2D and 3D Object Measurement for Control and Quality Assurance in the Industry

Claus Gramkow
Odense Steel Shipyard Ltd.
Lindø, Denmark
Department of Mathematical Modelling
Technical University of Denmark

Lyngby 1999
IMM-PHD-1999-71

IMM

ISSN 0909-3192

Printed by IMM, Technical University of Denmark

Preface

This dissertation has been prepared at Odense Steel Shipyard Ltd. and the Institute of Mathematical Modelling, Technical University of Denmark, in partial fulfilment of the requirements for the degree of industrial Ph.D. in engineering. The project was carried out in the period January 1997 through December 1999.

The project is the result of a cooperation with Erhvervsforskerudvalget, Akademiet for de Tekniske Videnskaber, ATV, and has been funded by Erhvervsfremme Styrelsen. The project number is EF660. An analysis of the commercial aspects of the project and the innovative process has been published in the report *"Teknologiledelse og Teknologiskabelse - ATV Erhvervsrapport"*.

The title of the project is

Object Measurement

The dissertation describes the problems associated with implementation of vision technology in large-scale industrial applications. The discussion embraces the subjects: camera hardware, camera models, projective geometry, calibration, feature detection, and applications. The theory is supported by examples from vision installations at Odense Steel Shipyard. The dissertation is based on internal reports and published papers prepared during the project.

Odense, December 1999

Claus Gramkow

Acknowledgements

First, I would like to thank my close colleagues at Odense Steel Shipyard. Especially Ole Knudsen Neckelmann, with whom I have had many interesting discussions on the theoretical and the practical problems of industrial vision. The anticipation of cooperation with him partly brought me to Odense and he has been very supportive during the whole project. Thanks also to Peter Hauge, who is the third member of the vision group. In addition to many other important modules, he has programmed most of the interfaces to the cameras and frame grabbers. He was very helpful in the design of the GUI for the calibration program. The persistent work by Ole and Peter was indispensable for achievement of the results in this presentation. They have been the driving force in installing hardware in the production and have managed the information interface to other groups at the yard. Also Ole Spielberg, who is no longer at the yard, is acknowledged for his cooperation.

I would also like to thank my supervisors Bjarne K. Ersbøll, Knut Conradsen, and Carl-Erik Skjølstrup. Bjarne and I have had some interesting theoretical discussions and he has been helpful with the paper work associated with the project. His proofreading of this presentation was a great help. I also enjoyed his visit to France, when I stayed there for half a year. Knut is the very reason that I began this industrial Ph.D. project. He established the contact to Ole Knudsen, and was very helpful in defining the project. Carl-Erik Skjølstrup, who unfortunately is no longer at the yard, has been my industrial supervisor. He is thanked for his dedication and support when projects ran into problems.

I would also like to express my gratitude to Odense Steel Shipyard and to ATV. They made the project possible. Henrik I. Christensen and Hans Henrik Thodberg are thanked for reviewing the project.

Naturally, I would also like to thank my colleagues at the Section for Image

Analysis, where I spent the first fifteen months of the project. I enjoyed the discussions on many different image analysis problems. Special acknowledgements to Jørgen Folm-Hansen, who introduced me to the calibration model used in this work. We have had many fruitful discussions on calibration, feature detection, and interpolation. His calibration software was used in the beginning of the project.

Half-way through the project, I visited INRIA in Grenoble, France, for six months. I would like to thank Radu Horaud for receiving me and providing me with a large amount of relevant literature on projective geometry. Also thanks to Roger Mohr and the entire MOVI group for making the stay so pleasant. In particular I would like to thank Stéphane Christy, who let me use his reconstruction software. It enabled me to get acquainted with the theory and arrive at some preliminary reconstruction results with an OSS mock-up. It would also like to acknowledge the funding that I received from *T3net*.

I would like to acknowledge the inspiring network with which I was provided through the EC funded projects ESPRIT-IV LTR 26247 VIGOR (Visually Guided Robots using Uncalibrated Cameras) and ESPRIT 28867 ROBVISION (ROBust VIsion for Sensing in Industrial Operations and Needs).

Finally, I would like to thank my family. My father, Bjarne, has been very helpful in checking my English. Many thanks goes to my girlfriend, Mette, for her support and patience during the entire project.

This dissertation is typeset in LaTeX. The figures were prepared with **gnuplot**.

Summary

The subject of this dissertation is object measurement in the industry by use of computer vision.

In the first part of the dissertation, the project is defined in an industrial frame. The reader is introduced to Odense Steel Shipyard and its current level of automation. The presentation gives an impression of the potential of vision technology in shipbuilding.

The next chapter describes different important properties of industrial vision cameras. The presentation is based on practical experience obtained during the Ph.D. project.

The geometry that defines the link between the observed world and the projected image is the subject of the two next chapters. The first chapter gives a short introduction to projective algebra, which is extremely useful for modelling the image projection and the relation between more images of the same object viewed from different positions. It provides a basis for understanding many of the results later in the dissertation. In the second chapter a variety of different camera models are described. The relation between different models is explained and a guide is given to the interpretation of the model parameters.

The following chapter deals with the problem of camera calibration. Different issues related to residual analysis are discussed and a calibration example is shown. The presentation is based on a software program that has been developed during the project. It is shown that the used cameras can be calibrated down to $1/20\,pixel$.

An accurate description of the geometry is only relevant if features can be detected accurately in the images. This is the subject of the next chapter, where reference mark detection and straight edge detection are treated in two separate sections. The detection of reference marks is based on a parametric

model, and it is shown that marks in synthetic images can be detected with an accuracy of 1/100 *pixel*. Two new methods for straight edge detection are presented. They aim at using all the image information in a global optimization. The paper *"From Hough Transform to Radon Transform using Interpolation Kernels"*, which is included in a later chapter, is introduced. The new methods are compared with more conventional approaches on a number of synthetic images. It is shown that local edge detection by Gaussian convolution with a subsequent regression through the edge pixels is superior to the other methods in accuracy as well as in speed.

The next chapter introduces the paper *"On Averaging Rotations"* that was presented at the 11'th Scandinavian Conference on Image Analysis, SCIA'99, in Greenland. The paper makes a theoretical comparison of two normally used linear averages to a recently proposed method that considers the structure of the non-linear manifold of rotations. The conclusion is that the behaviour of the three different methods is very similar.

The following four chapters describe the practical results of the project. The first chapter gives a short introduction to a report entitled *"Reconstruction and Matching of OSS Mock-Up"*. This report describes a preliminary attempt to apply a method of Euclidean reconstruction from a sequence of images on a ship block. The other three chapters describe vision installations that have been made at Odense Steel Shipyard. The first installation uses vision for check-in and quality control on a plasma cutting station. The second installation was designed for check-in on the quay. Finally, the third installation does check-in and quality control on a laser station. It is shown that measurements can be made with an accuracy of 1 *mm* at a distance of 10 *m* under favourable conditions. When the camera is only 1–2 *m* from the object the accuracy is better than 1/3 *mm*. It is also shown that it is very difficult to obtain ideal conditions and that the vision measurements are very sensitive in the large-scale installations. Only the installation on the laser cutter is currently used in production.

Finally, a chapter gives some concluding remarks about the results of the project.

Resumé

Denne afhandling omhandler brugen af computer vision til objektopmåling i industrien.

Den første del af afhandlingen placerer projektet i en industriel ramme. Læseren bliver introduceret til Odense Staalskibsværft og værftets nuværende automationsniveau. Fremstillingen giver et indtryk af potentialet for visionteknologi i skibsbygning.

Det følgende kapitel beskriver forskellige vigtige egenskaber ved industrielle visionkameraer. Beskrivelsen er baseret på praktiske erfaringer opnået i løbet af projektet.

Geometrien, der definerer båndet mellem den betragtede verden og det projicerede billede, bliver beskrevet i de to følgende kapitler. Det første kapitel giver en kort introduktion til projektiv geometri, som er et utroligt nyttigt værktøj til modellering af den perspektiviske projektion og de geometriske bånd, der eksisterer imellem flere billeder af det samme objekt betragtet fra forskellige synsvinkler. Dette afsnit giver det matematiske grundlag for mange af resultaterne senere i afhandlingen. Det andet kapitel præsenterer en række forskellige kameramodeller. Forbindelsen mellem disse modeller bliver forklaret, og der gennemgås en detaljeret fortolkning af modelparametrene.

Det følgende kapitel omhandler kamerakalibrering. Forskellige forhold omkring residualanalyse bliver diskuteret, og der gennemgås et kalibreringseksempel. Fremstillingen er baseret på et kalibreringsprogram, som er udviklet i forbindelse med projektet. Det bliver vist, at de anvendte kameraer kan blive kalibreret ned til $1/20\,$*pixel*.

En præcis beskrivelse af kamerageometrien er kun anvendelig, hvis objekter kan blive detekteret i billederne med høj nøjagtighed. Dette er netop emnet i det følgende kapitel, hvor referencemærkedetektion og detektion af retlinede kanter bliver beskrevet i to separate afsnit. Referencemærkedetektionen er

baseret på en parametrisk model af mærket i billedet, og det bliver vist at et mærke i et syntetisk billede kan lokaliseres med en nøjagtighed omkring $1/100\,pixel$. To nye metoder til detektion af retlinede kanter bliver beskrevet. Metoderne sigter mod at anvende al tilgængelig billedinformation samtidigt i en global optimering. Artiklen *"From Hough Transform to Radon Transform using Interpolation Kernels"*, som er inkluderet bagest i afhandlingen, bliver introduceret. De to nye metoder bliver sammenlignet med nogle mere almindelige algoritmer på en række syntetiske billeder. Det bliver vist, at lokal kantdetektion med Gaussisk foldning og en efterfølgende regression igennem kantpixels er de andre metoder overlegen med hensyn til såvel nøjagtighed som hastighed.

Det næste kapitel giver en introduktion til artiklen *"On Averaging Rotations"*, som blev præsenteret på den ellevte Scandinavian Conference on Image Analysis, SCIA'99, på Grønland. Artiklen sammenligner to almindeligt anvendte lineære gennemsnit af rotationer med en nyligt publiceret metode, som tager hensyn til strukturen af den ikke-lineære manifold, der beskriver de tilladte rotationer. Konklusionen er, at de tre metoder producerer meget ens resultater.

De følgende fire kapitler beskriver de praktiske resultater som er opnået i projektforløbet. Det første kapitel giver en kort introduktion til en rapport der bærer titlen *"Reconstruction and Matching of OSS Mock-Up"*. Denne rapport beskriver et indledende forsøg på et fremstille en Euklidisk rekonstruktion af en skibsblok ud fra en sekvens af billeder. De andre tre kapitler beskriver visionsystemer som er installeret på Odense Staalskibsværft. Den første installation benytter vision til check-in og kvalitetskontrol på en plasmaskærestation. Den anden installation opmåler stålpladerne, der ankommer til værftet, med det formål at lave automatisk opslag i en database og efterfølgende opmærkning. Den sidste installation bliver benyttet til check-in og kvalitetskontrol på en laserskærestation. Det bliver vist, at opmålinger kan gennemføres med en nøjagtighed på $1\,mm$ over en afstand på $10\,m$ under optimale forhold. Når kameraet befinder sig 1–2 m fra objektet kan der opnås en nøjagtighed på under $1/3\,mm$. Resultaterne viser også at det er meget svært at opnå tilstrækkeligt gode måleforhold og at visionopmålingsnøjagtigheden er meget følsom ved installationer i stor skala. Det er kun installationen på laserskærestationen der på nuværende tidspunkt benyttes i produktionen.

