a mobile PC, he could have access to real-time augmentation at no additional engineering cost on the model. This real-time augmentation could be used to find a malfunctioning system or support the communication with a remote expert that tries to perform an offsite diagnosis. Early results from Kaiser [2010] have shown that such an application offers challenges. One needs not only to have real-time tracking algorithms but it also requires a fast re-localization procedure when the tracking is lost. Using our *reality model* would finally make available to the industry a scalable solution for tracking that would not require scene engineering (installation of markers or external tracking system). The environment engineering would be seamless as the reality model would be prepared while the civil engineers are performing their discrepancy check tasks.

Part V

Appendix

APPENDIX

A

INDUSTRIAL AUGMENTED REALITY

For the AR community, the industry was always one of the steering forces for research. *Boieng* is the company that defined AR and pushed it forward [Mizell, 1994]. Since then many researchers, projects and companies followed this path. They all tried to apply the concept of aligning virtual information with the real context for the user's benefit [Azuma, 1997, Azuma et al., 2001]. AR applications are everywhere: medical, military, manufacturing, design, advertisement, etc. But only few of the developed ideas made it into products. The entertainment industry is using AR techniques for televised sport (e.g. the 10 yard line in US football) [Azuma et al., 2001] and for theme parks [Stapleton et al., 2003]. The printed media are toying around with AR [Magazines, 2009]. In the medical field the only concept, which finally is in a tryout phase, is the Camera Augmented C-Arm (CAMC). This device augments the physician view with X-ray vision [Navab et al., 2009]. In the car industry, the intelligent welding gun is the only concept that was fully put into a product and is being used [Echtler et al., 2003], see Section A.2.1 for a detailed description.

Augmented Reality can be applied in many different scenarios within the same field. Robotic was one of the early field of applications for AR. It was first used as a better user interface for robot telepresence. In the ARGOS (AR through Graphic Overlays on Stereovideo) project, augmentations are delivered in a stereoscopic display to support the operator in guiding a robotic arm. The manipulator has access to virtual measuring tool: a pointer and tape measure to help him define the robot path by increasing his awareness of depth. Ultimately they expect that it should decrease the manipulator learning period [Zhai and Milgram, 1991, Drascic et al., 1993, Milgram et al., 1993]. AR for robot manipulation was used as a predictive display, where the result of a command is simulated in the current context and can be evaluated before being executed [Kim, 1993, 1996]. AR was used to create an enhanced task representation for planing remote mining drill. This is based on AR visualization and haptic feedback. By reconstructing the scene, they can perform for remote planing of the drill sequence [Gu et al., 2002]. AR is also used in more advanced scenarii where the robot has to interact with complex virtual objects that have advanced behaviors (e.g. other robots) [Kim et al., 2008].

AR was not only deployed as a new user interface but also to change workflows [Bischoff and Kazi, 2004, Bischoff and Kurth, 2006]. *KUKA*[1] for human-robot interactions presented a set of possible applications of AR. They do not only want to simplify the manipulation via a better interface. They propose to simplify the programming and training by helping the manipulator to better understand the coordinate system and the usage of a space-mouse. They also want to support the facilities upgrade using virtual model in situ of the new robots, to check whether they fit and can operate. This includes preparing the introduction of new prototypes in the existing workflow by performing test in a MR world. This allows the manufacturer to prepare the robot programing ahead of time using CAD model of the object. Finally, they propose to use AR for supporting a maintenance task by displaying instructions and highlights on the object being inspected.

These different prototypes show the broad applicability of Augmented Reality for the industry. In this chapter, we focus on AR applications that support a product during its life-cycle : design (c.f. Section A.1), manufacturing (c.f. Section A.2), commissioning (c.f. Section A.3), maintenance (c.f. Section A.4) and decommissionning (c.f. Section A.5).

A.1 Product Design

First we discuss the early use of single image augmentation for architectural design. This is mainly performed to evaluate the visual impact of a new construction (Section A.1.1). Then we present advanced AR and MR systems that allow live interaction between users and advanced simulation (Section A.1.2).

A.1.1 Photo-Montage

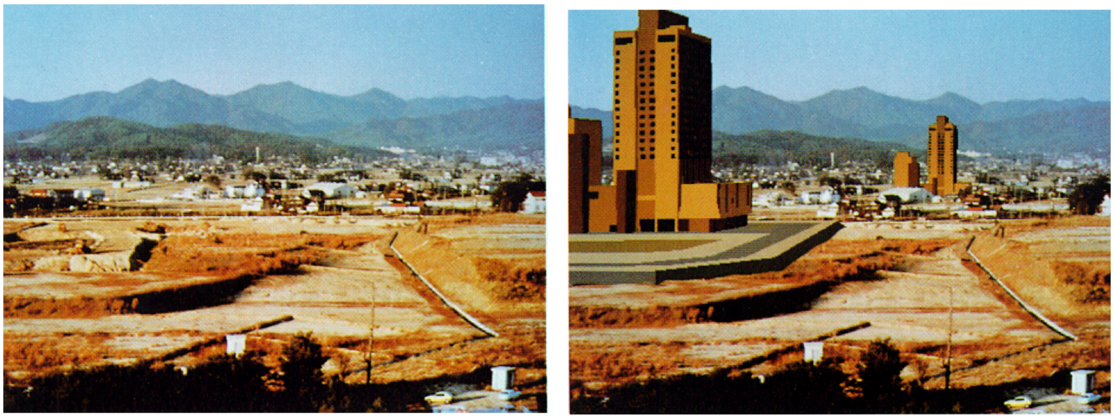
Uno and Matsuka [1979] are the first, from our knowledge, to present a CAD software that can use real photographs called A-IDAS (advanced integrated designer's activity supports). Digitalized photographs can be displayed as background images. Their software includes a set of routine to distort the image perspectively. The obtained image could then be used for image composition that is "superimposing of one image onto another".

In photo-montage, most of the work was based on manual interactions. Images are scanned and their textures is used to obtain realistic views [Feibush et al., 1980]. This can be seen as a pure VR compositing where everything is a virtual object rendered by the computer [Porter and Duff, 1984]. Photo-montage is useful in architectural simulation. By combining CAD models and background photograph, it can give a perspective of the visual impact of a new construction [Nakamae et al., 1986]. A typical result of a photo-montage system is visible in Figure A.1.

Kaneda et al. [1989] extend static 2D montage by allowing some images navigation by generating near view images. This is achieved by using a 3D model of the real world generated from cartographic images, which is textured using aerial images.

All these systems are all based on a manual alignment, which limits their applicability. This problem is first tackled by integrating an automatic method to align image with their relative 3D Model [Chevrier et al., 1995, Berger et al., 1996, 1999]. This alignment is used

[1] http://www.kuka.com/



(a) Original Photograph

(b) Photo-montage

Figure A.1: **Photo-montage for an Architecture Project**: Early work in graphic rendering was used to illustrate the visual impact of architectural project by rendering a virtual model onto the image. *Courtesy of Eihachiro Nakamae, Koichi Harada, Takao Ishizaki and Tomoyuki Nishita [Nakamae et al., 1986] (awaiting agreement).*

to evaluate different illumination projects fir architectural landmark (e.g. bridges). This allows to visualize architectural projects in their existing urban environment and to show the impact of different projects to the decision makers directly on-site.

Klinker et al. [1998] also study architectural impact by introducing diminished reality, where they allow users to remove an object that is to be replaced. The use case is also the insertion of a bridge.

A.1.2 AR Supported Collaborative Design

A lot of research has done in AR for collaborative design where several persons would interact with the virtual and the real scene to achieve the best possible design. We first discuss architectural applications and then some product oriented applications.

A.1.2.1 Collaborative Architectural Design

Collaborative architectural design is a classic application for Augmented Reality that the European Union funded extensively through different projects such as VANGUARD [Mohr et al., 1998], ARTHUR[2], IPCity[3], CICC[4],etc.

AR supported interior design application was first studied by the ECRC [Ahlers et al., 1995, Koller et al., 1997b, Schumann et al., 1998, Klinker et al., 1997]. A customer is supported using an AR system to design and evaluate new interior arrangements. The customer is in contact with an interior architect that helps him decides for new furnitures. They both can manipulated the furniture in the mixed world. The customer can contact

[2] `http://www.vr.ucl.ac.uk/projects/arthur/`

[3] `http://www.ipcity.eu/`

[4] `http://www.vers.co.uk/cicc.htm`

friends or colleagues to request opinions. This allows him to create a design that can be validated in context and that he is confident with. The software client has the appearance of an regular CAD software with a Mixed view and it can be used to order new furniture directly.

The MIT developed a table top augmentation system: URP (Urban Planning and Design) [Underkoffler and Ishii, 1999, Ben-Joseph et al., 2001]. They let the users arrange the building layout. It employs tangible wire-frame model to represent real mock-ups of the different components. This was later extended in the Luminous Table [Ishii et al., 2002], which integrates 2D drawings, 3D physical models and digtal simulations. They use this unique platform to simplify the mind merging process. They propose complex simulation to evaluate their design based on sun exposure, cast shadows, wind pattern and traffic congestion. For example, they use their system to maximize the amount of sun each building gets at the same times and still enforce a sense of harmony in the geometric arrangement. To interact, the user can create mixed views using a handheld tracked camera to share his "point of view" with his collaborators. The round table setting encourages social interactions and forms a creative space.

In the ARTHUR project (Augmented Round Table for Architecture and Urban Planning), the use of projection based display is replaced by personal displays [Broll et al., 2003, 2004, Ohlenburg et al., 2004, Aish et al., 2004] Each user is equipped with a HMD, which allows to examine the site by walking around. They use their system in different architectural scenarii: evaluate the positioning of a high-rise within a city, evaluate exit routes layout using crowd simulation software, etc. They emphasize on the need to be tightly integrated with the real CAD system to allow constant design discussion and update. If a modification is proposed it can be directly implemented and visualize. This direct feedback mechanism offers better collaborations. Also using HMDs, Kato et al. [2003] use AR as a tangible interface for city planning where everyone can modify the virtual world being models. The designers can layout (building, trees, etc) to quickly evaluate the visual impact of the different setups.

All these methods are designed for off-site collaborations. They restrict their use to labs or in meeting rooms. This limitation was tackled by Moloney [2008] who propose to visualize the environmental impact of the project in the early stage of the design. They want to realize how the atmospheric condition impact the aesthetic effect. For this, they use strollAR a mobile platform that they bring on-site to evaluate their design. They handle realistic lighting for the virtual model to offer convincing augmentation. This has shown to be a good evaluation tool before meeting the stakeholder. Similarly Sareika and Schmalstieg [2007] and Maquil et al. [2009] try to improve the communication among stakeholders by going on site. They develop the concept of an MR tent as a location for on-site communication, which incorporates all necessary tools and interactions to help stakeholders deciding on an urban renovation project. They allow the modification of images captured from the scene by sketching. They want to create a collaborative space where everyone can interact properly even though they do not share the same educational background. All this happens on the site (e.g. renovation site), which forces everyone to be on the same level as it disrupts normal behaviors that would happens in a regular meeting. This tends to help communication and the emergence of new concepts.

A.1.2.2 AR for Product Design

AR is not only used for architectural design but also to ease the development of other products such as cars and planes. For example for the car industry, Tamura et al. [2001] and Ohshima et al. [2003] propose to evaluate the design while sitting in a car. The user is in presence of a car skeleton (seat, steering wheeling, on-board commands). There are physically present in this MR world. It improves the immersion and their understanding of the virtual world by enhancing their sense of distance. The user can switch between option and version to evaluate the best fit. Similarly, Klinker et al. [2002] augment a car mock-up with different light optics to evaluate in-situ the visual results. This offers the possibility to navigate around an augmented mock-up. They emphasize on the need to integrate AR into the designing process. As a goal they hope to reduce the need (or at least the numbers) of expensive clay mock-ups. [Regenbrecht et al., 2002] also try to bring realism into car design by integrating augmentations to physical mock-ups during meetings. Nölle and Klinker [2006] develop a system to verify that manufactured object matched the CAD data, which can be useful during the design period of a product, where multiple designs exist and it is sometimes hard to keep track which manufactured piece correspond to which 3D model.

AR is not only use for ecstatic evaluation. Regenbrecht and Specht [2000] use hand held devices to evaluate the functionality of a design. They interpret air flow data around a passenger car seat resulting of a particular design via visualizing augmentation. Regenbrecht et al. [2005] support a customer when selecting option for an airplane cabin. Using an AR capable trolley, they display simulation data on a real size airplane mock-up. They also propose to improve functionality, ergonomy and safety of the cockpit design. The designer gets to place virtual instruments and commands in a real size cockpit to develop a more efficient layout. Using a similar concept, Balcisoy et al. [2000] test prototypes not only by looking at the 3D model but also its behavior. The user can interact with the prototype in a mixed world to evaluate its practicality. Nölle [2002] propose to validate crash tests simulation using AR by comparing them with real experiments. The ultimate goal is to replace some of the real crash tests by simulation to cope with the shorter life cycles of cars.

Furthermore AR can be used to optimize a design. For example, Dunston et al. [2002] display pipe layout using AR to optimize their arrangements. The use of a tangible interface allows the user to better understand the complexity of the model. Webel et al. [2007] pushes this concept further by integrating it tightly in the design process. The design of submarines piping system is complex, as a lot of pipes have to go through a restricted space. Therefore the designers are forced to use mock-ups to optimize the pipe layout for it to take the smallest amount space possible. By using AR, the engineers can verify that the current mock-up matches the design. The engineers can physically change the mock-up and integrate to the CAD model this modification using vision based reconstruction. This tight integration of AR in the design workflow allows to close the loop between the real and the virtual mock-up to create a more efficient development process.

Most of the previously mentioned approaches are designed to work in a prepared environment. This constraint is lifted by Thomas et al. [1999], Thomas [2008]. They propose to use a HMD combined with a wearable computer to visualize design data on-site by aligning CAD data to the real world. They present tools to modify the design and to model existing object that are not yet represented in the virtual data.

Once the design of an object has been validated AR can be used to plan its production. For example, Behzadan and Kamat [2005] develop an AR system for outdoor construction sites where they emphasize on the animation of 3D models. They want to verify simulation results on-site before implementing them. VR helps to understand the subtleties of such plan but does not give any contextual information. Additionally VR has a lot of overhead in order to model features not presents in the CAD data. They demonstrate their system to simulate a bridge construction and verify if their plan was realistic in the context of the real target site. In the next section, we discuss similar ideas to support the manufacturing process using AR techniques.

A.2 Manufacturing

When the design of a new product is finalized, its production can be launched. Augmented Reality is not only used to support worker for an assembly task (Section A.2.1) and to train an operator to produce a new object (Section A.2.2). It is also employed to design new factory floor to prepare a plant for a new item production (Section A.2.3).

A.2.1 AR for Assembly Guidance

Caudell and Mizell [1992], when developing the first AR application, tries to offer support for an assembly task using a see through display. Their system was originally proposed to reduce storage requirements of foam boards. These foam boards are used as real size map to guide the assembly worker when preparing wire bundle. Each type of wire bundle requires a different foam board. Using AR they can replace the need for specific foam boards by augmenting with wire bundle specific information a generic foam board. Their system does not only removes the need for specific foam board storage space but it helps the work to perform their task faster [Sims, 1994, Mizell, 1994]. Regenbrecht et al. [2005] also propose to use AR for montage of highly customizable objects. They specifically look at fuse boxes assembly for trucks that are based on the options selected by the client. This makes each truck unique. Therefore they present to the worker model-specific instruction using AR. This simplifies the workflow of the assembly as the worker is not required to refer to a generic paper-based instructions manual and can directly follow the model specific instruction.

AR is also used to support the manufacturing of larger goods such as power-plants. Webster et al. [1996] uses an AR system to support construction. Using AR they offer an x-ray view to visualize hidden structures such as pipes installed in between walls that should not be damaged. A similar solution is proposed by Klinker et al. [1999] to display the most recent construction plan to the worker, as seen in Figure A.2. For unmanned construction sites, Fujiwara et al. [2000] propose an AR system that displays virtual property lines on a video stream. This additional information helps the construction

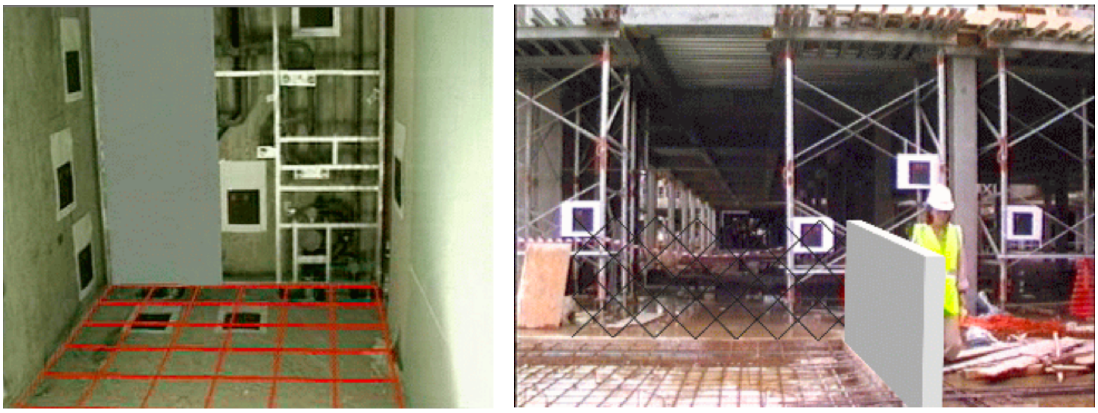


Figure A.2: **AR in Large Building Construction**: AR is often used to display up-to-date design information and to verify the correctness of the current construction in comparaison to the planing data. *Courtesy of Gudrun Klinker, Didier Stricker and Dirk Reiners [Klinker et al., 2001b].*

worker to properly performs this task. The need for remote operation can be justified when the construction site is hazardous, in their case an active volcano.

AR can be used as a replacement for paper based assembly instruction manual. The development overhead for such new manual can be justified because products' life cycle is constantly getting reduced. Ever changing product lines constantly force workers to be more flexible in manufacturing new models. For example, during the CICC project AR is deployed to support a car door assemblage [Reiners et al., 1998, Klinker et al., 1999]. This project sparked the interest of the European industry for AR. Molineros et al. [1998] study the applicability of AR for assembly tasks. They offer step by step instructions to the worker. By sensing the current state of the assembly, they offer the right information to the assembly worker. The construction process is modeled as a state graph, which represents evolution of the object being assembled. They use multiple hypotheses verification method to determine the evolution of the assembly. By studying this graph, they can determine when the worker performs a step that will block him to finalize the construction. Using this technique, they can evaluate a set of instructions to find the optimal set[Raghavan et al., 1999, Sharma and Molineros, 2001]. For the same task, Zauner et al. [2003] propose an MR system, where instructions are displayed to explain each step of the assembly. They do not use an instruction graph but a more object oriented approach. For each object, animations are available to describe its assembly. Each object can be detected by the system and when pieces are combined they form a new object with its own set of instructions. Barakonyi et al. [2004] present a virtual agent to guide the user to build an object (e.g. a Lego set). The agent presents required pieces (e.g. new blocks) and display an animation on the current built object to explain the next step. Mobile phones have also been proposed as an interface for AR-based assembly instruction [Hakkarainen et al., 2008].

AR is also used to support logistic application. This is investigated by the FORLOG[5]

[5] `http://www.bayfor.org/en/portfolio/research-cooperations/world-of-culture/forlog`.

project. When assembling complex systems, such as cars, specific pieces need to be available on the production lines. These items are picked up in a warehouse by a worker that follows a item picking list often paper based. In order to reduce errors that can have a certain impact such as delays on the production lines, Schwerdtfeger and Klinker [2008] propose a new guidance system for order picking using augmentation displayed in a HMD. The augmentation points the user to a target location where an item need to be picked up. This offers advantages in terms of lowering mistakes and automatic reporting as the system is linked to the IT infrastructure.

AR can not only be used to support unskilled workers but it can also considered for highly trained operators who use complex machinery. For example, Olwal et al. [2005] support a lathe[6] operator. Their system displays sensor readings in-situ such as cutting forces, RPMs, temperatures, etc. This allows the workers to stay focus on the piece being manufactured and access readings that require constant monitoring.

Following a similar trend of supporting skilled workers, many AR welding projects have been developed [Tschirner et al., 2002, Aiteanu et al., 2003]. Many manual welding procedures have been replaced by programmable robots, for example in the car industry. Unfortunately, for some complex and not repeated tasks, manual operators cannot be replaced, for example on shipyard where specific procedures require highly trained operators. By using AR, researchers want to improve welding seam quality, decrease rejection rate and therefore reduced cost. The usual setup integrates in a welding shield a HMD and a pair of High Dynamic Range (HDR) cameras [Tschirner et al., 2002]. The direct view through the darkened lens is replaced by a view captured by the cameras. Instructions and sensors information are displayed on this video view. It can inform the worker about electrical welding parameters (e.g. current and voltage). This constant monitoring of the worker actions offers the possibility to create on-line a documentation relative to the manufacturing process. This documentation can give hints about mistakes that could have happened and how to avoid them in the future. Both these projects stayed at the prototypical level.

On the other hand Echtler et al. [2003] with their *Intelligent Welding Gun* introduce a new product and a related workflow for the industry benefit. The target is to help welders shoot studs with high precision for experimental car (i.e. prototype) where robot cannot be programmed for, as it would require too much time. These prototypes are mainly hand built. A regular welding gun is tracked using external sensing devices and is augmented with a display that provides guidance for the worker to find designated studs location. In their application they are trying to find the best stud's placements. The produced prototype can be evaluated and a stud position can be validated or modified as a result when a proper arrangement is found. This new workflow replaces an cumbersome procedure that require one worker to manipulate a probe sensor to find a stud location that he reads from an instruction sheet, while a second worker marks the position and then studs. Clearly an AR setup is more effective as it only requires one operator. They have demonstrate that using their setup they can be four times faster while sustaining the same precision. This AR project is one of the only publicly known ones, which is currently

`html`

[6] A spinning tool that can perform various tasks such as carving, drilling, sanding, etc.

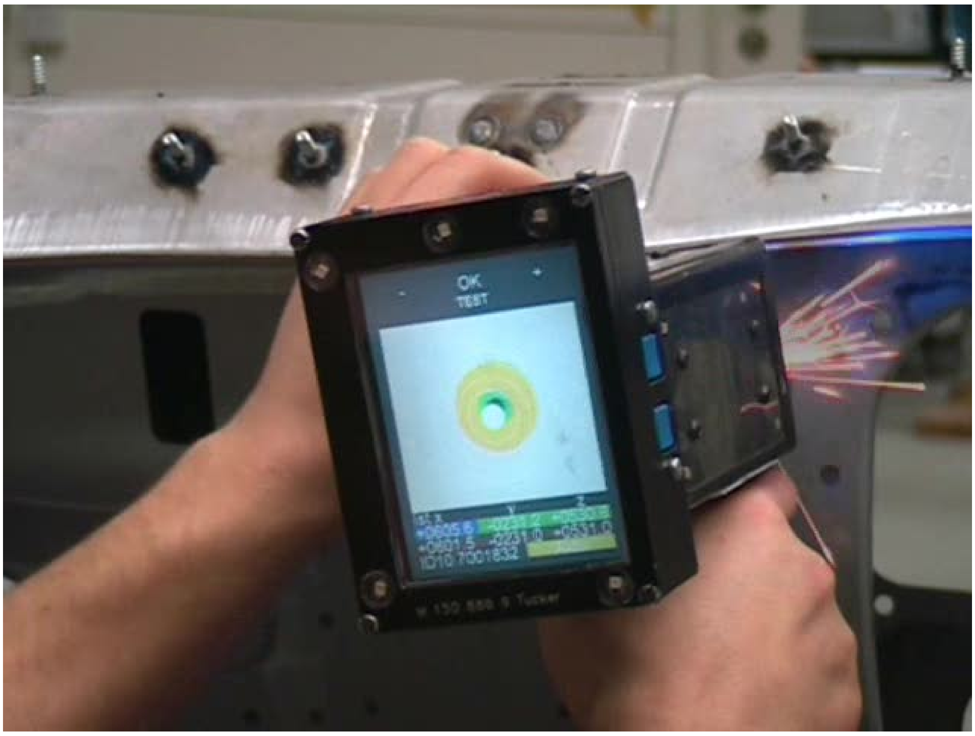


Figure A.3: **The Intelligent Welding Gun** is the first AR based product to be used in the manufacturing industry. The welder has access to navigation information on a screen attached to his welding gun to localize the next stud location. *Courtesy of Kudrun Klinker.*

deployed and used by a company (*BMW*). A picture of the final product is visible in Figure A.3. For the same application, Schwerdtfeger et al. [2008] recently propose to replace the gun mounted display by a laser projected that would reduced the gun size and make it handier.

This various projects and area of application demonstrate the high applicability of AR. It also shows the difficulty to go from the lab to a product.

A.2.2 AR based Training

Here we present applications of AR that try to improve workers training to manufacture new items.

For example, AR is not only used to support welders but also to train new welders, which is a complex procedure to learn [Kobayashi et al., 2001]. Welders need to be properly train as the strength of a welded product depends on the operators' skills. This project proposes an AR simulator in a safe and efficient environment because welders learning is a complex process that can be harmful. Additionally, the number of good teachers is limited. Their system uses a similar display setup as [Tschirner et al., 2002, Aiteanu et al., 2003] and offers additional haptic and sound feedback.

Schwald and Laval [2003] propose to support training for a complex assembly task.

They hope to save on training time by using AR. The user obtains visual augmentation via a HMD and instruction via audio. He can request information via vocal input using a microphone.

AR is also used to design new type of instructions manual for workers training [Haringer and Regenbrecht, 2002]. They look at the creation of an AR ready manuals for car mechanic support. A basic workflow of the repair is sketched as a set of 2D slides using powerpoint For each step (i.e. slide) a 3D layout of the instruction is elaborated based based on the set of 2D instructions. Then the order and relations between steps is finalized. Each instructios manual is then tested and modified until it reaches an acceptable quality.

A.2.3 AR Supported Factory Planning

In the industrial process, a lot of care is given to factory setup, when a new item needs to be produced. This factory design can happen in new compound or in already existing production line that needs to be evaluated to verify if they can produce a new item and to decide if eventually they require revamping. This process is called *factory planning.*

The first demonstrator for planing activity for plant design is proposed by Rauterberg et al. [1998]. The designers sit around a table with a virtual orthogonal view of the plant currently being designed superimposed on real objects. A full perspective view of the virtual world is accessible on an additional display. The designers can select an object to change its position or to delete it. They can also manipulate the viewpoint of the perspective view. This offers a more immersive and collaborative experience than a VR system.

Gausemeier et al. [2002] propose not only to assemble 3D components using AR but also to consider some semantic knowledge of the plant (e.g. water and electrical access) and for each component the minimum and maximum distance required to its adjacent module. Components that need to be positioned are materialized by markers that the designers manipulate to create a proper design. The created plant design can be tested in a production simulation tool to verify its efficiency.

For the positioning of components and system in new factory, Siltanen et al. [2007] propose to use an iterative process where an plant operator (e.g a mechanics) requests a plant alteration that he believes is best suited. This request is generated from the factory floor for the plant designer. Using an augmented view of the current plant and the proposed design alteration, the operator can evaluate them. This can lead either to a validation of the proposed design and to its implementation or to further design modifications in order to obtain an optimal plant design. This method allows the plant operator to communicate from the place he feels the most confident: the factory floor. He can directly explain his requests by showing the reality of the factory to the designer and clearly describe problems he founds on-site.

When new items need to produce in existing plants, the plant need to be verified to know whether they can handle the new production line and if they need alterations. Doil et al. [2003] introduces methods to plan the upgrade of a factory not only in a VR system but in an actual plant. They hope to validate the planning faster, to improve the data quality, and to avoid collision between new components and the once setup. This should minimize re-planning activities. In their first prototype they are only interested in knowing whether new components fit the current plant. They allow designers to organize the shop floor, to perform measurement in the mixed view (e.g. to verify security space requirements) and to test the ergonomics of the new workplace (e.g. check whether workers are not required un-necessary movement). In a second iteration of this project, Pentenrieder et al. [2007] propose to reduce errors in factory planing by creating an up-to-date and complete CAD model of the current plant. They realized that most of the available CAD data for plants are not correct compared to the current plant state because it is a extremely complexed task to synchronize the CAD model with the plant. They focus their work on the accurate alignment of the CAD model with the reality to offer a precise augmentation to plant designers. After several iteration of their system, they settle for photo-based augmentation because they found it to be the most accessible technique for the plant designers. In their system (ROIVIS), they offer precise (verified and bounded) measurement functionality and collision detection between a plant update and the current plant state. They offer comprehensive documentation by saving AR screen-shots of the plant images, that can be later used to inform someone about design acceptance or rejection.

A.3 AR Supported Commissioning: Validation and Documentation

After its production and before its use an manufactured item needs to be verified and documented during a process called commissioning. This quality control is done for small items (e.g. micro-processors and cell-phones) and for larger systems (e.g. ships and power-plants).

Klinker et al. [2001b] introduce an AR system for the construction business. During and after construction using their system one can visualize design modification directly on the building site and verify its correctness.

For the commissioning of offshore structures, Lee et al. [2010] propose to support inspection using a mixed reality to guarantee the quality of the delivered product.

Navab et al. [1998, 1999b] propose to create an as-built documentation that could offer new application to the industry. For example, it could simplify the maintenance planning and execution, where a precise 3D model is required for the plan to be realistic. They develop a software platform (*Cylicon*) to register industrial drawing and perspective images. A decade ago drawings were the only documents used during the complete plant life cycle from design to decommissioning. They create a better documentation of the plant, where images have hyperlinks (inherited from the drawings) to meta-information (e.g. inventory status and past maintenance logs). They call this new type of document: *the transparent factory.* Their solution does not only focus on floor maps but also wiring schematic and factory layout. Using *Cylicon* one can create an as-built 3D model based on the fusion of industrial drawing and perspective images [Navab et al., 1999a, Appel and Navab, 2002]. Such a solution has great financial advantages with respect to delivery payment and quality control.

Schoenfelder and Schmalstieg [2008] develop a system for Augmented Reality Building Acceptance (ARBA). This task is sometimes known as plant walk-down, where the plant engineers want to document discrepancy existing between the new built plant and the planning documentation. For this they use *planar* a tracked touchscreen mounted on wheels that can be moved around the factory floor [Schoenfelder et al., 2005]. They tested their system on a plateform 300x150 meters multistorey factory floor where they compared the built plant against planning documents. They justify the need of such a system not only for guaranteeing the usability of the new plant's documentation, but also because discrepancies might lower the value of the building in terms of re-usability. A discrepancy might not affect its current operability but it could have an impact when the plant is re-factored. Their system can in situ superimpose CAD planning data to an image of the plant captured from a camera mounted on *planar*. They expect stack-holders to accept discrepancy more easily when viewed in-situ as their impact might be evaluated in context. Discrepancies are detected by changing the transparency of the image, the depth of image plan. Documentations of the discrepancy are done using a stylus to annotate the augmentations after the inspection. The documentation data is gathered to assemble a report. This report can be passed on to the contractors that have to deliver revised 3D data. If necessary they can use *planar* to change the position of CAD components to obtain an as-built. This system offers a limited precision due to the use low resolution camera and can be only applied to some hotspot as it requires a external tracking system to be installed before end.

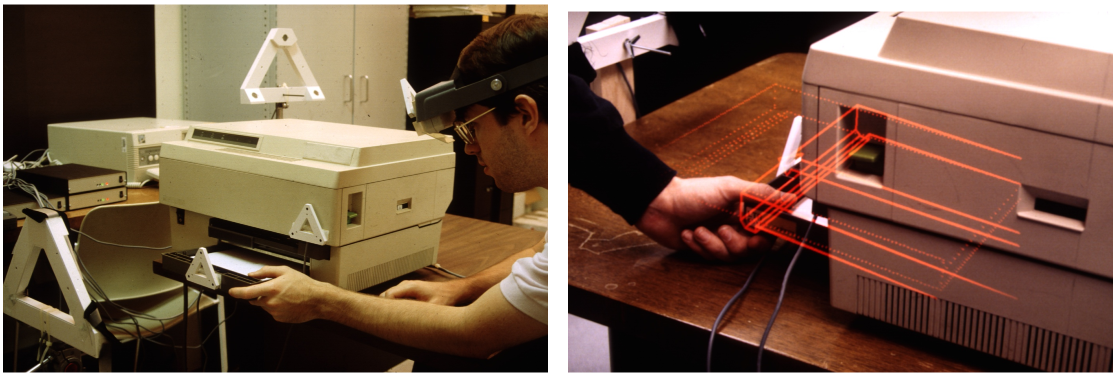
A.3.1 AR Ready and Accessible Documentation

In this section, we discuss how existing documents or documentation created with the help of an AR system can be accessed on-site.

In order to help construction workers, Klinker et al. [1999] present the latest document to minimize mistakes generated from outdated or inaccurate planning data. This system informs worker during the erection by augmenting the current construction state with virtual components that are left to be built. It offers x-ray view to access invisible features documented in the model.

Dodson et al. [2002] develop a system to help field works to localize sub-terrestrial information such as pipes (gas and water), contaminated soils, geological structures, power-cables, communication hubs, etc. This system should help the worker as it is difficult to apprehend the relation of 2D maps with the real world and a misinterpretation could lead to extra excavations. Their system uses virtual goggles that aligns digitally stored maps using GPS and gyroscopes This idea was extended by Schall et al. [2008] to allow field worker to annotate the digital map. This system is based on an ultra mobile pc.

On-site data access was demonstrated by Reitmayr et al. [2005] to support firefighters and rescue squads in their tasks by augmenting maps with live information such as CCTV feeds and real-time wind patterns. A similar system (AR Vino) was developed to superimpose viticulture GIS data on to vineyards to help viticulturists understanding the effect of environmental parameters on the quality of grapes.



(a) Setup

(b) Display

Figure A.4: **The KARMA Printer Repair Project**: an operator wearing a HMD in which is displayed animation to describe the step to follow to achieve the task of filling the paper tray. *Courtesy of Steven Feiner, Blair MacIntyre, and Doree Seligmann, Columbia University* ***http://graphics.cs.columbia.edu/projects/karma/karma.html***.

Proper documentation is not only useful for professional, it can also be beneficial for the customer. For example, Geiger et al. [2001] present an AR agent that explains how to install a memory card in a digital camera. This new type of documentation presentation is an interesting feature that a consumer could prefer in comparison to a regular manual in PDF.

An AR system if well designed can improve workflow. For example, Klinker et al. [2001a] gather information available in CAD systems and instruction manuals to create an AR ready document that could be used in many different scenarii. They present a prototype to support the maintenance of a nuclear power-plant maintenance, driven by the idea that if the right information is given at the right time then the worker should be more efficient and therefore the downtime of the power-plant could be shortened or at least be on schedule.

The work presented in this thesis is directly related to this body of work on AR supported commissioning. We proposed a global solution to offer verification tool, reporting capability and accessible documentation to construction workers.

The applications of AR as the ultimate interface for a maintenance task are plethoric and it is the focus of the next section.

A.4 Inspection and Maintenance

Many system have been develop to support the maintenance of manufactured goods for example: for radar control devices [Regenbrecht and Specht, 2000], for nuclear power-plants [Ishii et al., 2004, Shimoda et al., 2005], for airplanes [Mizell, 2000, Majoros and Neumann, 2001], for streetcars [Regenbrecht et al., 2005], for cars [Platonov et al., 2006], etc. In this section we describe some of these projects that cover most of the applications and themes developed over the years.

Feiner et al. [1993] describe the first maintenance and repair task supported by AR . The KARMA (Knowledge-based Augmented Reality for Maintenance Assistance) system guides graphically the user through the repair of a printer (charge paper, change cartridge, etc). The system automates the design of augmentations that explain how to perform a 3D task with a set of methods (related to display) and evaluators (related to the accomplishment of a task). An action in the world is recognized by KARMA and interpreted to change the state of the system. For example, a new augmentation is displayed corresponding to the next step. Augmentations, displayed with a HMD, help the worker in localizing and identifying action to be performed using highlights, labels, and animations as show in Figure A.4.

For mechanics training, Rose et al. [1995] propose to display names and function of the different part of an engine. They present typical procedures to train workers by displaying visual augmentations. Using a tracked object, the trainee can query information about the real objects or using a 2D mouse for the virtual parts [Klinker et al., 1997]. The system can present a variety of information, such as meta data (e.g. repair logs). They emphasize on the need for object interaction. For example, an AR system should understand the modification applied to the scene (e.g. when a piece is removed during the repair).

Reading paper-based documentation to perform a complex maintenance task is a long and accepted tradition in the industry even if it is not the most productive method. Neumann and Majoros [1998] propose an IT system that would support a maintenance worker to test the circuitry of an aircraft by displaying augmentation. This AR system gives information on the task to perform (testing) and can sense the step of the process (e.g. a dust cap has been removed, which calls for the next step in the process). It can also show hidden objects (e.g. give a preview of what is under the dust cap). They test their system using an aircraft mock-up and demonstrate that AR is particularly attractive as an information technology.

For the inspection of a water distribution system, [Goose et al., 2003, 2004] develop SEAR (Speach enabled AR) system. The worker interact with the system using vocal command. A technician, performing a servicing task, is supported by a PDA that can sense his location. This location triggers an augmentation of the current view with a virtual model and avails context aware speech interaction. For example he can "vocally" ask a valve for its pressure or a tank for its temperature, this triggers a query in the plant managing software to check on this specific status. The combination of a simple interaction and a tight integration to the IT structure is clearly beneficial for the worker, as he has information constantly and immediately available.

Some systems try to integrate measurements reading (e.g. oscilloscopes) to the augmentations. This is to avoid task switching. For example, Sato et al. [1999] present two prototypes. They develop a desktop-based system that uses a half tainted mirror to supervise the maintenance of a on a Printed Circuit Board (PCB). The PCB is tracked in real-time and each steps is validated by the MR system, which is reading the measurement from instruments output. They also develop a backpack MR based system to support electrical parts inspection in a industrial compound. Similarly Klinker et al. [2004], in FixIT, uses the current pose of the robot being inspected and the robot sensor to indicate malfunction. The current state of the robot is overlaid to support the worker to find the malfunction.

Regenbrecht et al. [2005] propose maintenance system which uses augmented reality as the user interface to guide astronaut in changing filter for the international space station filter. This was only an earth-located demo because of the constraints that are related to performing a demo in outer-of-space.

AR Maintenance systems can close the gap between the diagnosis software and the malfunction documentations because while supporting the worker in performing his task the system can document the procedure. This automatic documentation is a clear benefit for the worker.

A.4.1 AR Supported Maintenance with Remote Experts

In the previous section, we focused our attention on AR systems that directly support the users. Another popular approach is to support the user by giving him access to an expert. In this case, the worker in charge of the maintenance can take care of the repair by himself, but sometimes he does not find the problem and he would benefit from the knowledge of an expert. The interaction between the expert and the worker needs to be effective. The expert needs to understand the problem that the worker is facing and the worker needs to understand the instructions given by the expert. It is why the audio communication between them is often augmented with a video feed. The access to a video is the perfect scenario to demonstrate the benefit of AR. AR for remote expert system is first sketched for tele-training by Rekimoto and Nagao [1995] as a general purpose system. It has been implemented for support specific task such as electronic switchboard repair [Zhong et al., 2003], AC repair [Comport et al., 2003], electronic diagnosis [Ladikos et al., 2008], etc. We describe in this section the most elaborated remote expert applications that use Augmented Reality.

To fight the constant increase in complexity of maintenance tasks (preventive, repair, upgrade) Lipson et al. [1998] propose to use an on-line product maintenance system that would not require the field worker to be an expert. This would avoid for the expert to be flown for diagnosing the problem or performing the repair. The expert could support several complex repairs using his advance knowledge in different remote locations at the same time. This is clearly beneficial for products, which need constant maintenance such as aircrafts, medical equipment and production plants. They demonstrate their ideas to test a hard-disk cabinet. Their system can help to guide the field worker using augmentation, additionally it reports directly back to the head-quarter for an automatic log of the maintenance.