Afhandlingen bliver rundet af med nogle konkluderende bemærkninger om de opnåede resultater.

Contents

List of Papers

I C. Gramkow, *From Hough Transform to Radon Transform using Interpolation Kernels*, Submitted to Journal of Computer Vision and Image Understanding (CVIU), October, 1999.

II C. Gramkow, *Averaging Rotations*, Proceedings of the 11'th Scandinavian Conference on Image Analysis (SCIA'99), Greenland, June, 1999. Invited for a special issue of Journal of Mathematical Imaging and Vision (JMIV), 2000.

III C. Gramkow, *Reconstruction and Matching of OSS Mock-Up*, Internal Report, VIGOR, Esprit-IV reactive LTR project, number 26247, September, 1998.

List of Figures

List of Tables

Chapter 1

Introduction

The European shipyards have lost significant market shares to the yards in the Far East during the past few decades. Especially the yards in Japan and Korea are expanding rapidly and they are very competitive benefiting from low production costs. To survive on the market, the European yards need to improve their productivity.

Odense Steel Shipyard (OSS) has therefore made great efforts to improve the efficiency of the production in the past 15 years. The increased productivity is based on the introduction of Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM). In fact, the ships that are built at OSS are entirely modelled in CAD systems. Besides the immediate advantages in the design phase, the modelling of the ships creates the basis for extensive automation in the production. Thus, OSS has been able to introduce robot welding and cutting in a large scale in the production.

The numeric representation of each ship can also be used as a means to increase the productivity through improved time scheduling and better logistics. Especially if an updated CAD model is maintained based on a continuous monitoring of the ship parts after each process in the production. The information that is acquired from the actual process can be fed forward to allow a compensation on adjacent parts or simply a re-scheduling of tasks, or it can used in a feedback to reveal tools that are damaged or need calibration. The CAD model that is augmented with information from the physical product is denoted the Product State Model (PSM).

The objective of the current project is to implement vision technology in shipbuilding to provide information for the PSM and to take advantage of the non-invasive properties of vision cameras as sensors for the manufacturing tools. The challenge in the applications at the shipyard compared to typical vision applications in the industry is the very large scale, which calls for special demands on calibration, robustness, and illumination. For instance, the vision measurements at the yard must be based on full three-dimensional models of the geometry of the installation. Compared to the use of vision technology seen at other large-scale industries, the present work differs in the requirement that markers or reflectors cannot be placed on the object to be measured.

1.1 Outline of the Dissertation

The dissertation is organized in such a way that the first chapters deal with background and theory while the last chapters present various practical projects.

Chapter 2 defines the project in relation to the technological developments and strategies at the shipyard. The presentation will give the reader an impression of the potential of vision technology in the shipbuilding production line.

Different properties of camera hardware are described in chapter 3 with special emphasis on the characteristics that determine the accuracy of a vision system. Problems that are often encountered are quantified with examples based on the cameras used in the project. The different camera configurations that are used in the applications later in the dissertation are presented.

The following three chapters are concerned with the geometry related to image acquisition and the placement of the cameras in a vision installation. Chapter 4 gives a general introduction to the subject of projective geometry and defines the homogeneous representations for points, lines, and planes in projective space. Points at infinity and maps between projective spaces are also defined. Furthermore, the chapter introduces some helpful tools to perform calculations in homogeneous coordinates. Chapter 5 describes different linear and non-linear camera models. The linear camera models in two and three dimensions are described with sections on estimation, interpretation, and factorization. The complete camera model used in the project is obtained by adding a model of the non-linear lens based on radial and decentering distortion. It is shown how the inverse of the distortion can be calculated. Finally, chapter 6 deals with the problem of camera calibration. The quality of the calibration is evaluated by different approaches and it is argued that the calibration residual is in itself an insufficient quality measure. The example in the chapter is based on a calibration software that was developed in the course of the project.

Calibration as well as measurement is based on features detected in the images. Feature detection is the subject of chapter 7. The first section describes reference mark detection, which is based on fitting a parametric model. It is shown that the dispersion of the parameters can be obtained from the estimation, and that it gives a satisfactory approximation to the empirical dispersion. The effect of trend in the illumination and the size of the reference marks are described and quantified. The second section describes straight edge detection. Two new methods are presented, that are based on a simultaneous use of all available information. The first method is based on smoothing the Hough space, that is obtained from a grey level Hough transformation of the image. The theory is described in a paper included in the back of the dissertation. The second method is based on a normalized Gaussian convolution of the projected image. Those two methods are compared with conventional edge detection techniques. It is concluded that Gaussian convolution with subsequent line fitting is superior to the other methods in terms of both accuracy and speed.

The two following chapters describe work that has been done in the project, but does not relate directly to the installations at the yard. Chapter 8 gives a short introduction to a paper on averaging rotations and chapter 9 shows the preliminary results that was obtained on the problem of Euclidean reconstruction of a mock-up from the shipyard.

Chapters 10, 11, and 12 present the installations that have been implemented on the shipyard in the course of the project in terms of physical dimensions, obtained results, and encountered problems. It is shown that steel plates can be measured with an accuracy of $1\,mm$ at a distance of $10\,m$ under favourable conditions. At a range of 1–$2\,m$ the accuracy is as high as 0.1–$0.3\,mm$.

Finally, some concluding remarks to the work in the dissertation are added in chapter 13.

The dissertation contains three papers and two appendices included in the back. The papers are included as chapters 14-16. Appendix A describes the solutions to two point matching problems in the plane, that were used in the presentation of the results in the applications above. Different issues on the analysis of variance are treated in appendix B.

1.2 Notation

The notation used in the dissertation will to some degree be dependent on the context and will be introduced in the appropriate sections. It is however based on a few simple guidelines.

Matrices and vectors are denoted by bold face letters with matrices in capitals, e.g. $\mathbf{A}$, and vectors generally in lower case, e.g. $\mathbf{b}$. Scalars and the scalar entities of vectors and matrices are denoted by normal case italic letters, e.g. a_{ij} and b_i.

Image coordinates are denoted $\mathbf{u} = (u, v)$ and the image intensity is called $i(u, v)$. Coordinates in the three-dimensional camera frame are denoted $\mathbf{x} = (x, y, z)$ and coordinates in the object frame are denoted by capital letters, $\mathbf{X} = (X, Y, Z)$, to make a distinction from the camera coordinates.

$\mathbf{P}$ is used to represent the linear calibration matrix. When it is a 3×4 matrix, the first 3×3 matrix is denoted $\mathbf{M}$. The upper triangular matrix of intrinsic camera parameters is called $\mathbf{K}$. The rotation and the translation of the camera are denoted $(\mathbf{R}, \mathbf{x}_0)$.

Homogeneous coordinates are represented by capital letters with no respect to their coordinate system. That is, $\mathbf{L}$ is a line in the plane and $\mathbf{P}$ is a plane in three-dimensional space. It should be clear from the context when $\mathbf{P}$ is a plane and when it is the calibration matrix.

Chapter 2

An Industrial Project

This dissertation is the outcome of an industrial Ph.D. project, i.e. the work has been co-funded and hosted by an industrial company. As such the present work has been carried out within an employment at Odense Steel Shipyard Ltd. The close industrial contact gives some immediate advantages in terms of defining the project, production environment accessibility, funding of experiments, and assessment of cost saves. The disadvantage, compared to a conventional purely academic Ph.D. project, is the resources required to implement installations that reach a state where they can be handed over to the production department. At the same time, however, this task gives invaluable insight and experience. The problems that are faced include interfacing to other software components at the Yard, getting access to production facilities, taking into account all sorts of exceptions in the production environment, and making the product fail-safe.

The dependency on the running production and on other people at the yard has sometimes prevented the desired data collection, or limited the extent of testing of the developed installations. The statistics reported in this dissertation should, however, give a clear picture of the developed methods and software.

This chapter will first give an introduction to OSS. Subsequently, the industrial problems that have been addressed during the project will be introduced to give some background for the following theoretical chapters. Some potential areas for computer vision at the shipyard, which have not been addressed, will also be listed. The problems will be viewed in relation to existing solutions on the market. Finally, a section will cover the very interesting EC funded projects that I have been involved in during the project.

2.1 Odense Steel Shipyard

OSS is part of the A. P. Moller group, which was founded in 1904. The original yard was founded in Odense in 1917/18. In 1959 the new yard was built in Lindø and this yard gradually replaced the yard in Odense.

Since the beginning OSS has delivered more than 300 ships to customers all over the world. Recently, OSS has delivered a series of the world's largest and most technologically advanced 6.600 TEU container ships to A.P. Moller, see figure 2.1.

Figure 2.1: Regina Maersk. 6.000 TEU container ship delivered to A.P. Moller.

The shipbuilding industry is characterized by very tough competition. There is an aggressive struggle for market shares, and the world prices are heavily influenced by different kinds of subsidies. It has been said that the world market suffers from up to 30% overcapacity. To match the competition from the Far East, OSS has put a lot of effort into developing a highly technological production line. This way OSS has become one of the most modern yards in Europe, and is one of the few surviving yards in Denmark.

The largest technological advancement in the past 10–15 years has been the introduction of robots in the production. A large number of robots are now used for welding profiles and large assemblies resulting in an increased productivity of 3–6 times compared to manual welding. Figure 2.2 shows the biggest robot installation on the shipyard. The robot consists of twelve vertical axes placed two-by-two on six independent gantrys. Each axis is equipped with a five degrees of freedom industrial robot. All positions within a volume of $32\,m \times 22\,m \times 6\,m$ can be reached with a precision of $1\,mm$.

Figure 2.2: Large industrial robot at Odense Steel Shipyard.

There are different additional ways to improve the productivity on the yard. OSS has focused on the following key areas

- Improved data management (cell control, monitoring, adaptive production etc.)
- Improved machine control (open modular controllers)
- Improved equipment (multiple axes, flexible and modular robots)
- Quality assurance
- Increased production flow

In order to improve in these areas, OSS has introduced CAD systems in all phases of the production. The fact that shipbuilding is a one-of-a-kind production, makes the integrated design and manufacturing even more important. Optimal design can cause tremendous savings through improved logistics, off-line programming etc. Figure 2.3 shows the information flow at OSS. The ship is designed in CAD in the upper left-hand corner. The design is optimized with regard to the welding robots. From the design the robot movements are planned and programmed using simulation tools. On the shop floor the information is available for control and monitoring. It is the ambition to attach updated information to the CAD model after each step in the manufacturing, such as the 'true' dimensions of the part or process parameters. Owing to heat injection during welding or release of residual stresses during cutting, most manufactured parts are somewhat different from their ideal CAD model. Logging these deviations makes it possible to compensate or re-schedule in subsequent processes. Very large savings are expected from this closed information loop. The extended CAD model is denoted the Product State Model (PSM).

2.2 Vision Applications

The short introduction to OSS should give a better understanding of the perspectives of vision technology in shipbuilding. In an introductory manner, this section will describe some of the installations that have been worked upon during the Ph.D. project. Other potential areas for the use of vision technology will also be mentioned.