SLAM system are very popular for remote expert application. For example, Davison et al. [2003a,b] use their SLAM system to map the real environment to allow simple interaction. The expert can indicate an area of interest in a stabilized 3D world, in comparison with a jittery video stream. Reitmayr et al. [2007] pushed this idea further by allowing the expert to annotate the 3D world. They demonstrate their system to support the maintenance of a computer. The local geometry is estimated based on SLAM and the annotations sketched by the expert are snapped on the geometry. This allow to precisely describe the task to be performed as shown in Figure A.5.

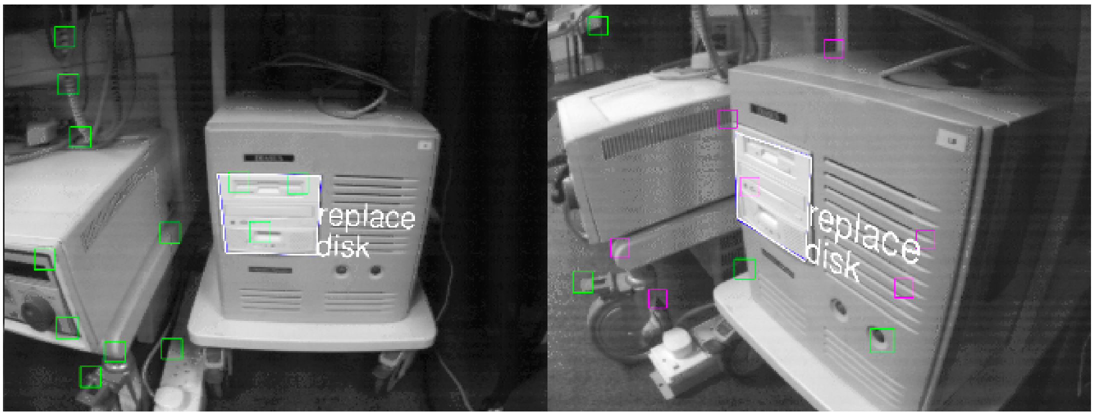


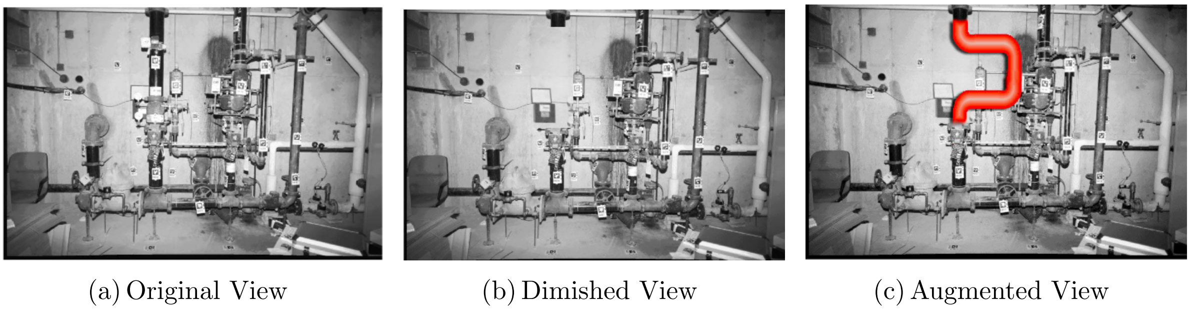
Figure A.5: **Remote Export Using Augmented Reality**. The expert indicates, in his video view, the part of the computer to be repaired. The annotation is transfered to the field work and will follow the user motion as it attached to the 3D world. *Courtesy of Gerhard Reitmayr, Ethan Eade and Tom W. Drummond* `http://mi.eng.cam.ac.uk/~gr281/slamannotations.html`.

A.5 Redesign and Decommissionning

When a product is reaching the end of life, it needs either to be recycled. This might be a revamping or decommissioning procedure. This process for large system are planned in advance not only to minimize the labor cost but also to limit the exposure to hazardous materials, for example when dismantling a nuclear power-plant. In this section, we describe AR systems that support the end of life cycle of a product.

Siemens Corporate Research is extremely active in trying augmented reality to support industrial processes. They played a major role in trying to change the workflow of traditional industry [Zokai et al., 2003]. For example, they look at how Augmented Reality could help to illustrate a revamping procedure. They allow maintenance planners to remove objects (e.g. a pipe) from the scene. This was possible because they have access to images registered to a CAD system. The real pipe would then be replaced by a virtual new pipe that would be designed using a CAD system. This helps the planners to know whether this could create a clash with an object not represented in the CAD data. Figure A.6 demonstrates the possibility offered by such a diminished reality system for revamping.

Augmented reality is also used to support decommissioning of nuclear power plant. This task is heavily regulated for obvious security reasons. It needs first to be planned and the feasibility of the process need to be verified. Then the actual dismantling occurs. Progress needs to be constantly documented. When the decommissioning is finished the work achievement is verified and the CAD model is annotated to reflect the current physical state of the plant. Finally the area, where the dismantling occurred is cleaned. Ishii et al. [2009] demonstrate the benefit of AR for the dismantling of an ion tower. They introduce new technologies for safe and efficient decommissionning work of contaminated zones. Their system support the work by ensuring that the cuts made to the surrounding pipes are localized where they are supposed to. It also monitors the work and record the



(a) Original View (b) Dimished View (c) Augmented View

Figure A.6: **Diminished Reality Used to Illustrate a Revamping Procedure**. The task planner can erase a pipe from a picture, using the information from neighboring views and superimpose the model of the replacement module (in red). *Courtesy of Nassir Navab, Siemens Corporate Research [Zokai et al., 2003].*

progress made. Finally it gives the field worker a direct access to the CAD data on site [Shimoda et al., 2007].

A.6 Pitfalls to Avoid When Creating a Real IAR Application

Most of the applications described in this chapter were only prototypes, few of them were field tested and one has become a product [Echtler et al., 2003]. It was claimed that the main reason for failure is related to hardware and tracking technology [Weidenhausen et al., 2003, Klinker et al., 2004]. But, it also failed because most researchers did not look at the existing industrial processes. The Industry requires a different set of time-lines, as any new developed system need to be beneficial the company in the near future. The industry is not always interest in "how clever" a technology is. It mainly wants that the new system improves their finances [Siltanen et al., 2007]. Overall, AR solution should be integrated in existing workflow [Navab, 2004] to guarantee an easier acceptance of the new work procedure, as the industry is often reluctant to completely change work procedures. The solution needs to be scalable for it to be applicable out of the a lab [Klinker et al., 2004, Navab, 2004]. This aspect is often overlooked, but it offers a great challenge, which is one of the reason why AR research is so different than other field. It is not sufficient for a solution to be cost beneficial, its design needs to involve the complete company and not only one department. It will be simpler to push a product out if the complete company sees the benefit and not only one department [Regenbrecht et al., 2005]. Another reason for failure is that the solution is not integrated into any workflow. It is to often designed to apply some algorithms or methods and is rarely thought before hand. Finally a new solution should not be an overkill [Navab, 2004, Regenbrecht et al., 2005]. Unfortunately, this is often the sin of AR researchers that are too excited about a technology that they completely forget about the arshe reality of the industrial context. Regenbrecht summarizes this precept by citing Albert Einstein "Everything should be made as simple as possible, but not simpler". This is true for industrial AR projects.

Ultimately using AR should enfold new applications. For example, as an AR system has access to all sort of measurements, it could offer after action review that could help the worker to perform his everyday job better but also allows for testing new workflows [Quarles et al., 2008]

In the application we developed during this thesis we tried to follow these precepts. Our system was integrated in the inspection workflow. Currently, the civil engineers that followed the construction acquired pictures to document the construction progress. By integrating these pictures with the CAD model not only we offered new possibilities for discrepancy evaluation, we also made use of what was available to us: pictures and CAD model. We have also been careful to stay in an environment that the user knows. We avoided as much as possible for them to learn new process in order to use our systems. For example, we did not used visual markers that would require a cumbersome installation. We developed our solution as a CAD model viewer, so they could keep their automatism. Our system was not limited to only verify that a power-plant is built as planned but in the future it could offer new applications such as maintenance planning based on the mixture of CAD data and plant images.

APPENDIX

B

VID FRAMEWORK

The project, developed around this thesis, followed strict software guidelines and implementation road-maps elaborated yearly in cooperation with Siemens CT and Areva NP.

In this Chapter, we discuss some of the resulting requirements and implementation details that, we think, could be useful for someone trying to reproduce some of the aspect of the VID system. Resulting hardware requirements are discussed in Section B.1. In Section B.2, we explain how we transfer the design information from a legacy database to a database that could be employed by our software. Then we discuss implementation details of VID in Section B.3. Finally, the design of the graphical user interface is discussed in Section B.4.

B.1 Hardware Requirements

The hardware necessary to use VID is a digital camera and a computer.

The camera used is preferably a high resolution DSLR[1] camera with a wide-angle lens. This is to minimize the number of images to acquire because one shot covers more area and the high resolution capture fine details. For example, it allows for accessing fine details such as reading component identification number. Because of limited light condition, it is recommended to employ a tripod to obtain sharp images even when a long exposure is used. Finally, the focal length should be fixed so the camera can be calibrated once. The calibration is usually performed before and the zoom and focus ring blocked using tape to avoid any movement of the ring. We mostly used during our field test a Canon EOS 400D mounted with wide lens and more recently we used a Canon EOS 50D.

The PC used to run the software does not require any specific hardware. It should have a dedicated graphic card for the renderer to have an interactive frame rate. Since the software is multi-threaded, better performances are available on multi-core machines. The majority of test machines were laptops.

[1] DSLR: Digital Single-Lens Reflex.

B.2 Data Storage

Our industrial partners used PDMS [Aveva] for the design of their virtual models. This powerful tool takes care of storage and versioning. For security reasons, we were not allowed to directly interact with PDMS. We could only create report (complex SQL request to the server) that contained all the information necessary for our use. In the next Section B.2.1, we discuss past approaches to retrieve data from legacy system for use in Augmented Reality. Then in Section B.2.2, we describe the method employed and the resulting database diagram. Finally in Section B.2.3, we present extensions made to the exported model to handle images and to store annotations.

B.2.1 Prior Work on Information Transfer from Legacy Database for AR

The first research group that pointed out the problem linked with exporting information from legacy database was the ECRC [Klinker et al., 1998, 2001b]. They emphasize on the unreliability of CAD and GIS. They also developed the idea of a *"Reality Model"* that is different from the heterogeneous data exported from the miscellaneous documents available. This would be a 3D model that is verified and that contains all the information necessary for an AR system. In their applications, Klinker et al. create this model by merging surveyed 3D points, 2D maps and images to create an as-built textured 3D model usable for the calibration of an AR system. They also point out that a properly created *Reality Model* should not loose meta-information linked to the original documents such as part's condition, images, video clips or instructions manual because it can facilitate the creation of new AR application such as AR supported maintenance.

The VTT group in [Siltanen et al., 2007] argues that industrial application of AR are guaranteed to fail if attention is not paid about existing industrial processes in the information management. In order to create scalable solutions, AR has to use automatic means to access available data. They support the plant management through its lifecycle. The developed plug-in and data access system for AR that directly connects to one unique central data warehouse. They produce meaningful information by studying the data related ontology. This way they are able to keep connection between model and meta-data for different AR supported applications such as installation and maintenance.

Managing models and information for AR is also a focus of the ICG group at TU Graz for many years [Schmalstieg et al., 2007, Mendez et al., 2008, Schall et al., 2008] . They point out the problem of handling semantic information that organizes the digital models. If, during export, the semantic is discarded too early, it diminishes the interaction possibility in the rendering pipeline, but interpretation of the model from its semantic is time consuming therefore a delicate balance has to be selected. Their system is flexible. It, for example, permits filtering actions (e.g. display only pipes). They generate their model from GIS data of underground structure (telephone line, gas pipe, water, etc) that they transcode in procedural models, which can be translated on the fly to a renderable 3D model with the desired properties (material, level of detail, etc). The language that they employ to achieve this task is GenerativeML. The scene itself is represented using a

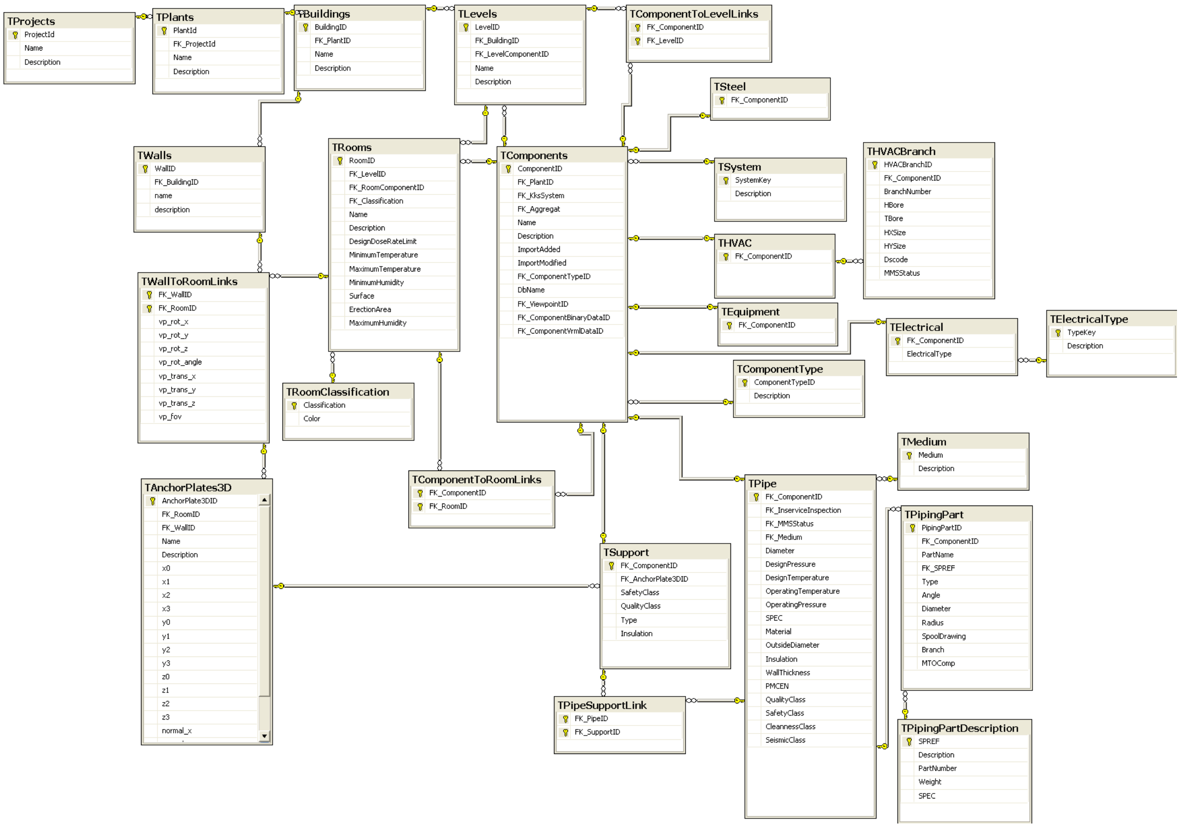


Figure B.1: **CAD model database diagram**: This represent all information exported from the legacy database used store the design information. See the hierarchy from Project to room and how the table TRooms and TComponents are central in this diagram.

scene graph for rendering performances. Additionally they linked business and touristic information to location or model part to create additional meta-data. They can also use authoring tool to extract the specific information they need and the required rendering property thus creating a highly flexible transfer pipeline.

B.2.2 Data Collection from PDMS

When developing our model export procedure, we followed the guidelines introduced by researchers in Industrial Augmented Reality, presented in Section B.2.1.
These are to:

- keep semantic information,
- export only proper/verified 3D information for registration,
- create models that can be rendered at an interactive framerate.

Additionally to these specifications and because of the large amount of data we have to handle, we need fast access to the information (e.g. Access to a room, load and render models and images).

The 3D model stored in PDMS can be accessed through complex report procedure. First, we export the semantic inherent of the model. For example, a project including several plants themselves include a set of buildings. These are organized by levels and these levels are divided by rooms. Keeping this information is extremely important for scalability purposes because of the amount of data to handle. To put in perspective the quantity of data we have to handle, in the full-scale scenario, the project includes a total of 50 buildings, 399 levels, 2678 rooms. The 3D model stored in our database is composed of 78,313 components. It would not be realistic that each time that the software is launched that we load the complete structure of the project, but using the semantic of the model, we can access progressively the model, thus minimizing the transfer from the database to the client.

Additionally to the hierarchy of the model, a complex classification exists with the components between air-conditioning system, supports, pipes, electrical, equipments (e.g. pumps), Steel structures, Anchor-plates. This helps the classification and accessibility of different components. Furthermore, because of the standardization of some industrial components, they can be divided in standard sub-parts; this is particularly true for air conduct and pipes. Using this structured representation, we could also generate on the fly the model to regulate the level of detail as in [Mendez et al., 2008]. But modifying the geometric properties of the model, which might alter the decision making on discrepancy, is to be avoided at all cost. Therefore each model represents a (or set of) complete component(s).

Our industrial partners created a complete procedure to export 3D models from PDMS to VRML, which was optimized to obtain light model. The use of VRML was decided at the early stage of the project and later on discarded because of performance issues related to the rendering pipeline. Instead we use Open SceneGraph model (OSG). We automatically transcode the VRML exported from PDMS to OSG. Each component is then stored and linked to the database.

The *Reality Model* used for registration as mentioned earlier is based on anchor-plates. They are therefore stored as a set of 3D points. Additionally, because their geometric property can be modified (e.g. after a survey), we do not additionally store a renderable model but generate them on-the-fly. This does not decrease performances because of their geometric simplicity: a rectangle.

The diagram of the exported model is visible in Figure B.1. We exported all the necessary information to maximize the usability of our software and we kept the semantic that binds the original model. An interesting “plus” of our database model is the semantic relation that we retrieved from the legacy model that enforces some consistency. For example, we do not store duplicates, a unique model is stored per component. Additionally, we can sort meta-information, for example access each pipes that has cold water for medium and that goes through a given level of the plant. In order to store the information used in the software we augmented the original diagram with table that stores images, cameras, etc. This is discussed in the next Section B.2.3.

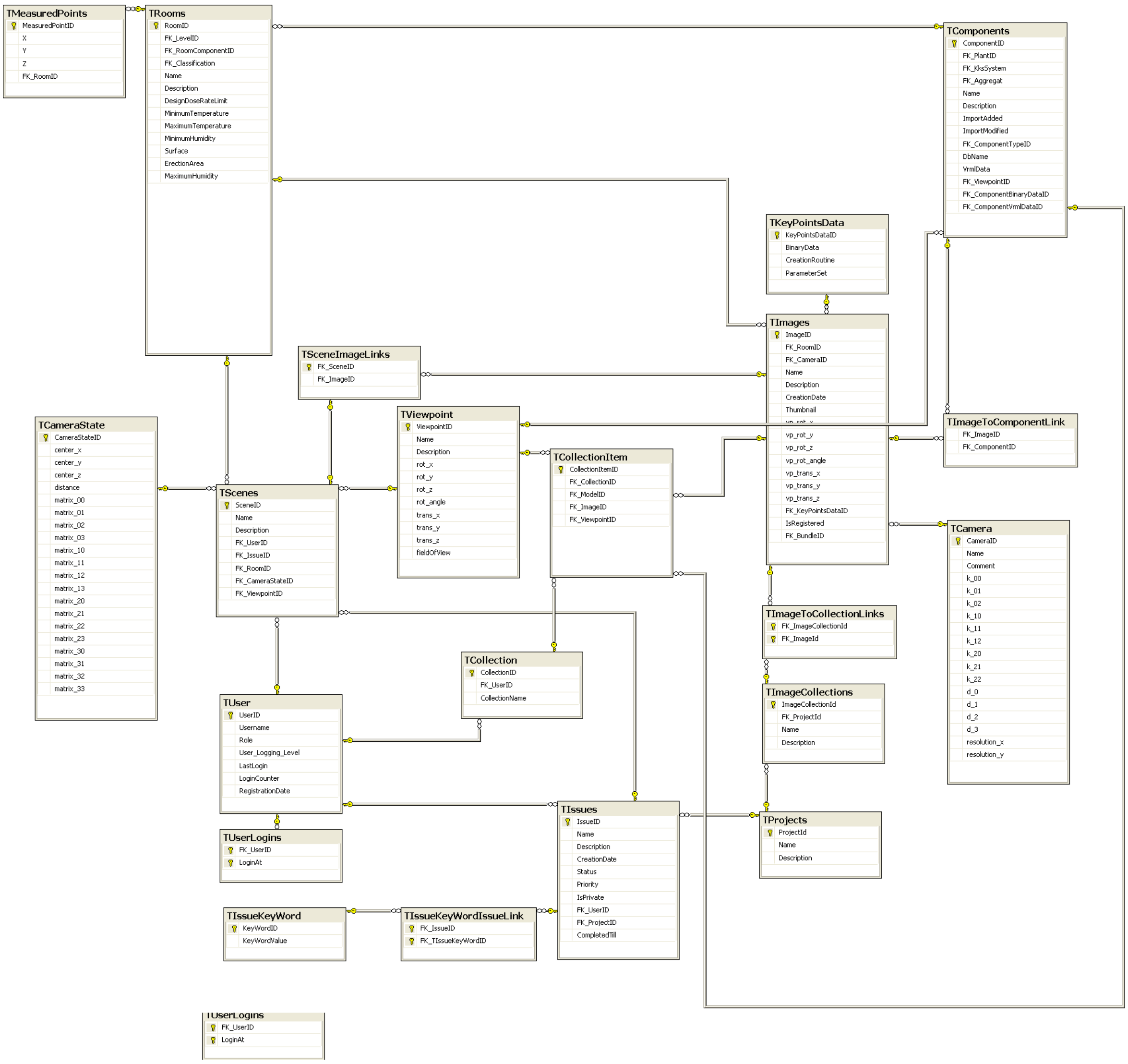


Figure B.2: **Extension of the CAD Model Database** to store information used and generated by VID such as images, image's poses, calibration information. The new information has been completely integrated in the current data model to facilitate the data interaction (e.g. link image to 3D model to document discrepancy).