2.2.1 Check-In and Quality Control on Plasma Cutter

The majority of the steel plates that are cut at OSS are cut with plasma cutters. The plates measure up to 4 meters by 18 meters. They are positioned in the cutting berth where they are marked and cut. Normally, two of the edges are laid in a fixed position. Great savings are anticipated if the dimension of the plates could be verified before cutting to ensure that all cuts are placed within the plate. Similarly, a thorough measurement after cutting could reveal deviations from the expected shape caused by the release of residual stresses in the plate or by a damaged tool. In either case the proper precautions

Figure 2.3: The information flow in integrated CAD-based ship building. OSS graphic.

should save resources. The yard contains 18 such cutting stations giving the installation a very big economic potential. This project is described in more detail in chapter 10.

2.2.2 Check-In of Steel Plates on the Quay

All steel plates that arrive at the yard are ordered for a specific position in the final ship. The yard is therefore looking for a system to automatically check the plates into the database on arrival, and to perform a measurement to verify that each plate meet the specifications. The information will be used to mark the plates and can eventually be the basis for a totally automatic store control. This installation is conceptually very similar to the installation above. The project is described in chapter 11.

2.2.3 Check-In and Quality Control on Laser Cutter

The yard has recently installed a laser cutter mounted on an extremely accurate gantry system ($\pm 0.1mm$). Laser cutting is expected to be far more precise than plasma cutting, partly because laser cutting and welding is associated with much less heat injection. The objectives of this vision installation are similar to those of the plasma cutter installation. Plates are not placed in fixed positions, however, and the check-in is therefore a bit more advanced. The two projects mentioned in section 2.2.1 and 2.2.2 above ran into some difficulties that primarily were caused by the very large scale. In those two installations the cameras were mounted in the ceiling and on a very tall beam, respectively. To avoid similar problems on the laser cutter installation the measurements are instead based on one or more cameras mounted on the gantry. This set-up gives a much better image resolution, better lighting control, and in the case of stereo vision easier arrangement of cameras in terms of a reasonable base-line. The laser cutter installation is described in chapter 12.

2.2.4 Potential Areas for Vision Technology

The three projects described above have been worked upon actively in the Ph.D. project. The research, however, is viable in a number of other potential installations at the yard as well. The most obvious use is naturally check-in and quality control of steel plates at other sites at the yard, e.g. by the milling machines. But the research also extends to very different applications. Some of those are described in the following.

The welding performed by the robot installation in figure 2.2 is based on very advanced off-line programming. The planning optimizes the welding speed while taking into account the lay-out of the individual robots and the problem of collision avoidance. Due to this dependency on the robot configuration, the programs are not easily transposed if the assembly is not positioned as in the planning stage. To avoid this problem the assemblies are positioned very accurately in the work area. Currently, manual measurement devices are used for this purpose, giving a very time-consuming overhead. With multiple vision cameras over-looking the work area, it should be possible to perform an automatic position measurement on the assembly. It might be possible, furthermore, to extend this design with quality control measurements after welding.

A similar vision system could be of interest in the smaller work cells, where several smaller assemblies are welded at the same time. In this case the vision system would be able to localize as well as identify, from a small set, the individual assemblies in the work area. Again, the outcome would be a faster initialization of the actual value adding process. The vision group at the yard has already developed some software for the work cell B13, but the project has now been given a lower priority.

Vision technology could also present an alternative to the touch sensors currently used on the robots. Before welding, the robots determine the beginning of the weld seam by performing a series of movements. This is done by touching the three planes that define the joint where the seam begins. Mounting a calibrated camera on the robot, it is possible to localize the joint in the image, and move the robot directly to the beginning of the seam. To be of any interest the vision system should be superior to the current approach, which takes approximately $15secs$. Possibly, such a system could be extended to give feed-back control during welding as well. However, the engineering problem of seeing and welding at the same time should not be underestimated.

Those were only a few of the potential areas for vision at the shipyard. There are of course other possible applications that may be summarized under the headlines: tool calibration, on-line sensing, check-in, and quality assurance.

2.3 Solutions on the Market

The problems mentioned above have, to my knowledge, not been solved by the vision companies on the market. Most companies focus on a niche with a great market potential and little research. Typically, vision systems are developed for assembly lines in moderate scales, with unproblematic lighting conditions, and large homogeneity. However, some solutions have been developed by companies that specialize in photogrammetry. For instance, Imetric SA from Switzerland has been active in the Esprit reactive long term research (LTR) project CUMULI, where accurate measurement at shipyards has been addressed. Similarly, the German company AICON has developed large scale accurate measurement systems for shipyards. Unfortunately, all these solutions are based on markers on the measured object. This gives a considerable advantage in terms of accuracy, but also a serious drawback in terms of manpower and time.

2.4 EC Funded Projects

During the project I have been fortunate to participate in three EC funded vision projects. This has given an outstanding access to European computer vision research and the opportunity to develop a personal network throughout the European countries. Furthermore, these contacts have given feed-back to the present project. The projects are described in the following.

2.4.1 VIGOR

VIGOR is a three year Esprit reactive long term research (LTR) project that began in February 1998. The partners are the University of Cambridge, England, the Hebrew University of Jerusalem, Israel, INRIA Rhône-Alpes, France, IITB, Karlsruhe, Germany, Sinters SA, France, and Odense Steel Shipyard. The objective of the project is to develop uncalibrated visually guided robots, meaning that the robot control is based only on visual input. The key idea is to teach the vision system the configuration of visual features at the target position and then servo the robot until this configuration is achieved. OSS is one of the end users of the project and will be in charge of the demonstrator at the yard.

2.4.2 RobVision

RobVision is a two year Esprit project with INFA, Wien, Austria, Portech, England, Aalborg University, Denmark, DIST, Genova, Italy, and Odense Steel Shipyard. The aim of the project is to integrate CAD model data with visual features to perform robust navigation in industrial environments. The demonstrator will be based on a climbing robot already developed at Portech. This robot will enter a ship block, position itself relative to a target position specified in the CAD model, and return to a safe position. OSS is the end user of the project and is responsible for the demonstrator at the yard.

2.4.3 QualiGlobe

QualiGlobe is a Brite-EuRam project. The partners are Det Norske Veritas, Norway, Fincantieri, Italy, Institut für Schweißtechnische Fertigungsverfahren, RWTH-Aachen, Germany, Migatronic, Denmark, Robotiker, Spain, and Odense Steel Shipyard. The subject of the project is fabrication based on the state of the product quality. The project is closely related to the product state model (PSM) project at the shipyard. Although OSS again is the end user, the vision group at OSS has been responsible for the vision software in the project. The work is described in chapter 12.

Chapter 3

Vision Cameras

Before entering the theoretical aspects of the project, it is worthwhile to include a few remarks about image acquisition. It is not without problems to obtain high quality images in an industrial environment like OSS. Vision installations at the shipyard differ from the typical vision solutions, as to the very large scale there has to be dealt with, which makes it very difficult to control the light. Very often the surroundings introduce varying illumination of the scene, electrical noise, and limitations to the placement of the camera. In addition, many cameras on the market are not suited for photogrammetric measurements. Some of the problems that have been encountered in the project are described in the following sections.

3.1 Optimal Grey Values

During the project, vision systems have been installed indoors as well as outdoors. In the latter case the system was required to be operational both night and day. This variation of light places a great demand on a camera. As a guideline office lighting produces about 750 *lux* whereas sunshine is as much as $80,000$ *lux*. This variation appears to be more modest to the human eye owing to its amazing ability to accommodate. A camera is linear, however, and the accommodation must be achieved through the shutter time. The camera must therefore have an electronically controlled shutter, and the available settings should range over 3–4 factors. There are other means of controlling the intensity distribution of the image. Some cameras are equipped with an auto-iris, that adjust the aperture automatically. The aperture setting, however, tends to affect the internal parameters of the camera, thus making the calibration invalid. Other cameras are equipped with either automatic gain or electronically controlled gain. The drawback of this feature is a lower signal-to-noise ratio. It is therefore strongly recommended to optimize the grey values of the image through the shutter time.

Once the image is acquired it is important to verify that the image does not have any underflow or overflow near the features that are to be measured. Overflow or saturation occurs when too many photons have hit the CCD chip. Typically, this causes a leakage of electrons to the neighboring pixels, which is seen as a smearing of the image. Naturally, the affected pixels cannot be used for measurements. In fact, leakage is sometimes seen before actual saturation occurs, and it is therefore wise to avoid very high pixel values in the image. Underflow or black level cut-off is a similar problem in the dark end of the spectrum. It is related to the quantizing of the video signal. The potential on the CCD chip that is mapped to pixel value zero in the image is controlled by the offset setting on the camera. The offset value should be chosen so that all pixel values are greater than zero to ensure a proper quantizing of the small potentials.

Saturation as well as black level cut-off introduces non-linearities in the image intensities. If a signal with these artifacts is used for position measurement, the results will be biased. Figure 3.1 shows an edge in an image that has been acquired with three different camera settings. The left column shows the same region of the image and the right column shows an intensity profile across the edge, sampled from one row in the image. The rows show a good quality image, an image with saturation, and an image with black level cut-off, respectively. To illustrate the effect of the artifacts, the edge location has been determined by the maximum of the Gaussian derivative, cf. chapter 7. Note, how the detection is biased towards the darker side in the saturation example and towards the lighter side in the case of black level cut-off. The absolute position of the edge should not be compared between images, as the camera may have moved. In general the edge detector tends to locate the edge midway on the transition from light to dark. If part of the signal has been truncated, the detection will be biased in the opposite direction.

To avoid the problems discussed in this section, an iterative scheme has been used to acquire images on the installations at the shipyard.