B.2.3 Database Model Extension for AR Specific Data Storage

The diagram presented in Section B.2.2 is extended to store information used or generated by VID such as cameras, images, and issues. We discuss here some of the reasons and ways to store this data to obtain a coherent and practical relational model. These extensions to the exported database diagram are visible in Figure B.2.

Cameras used in VID have to be calibrated. This is done using a task specific GUI visible in Figure B.3. We store the focal length in x and y, the principal point's shift and the distortion parameters. Optionally information such as the date and who has performed the calibration can be also saved.

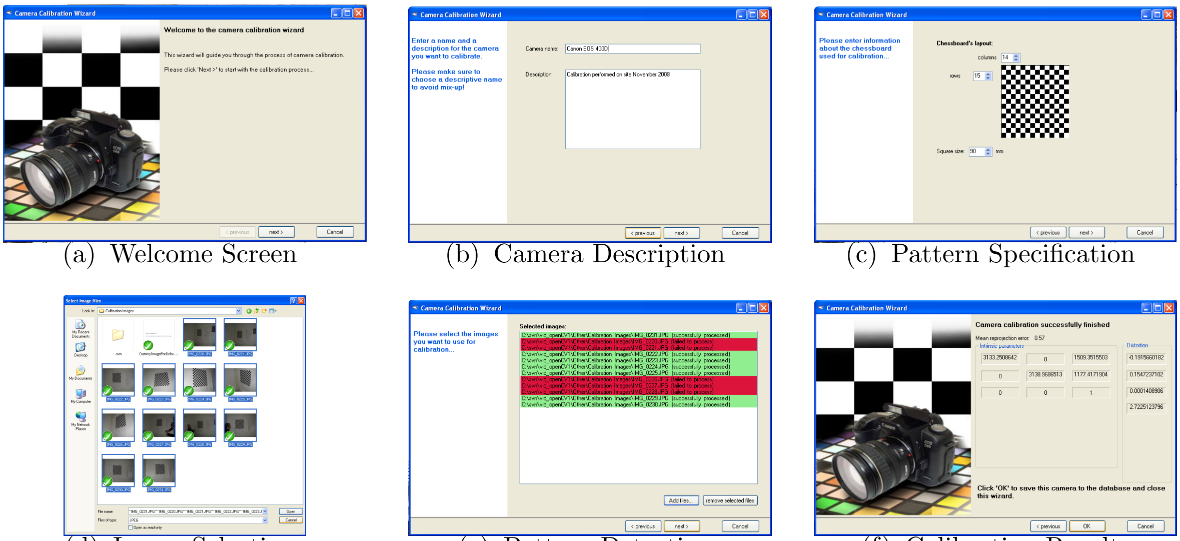
Pictures of the plant are stored in a specific table (TImage). Each picture is linked to a unique camera and a unique room. This allows accessing pictures room-wise and therefore accessing only the components that might be visible in that picture. After registration, its pose is also stored in the database. A time-stamp is also stored to allow a search by date, which allows the user to separate images, for example by construction stage: foundation layout, pipe installation, etc. Optionally meta-data, such as annotation or local features extracted from the image, can be stored within this table.

Bookmarks and Meta-data can be created (automatically or by the users) and stored for later access. Each user can store a collection of images (TCollection). VID can also create a link between images and components based on visibility (TImageToComponentLink).

Issues creation and modification is at the core of VID. They store problems discovered with the model: discrepancies and are organized room-wise. They contain images collections, a list of components and of different viewpoints. This should offer a good documentation system. Additionally annotated screenshots and textual comments can be stored in the database. The GUI developed to edit issues is visible in Figure 4.7.

B.3 Augmented CAD Software - Implementation Details

The software was developed for Microsoft Windows XP, this was a requirement from target user of the software. It uses the .Net framework 3.0 to have a similar look and feel than the software suites that they already use. It is developed using the C++ and C++/CLI under Microsoft Visual Studio 2008. All the data is accessed via NHibernate (a port of Hibernate to the .Net framework [Kuaté et al., 2009]). Information is accessed from the database only when required by the user. The database is stored on a Microsoft SQL Server 2008. Each graphical user interface is a distinct reusable component. Communication between components is done using .Net delegate that takes care of message dispatching [Skeet]. Each component can raise and catch messages.



Welcome to the camera calibration wizard

This wizard will guide you through the process of camera calibration.

Please click 'Next >' to start with the calibration process...

(a) Welcome Screen

Enter a name and a description for the camera you want to calibrate.

Please make sure to choose a descriptive name to avoid mix-up!

Camera name: Canon EOS 400D

Description: Calibration performed on site November 2008

(b) Camera Description

Please enter information about the chessboard used for calibration...

Chessboard's layout:

columns 14

rows 15

Square size: 90 mm

(c) Pattern Specification

.svn DummyImageForDebu... IMG_0220.JPG IMG_0221.JPG IMG_0222.JPG IMG_0223.JPG IMG_0224.JPG IMG_0225.JPG IMG_0226.JPG IMG_0227.JPG IMG_0228.JPG IMG_0229.JPG IMG_0230.JPG IMG_0231.JPG

File name: "IMG_0231.JPG" "IMG_0220.JPG" "IMG_0221.JPG" "IMG_0222.JPG" "IMG_0223.J

Files of type: JPEG

□ Open as read-only

(d) Image Selection

Please select the images you want to use for calibration...

Selected images:

```
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0231.JPG (successfully processed)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0220.JPG (failed to process)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0221.JPG (failed to process)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0222.JPG (successfully processed)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0223.JPG (successfully processed)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0224.JPG (successfully processed)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0225.JPG (successfully processed)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0226.JPG (failed to process)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0227.JPG (failed to process)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0228.JPG (failed to process)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0229.JPG (successfully processed)
C:\svn\vid_openCV1\Other\Calibration Images\IMG_0230.JPG (successfully processed)
```

(e) Pattern Detection

Camera calibration successfully finished

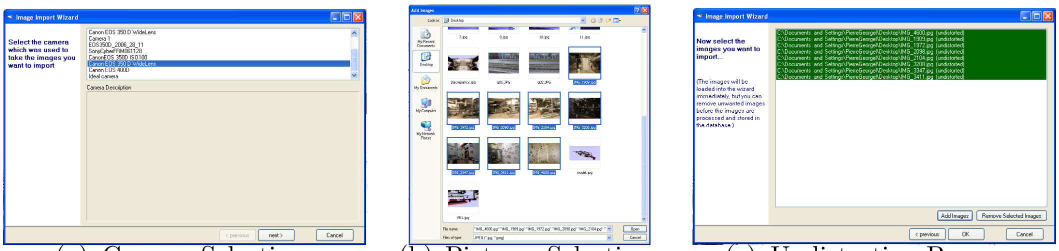
Mean reprojection error: 0.57

Intrinsic parameters			Distortion
3133.2508642	0	1509.3515503	-0.1915660182
0	3138.9686513	1177.4171904	0.1547237102
0	0	1	0.0001408906
			2.7225123796

Click 'OK' to save this camera to the database and close this wizard.

(f) Calibration Result

Figure B.3: **Camera Calibration GUI**. The user is guided through a wizard to obtain the internal parameters of the camera. The results are stored in the database.



Select the camera which was used to take the images you want to import

Canon EOS 350 D WideLens
Camera 1
EOS350D_2006_28_11
SonyCyberFRM061128
CanonEOS 350D ISO100
Canon EOS 350 D WideLens
Canon EOS 400D
Ideal camera

Camera Description

(a) Camera Selection

7.jpg 8.jpg 10.jpg 11.jpg Discrepancy.jpg g01.JPG g02.JPG IMG_1909.jpg IMG_1972.jpg IMG_2098.jpg IMG_2104.jpg IMG_3208.jpg IMG_3347.jpg IMG_3411.jpg IMG_4600.jpg model.jpg VR1.jpg

File name: "IMG_4600.jpg" "IMG_1909.jpg" "IMG_1972.jpg" "IMG_2098.jpg" "IMG_2104.jpg" "

Files of type: JPEG (*.jpg, *.jpeg)

(b) Pictures Selection

Now select the images you want to import...

(The images will be loaded into the wizard immediately, but you can remove unwanted images before the images are processed and stored in the database.)

```
C:\Documents and Settings\PierreGeorge\Desktop\IMG_4600.jpg (undistorted)
C:\Documents and Settings\PierreGeorge\Desktop\IMG_1909.jpg (undistorted)
C:\Documents and Settings\PierreGeorge\Desktop\IMG_1972.jpg (undistorted)
C:\Documents and Settings\PierreGeorge\Desktop\IMG_2098.jpg (undistorted)
C:\Documents and Settings\PierreGeorge\Desktop\IMG_2104.jpg (undistorted)
C:\Documents and Settings\PierreGeorge\Desktop\IMG_3208.jpg (undistorted)
C:\Documents and Settings\PierreGeorge\Desktop\IMG_3347.jpg (undistorted)
C:\Documents and Settings\PierreGeorge\Desktop\IMG_3411.jpg (undistorted)
```

(c) Undistortion Process

Figure B.4: **Picture Importation GUI**. The user is guided through a wizard to import newly acquired pictures. First, he selects the camera used, then the images from the disk, which are then automatically distorted. The new undistorted images are stored in the database.

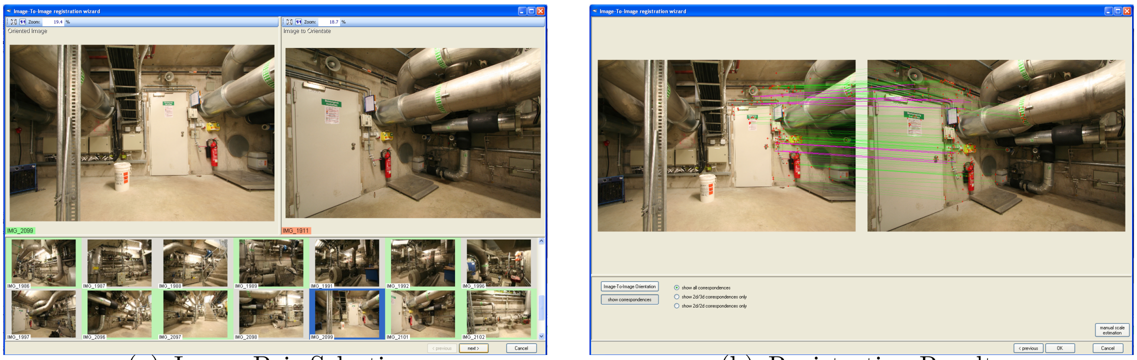


Image-To-Image registration wizard
Zoom: 19.4 %
Oriented Image
Zoom: 18.7 %
Image to Orientate
IMG_2098
IMG_1911
Image-To-Image registration wizard
Image-To-Image Orientation
show correspondences
● show all correspondences
○ show 2d/3d correspondences only
○ show 2d/2d correspondences only
manual scale estimation
< previous
OK
Cancel

(a) Image Pair Selection (b) Registration Result

Figure B.5: **Keyframe-based Registration GUI**. The user selects a keyframe and the image to be registered. The internal parameters and the keypoints are then loaded from the database. Then the essential matrix is computed and extended to a full pose. The resulting camera pose is stored in the database.

All computations are handled within a vision library, which is based on OpenCV (a C/C++ library of computer vision [Bradski and Kaehler, 2008]). This is the functional layer. A wrapper has been developed as part of the functional layer to translate object to managed code usable in .Net.

A review of the development of the application including design history is given in [Schroeder, 2009].

B.3.1 3D Renderer Description

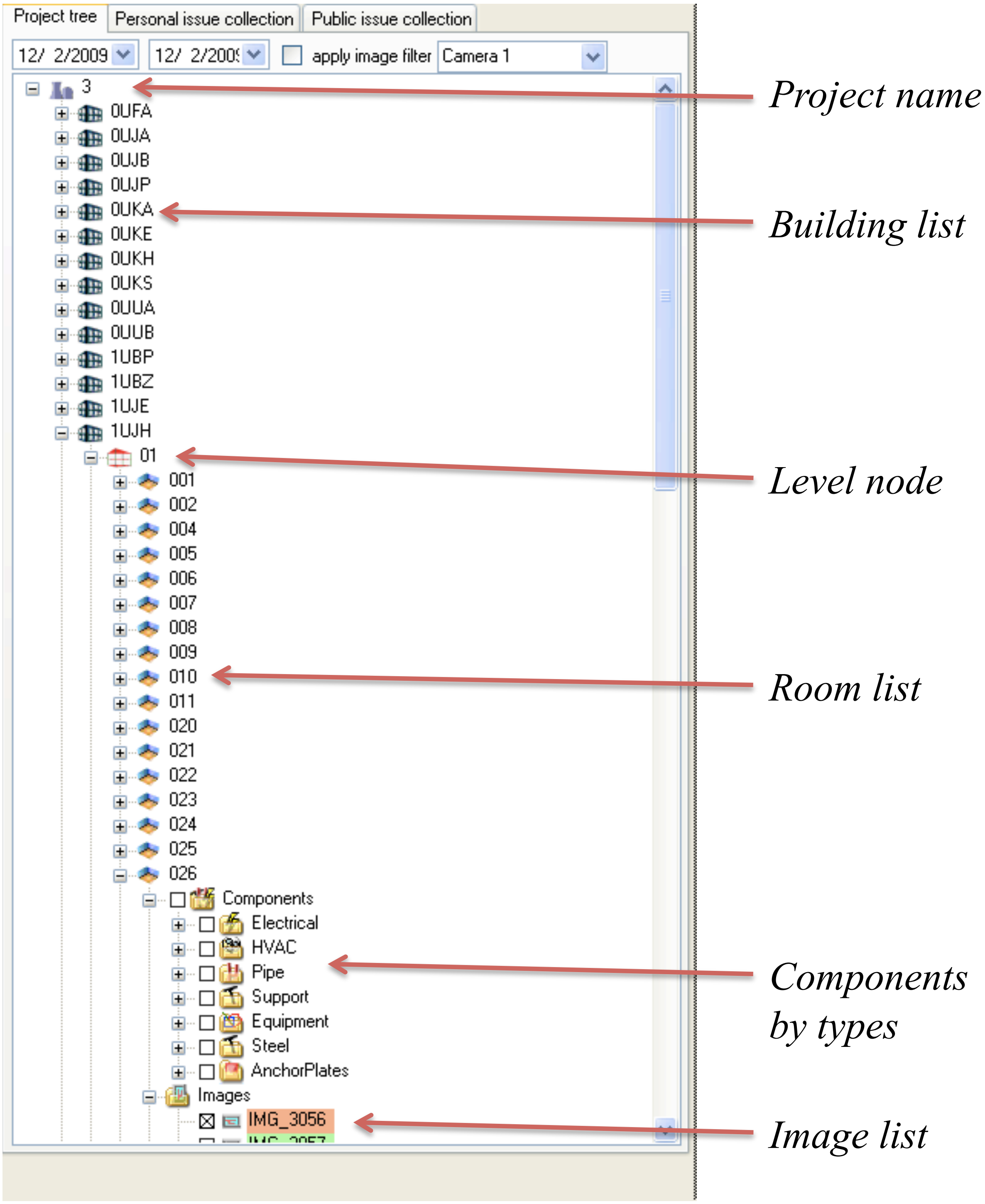
The renderer is based on the open scene graph library because it allows for fast loading of the models and interactive framerate. The scenegraph does automatic view frustum culling (only visible objects actually get rendered), small feature culling (very small features that would only be rendered as dot are discarded) and automatic load balancing of the model with a preference for VRAM (RAM on the graphic card) instead of RAM. Every components material can be conveniently changed on the fly so that the transparency property can be modified. It offers also useful function for ray casting. The images are displayed in the renderer as textured quadrangle.

Some specific components are not directly stored as 3D model with material property, but with only geometric information. For this particular objects procedural rendering functions create the model that are added to the rendering pipeline. In the current system, only anchor-plates (see Section 6.1) are not stored as models, because they are used as geometric information for registration. This was done in order to avoid to have duplicate in the database. Additionally if the 3D data that defines the anchor-plates are modified (for example after a survey), it does not create a discrepancy between the model and the geometric information. This procedural rendering allows us to define the visual property of this specific entity. More information on the implementation of the renderer can be found in [Kaiser, 2009].

B.4 Graphical User Interface

In order to reduce the quantity of work for the end user, VID includes all necessary tools to perform the task at hand. This means that no second software is necessary, we believe that this simplify the use of our platform. For this reason, we tried to automate most of the process when user interactions were necessary. For example to import new images in the system, we use step by step wizard to support the user in the task he performs (c.f. Figure B.4. The GUI to calibrate a camera is presented in Figure B.3 an the one to register an image using a keyframe in Figure B.5 Usually these tasks are not directly linked to documentation or discrepancy inspection, but are necessary in order to later perform them, such as image registration. To create a usable tool that can handle properly such a large amount of data that are CAD models and image collections, we had to design all components with care. We discuss here the tree-view because of the challenges it was to develop.

The tree-view is a custom built tree-view. It does not work as a standard tree-view would because a leaf can have several instances. For example, a component can be part of several rooms therefore it should be carefully loaded. In order to handle this particularity, the communication is message based so information is passed on different leaf instances of an object. For example, if a component is displayed using one of its leaf instance it passes a message to delegate. The tree-view catches it and updates all the leaf instances corresponding to this component. This way, messages can also be sent to Delegate by other GUI components (e.g. using the contextual menu of a thumbnail to make a image visible in the renderer changes the status of its leaf in the tree-view). Delegate makes its messages available to any listeners that needs them. When a node is expended it sends a message to load all its children. This task is then carried out by the data access layer, which uses Nhybernate. We took a special care that Nhybernate only loads next levels of nodes and not the full dependencies. When the loading is finished, it calls for a repaint. The data access layer avoids also creating duplicates.



12/ 2/2009 | 12/ 2/2009 | ☐ apply image filter | Camera 1

⊟ 3
⊞ 0UFA
⊞ 0UJA
⊞ 0UJB
⊞ 0UJP
⊞ 0UKA
⊞ 0UKE
⊞ 0UKH
⊞ 0UKS
⊞ 0UUA
⊞ 0UUB
⊞ 1UBP
⊞ 1UBZ
⊞ 1UJE
⊟ 1UJH
⊟ 01
⊞ 001
⊞ 002
⊞ 004
⊞ 005
⊞ 006
⊞ 007
⊞ 008
⊞ 009
⊞ 010
⊞ 011
⊞ 020
⊞ 021
⊞ 022
⊞ 023
⊞ 024
⊞ 025
⊟ 026
⊟ ☐ Components
⊞ ☐ Electrical
⊞ ☐ HVAC
⊞ ☐ Pipe
⊞ ☐ Support
⊞ ☐ Equipment
⊞ ☐ Steel
⊞ ☐ AnchorPlates
⊟ Images
☒ IMG_3056
☐ IMG_3057

Figure B.6: **VID Custom Tree-View**: it allows the users to navigate through the project hierarchy: plants, levels, rooms. Components are sorted by types under a room-node along with images and issues.

APPENDIX

C

AUTHORED AND CO-AUTHORED PUBLICATIONS

i. An Industrial Augmented Reality Solution For Discrepancy Check, *Pierre Georgel, Pierre Schroeder, Selim Benhimane, Stefan Hinterstoisser, Mirko Appel, Nassir Navab*, International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan, November 2007

ii. A Unified Approach Combining Photometric and Geometric Information for Pose Estimation, *Pierre Georgel, Selim Benhimane, Nassir Navab*, British Machine Vision Conference (BMVC), Leeds, UK, September 2008

iii. How to Augment the Second Image? Recovery of the Translation Scale in Image to Image Registration, *Pierre Georgel, Pierre Schroeder, Selim Benhimane, Mirko Appel, Nassir Navab*, International Symposium on Mixed and Augmented Reality (ISMAR), Cambridge, UK, September 2008

iv. Estimation of Location Uncertainty for Scale Invariant Feature Points, *Bernhard Zeisl, Pierre Georgel, Florian Schweiger, Eckehard Steinbach, Nassir Navab*, British Machine Vision Conference (BMVC), London, UK, September, 2009

v. Photo-based Industrial Augmented Reality Application Using a Single Keyframe Registration Procedure, *Pierre Georgel, Selim Benhimane, Jürgen Sotke, Nassir Navab*, International Symposium on Mixed and Augmented Reality, Orlando, US, October, 2009

vi. Maximum Detector Response Markers for SIFT and SURF, *Florian Schweiger, Bernhard Zeisl, Pierre Georgel, Georg Schroth, Eckehard Steinbach, Nasir Navab*, Vision, Modeling and Visualization Workshop (VMV), Braunschweig, Germany, November 2009

vii. Simultaneous In-Plane Motion Estimation and Point Matching Using Geometric Cues Only, *Pierre Georgel, Adrien Bartoli, Nassir Navab*, IEEE Workshop on Motion and Video Computing (WMVC), Snowbird, USA, December 2009

viii. Recovering the Full Pose from a Single Keyframe, *Pierre Georgel, Selim Benhimane, Jürgen Sotke, Nassir Navab*, IEEE Workshop on Applications of Computer Vision (WACV) , Snowbird, USA, December 2009

ix. Navigation Tools for Augmented CAD Viewing, *Pierre Georgel, Pierre Schroeder, Nassir Navab*, IEEE Computer Graphic and Application (CG&A), special issue on 3D User Interfaces, November-December 2009

Patent Applications

i. Method for comparison of 3D computer model and as-built situation of an industrial plant, *Mirko Appel, Pierre Georgel, Ralf Keller, Nassir Navab*, European Patent Application, filed 2007

APPENDIX

D

ABSTRACTS OF PUBLICATIONS DISCUSSED IN THE DISSERTATION

An Industrial Augmented Reality Solution For Discrepancy Check

Pierre Georgel, Pierre Schroeder, Selim Benhimane, Stefan Hinterstoisser, Mirko Appel, Nassir Navab

International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan, November 2007

Construction companies employ CAD software during the planning phase, but what is finally built often does not match the original plan. The procedure of validating the model is called "discrepancy check". The system proposed here allows the user to easily obtain an augmentation in order to find differences between the planned 3D model and the built items. The main difference to previous body of work in this field is the emphasis on usability and acceptance of the solution. While standard image-based solutions use markers or rely on a "perfect" 3D model to find the pose of the camera, our software uses Anchor-Plates. Anchor-Plates are rectangular structures installed on walls and ceiling in the majority of industrial edifices. We are using them as landmarks because they are the most reliable components often used as reference coordinates by constructors. Furthermore, for real industrial applications, they are the most suitable solutions in terms of general applicability. Unfortunately, they have not been designed with Computer Vision applications in mind. On the contrary, they are often made or painted in such way that they are not easily popping out. They are therefore difficult targets to segment and to track. This paper proposes a solution to extract and match them to their 3D counterparts. We created a software that uses the detected structures for pose estimation and image augmentation. The software has been successfully employed to find discrepancies in several rooms of two industrial plants.

A Unified Approach Combining Photometric and Geometric Information for Pose Estimation

Pierre Georgel, Selim Benhimane, Nassir Navab

British Machine Vision Conference (BMVC), Leeds, UK, September, 2008

In this paper, we present a novel approach for the relative pose estimation problem from point correspondences extracted from image pairs. Unlike classical algorithms, such as the Gold Standard algorithm, the proposed approach ensures that the matched points are photo-consistant throughout the pose estimation process. In fact, common algorithms use the photometric information to extract the feature points and to establish the 2D point correspondences. Then, they focus on minimizing, in a non-linear scheme, geometric distances between the projection of reconstructed 3D points and the coordinates of the extracted image points without taking the photometric information into account. The approach we propose in this paper merges geometric and photometric information in a unified cost function for the final non-linear minimization. This allows us to achieve results with higher precision and also with higher convergence frequency. Extensive experiments with ground truth on synthetic data show the superiority of the proposed approach in terms of robustness and precision. The simulation results have been confirmed by several tests on real image data.

Estimation of Location Uncertainty for Scale Invariant Feature Points

Bernhard Zeisl, Pierre Georgel, Florian Schweiger, Eckehard Steinbach, Nassir Navab

British Machine Vision Conference (BMVC), London, UK, September, 2009

Image feature points are the basis for numerous computer vision tasks, such as pose estimation or object detection. State of the art algorithms detect features that are invariant to scale and orientation changes. While feature detectors and descriptors have been widely studied in terms of stability and repeatability, their localisation error has often been assumed to be uniform and insignificant. We argue that this assumption does not hold for scale-invariant feature detectors and demonstrate that the detection of features at different image scales actually has an influence on the localisation accuracy. A general framework to determine the uncertainty of multi-scale image features is introduced. This uncertainty is represented via anisotropic covariances with varying orientation and magnitude. We apply our framework to the well-known SIFT and SURF algorithms, detail its implementation and make it available. Finally the usefulness of such covariance estimates for bundle adjustment and homography computation is illustrated.

Photo-based Industrial Augmented Reality Application Using a Single Keyframe Registration Procedure

Pierre Georgel, Selim Benhimane, Jürgen Sotke, Nassir Navab

International Symposium on Mixed and Augmented Reality, Orlando, US, October, 2009

In the recent years, many Industrial Augmented Reality (IAR) applications are shifting from video to still images to create a mixed view. This new type of application is called Photo-based Augmented Reality. In order to guarantee the success of these applications, a simple and efficient registration method is required. We present a new method to register an image to a CAD model using a single keyframe. This registration is based on sparse 3D information from the model linked to the keyframe during its offline registration. We demonstrate this method in our in-house IAR software for Visual Inspection and Documentation: VID.

Recovering the Full Pose from a Single Keyframe

Pierre Georgel, Selim Benhimane, Jürgen Sotke, Nassir Navab

IEEE Workshop on Applications of Computer Vision (WACV) , Snowbird, USA, December 2009

Photo-based Augmentation is a growing field in particular for Industrial Augmented Reality (IAR) applications. Registration is at the core of every photo-based AR software. This alignment of the image to the 3D model coordinate system is usually achieved with fiducial markers. When a single keyframe is used, the unknown baseline length has to be estimated in order to superimpose virtual models onto the image. In this paper, we develop an automatic algorithm to augment the relative pose, estimated using a single keyframe, into a full pose that will permit superimposition. This is performed by propagating known 2D-3D correspondences to the target image using perspectively corrected template matching and followed by a refinement of the estimated full pose that combines geometric and photometric information. The performance and the stability of the proposed method is extensively demonstrated on synthetic data and its applicability is shown within an industrial AR software for Visual Inspection and Documentation.

Navigation Tools for Augmented CAD Viewing

Pierre Georgel, Pierre Schroeder, Nassir Navab

IEEE Computer Graphic and Application (CG&A), special issue on 3D User Interfaces, November-December 2009

The creation of a computer aided design (CAD) model is the first step in the development of any modern physical product. This model will be used during the complete life cycle of the product: prototyping, fabrication, maintenance and upgrade. During the construction, a discrepancy between the model and the object can occur. In order to maintain and upgrade the object it is mandatory to have a model that represents the reality. So that one can have an up-to-date model one has to verify it and sometimes update it. We propose a scalable solution where CAD software has been augmented with pictures of the object. Still images have been aligned to the model allowing visualization of the model and the object at the same time. This creates what can be called a mixed view. The virtual camera that renders the model in a mixed view is restricted by the still image because the alignment between the image and the model has to be maintained. We developed tools to navigate in this mixed world. We transposed the zoom and pan from 2D user interfaces in order to navigate in the mixed view. Additionally we introduced tools for intuitive navigation within a set of mixed views.

APPENDIX

E

EXTENDED ABSTRACTS OF MAJOR PUBLICATIONS NOT DISCUSSED IN THE DISSERTATION

How to Augment the Second Image? Recovery of the Translation Scale in Image to Image Registration

Pierre Georgel, Pierre Schroeder, Selim Benhimane, Mirko Appel, Nassir Navab

International Symposium on Mixed and Augmented Reality (ISMAR), Cambridge, UK, September 2008

In this paper, we present an automatic pose estimation (6 DoF) technique to augment images using keyframes pre-registered to a CAD model. State of the art techniques recover the essential matrix (5 DoF) in an automatic manner, but include a manual step to align the image with the CAD reference system because the essential matrix does not provide the scale of the translation. We propose using planar structures to recover this scale automatically and to offer immediate augmentation. These techniques have been implemented in our augmented reality software. Qualitative tests are performed in an industrial environment.

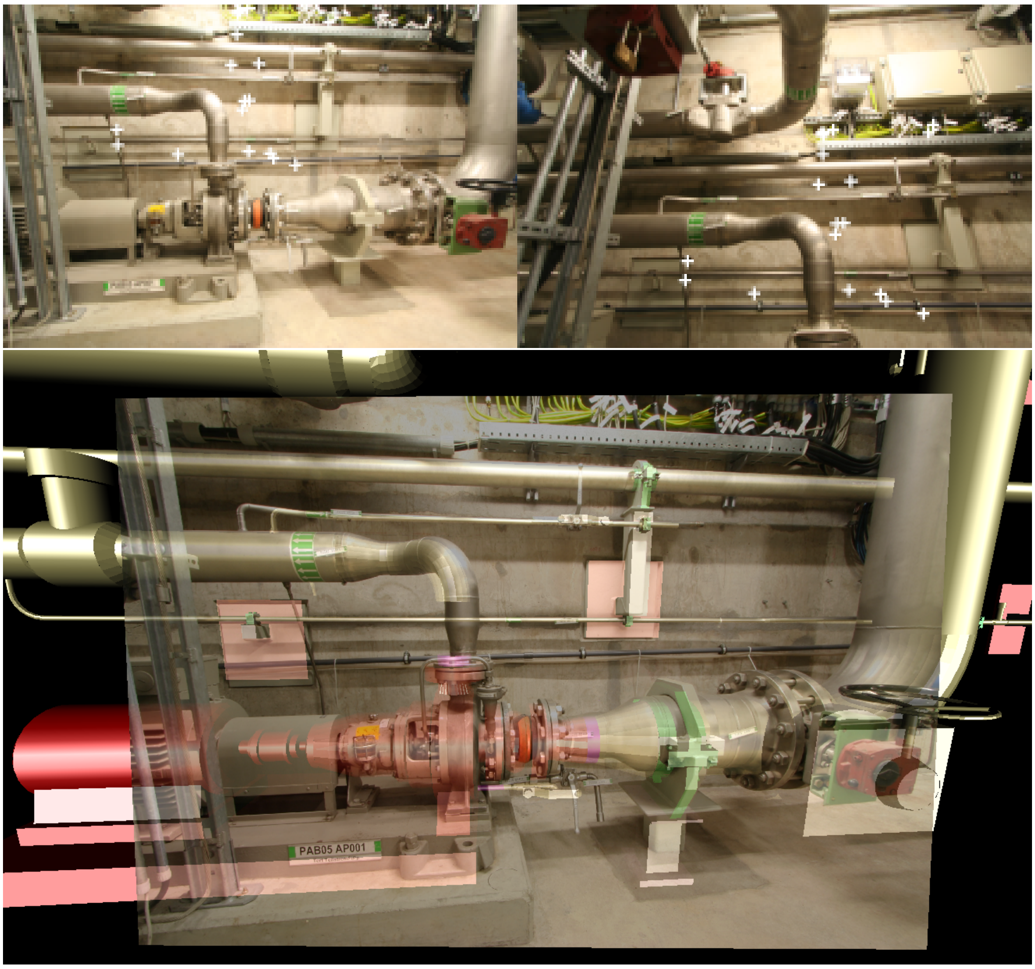


Figure E.1: **Full Pose Estimation from a Single Keyframe Using Planes** Matched features points are triangulated and planar structures a extracted and matched to the CAD model. The distance from the plane can be used to extend the relative pose to a full pose. Features matched to a CAD model plane are marked in white in the (top) images and the resulting augmentation is visible on the (bottom).

Maximum Detector Response Markers for SIFT and SURF

Florian Schweiger, Bernhard Zeisl, Pierre Georgel, Georg Schroth, Eckehard Steinbach, Nasir Navab

Vision, Modeling and Visualization Workshop (VMV), Braunschweig, Germany, November 2009

In this paper, we introduce optimal markers to be used with the SIFT and SURF feature detectors. They can be applied to trigger the detection of feature points at desired locations. Unlike conventional marker systems, we do not propose a standalone solution

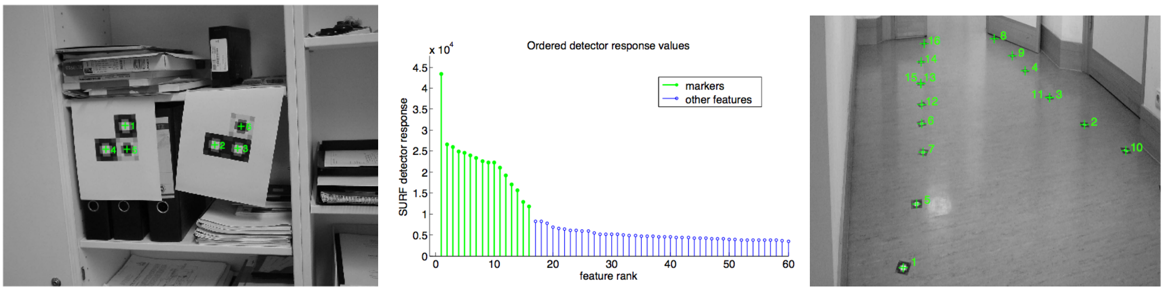


Figure E.2: **Optimal SURF Markers**: (left) two typical markers composed using our optimal image feature (center) the SURF detector response for the synthetic features is distinctive compared to image features (right) Sinthetic featres are detected even when a large perspective deformation is present.

comprising a set of markers and a thereto adapted detection algorithm. Instead, our markers are adapted to existing and established detectors. In particular, we introduce markers optimally suited for SIFT and SURF. We derive the optimal design and show their high detectability within a wide range of different imaging conditions in experiments on both synthetic and real data

Simultaneous In-Plane Motion Estimation and Point Matching Using Geometric Cues Only

Pierre Georgel, Adrien Bartoli, Nassir Navab

IEEE Workshop on Motion and Video Computing (WMVC), Snowbird, USA, December 2009

In this paper, we present a novel approach that, given two sets of unmatched keypoints, simultaneously estimates the in-plane camera motion and keypoint matches without using photometric information. Standard approaches estimate the epipolar geometry based on putative matches, first established with photometric information, then accepted or rejected using the epipolar constraint. Our method discretizes the space of essential matrices at different levels. It searches for the essential matrix and keypoint matches which are the most geometrically coherent. We maximize geometric coherence, that we define as the number of points that can be matched based on the epipolar and unicity constraints. We applied this general framework to sets of images acquired by a moving tripod. We present promising results on simulated and real data.

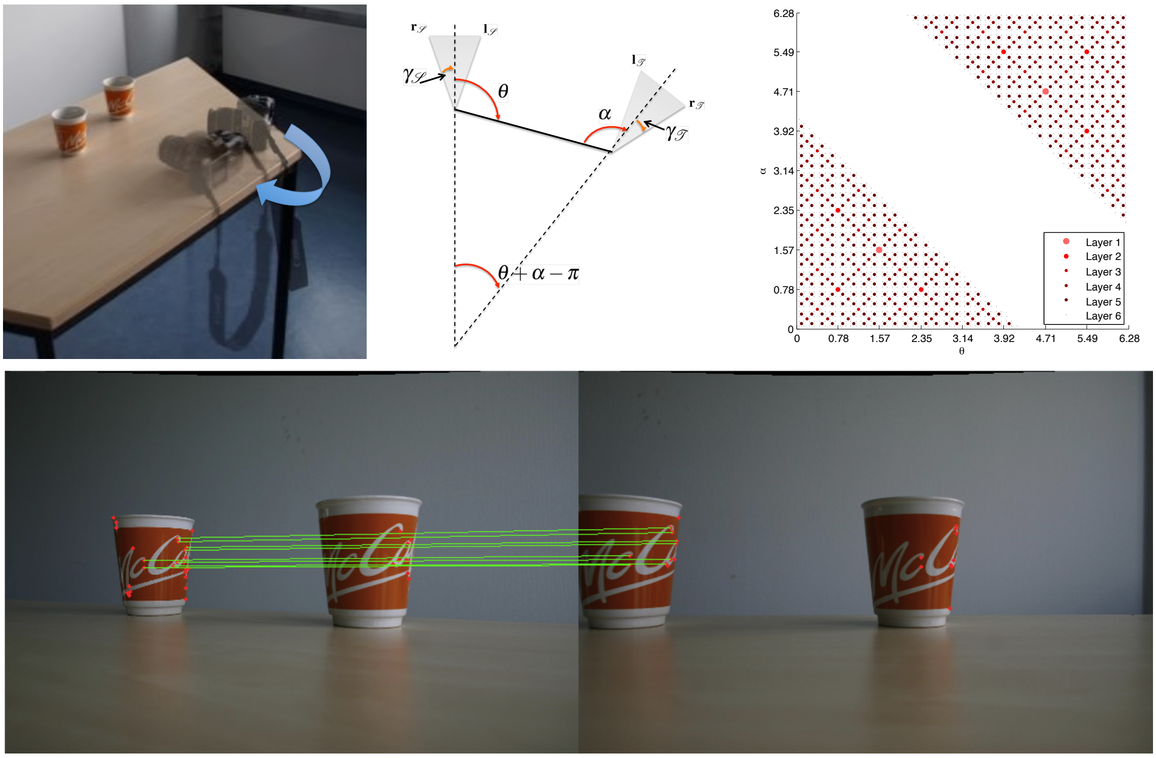


Figure E.3: **Geometric Only Simultaneous Pose and Matching Estimation** (upper left) Exemplary in-plane motion used for this system (upper middle) Top view of the 2-parameter camera setup we consider. (upper right) A quad-tree subdivision to 6 layers of the essential matrix space $\mathbb{E}$ for the camera setup we consider. The empty diagonal comes from the non-overlapping criterion. (center) Registration and matching results using unmatched Harris corners as input using this method.