Another issue concerning the light intensities of the image is the choice of

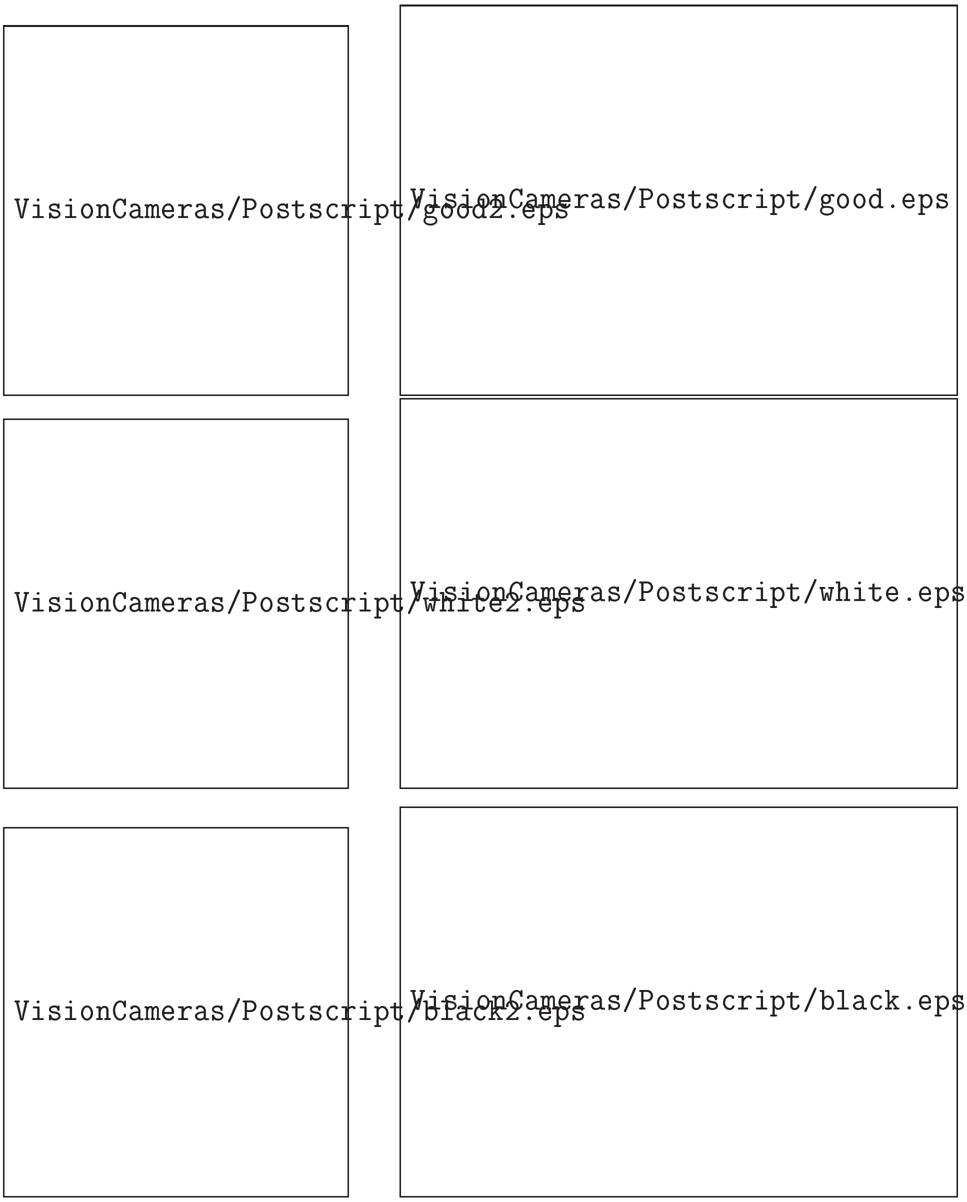
VisionCameras/Postscript/good2.eps
VisionCameras/Postscript/good.eps
VisionCameras/Postscript/white2.eps
VisionCameras/Postscript/white.eps
VisionCameras/Postscript/black2.eps
VisionCameras/Postscript/black.eps

Figure 3.1: The left column shows the same edge acquired with three different camera settings and the right column shows the intensity profile across the edge, sampled from one row in the image.

lighting. In a particular installation neon tubes where used. The shutter of the camera is so fast, however, that the crests and troughs of the light could be seen from one image to the other. To avoid this problem, high frequency neon light was used instead.

3.2 The Video Signal

The quality of the video signal is another cause for concern. Two circumstances make this a key issue in an industrial environment. First, the camera typically will be placed far away from the frame-grabber and the computer. Therefore cables often have to be as long as 25–30 meters. This distance is sufficient to reduce the quality of the signal. Second, the environment is characterized by a lot of electrical noise. This problem is aggravated when the video cable typically must be laid along with many other cables.

It is difficult to quantify the effect of those problems in the image. Even though they are not always visible, they can easily affect the accuracy of a vision measurement. If the noise is severe it may corrupt the synchronization of the signal, causing a displacement of the fields or lines of the image. This is known as field or line jitter. If the field pulse is corrupted, the two entire half-fields are misaligned whereas bad line synchronization causes the individual lines to be misaligned. Figure 3.2 shows an example of field jitter. Note, how every second line is shifted 1–2 pixels. The same behaviour could be experienced with a non-steady camera, e.g. a camera held by hand. In the present case the camera was fixed on a tripod. Obviously, the image is not suited for accurate measurements.

Pixel jitter may also occur [48]. This is caused by inaccuracies in the pixel clock of the camera or the frame-grabber. As a result the pixel is digitized late or early in the analog signal resulting in an inaccurate pixel representation.

The quality of the synchronization can be improved by using a HD/VD camera with separate signals for the horizontal and vertical synchronization. Separate signals are less sensitive to noise than the composite signal where the synchronization is mixed with the image signal.

The video signal may be unsuited for measurement for other reasons as well. Many off-the-shelf cameras are not meant for computer vision. Most of these cameras have built-in features to produce an image that satisfies the human

Figure 3.2: Field jitter. The two half-fields are misaligned.

eye. An obvious example is cameras designed for surveillance. Two typical features are gamma correction and edge enhancement. Gamma correction is defined by a non-linear adjustment of the intensities in the image. It is defined by

$$i(u,v) := i_{max}\left(\frac{i(u,v)}{i_{max}}\right)^{\gamma}, \tag{3.1}$$

where $i(u,v)$ is the intensity value of a given pixel and i_{max} is the maximal grey value, i.e. typically 255. For $\gamma < 1$ the dark end of the spectrum is stretched to produce a better contrast while the contrast in the light end of the spectrum is reduced. This pleases the human eye, which is less sensitive to variations in dark grey levels.

Figure 3.3 shows the edge detection error introduced by gamma correction. The left plot displays the edge profile for three typical values of γ: 0.6, 1.0, and 0.6^{-1}. The edge determined by the maximum Gaussian derivative is also shown. The right plot shows how the edge detection varies as a function of γ around the location determined for $\gamma = 1.0$. Note, how the location of the edge changes as much as 0.2 pixel due to gamma correction.

Gamma correction can naturally be removed in software by inducing the inverse mapping of equation 3.1, but it should rather be turned off in hardware

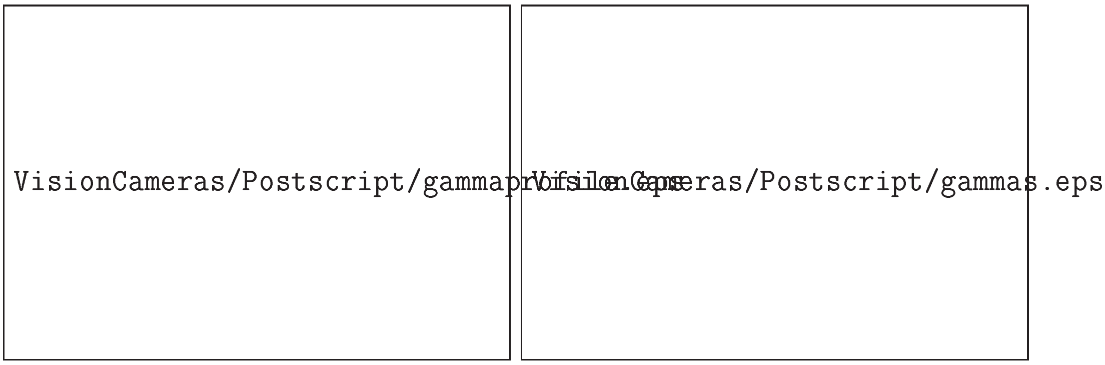
VisionCameras/Postscript/gammaprofile.eps
VisionCameras/Postscript/gammas.eps

Figure 3.3: Left: the edge profile for three different values of γ. Right: the edge detection error as a function of γ relative to $\gamma = 1.0$.

to avoid the calculations and the round-off errors.

The presence of gamma correction in the image may be revealed during calibration. Figure 3.4 shows the calibration residuals in an image with and without gamma correction. This camera had a built-in gamma correction of $\gamma = 0.6$. The corners of the rectangles are used for calibration and their positions are estimated by detection of the edges. The systematic bias in the edge detection results in systematic calibration residuals.

VisionCameras/Postscript/res1.nogam.eps
VisionCameras/Postscript/res1.gamma.eps

Figure 3.4: Left: calibration residuals with gamma correction. Right: calibration residuals with gamma correction removed. Residuals are shown exaggerated.

Edge enhancement is another feature that pleases the human eye. It is implemented as an amplification of the higher frequencies in the video signal. The result is a small overshoot of the intensities near an edge. Actually, the human eye does itself produce such a response to a change in the light intensity. Normally, edge enhancement is added to the analog signal in hardware. Consequently, it only appears in the horizontal direction of the image as the image is transmitted line by line. Figure 3.5 shows an example. The edge enhancement can be seen as shadows in the horizontal direction of the image. The left plot shows the intensity profile across the reference mark in the vertical direction and the right plot shows the intensities in the horizontal direction.

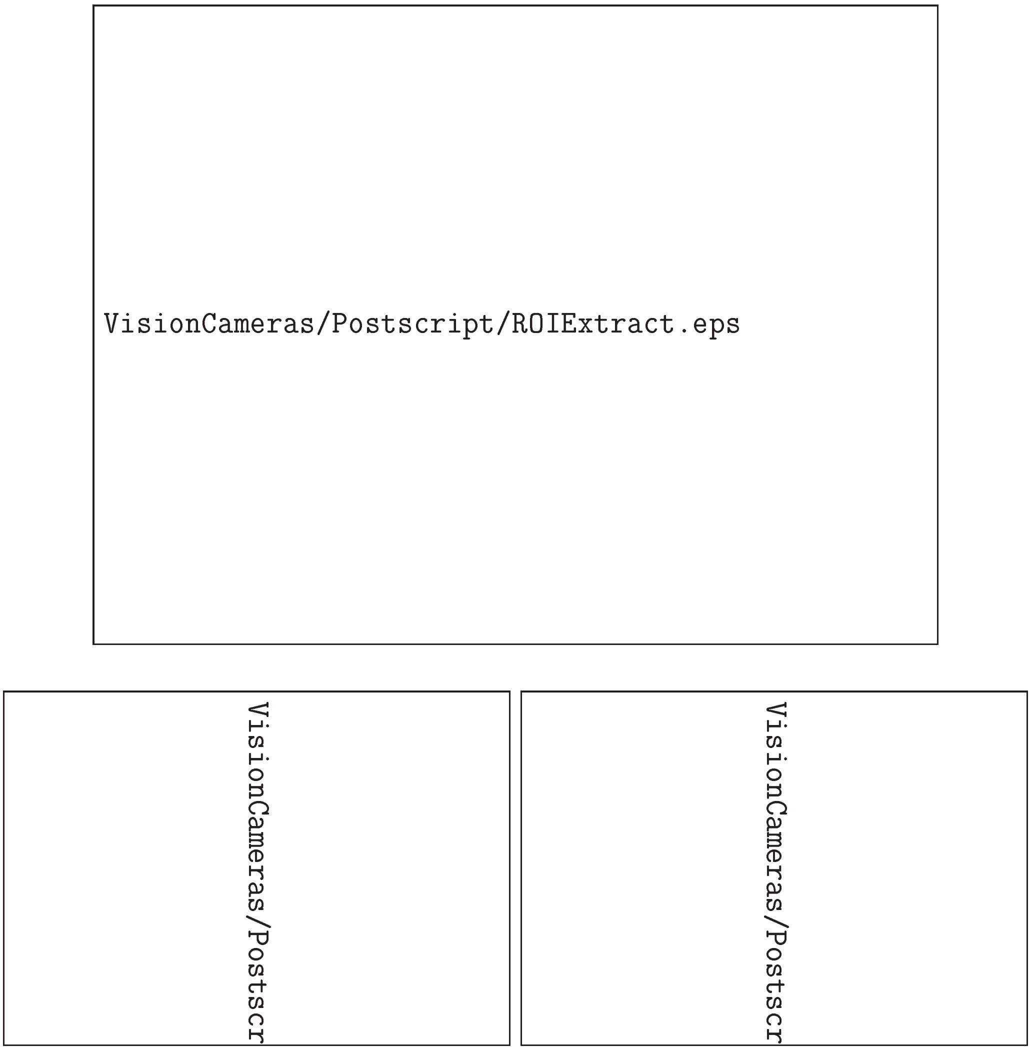
VisionCameras/Postscript/ROIExtract.eps
VisionCameras/Postscript/u30.ps
VisionCameras/Postscript/v30.ps

Figure 3.5: Edge enhancement in an image. Left: the intensity profile across the reference mark in the vertical direction. Right: the intensity profile in the horizontal direction.

Dedicated analog vision cameras without these features exist on the market. But another very attracting alternative to the analog cameras is the new digital cameras. These cameras quantize the potential on the CCD chip and transmit a digital video signal directly. For comparison the analog cameras make an analog signal from the discrete read-out on the CCD chip. This signal is then transmitted and A/D converted in the frame-grabber. The result is an extra D/A and A/D conversion.

According to the camera specification, very long cables are critical for digital cameras. Nevertheless, we have successfully transmitted images over 25–30 meters. Compared to the noise in images from analog cameras the noise in digital images is very easy to identify. It is typically seen as salt and pepper noise.

Still, digital cameras are more expensive than analog cameras, but the growing demand is beginning to show in the price.

The last feature to mention concerning the video signal is the ability to trigger the camera. This is particularly interesting on stereo heads or in applications where flashed light is used. Not all cameras possess an external trigger for simultaneous image acquisition.

3.3 Photogrammetric Quality

The lens system and the resolution of the camera are naturally also very important when the camera is used for photogrammetry. Most cameras are equipped with standard lenses, not produced with photogrammetric measurements in mind. The optical distortion of the lenses is typically rather nonlinear making the calibration more difficult. It is therefore recommended to use professional lenses, where the images can be rectified more successfully. This discussion is of course aimed at high spatial accuracy in the entire image. A wider range of lenses is acceptable if the camera is used for visual servoing in a feedback loop, or if only the center region of the image is used.

To avoid drifting in the optical properties of the lens as a result of temperature variations, a heating element can be placed adjacent to the lens.

The resolution of the image is determined by the CCD chip in the camera. Standard cameras produce images digitized in approximately 600 rows and 800 columns. High resolution cameras with 1300 × 1000 pixels are becoming more common and they are sold at reasonable prices. Even higher resolutions as $2k \times 2k$ and $4k \times 4k$ are also available, but those cameras are still very

	Brand	Type	D/A	Resolution	Price
1)	Basler	A113(p)	D	1296 × 1030	< kr. 27,000
2)	JAI	M10	A	768 × 576	< kr. 9,000
3)	JAI	M50	A	768 × 576	< kr. 3,400
4)	Burle	TC350A	A	720 × 540	< kr. 3,000

	Lens	Price
1)	Nikon F-mount	< kr. 15,000
2,3,4)	Std. C-mount	< kr. 1,000

	Brand	Type	D/A	Price
1)	ITI	MVC 150/40	D	< kr. 20,000
2,3)	DataTranslation	DT152-A	A	< kr. 10,000
4)	Sun	?	A	< kr. 5,000

Table 3.1: Cameras configurations used in the present project. The numbers to the left specify which components were used together. Configuration 4 was only used very early in the project. D/A means digital or analog.

expensive.

Table 3.1 shows the camera, lens, and frame-grabber combinations that have been used in the present work. Configuration 4 was only used in the beginning of the project. The prices are rough estimates (7 *kr.* ∼ 1 *euro* ∼ 1 *$*).

Chapter 4

Projective Geometry

The first chapters have concentrated on the practical aspects of the project. The theoretical foundation of the work will be introduced in the following chapters, that will focus on projective geometry and camera models.

4.1 Introduction

Projective geometry has attracted much interest in the image analysis community over the past one or two decades. It is out of the scope of this dissertation to give a strict mathematical introduction to the subject, but the following sections will give some very important results that form the basis for the understanding of the geometry in the rest of the present work. Projective geometry defines the laws of collinear mappings between vector spaces which includes the perspective projections. This is useful for describing the imaging process, where the three-dimensional object space is mapped into the two-dimensional image space, but is, however, only a special case of the theory. In its general form it explains many important dependencies between the world and its linear projections. For instance, two images of the same scene are subject to a series of mathematical constraints. In fact, every pixel in one image is constrained to a line in the other image. This relationship between two images is governed by a matrix called the fundamental matrix. Similar constraints can be formulated for three and four images through the trifocal and the quadrifocal tensors. The study of these constraints and their singularities gives an important understanding of critical configurations of cameras and object points, in terms of reconstruction and measurement.

4.2 Literature

The very brief introduction to projective analysis in this chapter is based on the results and observations in a large number of papers and lecture notes on the subject. In this section I will only list a few of these. The interest in projective geometry emerged from the observation that the projection of a point in two different images is geometrically constrained. We observe $2 \times 2 = 4$ image coordinates, but the object point is located in three-dimensional space, and can therefore only span three dimensions. Observing the point in one image therefore constrains the projection in the other image to a curve, that proves to be a line owing to the linearity of projective algebra. Whether the cameras are calibrated or not or have equal or different intrinsic parameters makes no difference. These constraints were described by *Faugeras et al.* [18, 16]. The problem of estimating the parameters of the constraint has been addressed by *Hartley and Gupta* [25], *Lawn and Cipolla* [43], *Luong et al.* [46], and *Zhang et al.* [79]. The latter paper describes a very robust method for estimating this epipolar geometry. *Hartley* [22] addresses the problem of proper minimization subject to geometric constraints.

A great deal of the research has focused on the use of uncalibrated cameras for recovering the structure of the observed world. In this set-up there is no distance measure in the object space. How to perform proper triangulation under these circumstances was addressed by *Hartley and Sturm* [23].

Recent papers have dealt with the study of the geometry and algebra of line correspondences and the extension to three and more cameras, cf. *Faugeras and Mourrain* [17], *Triggs* [70], and *Quan* [63]. It turns out that no constraints exist that involve more than four cameras simultaneously.

The problem of auto-calibration has also attracted a lot of interest. Different combinations of the number of cameras, the number of points, and their configuration have been investigated, and the associated singularities have been described. A few examples are *Zisserman et al.* [80], *Hartley* [24], and *Triggs* [71]. The latter paper gives a nice overview of different ways to represent constraints, and accounts for the number of degrees of freedom in each representation.

For a first introduction to the subject I should refer to *Mohr and Triggs* [51] and *Gros et al.* [19]. The former gives a mathematical introduction while the latter discusses the advantages of the projective representations. The dissertation by *Åström* [2] also gives a thorough overview of the algebra. Most of the observations reported in this chapter are discussed in these papers.

4.3 Homogeneous Coordinates

Calculations in projective geometry are based on homogeneous coordinates. Given an n-dimensional Euclidean coordinate $\mathbf{x}$, the $(n+1)$-dimensional homogeneous coordinate $\mathbf{X}$ is obtained by adding an extra coordinate equal to one. Typically, the homogenizing coordinate is appended at the end of the vector

$$\mathbf{x} \quad \mapsto \quad \mathbf{X} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \tag{4.1}$$

Two points in homogeneous space, $\mathbf{X}_1$ and $\mathbf{X}_2$, are said to be equal if they are equal up to scale and we write

$$\mathbf{X}_1 \sim \mathbf{X}_2 \quad \Leftrightarrow \quad \mathbf{X}_1 = \lambda \mathbf{X}_2, \tag{4.2}$$

where λ is the scale difference. To get from the homogeneous coordinate to the Euclidean coordinate we simply normalize with respect to, and remove, the homogenizing coordinate

$$\mathbf{X} \quad \mapsto \quad \mathbf{x} = \frac{1}{X_{n+1}} \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \tag{4.3}$$

We see that the extra coordinate homogenizes an inhomogeneous equation

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad [\mathbf{A} \quad -\mathbf{b}]\,\mathbf{X} = \mathbf{0}, \tag{4.4}$$

where $\mathbf{A}$ is a matrix and $\mathbf{b}$ is a vector. The introduction of homogeneous coordinates makes it possible to express collinearity linearly. That is, the line between two points is spanned by linear combinations of their homogeneous coordinates. Homogeneous coordinates are, in fact, similar to barycentric coordinates, with the only difference that the coordinate constraint is maintained in the homogenizing coordinate

$$\begin{aligned} & \mathbf{X} = a_1\mathbf{X}_1 + a_2\mathbf{X}_2, \\ \Downarrow & \\ & \mathbf{x} = \tfrac{a_1}{a_1+a_2}\mathbf{x}_1 + \tfrac{a_2}{a_1+a_2}\mathbf{x}_2 \\ \Downarrow & \\ & \mathbf{x} = a_1'\mathbf{x}_1 + a_2'\mathbf{x}_2, \quad a_1' + a_2' = 1 \end{aligned} \tag{4.5}$$

where the a's are scalars and $\mathbf{x}$ represent the points on the line through $\mathbf{x}_1$ and $\mathbf{x}_2$. Obviously, three and more points can be used to span planes and hyper-planes.

4.4 Duality

The duality between points and lines in the plane is easily formulated in homogeneous coordinates. Let $\mathbf{L} = [l_1\ l_2\ l_3]$ be the parameters of a line. The line equation is then

$$\begin{array}{ll} & l_1x_1 + l_2x_2 + l_3 = 0 \\ \Updownarrow & \qquad\qquad\qquad\qquad , \\ & \mathbf{LX} = 0 \end{array} \tag{4.6}$$

where row vectors are used for lines and planes. The same duality extends to points and planes in n-dimensional space. The duality implies that the linear combination expressed in equation 4.5 has an equivalent for lines. The linear combinations of two lines $\mathbf{L}_1$ and $\mathbf{L}_2$, in fact, express all the lines $\mathbf{L}$ through their intersection. If we let $\mathbf{X}$ be the point of intersection, we have

$$\begin{array}{ll} & \mathbf{L}_1\mathbf{X} = \mathbf{L}_2\mathbf{X} = 0 \quad \wedge \quad \mathbf{L} = a_1\mathbf{L}_1 + a_2\mathbf{L}_2 \\ \Downarrow & \\ & \mathbf{LX} = 0, \end{array} \tag{4.7}$$

where the a's are scalar coefficients. The spanned set of lines is called a pencil of lines. Similarly, two planes span a pencil of planes with a common line of intersection.

4.5 Point, Line, and Plane at Infinity

Until now, we have only considered the case where the homogenizing coordinate is different from zero. What makes projective geometry so powerful, however, is that a homogenizing coordinate equal to zero can be used to represent points at infinity in a well-defined manner. If we let the homogenizing coordinate of a point tend to zero we see that the Euclidean coordinate will tend to infinity. In fact, the homogeneous representation gives a well-behaved definition of the points at infinity. The direction to the point is given by the first n entries of the homogeneous coordinate. As a matter of fact, we can also define the line at infinity as the line containing all the points at infinity

$$\begin{aligned} & \mathbf{X} = \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} \quad \wedge \quad \mathbf{L}\mathbf{X} = 0 \\ \Downarrow & \\ & \mathbf{L} \sim \begin{bmatrix} \mathbf{0} & 1 \end{bmatrix} \end{aligned} \tag{4.8}$$

Note, that the normal vector of the line at infinity is undefined. The plane at infinity is defined in the same manner in n-dimensional space.