LIST OF FIGURES

REFERENCES

Carlos Acero. Estimation of internal parameters of a camera using a combination of geometric and photometric information. *Diploma Arbeit TU Muenchen*, pages 1–24, Jun 2009. 128, 138, 166

Klaus H Ahlers, Andre Kramer, David E Breen, Pierre-Yves Chevalier, Chris Crampton, Eric Rose, Mihran Tuceryan, Ross T Whitaker, and Douglas S Greer. Distributed augmented reality for collaborative design applications. *Technical Report ECRC*, 95-03, Dec 1995. URL http://eprints.kfupm.edu.sa/35443/. 173

Francis Aish, Wolfgang Broll, Moritz Störring, Ava Fatah, and Chiron Mottram. Arthur - an augmented reality collaborative design system. *European Conference on Visual Media Production (CVMP)*, Dec 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1374693. 174

Dorin Aiteanu, Bernd Hillers, and Axel Gräser. A step forward in manual welding: demonstration of augmented reality helmet. *Demonstration at the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Dec 2003. URL http://portal.acm.org/citation.cfm?id=946818. 178, 179

P Anandan and Michal Irani. Factorization with uncertainty. *International Journal of Computer Vision*, 49(2):101–116, 2002. 106

Mirko Appel. *From Images and Technical Drawings to 3D Models: A Novel Approach to As-built Reconstruction.* ibidem, 2005. viii, 47

Mirko Appel and Nassir Navab. Registration of technical drawings and calibrated images for industrial augmented reality. *Machine Vision and Applications*, 13(3):111–118, 2002. 5, 68, 181

AREVA Press Office. The AREVA and Siemens consortium is awarded by TVO a contract to build an EPR nuclear power plant , December 2003. URL http://www.areva-np.com/scripts/press/publigen/content/templates/show.asp?P=326&L=US. 3

ARTESAS. Advanced Augmented Reality Technologies for Industrial Service Applications, 2004. URL http://www.artesas.de/site.php?lng=en. 10

ARVIKA. Augmented Reality for development, production and service - Short Description Flyer, 2001. URL `http://www.arvika.de/`. 10

Selen Atasoy, Ben Glocker, Stamatia Giannarou, Diana Mateus, Alexander Meining, Guang-Zhong Yang, and Nassir Navab. Probabilistic region matching in narrow-band endoscopy for targeted optical biopsy. *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 1–8, Jun 2009. 103

Aveva. PDMS. URL `http://www.aveva.com/products_services_aveva_plant_pdms.php`. 190

Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997. 9, 171

Ronald T Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. Jan 2001. URL `http://citeseer.ist.psu.edu/622761`. 171

Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, pages 1–54, Jun 2004. 103, 127

Selim Balcisoy, Marcelo Kallmann, Pascal Fua, and Daniel Thalmann. A framework for rapid evaluation of prototypes with augmented reality. *ACM symposium on Virtual reality software and technology (VRST)*, Dec 2000. URL `http://portal.acm.org/citation.cfm?id=502390.502403`. 175

István Barakonyi, Thomas Psik, and Dieter Schmalstieg. Agents that talk and hit back: Animated agents in augmented reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2004. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1383051`. 22, 177

Adrien Bartoli. Reconstruction et alignement en vision 3d : points, droites, plans et caméras. *Ph.D. Thesis*, pages 1–279, Mar 2003. 27

Adrien Bartoli. Groupwise geometric and photometric direct image registration. *British Machine Vision Conference (BMVC)*, (17), 2006. 103, 127

J Battle, E Mouaddib, and J Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: A survey. *Pattern Recogintion*, 31(7):1–20, Jul 1998. 8

Martin Bauer, Bernd Bruegge, Gudrun Klinker, Asa Macwilliams, Thomas Reicher, and Martin Wagner. Design of a component-based augmented reality framework. *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, Nov 2001. URL `http://citeseer.ist.psu.edu/462045`. 22

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 38, 43, 46, 104, 110, 210

L. Bazizin, J-F. Monier, B. Tintognac, and C. Dechassey. Naissance d'une voiture (Video Documentary), 2009. 3

Reinhold Behringer, Gudrun Klinker, and David W. Mizell. *Augmented Reality - Placing Artificial Objects in Real Scenes- Proceeding of IWAR'99*. A K Peters, 1999. 10

Amir Behzadan and Vineet Kamat. Visualization of construction graphics in outdoor augmented reality. *WSC*, Dec 2005. URL http://portal.acm.org/citation.cfm?id=1162708.1163041. 23, 176

Eran Ben-Joseph, Hiroshi Ishii, John Underkoffler, Ben Piper, and Luke Yeung. Urban simulation and the luminous planning table: Bridging the gap between the digital and tangible. *Journal of planning Education and Research*, Dec 2001. URL http://jpe.sagepub.com/cgi/content/abstract/21/2/196. 20, 174

Selim Benhimane. Vers une approche unifiee pour le suivi temps-reel et l'asservissement visuel. *These de l'Ecole Nationale Superieure des Mines de Paris*, page 181, Jan 2007. 27, 32, 104, 127

Selim Benhimane, Alexander Ladikos, Vincent Lepetit, and Nassir Navab. Linear and quadratic subsets for template-based tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, Apr 2007. 127

Marie-Odile Berger, Gilles Simon, S Petitjean, and B Wrobel-dautcourt. Mixing synthesis and video images of outdoor environments: Application to the bridges of paris. *International Conference on Pattern Recognition (ICPR)*, Nov 1996. URL http://citeseer.ist.psu.edu/344560. 24, 172

Marie-Odile Berger, B Wrobel-dautcourt, S Petitjean, and Gilles Simon. Mixing synthetic and video images of an outdoor urban environment. *Machine Vision Applications (MVA)*, Jan 1999. URL http://citeseer.ist.psu.edu/96071. 24, 172

A. Bhattacharyya. On a measure of divergence between two statistical populations defined by probability distributions, vol. 35. *Bulletin of the Calcutta Mathematical Society*, 1943. 115

Rainer Bischoff and Arif Kazi. Perspectives on augmented reality based human-robot interaction with industrial robots. *IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS)*, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1389914. 20, 171

Rainer Bischoff and Johannes Kurth. Concepts, tools and devices for facilitating human-robot interaction with industrial robots through augmented reality. *ISMAR Workshop on Industrial Augmented Reality*, pages 1–35, Nov 2006. 172

Gabriele Bleser, Yulian Pastarmov, and Didier Stricker. Real-time 3d camera tracking for industrial augmented reality applications. *Journal of WSCG*, Dec 2005. URL http://www.uni-koblenz.de/~cg/Veroeffentlichungen/wscg05_bleser.pdf. 25, 144

Tobias Blum, Oliver Knut Haeberle, Mirko Appel, and Helmut Krcmar. Estimating the financial consequences of using augmented reality in the construction of power plants. *Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*, pages 1–12, Aug 2006. 11

Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab. URL http://www.vision.caltech.edu/bouguetj/calib_doc/. 36

Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly Media Inc., 2008. URL http://oreilly.com/catalog/9780596516130. 196

Wolfgang Broll, Moritz Störring, and Chiron Mottram. The augmented round table-a new interface to urban planning and architectural design. *INTERACT*, Dec 2003. URL http://www.idemployee.id.tue.nl/g.w.m.rauterberg/conferences/interact2003/interact2003-p1103.pdf. 174

Wolfgang Broll, Irma Lindt, Jan Ohlenburg, Michael Wittkämper, Chunrong Yuan, Thomas Novotny, Ava Fatah gen Schieck, Chiron Mottram, and Andreas Strothmann. Arthur: a collaborative augmented environment for architectural design and urban planning. *Journal of Virtual Reality and Broadcasting (JRVB)*, Dec 2004. URL http://jan-ohlenburg.de/pdf/BLOWYNFMS2004_jvrb.pdf. 20, 174

Michael J Brooks, Wojciech Chojnacki, Darren Gawley, and Anton van den Hengel. What value covariance information in estimating vision parameters? *IEEE International Conference on Computer Vision (ICCV)*, pages 1–7, May 2001. 106, 107

Matthew Brown and David G Lowe. Invariant features from interest point groups. *British Machine Vision Conference (BMVC)*, Dec 2002. URL http://www.cs.ubc.ca/labs/lci/papers/docs2002/brown-02.pdf. 45

Martin Byrod, Klas Josephson, and Kalle Astrom. Fast optimal three view triangulation. *LECTURE NOTES IN COMPUTER SCIENCE*, Dec 2007. URL http://www.maths.lth.se/matematiklth/vision/publdb/user/publ/view_paper.php?paper_id=400. 50

John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, Apr 1986. 84

David Capel and Andrew Zisserman. Automated mosaicing with super-resolution zoom. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 885–891, Dec 1998. 103

Thomas P Caudell and David W Mizell. Augmented reality: an application of heads-up display technology tomanual manufacturing processes. *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, pages 1–11, Aug 1992. 176

Peng Chang. Robust tracking and structure from motion with sampling method. *Ph.D. Thesis - Carnegie Mellon University*, pages 1–172, Dec 2002. 128

Christine Chevrier, Salim Belblidia, Dominique Benmouoeek-Antoine, and Jean-Claude Paul. Visual assessment of urban environments. pages 1–6, Nov 1995. 24, 172

Kar Wee Chia, Adrian David Cheok, and Simon JD Prince. Online 6 dof augmented reality registration from natural features. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, 2002. 25, 144

Wojciech Chojnacki, Michael J Brooks, Anton van den Hengel, and Darren Gawley. Revisiting hartley's normalized eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Dec 2003. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1227992`. 125

A Clark and R Green. An adaptive algorithm switching system for image based object registration. *Conference on Image and Vision Computing New Zealand (ICVNZ)*, 2005. URL `http://pixel.otago.ac.nz/ipapers/59.pdf`. 127

Mark J Clayton, Robert E Johnson, Yunsik Song, and Jamal Al-Qawasmi. A study of information content of as-built drawings for usaa. *Technical Report CRS Center/USAA*, pages 1–24, Sep 1998. 55

Brian Clipp, Jae-Hak Kim, Jan-Michael Frahm, Marc Pollefeys, and Richard Hartley. Robust 6dof motion estimation for non-overlapping, multi-camera systems. *IEEE Workshop on Application of Computer Vision (WACV)*, pages 1–8, 2008. 143

Brian Clipp, Christopher Zach, Jan-Michael Frahm, and Marc Pollefeys. A new minimal solution to the relative pose of a calibrated stereo camera with small field of view overlap. *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, Sep 2009. 143

Frédéric Compain and Bruno Lancesseur. EADS-Airbus : une affaire d'États (Video Documentary), 2009. 5

Andrew Comport, Eric Marchand, and François Chaumette. A real-time tracker for markerless augmented reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2003. URL `http://portal.acm.org/citation.cfm?id=946787`. 24, 127, 185

Dan Curtis, David W Mizell, Peter Gruenbaum, and Adam Janin. Several devils in the details: making an ar application work in the airplane factory. *International workshop on Augmented Reality (IWAR)*, Dec 1998. URL `http://portal.acm.org/citation.cfm?id=322695`. 17

Andrew J Davison and David W Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (7):865–880, Jul 2002. 25

Andrew J Davison, Walterio W Mayol, and David W Murray. Real-time localisation and mapping with wearable active vision. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2003a. URL `http://www.igg.tu-berlin.de/~schaefer/AR/Proceedings2003/01240684.pdf`. 25, 185

Andrew J Davison, Walterio W Mayol, and David W Murray. Real-time visual workspace localisation and mapping for a wearable robot. *Demonstration at the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 1–2, Jul 2003b. 25, 185

Alain de Halleux. R.A.S. nucléaire rien à signaler (Video Documentary), 2009. 4

Alan Dodson, Andrew Evans, Bryan Denby, Gethin Wyn Roberts, Robin Hollands, and Simon Cooper. Look beneath the surface with augmented reality. *GPS World*, (Feb):1–3, Oct 2002. 23, 182

Fabian Doil, W Schreiber, Thomas Alt, and C Patron. Augmented reality for manufacturing planning. *EGVE*, Dec 2003. URL `http://portal.acm.org/citation.cfm?id=769953.769962`. 13, 20, 180

Leyza Baldo Dorini and Siome Klein Goldenstein. Unscented klt: nonlinear feature and uncertainty tracking. *Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 1–7, May 2006. 106

David Drascic, Julius Grodski, Paul Milgram, Ken Ruffo, Peter Wong, and Shumin Zhai. Argos: A display system for augmenting reality. *Conference on Human factors in computing systems table (INTERCHI)*, Dec 1993. URL `http://portal.acm.org/citation.cfm?id=169059.169506`. 171

Tom Drummond and Roberto Cipolla. Real-time tracking of complex structures with on-line camera calibration. *British Machine Vision Conference (BMVC)*, pages 1–10, Dec 1999. 24

Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, Oct 1972. 84

Phillip Dunston, Xiangyu Wang, Mark Billinghurst, and Ben Hampson. Mixed reality benefits for design perception. *International Symposium on Automation and Robotics in Construction (ISARC)*, pages 191–196, Sep 2002. URL `http://www.hitlabnz.org/publications/2002-ISARC-MixedReality.pdf`. 20, 175

Florian Echtler, Fabian Sturm, Kay Kindermann, Gudrun Klinker, Joachim Stilla, Jörn Trilk, and Hesam Najafi. The intelligent welding gun: Augmented reality for experimental vehicle construction. *Virtual and Augmented Reality Applications in Manufacturing*, pages 1–27, Sep 2003. 10, 22, 83, 171, 178, 187

The Economist. Augmented reality - reality, only better. *Technology Quarterly*, (December):1–24, Nov 2007. 11

Adel Fakih and John Zelek. Structure from motion: Combining features correspondences and optical flow. *International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008. 127

Olivier Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993. 27

Olivier Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? *European Conference on Computer Vision (ECCV)*, pages 563–578, May 1992. 48, 125

Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images.* MIT Press, 2001. 35, 126, 148

Eliot A Feibush, Marc Levoy, and Robert Cook. Synthetic texturing using digital filters. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Dec 1980. URL `http://portal.acm.org/citation.cfm?id=965105.807507`. 172

Steven Feiner, Blair MacIntyre, and Doree Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53 – 62, Jun 1993. URL `http://portal.acm.org/citation.cfm?id=159587`. 22, 183

Mark Fiala. Artag, a fiducial marker system using digital techniques. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec 2005. URL `http://www2.computer.org/portal/web/csdl/doi?doc=doi/10.1109/CVPR.2005.74`. 17

Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), May 1981. 104, 126, 128

Andrew Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec 2001. URL `http://doi.ieeecomputersociety.org/10.110910.1109/CVPR.2001.990465`. 103

R. Flecher. *Pratical Methods of Optimization - Second Edition.* Wiley, 1987. 30, 31

Wolfgang Förstner. Ein verfahren zur schätzung von varianz- und kovarianzkomponenten. *Allgemeine Vermessungs-Nachrichten (AVN)*, (86):446–453, Feb 1979. 132

Wolfgang Förstner. Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision. *Computer Vision, Graphics and Image Processing*, 40:273–310, May 1987. 104

Wolfgang Förstner and E Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. *ISPRS Intercommission Workshop*, pages 1–25, Apr 1987. 42, 105, 106

Eric Foxlin, Yury Altshuler, Leonid Naimark, and Mike Harrington. Flighttracker: A novel optical/inertial tracker for cockpit enhanced vision. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2004. URL `http://portal.acm.org/citation.cfm?id=1033718`. 22

Nobuyuki Fujiwara, Toshikazu Onda, Hideyoslii Masuda, and Kazuhiro Chayama. Virtual property lines drawing on the monitor for observation of unmanned dam construction site. *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, pages 1–4, Aug 2000. 20, 176

Yasutaka Furukawa and Jean Ponce. Dense patch models for motion capture from synchronized video streams. *Willow Tech. Report*, 02-07:1–9, Apr 2007. 138

J Gausemeier, J Fruend, and Carsten Matysczok. Ar-planning tool: designing flexible manufacturing systems with augmented reality. *EGVE*, Dec 2002. URL `http://portal.acm.org/citation.cfm?id=509714`. 20, 61, 180

Christian Geiger, Bernd Kleinnjohann, Christian Reimann, and Dirk Stichling. Mobile ar4all. *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, pages 1–2, Nov 2001. 183

Pierre Georgel, Pierre Schroeder, Selim Benhimane, Mirko Appel, and Nassir Navab. How to augment the second image? recovery of the translation scale in image to image registration. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2008. URL `http://ar.in.tum.de/pub/georgel2008ismar/georgel2008ismar.pdf`. 144

Stuart Goose, Sandra Sudarsky, Xiang Zhang, and Nassir Navab. Speech-enabled augmented reality supporting mobile industrial maintenance. *IEEE Pervasive Computing*, Dec 2003. URL `http://doi.ieeecomputersociety.org/10.110910.1109/MPRV.2003.1186727`. 7, 20, 184

Stuart Goose, Sinem Guven, Xiang Zhang, Sandra Sudarsky, and Nassir Navab. Paris: Fusing vision-based location tracking with standards-based 3d visualization and speech interaction on a pda. *International Conference on Distributed Multimedia Systems (DMS)*, Dec 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.4908&rep=rep1&type=pdf`. 19, 69, 184, 209

J Gu, E Augirre, and P Cohen. An augmented-reality interface for telerobotic applications. *IEEE Workshop on Application of Computer Vision (WACV)*, Dec 2002. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1182185`. 171

Oliver Knut Haeberle, Tobias Blum, and Helmut Krcmar. Evaluating an innovative technology in the presence of uncertainty. *Americas Conference on Information Systems (AMCIS)*, pages 1–9, May 2006. 11

Nate Hagbi, Oriel Bergig, Jihad El-Sana, and Mark Billinghurst. Shape recognition and pose estimation for mobile augmented reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–7, Sep 2009. 17

Andreas Haja, B. Jahne, and S. Abraham. Localization accuracy of region detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 104, 107

Mika Hakkarainen, Charles Woodward, and Mark Billinghurst. Augmented assembly using a mobile phone. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2008. URL `http://portal.acm.org/citation.cfm?id=1605305`. 20, 177

Robert Hanek, Nassir Navab, and Mirko Appel. Yet another method for pose estimation: A probabilistic approach using points, lines, and cylinders. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, Apr 1999. 104, 127

Marsha Jo Hannah. *Computer matching of areas in stereo images.* PhD thesis, Stanford, CA, USA, 1974. 42

Robert M Haralick. Propagating covariance in computer vision. *International Conference on Pattern Recognition (ICPR)*, 1994. URL `http://www.worldscinet.com/abstract?id=pii:S0218001496000347`. 104, 109

Matthias Haringer and Holger Regenbrecht. A pragmatic approach to augmented reality authoring. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2002. URL `http://portal.acm.org/citation.cfm?id=854991`. 180

Chris Harris and Mike Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 1–6, Jun 1988. 42, 43, 150

Richard Hartley. Estimation of relative camera positions for uncalibrated cameras. *European Conference on Computer Vision (ECCV)*, pages 579–587, May 1992. 48, 125

Richard Hartley. In defence of the 8-point algorithm. *IEEE International Conference on Computer Vision (ICCV)*, pages 1064–1070, 1995. 86, 125

Richard Hartley. Chirality. *International Journal of Computer Vision*, 26(1):41–61, 1998. 49

Richard Hartley and Fredrik Kahl. Global optimization through searching rotation space and optimal estimation of the essential matrix. *IEEE International Conference on Computer Vision (ICCV)*, 2007. 126

Richard Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997. 50, 134, 150

Richard Hartley and Andrew Zisserman. Multiple view geometry in computer vision (2ed,oup,2003). *Cambridge press*, page 672, Aug 2003. 27, 35, 47, 49, 50, 86, 125, 126, 128

Frank A Van Den Heuvel. Trends in cad-based photogrammetric measurement. *International Archives of Photogrammetry and Remote Sensing*, (33):1–12, Oct 2000. 8

Berthold K P Horn. Relative orientation. *Revised A.I. Memo*, 994-A:1–38, Jan 1989. 126

Berthold K P Horn. Recovering baseline and orientation from ‘essential’ matrix. pages 1–10, Mar 1990. 49

TS Huang and Olivier Faugeras. Some properties of the e matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12): 1310–1312, Dec 1989. 49

Hiroshi Ishii, Eran Ben-Joseph, John Underkoffler, Luke Yeung, Dan Chak, Zahra Kanji, and Ben Piper. Augmented urban planning workbench: Overlaying drawings, physical models and digital simulation. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2002. URL http://portal.acm.org/citation.cfm?id=854980. 174

Hirotake Ishii, Koji Matsui, Misa Kawauchi, Hiroshi Shimoda, and Hidekazu Yoshikawa. Development of an augmented reality system for plant maintenance support. *Cognitive System Engineering in Process Control (CSEPC)*, pages 1–8, Aug 2004. 4, 183

Hirotake Ishii, Hiroshi Shimoda, Toshinori Nakai, Masanori Izumi, Zhiqiang BIAN, and Yoshitsugu MORISHITA. Proposal and evaluation of a supporting method for npp decommissioning work by augmented reality. *Systemics, Cybernetics and Informatics (WMSCI)*, 2009. 4, 20, 186

Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu. Robust Video Denoising Using Low Rank Matrix Completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 113