Let us see what happens to equation 4.5 if $\mathbf{X}_2$ is a point at infinity

$$\begin{aligned} & \mathbf{X} = a_1\mathbf{X}_1 + a_2\mathbf{X}_2, \\ \Downarrow & \\ & \mathbf{x} = \mathbf{x}_1 + \tfrac{a_2}{a_1}\mathbf{x}_2, \end{aligned} \tag{4.9}$$

i.e. we simply obtain the usual point and vector parametrization of the line. Linear combinations with the line at infinity give rise to a set of parallel lines, since the line at infinity cannot contribute to a change of the normal vector. The same applies to the plane at infinity.

We see that the points, lines, and planes at infinity do not behave differently from other points, lines, and planes. This is an important characteristic of the projective space. It allows us to formulate a camera model where points at infinity, like the horizon, map to perfectly normal image points, and where normal points in the object space, sitting on the focal plane, map to image points at infinity. A consequence of this uniformity is that there is no notion of distance in the projective space.

4.6 Simple Algebra

The algebra of points, lines, and planes is particularly simple in the homogeneous formulation. Since we will need to determine intersections of planes and do similar geometry in the following chapters, this section will show a few methods and examples.

Assume that we need to determine the intersection of two planes $\mathbf{P}_1$ and $\mathbf{P}_2$, i.e. we search for the points $\mathbf{X}$ that satisfy

$$\mathbf{P}_1\mathbf{X} = \mathbf{P}_2\mathbf{X} = 0 \tag{4.10}$$

A very simple solution to this problem is to form the matrix

$$\mathbf{P}_{12} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix}, \tag{4.11}$$

which is a 2 × 4 matrix with the plane coordinates as rows. Assuming the planes are different, this matrix has rank 2. In other words the dimension of the null-space is $4 - 2 = 2$. The null-space is easily found by singular value decomposition (SVD). Let us denote the two vectors that span the null-space by $\mathbf{X}_1$ and $\mathbf{X}_2$, i.e.

$$\mathbf{P}_{12} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \end{bmatrix} = \mathbf{0} \tag{4.12}$$

Consequently, we have

$$\mathbf{X} = a_1\mathbf{X}_1 + a_2\mathbf{X}_2, \tag{4.13}$$

which is valid even if one of the points is at infinity. Using the same approach, we can find the intersection of three planes as the one-dimensional null-space of a 3 × 4 matrix, or even three points that span a plane from the three-dimensional null-space of a 1 × 4 matrix.

Singular value decomposition also works in the case of lines and points. However, the problem can be solved directly in three dimensions by use of the cross product of vectors '×'. Assume that we want the intersection $\mathbf{X}$ of two lines $\mathbf{L}_1$ and $\mathbf{L}_2$. We have

$$\begin{aligned} & \mathbf{L}_1\mathbf{X} = 0 \quad \wedge \quad \mathbf{L}_2\mathbf{X} = 0 \\ \Downarrow & \\ & \mathbf{X} \sim \mathbf{L}_1 \times \mathbf{L}_2 \end{aligned} \tag{4.14}$$

From duality we can also derive the line $\mathbf{L}$ that passes through two points $\mathbf{X}_1$ and $\mathbf{X}_2$

$$\mathbf{L} \sim \mathbf{X}_1 \times \mathbf{X}_2 \tag{4.15}$$

Successive application of these rules is useful for finding the intersection $\mathbf{X}$ of the diagonals in a quadrilateral defined by four points $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$, and $\mathbf{X}_4$

$$\mathbf{X} \sim (\mathbf{X}_1 \times \mathbf{X}_3) \times (\mathbf{X}_2 \times \mathbf{X}_4) \tag{4.16}$$

4.7 Collineations

A linear mapping by a square full rank $(n+1) \times (n+1)$ matrix of a projective space into another is called a collineation. The map naturally preserves collinearity. The matrix is homogeneous, i.e. scale invariant, and therefore has $(n+1)^2-1=n(n+2)$ degrees of freedom. That is, we need $n+2$ points to define a collineation, as each point represents n degrees of freedom. Hence, four points define a collineation in the plane. For comparison, an affine map is defined by only three points, while a Euclidean map needs only one point and a direction. In fact, the Euclidean and affine maps are embedded in the set of collineations. An affine map $(\mathbf{A},\mathbf{t})$ can be written as the collineation

$$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{4.17}$$

We see that the affine maps are exactly the collineations that preserve the plane at infinity. The n degrees of freedom difference between the affine maps and the collineations are explained by this constraint. The subspace of Euclidean maps is characterized by $n(n+1)/2$ more constraints from the orthogonality of the rotation matrix. Another way to state this is that Euclidean collineations preserve the absolute conic, which is a complex conic on the plane at infinity. This is a very useful formulation in the study of the nature of projective space, but it will not be used further in this dissertation.

Chapter 5 will present the 2D-to-2D camera matrix, which is an example of a collineation.

4.8 The Cross-Ratio

Even though the general collineation has more degrees of freedom than the affine map, it still preserves a lot of the structure in the mapped space. We have already seen that collineations preserve collinearity. Besides that, a measure called the cross-ratio of four points, is invariant to the projective transformations. Assume that we have four points on a line $\mathbf{X}_i=(\mathbf{x}_i,1)$, $i=1,2,3,4$. The distances between the points are defined by $d_{ij}=|\mathbf{x}_i-\mathbf{x}_j|$. The cross-ratio of the four points is calculated as

$$c(\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3,\mathbf{X}_4)=\frac{d_{13}d_{24}}{d_{14}d_{23}} \tag{4.18}$$

and is invariant to projective transformations. The proof follows from straightforward calculations. If one of the points is at infinity, say $\mathbf{X}_4$, we can take the limit of the fraction to obtain

$$c(\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3,\mathbf{X}_4)=\frac{d_{13}}{d_{23}} \tag{4.19}$$

It is interesting to notice that if this point stays at infinity under the transformation, the map is effectively affine and 4.19 simply express the usual affine invariant.

The cross-ratio is a useful tool to find corresponding points between two different projective representations. Given three points on a line, one can always calculate the forth point by using the invariant cross-ratio. Another application is shape recognition, where the cross-ratio can be used to distinguish different shapes.

Owing to duality the cross-ratio is naturally also defined for lines.

Chapter 5

Camera Models

In order to be able to use a camera as a measuring device, it is necessary to model the relationship between the observed world and the projected image. Depending on the required accuracy this model can be more or less advanced. In some vision applications, the camera repeatedly measures one distance on the same type of object and in the same position. In this case the camera can be modelled by a single scale factor, mapping from metric coordinates to pixel coordinates. This simple model can be extended to affine and polynomial models, that are able to account for more variability. These models are non-physical in the sense that their parameters have no interpretation in terms of camera orientation, focal length, image center etc. As a drawback they do not tend to generalize to the entire image and they sometimes behave strangely between calibration points.

Instead, a physical model, that can be calibrated very accurately to the entire image, is preferred. This type of model is typically adopted by photogrammetrists and vision researchers. The parameters can be chosen more carefully in the physical model, and their interpretation is very useful. The model most widely used is introduced in the following sections. It is a linear model in projective space and its parameters are easily interpreted when the results in the previous chapter are used.

In the final section of this chapter, the model will be extended with some non-linear correction terms that account for the image distortion caused by the camera optics. The model in its total form is obtained from *Heikkilä and Silvén* [26]. Basically the same model is used by *Kraus* [42], *Melen* [49], *Weng et al.* [76], *Tsai* [72], *Li and Lavest* [45], *Willson and Shafer* [77], *Nomura et* *al.* [53], and *Horn* [31].

5.1 Linear Model from 3D to 2D

This section will describe the pin-hole camera model, that maps points in a three-dimensional object coordinate system to points in the two-dimensional image plane. This is a linear model with a physical interpretation of the parameters. The model is typically not sufficiently detailed for very accurate photogrammetry, but it forms the conventional basis for more comprehensive camera models.

5.1.1 From World Point to Image Point

Physical camera models are divided into extrinsic and intrinsic camera parameters. The extrinsic parameters describe the position and orientation, called the pose, of the camera in relation to the observed world. In a multi-camera installation they also describe the relationship between the cameras. The intrinsic parameters model the projection of points in the three-dimensional camera coordinate system into the two-dimensional image plane.

Let us first describe the external part. The extrinsic parameters cause a coordinate change from a coordinate system centered in object space to a coordinate system centered in the camera. Let us denote the object space coordinates by capital letters, $\mathbf{X} = (X, Y, Z)$, and camera space coordinates by small letters, $\mathbf{x} = (x, y, z)$. The coordinates of the origin of the object space coordinate system $\mathcal{O}_\mathbf{X}$ in camera space are denoted $\mathbf{x}_0$. Similarly, we denote the camera space origin $\mathcal{O}_\mathbf{x}$ by $\mathbf{X}_c$ in the object coordinate system. The orientation of the camera relative to the object coordinate system is represented by a rotation matrix $\mathbf{R}$. The three-dimensional Euclidean transformation of a point $\mathbf{X}_i$ can now be written

$$\mathbf{x}_i = \mathbf{R}\mathbf{X}_i + \mathbf{x}_0 \tag{5.1}$$

Note, that the equation holds for $\mathbf{X}_0 = \mathbf{0}$. We see immediately that

$$\mathbf{X}_c = -\mathbf{R}^T\mathbf{x}_0 \tag{5.2}$$

It is important to realize that $\mathbf{R}$ does not rotate the geometric vector but only performs a coordinate change from one system to another. This is emphasized

Geometric	Object Space	Camera Space
$\mathcal{O}_{\mathbf{X}}\mathcal{O}_{\mathbf{x}}$	$-\mathbf{R}^T\mathbf{x}_0$	$-\mathbf{x}_0$
$\mathcal{O}_{\mathbf{x}}\mathcal{O}_{\mathbf{X}}$	$\mathbf{R}^T\mathbf{x}_0$	$\mathbf{x}_0$
$\mathbf{e}_x$	$\mathbf{R}_{1\cdot}$	$(1,0,0)^T$
$\mathbf{e}_y$	$\mathbf{R}_{2\cdot}$	$(0,1,0)^T$
$\mathbf{e}_z$	$\mathbf{R}_{3\cdot}$	$(0,0,1)^T$
$\mathbf{e}_X$	$(1,0,0)^T$	$\mathbf{R}_{\cdot 1}$
$\mathbf{e}_Y$	$(0,1,0)^T$	$\mathbf{R}_{\cdot 2}$
$\mathbf{e}_Z$	$(0,0,1)^T$	$\mathbf{R}_{\cdot 3}$
$\mathcal{O}_{\mathbf{x}}\mathbf{P}_i$	$\mathbf{R}^T\mathbf{x}_i$	$\mathbf{x}_i$
$\mathcal{O}_{\mathbf{X}}\mathbf{P}_i$	$\mathbf{X}_i$	$\mathbf{R}\mathbf{X}_i$

Table 5.1: Coordinate Representations. $\mathbf{P}_i$ is a point in space.

by equation 5.2 where $\mathbf{X}_c$ and -$\mathbf{x}_0$ are indeed the same geometric vector, only described in different coordinate systems. Table 5.1 shows how the different geometric vectors are represented in the two coordinate systems.

Often, $\mathbf{R}$ is represented in a more compact form by three subsequent rotations around the coordinate axes. This way the three degrees of freedom are represented by the primary angle ω around the x-axis, the secondary angle φ around the y-axis, and finally the tertiary angle κ around the z-axis. The factorization is written as

$$\mathbf{R} = \mathbf{R}_\kappa \mathbf{R}_\varphi \mathbf{R}_\omega \tag{5.3}$$

$$= \begin{bmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix} \tag{5.4}$$

Note, that the matrix product is sensitive to the order of the rotations. The elements of $\mathbf{R}$ become

$$\begin{aligned} r_{12} &= \sin\omega\sin\varphi\cos\kappa - \cos\omega\sin\kappa & r_{11} &= \cos\varphi\cos\kappa \\ r_{22} &= \sin\omega\sin\varphi\sin\kappa + \cos\omega\cos\kappa & r_{21} &= \cos\varphi\sin\kappa \\ r_{13} &= \cos\omega\sin\varphi\cos\kappa + \sin\omega\sin\kappa & r_{31} &= -\sin\varphi \\ r_{23} &= \cos\omega\sin\varphi\sin\kappa - \sin\omega\cos\kappa & r_{32} &= \sin\omega\cos\varphi \\ r_{33} &= \cos\omega\cos\varphi \end{aligned} \tag{5.5}$$

Once the object points are described in the camera's coordinate system, the actual projection can be performed. The pin-hole model assumes a central

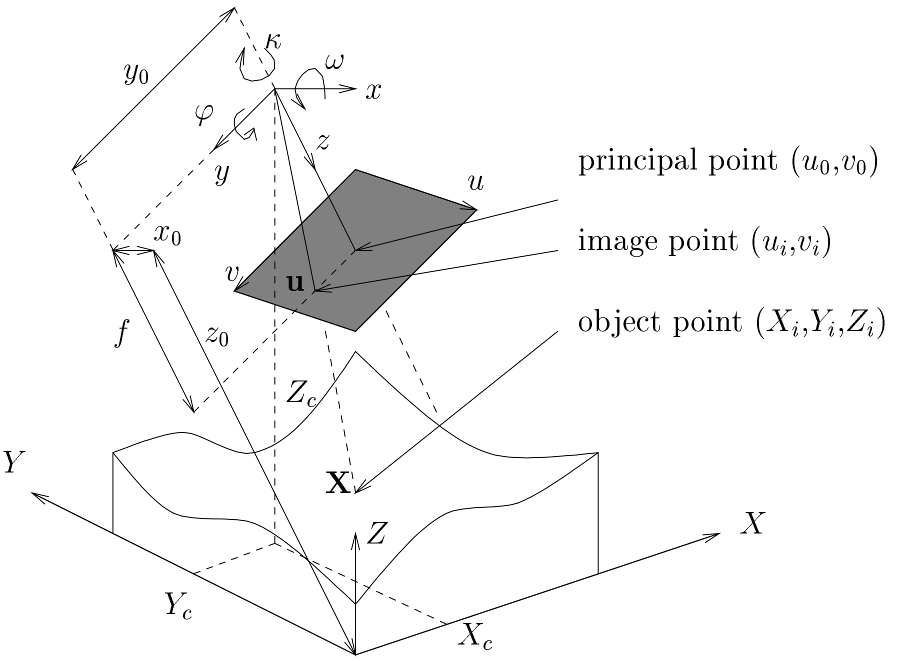

Figure 5.1: Sketch of the camera-world relationship with specification of the extrinsic and intrinsic camera parameters.

projection of the object points. That means that a point is projected in a straight line through the *focal center*, being the origin of the camera coordinate system, onto the *image plane*. The image plane is parallel to the xy-plane, which is called the *focal plane*. The distance between the focal plane and the image plane is the *focal length* of the camera and is denoted f. The z-axis, which is perpendicular to the image plane and passes through the focal center is called the *principal axis*.

If we denote by $\tilde{\mathbf{u}}_i = (\tilde{u}_i, \tilde{v}_i)$ the two-dimensional metric image coordinate of the projected point $\mathbf{x}_i$, the collinearity of the central projection gives

$$\lambda \begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ f \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}, \qquad \lambda \in R \tag{5.6}$$

$$\Updownarrow$$

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \frac{f}{z_i} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{5.7}$$

The principle of the projection is sketched in figure 5.1.

Now pixels in the image are referred to by their row and column indices and not by their metric coordinates. We therefore introduce the pixel coordinates $\mathbf{u}_i$ and parametrize the conversion as

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} D_u s_u & \beta \\ 0 & D_v \end{bmatrix} \begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{5.8}$$

D_u and D_v are two unit change factors, that map meters to pixels. They are assumed to be known from the camera's specifications and should not be estimated. The aspect ratio s_u models a possible deviation in the ratio between D_u and D_v. The exact values of D_u and D_v are not important as they are totally correlated with s_u and f. Actually, some models ignore these factors and express the scale and aspect ratio by two focal lengths in pixel units, $f_u = D_u s_u f$ and $f_v = D_v f$. The origin of the metric image coordinate system has, in pixel units, been denoted $\mathbf{u}_0 = (u_0, v_0)$. The final parameter, β, accounts for the skewness of the CCD chip. It is safe to assume that $\beta = 0$, and the skewness factor has been ignored in this project.

5.1.2 Homogeneous Formulation

To see how this camera model relates to the results in the previous chapter, we rewrite equations 5.1-5.8 in homogeneous coordinates. First, the Euclidean transformation becomes

$$\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{x}_0 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix} \tag{5.9}$$

Next step is the central projection with focal length f in equation 5.7

$$\begin{bmatrix} \tilde{\mathbf{u}}_i \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \tag{5.10}$$

Finally, the metric image coordinates are mapped to pixel coordinates as in equation 5.8

$$\begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix} \sim \begin{bmatrix} D_u s_u & \beta & u_0 \\ 0 & D_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_i \\ 1 \end{bmatrix} \tag{5.11}$$

All this can be concatenated into one single 3 × 4 homogeneous calibration matrix $\mathbf{P}$ called the direct linear transformation (DLT)

$$\begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix} \sim \mathbf{P} \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix} \tag{5.12}$$

where

$$\begin{aligned}
\mathbf{P} &= \begin{bmatrix} D_u s_u & \beta & u_0 \\ 0 & D_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{x}_0 \\ \mathbf{0}^T & 1 \end{bmatrix} && (5.13) \\
&= \begin{bmatrix} D_u s_u f & \beta f & u_0 \\ 0 & D_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{x}_0 \end{bmatrix} && (5.14) \\
&= \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{x}_0 \end{bmatrix} && (5.15) \\
&= \begin{bmatrix} \mathbf{KR} & \mathbf{Kx}_0 \end{bmatrix} && (5.16)
\end{aligned}$$

where the homogeneous triangular matrix $\mathbf{K}$ contains all the internal parameters. Note, that if $\mathbf{K}$ is known, all the individual intrinsic parameters can be calculated.

5.1.3 Interpretation of the 3×4 Camera Matrix

When working with the 3×4 calibration matrix, it is very useful to be able to interpret the meaning of the different entries of the matrix without doing the factorization sketched in the previous section. As it turns out the matrix itself is actually very informative. The simple observations, given in the following, are easily used to verify the validity of the matrix in relation to the known physical set-up of an installation.

Let us denote the i'th row of $\mathbf{P}$ by $\mathbf{P}_{i\cdot}$. If we first examine which object space points $\mathbf{X}$ that map to the image line $u = 0$ we have

$$u = 0 \quad \Leftrightarrow \quad \mathbf{P}_{1\cdot} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = 0 \qquad (5.17)$$

That is, $\mathbf{P}_{1\cdot}$ is exactly the plane defined by $u = 0$ and the focal center. The plane coefficients are represented in object space coordinates and the normal vector is defined by the first three entries. Similarly, $\mathbf{P}_{2\cdot}$ defines the plane that maps to $v = 0$. Finally, all points in the plane $\mathbf{P}_{3\cdot}$ map to infinity in the image plane. To have this property, $\mathbf{P}_{3\cdot}$ must be a plane that is parallel to the image plane and intersects the focal center. This is exactly the focal plane. Therefore the vector consisting of the first three entries of $\mathbf{P}_{3\cdot}$ define the direction of the principal axis. The point at infinity on the principal axis is defined by this vector supplied with a homogenizing coordinate equal to zero, and the image of this point is exactly the principal point (u_0, v_0).

Taking the dot product with the i'th row does not give the metric distance to the plane $\mathbf{P}_{i\cdot}$ as the planes have not been normalized with respect to the normal vector. But even so, we can still interpret e.g. the u-coordinate as the weighted ratio between the distance to $\mathbf{P}_{1\cdot}$ and the distance to the focal plane $\mathbf{P}_{3\cdot}$.

In the same fashion we can interpret the columns $\mathbf{P}_{\cdot j}$ of the calibration matrix. The first column $\mathbf{P}_{\cdot 1}$ is the homogeneous coordinate of the image of the point at infinity on the X-axis, $\mathbf{X} = (1, 0, 0, 0)$. Similarly, $\mathbf{P}_{\cdot 2}$ and $\mathbf{P}_{\cdot 3}$ are the images of the vanishing points on the Y-axis and Z-axis, respectively. Finally, $\mathbf{P}_{\cdot 4}$ is the image of the origin of the object space coordinates. Using the results in the previous chapter, we can represent the projection of e.g. the entire X-axis by the image line parameters $\mathbf{P}_{\cdot 1} \times \mathbf{P}_{\cdot 4}$.

With these observations in mind the geometric effect of the intrinsic parameters can be described. Changing u_0 causes a linear mixing of the plane defined by $u = 0$ and the focal plane, so that the former is rotated around its intersection with the latter. The value of v_0 has the same effect on the plane defined by $v = 0$. The skewness parameter β causes a mixing of the first two rows of $\mathbf{P}$ so that the plane defined by $u = 0$ is rotated around its intersection with the plane defined by $v = 0$. Finally, the focal length f and the aspect ratio s_u define the distance measure associated with these two planes in relation to the focal plane.

The following interpretation is obtained from *Åström* [2]. If we are presented with an arbitrary full rank calibration matrix, it must have a null-space of dimension one that we can call $\mathbf{C}$. This is actually a homogeneous point with the property that all points $\mathbf{X}$ on a line through $\mathbf{C}$ map to the same image point

$$\mathbf{P}[\mathbf{C} + \lambda \mathbf{X}] \sim \mathbf{P}\mathbf{X}, \tag{5.18}$$

where $\mathbf{X}$ is also homogeneous. $\mathbf{C}$ is the homogeneous coordinate of the focal center. In the general case, the planes defined by the rows of $\mathbf{P}$ intersect in a finite point, $\mathbf{C} = (\mathbf{X}_c, 1)$, and we find

$$\mathbf{X}_c = -[\mathbf{P}_{\cdot 1}\mathbf{P}_{\cdot 2}\mathbf{P}_{\cdot 3}]^{-1}\mathbf{P}_{\cdot 4} \tag{5.19}$$

Note, how this result agrees with equations 5.16 and 5.2. A special case occurs if $\mathbf{C}$ is a point at infinity, $\mathbf{C} = (\mathbf{X}_k, 0)$, in which case $\mathbf{P}$ describes a parallel projection in the direction $\mathbf{X}_k$. A typical example is the affine camera model, where, in addition, the focal plane is also the plane at infinity. When the projection has only a very weak perspective effect, this model can be more robustly estimated than the full perspective model.