H Johnson and G Christensen. Consistent landmark and intensity-based image registration. *IEEE Transactions on Medical Imaging*, Dec 2002. URL http://www.engineering.uiowa.edu/~n-morph/publications/pdfs/Consistent_Landmarkand_Intensity-based_Image_Registration.pdf. 127

Hagen Kaiser. Implementation of a 3d visualization system for an augmented cad software. *System-Entwicklungs-Projekt TUM*, pages 1–21, Dec 2009. 196

Hagen Kaiser. Real-time structure from motion forindustrial Augmented Reality. Master’s thesis, TUM - CAMP, 2010. 166

Yoshinari Kameda, Taisuke Takemasa, and Yuichi Ohta. Outdoor see-through vision utilizing surveillance cameras. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2004. URL http://doi.ieeecomputersociety.org/10.1109/ISMAR.2004.45. 24, 144

Kenichi Kanatani. Optimal fundamental matrix computation: Algorithm and reliability analysis. *Symposium on Sensing via Image Information (SII)*, pages 291–298, 2000. 104, 106

Kenichi Kanatani. Uncertainty modeling and geometric inference. *Modeling Mathematics Computation*, pages 1–10, Dec 2004. 107, 113, 114

Kenichi Kanatani and Daniel D Morris. Gauges and gauge transformations for uncertainty description ofgeometric structure with indeterminacy. *IEEE Transactions on Information Theory*, 47(5):2017–2028, 2001. 107, 120

Kenichi Kanatani, Yasuyuki Sugaya, and Hirotaka Niitsuma. Triangulation from two views revisited: Hartley-sturm vs. optimal correction. *British Machine Vision Conference (BMVC)*, 2008. URL `http://www.comp.leeds.ac.uk/bmvc2008/proceedings/papers/55.pdf`. 50, 150

Yasushi Kanazawa and Kenichi Kanatani. Do we really have to consider covariance matrices for image feature points? *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 86(1), 2003. 106, 107, 111, 113, 115, 122, 164

Kazufumi Kaneda, Fujiwa Kato, Eihachiro Nakamae, Tomoyuki Nishita, Hideo Tanaka, and Takao Noguchi. Three dimensional terrain modeling and display for environmental assessment. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Jul 1989. URL `http://portal.acm.org/citation.cfm?id=74333.74354`. 172

Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. *International workshop on Augmented Reality (IWAR)*, pages 1–10, Oct 1999. 17

Hirokazu Kato, Keihachiro Tachibana, Masaaki Tanabe, Takeaki Nakajima, and Yumiko Fukuda. A city-planning system based on augmented reality with a tangible interface. *Demonstration at the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 1–2, Jul 2003. 20, 174

Hannes Kaufmann, Dieter Schmalstieg, and Michael Wagner. Construct3d: A virtual reality application for mathematics and geometry education. *Education and Information Technologies*, 5(4):263–276, Nov 2000. doi: 10.1023/A:1012049406877. URL `http://citeseer.ist.psu.edu/315014`. 21

Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Aug 2004. 166

Jae-Hak Kim, Richard Hartley, Jan-Michael Frahm, and Marc Pollefeys. Visual odometry for non-overlapping views using second-order cone programming. *Asian Conference on Computer Vision (ACCV)*, pages 1–11, Sep 2007. 143

Seungjun Kim, NP Mahalik, Anind K Dey, Jeha Ryu, and Byungha Ahn. Feasibility and infrastructural study of ar interfacing and intuitive simulation on 3d nonlinear systems. *Computer Standards & Interfaces*, 30:36–51, 2008. 171

Won S Kim. Advanced teleoperation, graphics aids, and application to time delay environments. *Industrial Virtual Reality (IVR)*, pages 202–207, Jul 1993. 171

Won S Kim. Virtual reality calibration and preview/predictive displays for telerobotics. *Presence: Teleoperators and Virtual Environments*, 5(2), 1996. URL http://trs-new.jpl.nasa.gov/dspace/handle/2014/30385. 171

Gary King, Wayne Piekarski, and Bruce Thomas. Arvino — outdoor augmented reality visualisation of viticulture gis data. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Oct 2005. URL http://portal.acm.org/citation.cfm?id=1104996.1105174. 23

Georg Klein. Visual tracking for augmented reality. *Ph.D. Thesis - University of Cambridge*, pages 1–193, Jun 2006. 32

Georg Klein and Tom Drummond. Robust visual tracking for non-instrumented augmented reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–10, Jul 2003. 24, 104, 127

Georg Klein and David W Murray. Parallel tracking and mapping for small ar workspaces. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–10, Aug 2007. 25

Gudrun Klinker, Klaus H Ahlers, David E Breen, Pierre-Yves Chevalier, Chris Crampton, Douglas S Greer, Dieter Koller, Andre Kramer, Eric Rose, Mihran Tuceryan, and Ross T Whitaker. Confluence of computer vision and interactive graphics for augmented reality. *Presence: Teleoperators and Virtual Environments*, Dec 1997. URL http://wwwbruegge.in.tum.de/static/publications/pdf/77/klinker1997presence.pdf. 173, 184

Gudrun Klinker, Didier Stricker, and Dirk Reiners. The use of reality models in augmented reality applications. *LECTURE NOTES IN COMPUTER SCIENCE*, Dec 1998. URL http://www.springerlink.com/index/CJA94BAE0LJW3RUN.pdf. 24, 83, 173, 190

Gudrun Klinker, Didier Stricker, and Dirk Reiners. Augmented reality: A balance act between high quality and real-time constraints. *IEEE and ACM International Symposium on Mixed Reality*, Dec 1999. URL http://ar.in.tum.de/pub/klinker1999ismr/klinker1999ismr.pdf. 176, 177, 182

Gudrun Klinker, Oliver Creighton, Allen H Dutoit, Rafael Kobylinski, Christoph Vilsmeier, and Bernd Bruegge. Augmented maintenance of powerplants: A prototyping case study of a mobile ar system. *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, Sep 2001a. URL http://citeseer.ist.psu.edu/657786. 20, 183

Gudrun Klinker, Didier Stricker, and Dirk Reiners. Augmented reality for exterior construction applications. *Augmented Reality and Wearable Computers (Eds: W. Barfield und T. Caudell)*, 2001b. URL http://wwwbruegge.in.tum.de/static/publications/pdf/83/klinker2001arbook.pdf. 177, 181, 190, 218

Gudrun Klinker, Allen H Dutoit, Martin Bauer, Johannes Bayer, Vinko Novak, and Dietmar Matzke. Fata morgana–a presentation system for product design. *IEEE and*

ACM International Symposium on Mixed Augmented Reality (ISMAR), Dec 2002. URL http://doi.ieeecomputersociety.org/10.1109/ISMAR.2002.1115076. 20, 175

Gudrun Klinker, Hesam Najafi, Tobias Sielhorst, Fabian Sturm, Florian Echtler, Mustafa Isik, Wolfgang Wein, and Christian Trübswetter. Fixit: An approach towards assisting workers in diagnosing machine malfunctions. *International Workshop on Design and Engineering of Mixed Reality Systems - MIXER*, Jan 2004. URL http://citeseer.ist.psu.edu/646298. 20, 165, 184, 187

Manfred Klopschitz and Dieter Schmalstieg. Automatic reconstruction of wide-area fiducial marker models. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–4, Nov 2007. 21

Kazuhiko Kobayashi, Shinobu Ishigame, and Hirokazu Kato. Simulator of manual metal arc welding with haptic display. *International Conference on Artificial Reality and Telexistence (icat)*, Dec 2001. URL http://www.vrsj.org/ic-at/papers/01175.pdf. 20, 179

Kevin Koeser and Reinhard Koch. Exploiting uncertainty propagation in gradient-based image registration. *British Machine Vision Conference (BMVC)*, Dec 2008. URL http://www.comp.leeds.ac.uk/bmvc2008/proceedings/papers/15.pdf. 106

Dieter Koller, Gudrun Klinker, Eric Rose, and David E Breen. Real-time vision-based camera tracking for augmented reality applications. *ACM symposium on Virtual reality software and technology (VRST)*, Dec 1997a. URL http://portal.acm.org/citation.cfm?id=261135.261152. 17

Dieter Koller, Gudrun Klinker, Eric Rose, David E Breen, Ross T Whitaker, and Mihran Tuceryan. Automated camera calibration and 3d egomotion estimation for augmented reality applications. *International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 199—206, Dec 1997b. URL http://www.cs.iupui.edu/~tuceryan/research/AR/caip-97.pdf. 17, 173

Daisuke Kotake, Kiyohide Satoh, Shinji Uchiyama, and Hiroyuki Yamamoto. A fast initialization method for edge-based registration using an inclination constraint. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–10, Nov 2007. 24

Peter Kovesi. MATLAB and Octave Functions for Computer Vision and Image Processing. URL http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/. 44

Ernst Kruijff and Eduardo Veas. Vesp'r – transforming handheld augmented reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–2, Sep 2007. 23

Pierre Henri Kuaté, Christian Bauer abd Gavin King Gavin King, and Tobin Harris. *NHibernate in Action*. Manning Publications, 2009. 194

Alexander Ladikos, Selim Benhimane, Mirko Appel, and Nassir Navab. Model-free markerless tracking for remote support in unknown environments. *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 1–4, Nov 2008. 185

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Apr 2006. 104

Jung-Min Lee, Kyung-Ho Lee, Dea-Seok Kim, and Chung-Hyun Kim. Active inspection supporting system based on mixed reality after design and manufacture in an offshore structure. *J Mech Sci Technol*, 24(1):197–202, Jan 2010. doi: 10.1007/s12206-009-1129-2. 181

Maarten Lens-Fitzgerald. Layar-was-there-movement-on-the-ar-hype-cycle. *Mobile AR Summit*, pages 1–2, Feb 2010. 165

Vincent Lepetit and Pascal Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Fondations and Trends in Computer Graphics and Vision*, 1(1):1–91, Sep 2005. 47

Vincent Lepetit, Luca Vacchetti, Daniel Thalmann, and Pascal Fua. Fully automated and stable registration for augmented reality applications. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–10, Dec 2003. 25

Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):1–12, Jul 2009. 47

Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(2):225–270, Mar 1994. 39

Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):1–53, Apr 1998. 40, 43, 104

H Lipson, M Shpitalni, F Kimura, and I Goncharenko. Online product maintenance by web-based augmented reality. *International CIRP Design Seminar on "New Tools and Workflow for Product Development"*, Dec 1998. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.6290&rep=rep1&type=pdf`. 20, 185

Yong Liu, Moritz Störring, Thomas B Moeslund, Clause B Madsen, and Erik Granum. Computer vision based head tracking from re-configurable 2d markers for ar. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2003. URL `http://portal.acm.org/citation.cfm?id=946845`. 20

Hugh Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two pro jections. *Nature*, 293:133–135, Jul 1981. 49, 125

M.I. A. Lourakis and A.A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009. doi: http://doi.acm.org/10.1145/1486525.1486527. 51, 52, 121

David G Lowe. Object recognition from local scale-invariant features. *IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, Jun 1999. 104

David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 38, 45, 46, 104, 110

Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, pages 1–6, Jun 1981. 106

Hearst Magazines, editor. *Esquire - Augmented Reality Issue*, volume 152, Dec. 2009. 21, 171

Anthony Majoros and Ulrich Neumann. Support of crew problem-solving and performance with augmented reality. *Bioastronautics Investigators' Workshop*, Dec 2001. URL `http://www.dsls.usra.edu/dsls/meetings/bio2001/pdf/sessions/abstracts/017.pdf`. 183

Valerie Maquil, Sareika Markus, Dieter Schmalstieg, and Ina Wagner. Mr tent: a place for co-constructing mixed realities in urban planning. *GI*, Dec 2009. URL `http://portal.acm.org/citation.cfm?id=1555880.1555927`. 174

Mark Billinghurst. Where's the Reality in Augmented Reality ?, 2007. 165

Lucie Masson, Frederic Jurie, and Michel Dhome. Contour/texture approach for visual tracking. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 661–668, Jan 2003. 127

Jiri Matas, Ondrej Chum, M Urban, and Tomas Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *British Machine Vision Conference (BMVC)*, pages 1–10, Jul 2002. 104

J Meidow. Consideration of uncertainty in computer vision: Necessity and chance. *Pattern Recogintion and Image Analysis*, 18(2):1–6, May 2008. 121

Erick Mendez, Gerhard Schall, Sven Havemann, Dieter Fellner, Dieter Schmalstieg, and Sebastian Junghanns. Generating semantic 3d models of underground infrastructure. *IEEE Computer Graphics and Applications (CGA)*, 28(3):48–57, Nov 2008. 5, 190, 192

Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, Jul 2004. 44

Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1–34, Jun 2005. 39

Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, T Kadir, and Luc Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 1(65):43–72, Apr 2005. 38, 41, 43, 105

Paul Milgram and Herman Jr Colquhoun. A taxonomy of real and virtual world display integration. *Mixed Reality, Ed. Yuichi Ohta Hideyuki Tamura*, pages 1–26, Jan 1999. 9

Paul Milgram, Shumin Zhai, David Drascic, and Julius Grodski. Applications of augmented reality for human-robot communication. *IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS)*, 1993. URL `http://vered.rose.utoronto.ca/people/daviddir/SMC89/smc89.pdf`. 23, 171

M J Milroy, D J Weir, C Bradley, and G W Vickers. Reverse engineering employing a 3d laser scanner: A case study. *International Journal of Advanced Manufacturing Technology*, 12(2):1–1, Dec 1996. 8

David W Mizell. Virtual reality and augmented reality for aircraft design and manufacturing. *WESCON*, pages 1–5, Sep 1994. doi: http://doi.ieeecomputersociety.org/10.1109/MCG.1994.10012. 8, 171, 176

David W Mizell. Augmented reality applications in aerospace. *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, pages 1–1, Jan 2000. 183

R. Mohl. *Cognitive Space in the Interactive Movie Map: An Investigation of Spatial Learning in Virtual Environments*. PhD thesis, Education and Media Technology, M.I.T., 1981. 69

R Mohr, R Buschmann, L Falkenhagen, Luc Van Gool, and Reinhard Koch. Cumuli, panorama, and vanguard project overview. *Springer*, 1998. URL `http://www.springerlink.com/index/2QT8D5DTM3LQ0287.pdf`. 173

Jose Molineros and Rajeev Sharma. Real-time tracking of multiple objects using fiducials for augmented reality. *Real-Time Imaging*, 7:495–506, Nov 2001. 20

Jose Molineros, Vijaimukund Raghavan, and Rajeev Sharma. Computer vision based augmented reality for guiding and evaluating assembly sequences. *IEEE Virtual Reality Annual International Symposium (VRAIS)*, Dec 1998. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=658496`. 177

Jose Molineros, Reinhold Behringer, and Clement Tam. Vision-based augmented reality for pilot guidance in airport runways and taxiways. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2004. URL `http://doi.ieeecomputersociety.org/10.1109/ISMAR.2004.66`. 24

Jules Moloney. Temporal context and concurrent evaluation - enhancing decision making at the early stages of architectural design with mixed reality technology. *Mixed Reality in Architecture, Design Construction (Eds: X. Wang and M.A. Schnabel)*, pages 135–153, Nov 2008. 174

Hans Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, page 584, August 1977. URL http://www.frc.ri.cmu.edu/~hpm/project.archive/robot.papers/1977/aip.txt. 42

Daniel D Morris and Takeo Kanade. Factorization algorithm for points, line segments and planes with uncertainty models. *IEEE International Conference on Computer Vision (ICCV)*, pages 1–7, Nov 1998. 104, 106, 127

Daniel D Morris, Kenichi Kanatani, and Takeo Kanade. Gauge fixing for accurate 3d estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Jun 2001. 120, 147

Nima Moshtagh. Minimum Volume Enclosing Ellipsoids. Technical report, University of Pennsylvania - School of Engineering & Applied Science, 2005. 76

David W Murray and James J Little. Patchlets: representing stereo vision data with surface elements. *IEEE Workshop on Application of Computer Vision (WACV)*, Dec 2005. URL http://doi.ieeecomputersociety.org/10.1109/ACVMOT.2005.90. 138

Hesam Najafi. Fast 3d object detection and pose estimation for augmented reality systems. *Disseration TUM*, pages 1–167, Feb 2007. 144

Eihachiro Nakamae, Koichi Harada, Takao Ishizaki, and Tomoyuki Nishita. A montage method: the overlaying of the computer generated images onto a background photograph. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Aug 1986. URL http://portal.acm.org/citation.cfm?id=15922.15909. 172, 173, 218

Marilyn Nashman, William Rippey, Tsai Hong Hong, and Martin Herman. An integrated vision touch-probe system for dimensional inspection tasks. *National Institute of Standards and Technology - NISTIR 5678*, pages 1–21, Nov 1995. 8

Nassir Navab. Developing killer apps for industrial augmented reality. *IEEE Computer Graphics and Applications (CGA)*, Dec 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1297006. 11, 187

Nassir Navab, Nick Craft, Sven Bauer, and Ali Bani-Hashemi. Cylicon: software package for 3d reconstruction of industrialpipelines. *IEEE Workshop on Application of Computer Vision (WACV)*, Dec 1998. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=732905. 181

Nassir Navab, Benedicte Bascle, Mirko Appel, and Echeyde Cubillo. Scene augmentation via the fusion of industrial drawings and uncalibrated images with a view to markerless calibration. *International workshop on Augmented Reality (IWAR)*, 1999a. URL http://doi.ieeecomputersociety.org/10.110910.1109/IWAR.1999.803813. 181

Nassir Navab, Echeyde Cubillo, Benedicte Bascle, Jurgen Lockau, Klaus-D Kamsties, and Martin Neuberger. Cylicon: a software platform for the creation and update of virtualfactories. *IEEE International Conference on Emerging Technologies and Factory Automation*, Dec 1999b. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=815391`. 11, 24, 68, 181

Nassir Navab, Sandro Michael Heining, and Joerg Traub. Camera augmented mobile c-arm (camc): Calibration, accuracy study and clinical applications. *IEEE Transaction on Medical Imaging*, 29(7):1412–23, 2009. 171

Ulrich Neumann and Anthony Majoros. Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance. *IEEE Virtual Reality Annual International Symposium (VRAIS)*, Nov 1998. URL `http://citeseer.ist.psu.edu/244560`. 184

Ulrich Neumann, Suya You, Youngkwan Cho, Jongweon Lee, and Jun Park. Augmented reality tracking in natural environments. *IEEE and ACM International Symposium on Mixed Reality*, Dec 1999. URL `http://graphics.usc.edu/~suyay/paper/ismr99.pdf`. 20

Joseph Newman, David Ingram, and Andy Hopper. Augmented reality in a wide area sentient environment. *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, Jan 2001. URL `http://citeseer.ist.psu.edu/588225`. 21

Kevin Nickels and Seth Hutchinson. Estimating uncertainty in ssd-based feature tracking. *Image and Vision Computing*, 20(1), Aug 2002. URL `http://citeseer.ist.psu.edu/296438`. 106

David Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004. 126

David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Apr 2006. 166

David Nister, Oleg Naroditky, and James Bergen. Visual odometry. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:652–659, Mar 2004. 143

Stefan Nölle. Stereo augmentation of simulation results on a projection wall by combining two basic arvika systems. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Sep 2002. URL `http://portal.acm.org/citation.cfm?id=850976.854968`. 23, 175

Stefan Nölle and Gudrun Klinker. Augmented reality as a comparison tool in automotive industry. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2006. URL `http://portal.acm.org/citation.cfm?id=1514251`. 13, 20, 175

Benjamin Ochoa and Serge Belongie. Covariance propagation for guided matching. *SMVP*, 2006. 104

Jan Ohlenburg, Iris Herbst, Irma Lindt, Throsten Fröhlich, and W Broll. The morgan framework: enabling dynamic multi-user ar and vr projects. *ACM symposium on Virtual reality software and technology (VRST)*, Dec 2004. URL `http://portal.acm.org/citation.cfm?id=1077568`. 174

Toshikazu Ohshima, Tsuyoshi Kuroki, Hiroyuki Yamamoto, and Hideyuki Tamura. A mixed reality system with visual and tangible interaction capability: application to evaluating automobile interior design. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2003. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1240722`. 22, 175

John Oliensis and Michael Werman. Structure from motion using points, lines, and intensities. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec 2000. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=854927`. 127

Alex Olwal, Christoffer Lindfors, Jonny Gustafsson, Torsten Kjellberg, and Lars Mattsson. Astor: an autostereoscopic optical see-through augmented reality system. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 24–27, 2005. 178

Umut Orguner and Fredrik Gustafsson. Statistical characteristics of harris corner detector. *IEEE Workshop on Statistical Signal Processing (SSP)*, pages 571–575, 2007. 106

Youngmin Park, Vincent Lepetit, and Woontack Woo. Multiple 3d object tracking for augmented reality. (ISMAR):4, Jul 2008. 104

Wouter Pasman and Charles Woodward. Implementation of an augmented reality system on a pda. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 276–277, Jul 2003. 20

Katharina Pentenrieder, Christian Bade, Fabian Doil, and Peter Meier. Augmented reality-based factory planning-an application tailored to industrial needs. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–9, 2007. 13, 20, 68, 83, 181

Juri Platonov and Marion Langer. Automatic contour model creation out of polygonal cad models for markerless augmented reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–4, Nov 2007. 24

Juri Platonov, Hauke Heibel, Peter Meier, and Bert Grollmann. A mobile markerless ar system for maintenance and repair. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 105–108, 2006. 10, 25, 144, 183

Thomas Porter and Tom Duff. Compositing digital images. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Jan 1984. URL `http://portal.acm.org/citation.cfm?id=800031.808606`. 172

Muriel Pressigout and Eric Marchand. A model free hybrid algorithm for real time tracking. *IEEE International Conference on Image Processing (ICIP)*, 3, 2005. 127

Muriel Pressigout and Eric Marchand. Real-time hybrid tracking using edge and texture information. *International Journal of Robotics Research*, 26(7):689–713, Jul 2007. doi: 10.1177/0278364907080477. 127, 133

Muriel Pressigout and Eric Marchand. Realtime plannar structure tracking: a contour and texture approach. *Technical Report IRISA*, 1698:25, Jan 2008. 127

John Quarles, Samsun Lampotang, Ira Fischler, Paul Fishwick, and Benjamin Lok. Collocated aar: Augmenting after action review with mixed reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 107–116, 2008. 188

Vijaimukund Raghavan, Jose Molineros, and Rajeev Sharma. Interactive evaluation of assembly sequences using augmented reality. *IEEE Transactions on Robotics and Automation*, 15(3):435–449, Nov 1999. 177

Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. Exploiting uncertainty in random sample consensus. *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, Jun 2009. 104

Mathias Rauterberg, Morten Fjeld, Helmut Krueger, Martin Bichsel, Uwe Leonhardt, and Markus Meier. Build-it: a planning tool for construction and design. *Conference on Human Factors in Computing Systems (CHI)*, Dec 1998. URL `http://portal.acm.org/citation.cfm?id=286498.286657`. 24, 180

Holger Regenbrecht and R Specht. A mobile passive augmented reality device-mpard. *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, Dec 2000. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=880926`. 175, 183

Holger Regenbrecht, M Wagner, and Gregory Baratoff. Magicmeeting: A collaborative tangible augmented reality system. *Virtual Reality*, Dec 2002. URL `http://www.springerlink.com/index/8D2R4K179735TAC3.pdf`. 20, 175

Holger Regenbrecht, Gregory Baratoff, and Wilhelm Wilke. Augmented reality projects in the automotive and aerospace industries. *IEEE Computer Graphics and Applications (CGA)*, 25(6):48–56, 2005. 20, 175, 176, 183, 185, 187

Dirk Reiners, Didier Stricker, Gudrun Klinker, and Stefan Müller. Augmented reality for construction tasks: Doorlock assembly. *International workshop on Augmented Reality (IWAR)*, 1998. URL `http://wwwbruegge.in.tum.de/publications/includes/pub/reiners1998iwar/reiners1998iwar.pdf`. 20, 177

Gerhard Reitmayr and Tom Drummond. Going out: Robust model-based tracking for outdoor augmented reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2006. URL `http://portal.acm.org/citation.cfm?id=1514223`. 24

Gerhard Reitmayr, Ethan Eade, and Tom Drummond. Localisation and interaction for augmented maps. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2005. URL http://portal.acm.org/citation.cfm?id=1104996.1105191. 24, 182

Gerhard Reitmayr, Ethan Eade, and Tom Drummond. Semi-automatic annotations in unknown environments. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2007. URL http://portal.acm.org/citation.cfm?id=1514339.1514390. 25, 185

Jun Rekimoto and Yuji Ayatsuka. Cybercode: Designing augmented reality environments with visual tags. *Conference on Designing Augmented Reality Environments (DARE)*, Feb 2000. URL http://citeseer.ist.psu.edu/330380. 17

Jun Rekimoto and Katashi Nagao. The world through the computer: Computer augmented interaction with real world environments. *UIST*, Dec 1995. URL http://portal.acm.org/citation.cfm?id=215585.215639. 185

Patrick Riess and Didier Stricker. Ar on-demand: a practicable solution for augmented reality on low-end handheld devices. *AR/VR Workshop of the Germany Computer Science Society*, 2006. URL http://www.ist-ultra.org/publications/AR_on_demand.pdf. 20, 24

Eric Rose, David E Breen, Klaus H Ahlers, Chris Crampton, Mihran Tuceryan, Ross T Whitaker, and Douglas S Greer. Annotating real-world objects using augmented reality. *Conference on Computer Graphics International (CGI)*, Dec 1995. URL http://www.cs.iupui.edu/~tuceryan/research/AR/ECRC-94-41.pdf. 184

Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. *IEEE International Conference on Computer Vision (ICCV)*, 2, 2005. 42

Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: a machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–15, Jun 2010. 42

Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. *International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 1–8, Mar 2001. 25

Markus Sareika and Dieter Schmalstieg. Urban sketcher: Mixing reality on site for consent in urban planning and architecture. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–4, Sep 2007. 174

Kosuke Sato, Yoshihiro Ban, and Kunihiro Chihara. Mr aided engineering: Inspection support systems integrating virtual instruments and process control. *Mixed Reality, Ed. Yuichi Ohta Hideyuki Tamura*, Dec 1999. URL http://sciencelinks.jp/j-east/article/200003/000020000399A0861532.php. 184

Frank Sauer, Fabian Wenzel, Sebastian Vogt, Yiyang Tao, Yakup Genc, and Ali Bani-Hashemi. Augmented workspace: Designing an ar testbed. *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, Dec 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=880922. 22

Gerhard Schall, Erick Mendez, and Dieter Schmalstieg. Virtual redlining for civil engineering in real environments. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4637332. 4, 5, 23, 182, 190

Daniel Scharstein, Richard Szeliski, and Ramin Zabith. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002. 146

Dieter Schmalstieg, Gerhard Schall, Daniel Wagner, István Barakonyi, Gerhard Reitmayr, Joseph Newman, and Florian Ledermann. Managing complex augmented reality models. *IEEE Computer Graphics and Applications (CGA)*, 27(4):48–57, Jun 2007. 190

Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, pages 1–46, Jun 2000. 38, 105

Holger Schnädelbach, Boriana Koleva, Martin Flintham, Mike Fraser, Shahram Izadi, Paul Chandler, Malcolm Foster, Steve Benford, Chris Greenhalgh, and Tom Rodden. The augurscope: A mixed reality interface for outdoors. *Conference on Human Factors in Computing Systems (CHI)*, pages 9–16, Jan 2002. 23

Ralph Schoenfelder and Dieter Schmalstieg. Augmented reality for industrial building acceptance. *IEEE Virtual Reality Conference (IEEE VR)*, pages 83–90, 2008. 13, 22, 182

Ralph Schoenfelder, Joachim Baur, and Frank Spenling. The planar: A mobile vr tool with pragmatic pose estimation for generation and manipulation of 3d data in industrial environments. *International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 462–469, Nov 2005. 182

Pierre Schroeder. Design of an augmented reality software for discrepancy inspection. *System-Entwicklungs-Projekt TUM*, pages 1–85, Nov 2009. 196

Hagen Schumann, Silviu Burtescu, and Frank Siering. Applying augmented reality techniques in the field of interactive collaborative design. *Workshop of ECCV on 3D structure from multiple images of large-scale evironments (SMILE)*, Dec 1998. URL http://www.springerlink.com/index/HFXD5LCAG4URBUGN.pdf. 20, 173

Bernd Schwald and Blandine De Laval. An augmented reality system for training and assistance to maintenance in the industrial context. *Journal of WSCG*, Jan 2003. URL http://citeseer.ist.psu.edu/586891. 22, 179

Florian Schweiger, Bernhard Zeisl, Pierre Georgel, Georg Schroth, Eckehard Steinbach, and Nasir Navab. Maximum detector response markers for sift and surf. *VMV*, pages 1–10, Sep 2009. 114

Bjoern Schwerdtfeger and Gudrun Klinker. Supporting order picking with augmented reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2008. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4637331`. 22, 178

Bjoern Schwerdtfeger, Daniel Pustka, Andreas Hofhauser, and Gudrun Klinker. Using laser projectors for augmented reality. *ACM symposium on Virtual reality software and technology (VRST)*, pages 1–4, Aug 2008. 179

Hélène Seingier. Nucléaire : prolonger la vie d'un rÃ©acteur en changeant sa cuve ?, 2009. URL `http://www.rue89.com/print/125006`. 4

Rajeev Sharma and Jose Molineros. Computer vision for guiding manual assembly. *International Symposium on Assembly and Task Planning*, Dec 2001. URL `http://www.csa.com/partners/viewrecord.php?requester=gs&collection=TRD&recid=A9737815AH`. 177

Jianbo Shi and Carlo Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Feb 1994. 106, 128, 144

Hiroshi Shimoda, Hirotake Ishii, Masayuki Maeshima, Toshinori Nakai, Zhiqiang BIAN, and Hidekazu Yoshikawa. Development of a tracking method for augmented reality applied to nuclear plant maintenance work:(1) barcode marker. *Workshop of Halden Project VR*, 2005. 4, 20, 183

Hiroshi Shimoda, Toshinori Nakai, Hirotake Ishii, Masanori Izumi, Zhiqiang BIAN, Yoshinori Kanehira, and Yoshitsugu MORISHITA. A feasibility study of decommissioning support method by augmented reality. *International Symposium on Symbiotic Nuclear Power Systems for 21st Century*, 2007. 187

Tobias Sielhorst, Marco Feuerstein, Joerg Traub, Oliver Kutter, and Nassir Navab. Campar: A software framework guaranteeing quality for medical augmented reality. *Computer Assisted Radiology and Surgery (CARS)*, page 5, Oct 2006. 22

Pekka Siltanen, Tommi Karhela, Charles Woodward, and Paula Savioja. Augmented reality for plant lifecycle management. *International Conference on Concurrent Enterprising (ICE)*, pages 4–6, 2007. 20, 180, 187, 190

Dave Sims. New realities in aircraft design and manufacture. *IEEE Computer Graphics and Applications (CGA)*, Dec 1994. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=267487`. 3, 176

Ajit Singh and Petter Allen. Image-flow computation: An estimation-theoretic framework and a unified perspective. *CVGIP: Image Understanding*, 56(2):152–177, Nov 1992. 106

Wladyslaw Skarbek and Michal Tomaszewski. Epipolar angular factorisation of essential matrix for camera pose calibration. *Lecture Notes in Computer Science - Computer Vision/Computer Graphics CollaborationTechniques*, 5496:401–412, Mar 2009. 138

Jon Skeet. Delegates and Events. URL http://www.yoda.arachsys.com/csharp/events.html. 194

Johan Skoglund and Michael Felsberg. Covariance estimation for sad block matching. *SCIA*, pages 1–9, Apr 2007. 106

Greg Slabaugh, Ron Schafer, and Mark Livingston. Optimal ray intersection for computing 3d points from n-view correspondences. *Technical Report*, Oct 2001. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.6117&rep=rep1&type=pdf. 49

S.M Smith and J.M.Brady. Susan - a new approach to low level image processing. *Technical Report TR95SMS1c*, pages 1–59, Jun 1995. 42

Noah Snavely, Steven Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 835–846, 2006. 68

Noah Snavely, Rahul Garg, Steven Seitz, and Richard Szeliski. Finding paths through the world's photos. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Aug 2008. URL http://portal.acm.org/citation.cfm?id=1399504.1360614. 68, 104

Cristopher Stapleton, Charles Hughes, and J. Michael Moshell. Mixed fantasy: Exhibition of entertainment research for mixed reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2003. URL http://portal.acm.org/citation.cfm?id=946248.946813. 171

R.M. Steele and C. Jaynes. Feature uncertainty arising from covariant image noise. In *IEEE CConference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2005. 106, 109

Henrik Stewenius, Frederik Schaffalitzky, and David Nister. How hard is 3-view triangulation really? *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, Jan 2005. 50

Didier Stricker. Tracking with reference images: a real-time and markerless tracking solution for out-door augmented reality applications. *International Symposium on Virtual Reality, Archaeology, and Intelligent Cultural Heritage (VAST)*, Nov 2001. URL http://portal.acm.org/citation.cfm?id=584993.585006. 25, 144

Didier Stricker and Thomas Kettenbach. Real-time and markerless vision-based tracking for outdoor augmented reality applications. *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, Dec 2001. URL http://doi.ieeecomputersociety.org/10.1109/ISAR.2001.970536. 144

Didier Stricker and Nassir Navab. Calibration propagation for image augmentation. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 95–102, 1999. 144

Frederic Sur. Robust matching in an uncertain world. *International Conference on Pattern Recognition (ICPR)*, pages 1–4, Apr 2010. 122

Frederic Sur, Nicolas Noury, and Marie-Odile Berger. Computing the uncertainty of the 8 point algorithm for fundamental matrix estimation. *British Machine Vision Conference (BMVC)*, 2008. 104

Ivan E Sutherland. The ultimate display. *International Federation for Information Processing Congress (IFIP)*, Dec 1965. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.5951&rep=rep1&type=pdf`. 8

Hideyuki Tamura, Hiroyuki Yamamoto, and Akihiro Katayama. Mixed reality: Future dreams seen at the border between real and virtual worlds. *IEEE Computer Graphics and Applications (CGA)*, Dec 2001. URL `http://doi.ieeecomputersociety.org/10.1109/38.963462`. 175

Bruce Thomas. Augmented reality visualisation facilitating the architectural process using outdoor augmented reality in architectural designing. *Mixed Reality in Architecture, Design Construction (Eds: X. Wang and M.A. Schnabel)*, 2008. URL `http://www.springerlink.com/index/nx77075679345115.pdf`. 175

Bruce Thomas, Wayne Piekarski, and Bernard Gunther. Using augmented reality to visualise architecture designs in an outdoor environment. *DCNET*, Dec 1999. URL `http://www.cis.unisa.edu.au/~cisbht/Brucepdf`. 23, 175

Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. *Technical Report CMU-CS*, 91(132):1–22, Mar 1991. 106

Bill Triggs. Factorization methods for projective structure and motion. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 845–851, Nov 1996. Well it is a factorization algorithm. 127

Bill Triggs. Detecting keypoints with stable position, orientation and scale under illumination changes. *European Conference on Computer Vision (ECCV)*, pages 1–13, May 2004. 104

Bill Triggs, Philip F McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment - a modern synthesis. *LECTURE NOTES IN COMPUTER SCIENCE*, 1883: 1–75, Aug 1999. 51

Roger Y Tsai. A versatile camera calibration techniaue for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, Sep 1987. 35

Petra Tschirner, Bernd Hillers, and Axel Gräser. A concept for the application of augmented reality in manual gas metal arc welding. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2002. URL `http://doi.ieeecomputersociety.org/10.1109/ISMAR.2002.1115098`. 178, 179

Mihran Tuceryan, Douglas S Greer, Ross T Whitaker, David E Breen, Chris Crampton, Eric Rose, and Klaus H Ahlers. Calibration requirements and procedures for a monitor-based augmented reality system. *IEEE Transactions on Visualization and Computer Graphics*, Feb 1995. URL `http://citeseer.ist.psu.edu/352033`. 24

John Underkoffler and Hiroshi Ishii. Urp: A luminous-tangible workbench for urban planning and design. *Conference on Human Factors in Computing Systems (CHI)*, Dec 1999. URL `http://portal.acm.org/citation.cfm?id=303114`. 174

Sakae Uno and Hideo Matsuka. A general purpose graphic system for computer aided design. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Dec 1979. URL `http://portal.acm.org/citation.cfm?id=965103.807421`. 172

Matthew Uyttendaele, Antonio Criminisi, Sing Bing Kang, Simon Winder, Richard Hartley, and Richard Szeliski. High-quality image-based interactive exploration of real-world environments. *IEEE Computer Graphics and Applications (CGA)*, 24(3):52–63, 2004. 69

Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Fusing online and offline information for stable 3–d tracking in real-time. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Mar 2003. 144

Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1385–1391, 2004a. 104, 144

Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Combining edge and texture information for real-time accurate 3d camera tracking. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2004b. URL `http://doi.ieeecomputersociety.org/10.1109/ISMAR.2004.24`. 127

Eduardo Veas and Ernst Kruijff. Vesp'r: design and evaluation of a handheld ar device. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, pages 1–10, Jun 2008. 23

Daniel Wagner and Dieter Schmalstieg. Artoolkitplus for pose tracking on mobile devices. *Computer Vision Winter Workshop (CVWW)*, pages 1–8, Jan 2007. 17

Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *SI3D*, Feb 1990. URL `http://portal.acm.org/citation.cfm?id=91385.91442`. 67

Sabine Webel, Mario Becker, Didier Stricker, and Harald Wuest. Identifying differences between cad and physical mock-ups using ar. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2007. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4538867`. 13, 23, 175

Anthony Webster, Steven Feiner, Blair MacIntyre, William Massie, and Theodore Kruegger. Augmented reality in architectural construction, inspection and renovation. *Workshop on Computing in Civil Engineering*, Dec 1996. URL `http://eprints.kfupm.edu.sa/27036/`. 21, 69, 176

Jens Weidenhausen, Christian Knoepfke, and Didier Stricker. Lessons learned on the way to industrial augmented reality applications, a retrospective on arvika. *Computers & Graphics*, (27):887–891, Oct 2003. doi: 10.1016/j.cag.2003.09.001. 10, 21, 187

Changchang Wu, Brian Clipp, Xiaowei Li, Jan-Michael Frahm, and Marc Pollefeys. 3d model matching with viewpoint-invariant patches (vip). *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec 2008. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4587501`. 154

F. C. Wu, Z. Y. Hu, and F. Q. Duan. 8-point algorithm revisited: Factorized 8-point algorithm. *IEEE International Conference on Computer Vision (ICCV)*, 1:488–494, 2005. doi: http://doi.ieeecomputersociety.org/10.1109/ICCV.2005.3. 126

Harald Wuest and Didier Stricker. Tracking of industrial objects by using cad models. *Journal of Virtual Reality and Broadcasting (JRVB)*, 4(1):1–9, Jul 2007. 10

Jürgen Zauner, Michael Haller, Alexander Brandl, and Werner Hartman. Authoring of a mixed reality assembly instructor for hierarchical structures. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2003. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1240707`. 20, 177

Bernhard Zeisl. Estimation and exploitation of localization uncertainty for scale invariant feature points. *Diploma Arbeit TU Muenchen*, pages 1–61, Jun 2009. 114

Shumin Zhai and Paul Milgram. A telerobotic virtual control system. *Proceedings of SPIE - Cooperative Intelligent Robotics in Space II*, 1612, Dec 1991. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.5742&rep=rep1&type=pdf`. 23, 171

Xiang Zhang and Nassir Navab. Tracking and pose estimation for computer assisted localization in industrial environments. *IEEE Workshop on Application of Computer Vision (WACV)*, pages 1–8, Aug 2000. 17

Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–198, Dec 1998. 49, 125

Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *IEEE International Conference on Computer Vision (ICCV)*, pages 666–673, Jul 1999. 35

Xiaowei Zhong, Peiran Liu, Nicolas D Georganas, and Pierre Boulanger. Designing a vision-based collaborative augmented reality application for industrial training. *it – Information Technology*, (45):1–13, Feb 2003. 20, 185

Siavash Zokai, Yakup Genc, Nassir Navab, and Julien Esteve. Multiview paraperspective projection model for diminished reality. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, Dec 2003. URL `http://portal.acm.org/citation.cfm?id=946248.946801`. 186, 187, 219