5.1.4 Direct Estimation

The conventional linear method to estimate the parameters of $\mathbf{P}$ will be described shortly in this section. The next chapter will show the more accurate approach used in the current project. The linear method is included here as it gives a good starting guess for the more advanced methods, and as it shows why this method degrades as the perspective effect increases.

The estimation is based on a rearrangement of the projection equations

$$u = \frac{(\mathbf{X}^T, 1) \cdot (P_{11}, P_{12}, P_{13}, P_{14})}{(\mathbf{X}^T, 1) \cdot (P_{31}, P_{32}, P_{33}, P_{34})} \tag{5.20}$$

$$v = \frac{(\mathbf{X}^T, 1) \cdot (P_{21}, P_{22}, P_{23}, P_{24})}{(\mathbf{X}^T, 1) \cdot (P_{31}, P_{32}, P_{33}, P_{34})} \tag{5.21}$$

If we multiply by the denominator and rearrange, we obtain the error term of the direct linear method

$$\epsilon_{u,direct} = (\mathbf{X}^T, 1) \cdot (P_{11}, P_{12}, P_{13}, P_{14}) - u(\mathbf{X}^T, 1) \cdot (P_{31}, P_{32}, P_{33}, P_{34}) \tag{5.22}$$

with an equivalent expression for $\epsilon_{v,direct}$. Given n corresponding object points and image points, $\mathbf{X}_1, \ldots, \mathbf{X}_n$ and $\mathbf{u}_1, \ldots, \mathbf{u}_n$, we can set up a linear set of equations

$$\begin{bmatrix} \mathbf{X}_1^T & 1 & \mathbf{0}^T & 0 & -u_1\mathbf{X}_1^T & -u_1 \\ \mathbf{0}^T & 0 & \mathbf{X}_1^T & 1 & -v_1\mathbf{X}_1^T & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{X}_n^T & 1 & \mathbf{0}^T & 0 & -u_n\mathbf{X}_n^T & -u_n \\ \mathbf{0}^T & 0 & \mathbf{X}_n^T & 1 & -v_n\mathbf{X}_n^T & -v_n \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{33} \\ P_{34} \end{bmatrix} = \begin{bmatrix} \epsilon_{u_1,direct} \\ \epsilon_{v_1,direct} \\ \vdots \\ \epsilon_{u_n,direct} \\ \epsilon_{v_n,direct} \end{bmatrix} \tag{5.23}$$

where the residuals can be minimized by the least squares method. The P's should be constrained to avoid the solution $\mathbf{P} = \mathbf{0}$. The easiest way is to apply the singular value decomposition. The solution is the vector corresponding to the smallest singular value.

The least squares method is, however, only statistically justified if the noise is modelled more carefully. In the case of calibration, the noise stems from the detection of the calibration points in the image. We should therefore rather minimize

$$\epsilon_{u,image} = \frac{(\mathbf{X}^T, 1) \cdot (P_{11}, P_{12}, P_{13}, P_{14})}{(\mathbf{X}^T, 1) \cdot (P_{31}, P_{32}, P_{33}, P_{34})} - u \tag{5.24}$$

with an equivalent expression for $\epsilon_{v,image}$. Unfortunately, this minimization requires iterative methods. We see that the difference between the two residuals is

$$\epsilon_{direct} = \epsilon_{image}(\mathbf{X}^T, 1) \cdot (P_{31}, P_{32}, P_{33}, P_{34}), \tag{5.25}$$

where the correction term is exactly the point's distance to the focal plane. The direct method therefore has its best performance when the points have a comparable distance to the focal plane, or in other words the depth of the object is small compared to the distance to the camera. The relation also shows that the algorithm can be iterated with a re-weighting based on the points' distances to the focal plane.

5.1.5 Separation of the Camera Matrix

Once the linear calibration matrix has been estimated, it is of interest to extract the values of the intrinsic and extrinsic parameters. This is straightforward, when applying the observations in the previous sections. We see from equation 5.16 that the first 3×3 matrix of $\mathbf{P}$ factors into the rotation and the intrinsic parameters. The factorization is seen to be a variant of a QR factorization. Once $\mathbf{K}$ is calculated, we can obtain $\mathbf{x}_0$ through back-substitution with the last column of $\mathbf{P}$. In this section the first 3×3 matrix of $\mathbf{P}$ is denoted $\mathbf{M}$.

To show how the QR factorization relates to the observations in the previous sections, I will sketch two algorithms.

Gram-Schmidt Orthogonalization

It was already stated that the third row of $\mathbf{P}$ contains the parameters of the focal plane, and that the normal vector points in the direction of the principal axis, being the z-axis of the camera. We therefore immediately have

$$\mathbf{R}_{3\cdot} = \frac{\mathbf{M}_{3\cdot}}{|\mathbf{M}_{3\cdot}|} \tag{5.26}$$

The second row of $\mathbf{P}$ is the plane defined by $v = 0$. As the skewness parameter works on u and not on v, this plane cuts the focal plane in the x-axis of the camera. The normal vector is therefore in the yz-plane of the camera coordinate system. Removing the z part, we get

$$\mathbf{R}_{2\cdot} = \frac{\mathbf{M}_{2\cdot} - (\mathbf{M}_{2\cdot} \cdot \mathbf{R}_{3\cdot})\mathbf{R}_{3\cdot}}{|\mathbf{M}_{2\cdot} - (\mathbf{M}_{2\cdot} \cdot \mathbf{R}_{3\cdot})\mathbf{R}_{3\cdot}|} \tag{5.27}$$

We could proceed and remove the contribution of $\mathbf{R}_{2\cdot}$ and $\mathbf{R}_{3\cdot}$ from $\mathbf{M}_{1\cdot}$, but it is easier just to take the cross product

$$\mathbf{R}_{1\cdot} = \mathbf{R}_{2\cdot} \times \mathbf{R}_{3\cdot} \tag{5.28}$$

The steps taken to obtain the factorization are recognized as the Gram-Schmidt orthogonalization of a non-orthogonal basis. It is seen that the rotation matrix is obtained from $\mathbf{M}$ doing only a few calculations. The intrinsic parameters are now given by

$$\mathbf{K} = \mathbf{M}\mathbf{R}^T \tag{5.29}$$

Givens' Method

If we would like the rotation to be factorized into primary, secondary, and tertiary rotations, Givens' method is more appropriate. This method is based on cancellation of the sub-diagonal elements of a matrix by a series of primitive one-dimensional rotations. In the present case, we can use $\mathbf{R}_\omega$, $\mathbf{R}_\varphi$, and $\mathbf{R}_\kappa$ as Givens' matrices

$$\mathbf{K} = \mathbf{M}\mathbf{R}_\omega^T\mathbf{R}_\varphi^T\mathbf{R}_\kappa^T \tag{5.30}$$

Now it is only a matter of choosing ω, φ, and κ, such that $\mathbf{K}$ becomes an upper triangular matrix. We begin by determining $\mathbf{R}_\omega$ so that m'_{32} becomes zero

$$\begin{aligned} \mathbf{M}' &= \mathbf{M}\mathbf{R}_\omega^T & (5.31) \\ &= \frac{1}{\sqrt{m_{33}^2 + m_{32}^2}} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & m_{33} & m_{32} \\ 0 & -m_{32} & m_{33} \end{bmatrix} & (5.32) \end{aligned}$$

We see that $\omega = \angle(m_{33}, m_{32})$, as $\mathbf{R}_\omega$ is transposed. Next, we rotate to make m''_{31} zero, while preserving the zero obtained by the first rotation

$$\mathbf{M}'' = \mathbf{M}'\mathbf{R}_\varphi^T \tag{5.33}$$

$$= \frac{1}{\sqrt{m_{33}^2 + m_{31}^2}} \begin{bmatrix} m'_{11} & m'_{12} & m'_{13} \\ m'_{21} & m'_{22} & m'_{23} \\ m'_{31} & 0 & m'_{33} \end{bmatrix} \begin{bmatrix} m'_{33} & 0 & m'_{31} \\ 0 & 1 & 0 \\ -m'_{31} & 0 & m'_{33} \end{bmatrix}, \quad (5.34)$$

This gives $\varphi = \angle(m_{33}, -m_{31})$. Finally, $\mathbf{R}_\kappa$ is determined to cancel the final sub-diagonal non-zero element

$$\mathbf{K} = \mathbf{M}''\mathbf{R}_\kappa^T \quad (5.35)$$

$$= \frac{1}{\sqrt{m_{22}^2 + m_{21}^2}} \begin{bmatrix} m''_{11} & m''_{12} & m''_{13} \\ m''_{21} & m''_{22} & m''_{23} \\ 0 & 0 & m''_{33} \end{bmatrix} \begin{bmatrix} m''_{22} & m''_{21} & 0 \\ -m''_{21} & m''_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (5.36)$$

where $\kappa = \angle(m_{22}, m_{21})$. Note, that we develop the individual rotation matrices to be direct rotations, i.e. $det(\mathbf{R}_\omega) = det(\mathbf{R}_\varphi) = det(\mathbf{R}_\kappa) = 1$. We see that Givens' method is very well suited for the factorization into three rotations around the three coordinate axes.

5.1.6 Constrained Back-Projection

It is often of interest to perform back-projection, i.e. to find an object point that corresponds to a given image point. A single solution is only defined if we constrain the back-projection to e.g. a plane in object space. If we denote the homogeneous image point $\mathbf{u}$, and constrain the back-projected point $\mathbf{X}$ to the plane $\mathbf{P}_x$, we have

$$\mathbf{u} \sim \mathbf{P}\mathbf{X} \qquad \mathbf{P}_x\mathbf{X} = 0 \quad (5.37)$$

The solution can be found by collecting the two constraints in a single equation

$$\begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{P} \\ \mathbf{P}_x \end{bmatrix} \mathbf{X}, \quad (5.38)$$

where solving for $\mathbf{X}$ and normalizing give the Euclidean coordinates. Note, that the problem is well-posed if $\mathbf{P}_x$ is not spanned by the rows of the camera matrix $\mathbf{P}$, which means that $\mathbf{P}_x$ cannot intersect the focal center.

5.2 Linear Model from 2D to 2D

Sometimes the observed object space is two-dimensional. This is for instance the case when the camera looks at flat objects lying on a flat table. In this situation we use a 3×3 direct linear transformation to map back and forth between the object space and the image space. Since this camera matrix is a collineation, it defines a one-to-one map between the two spaces. It is not possible unambiguously to derive the pose or the intrinsic parameters of the camera from the two-dimensional DLT. In the present project the 3×3 camera matrix has been used to map image points back onto a plane in three-dimensional space. In this case the 3×3 matrix is derived directly from the three-dimensional camera model. This is shown in the next section.

5.2.1 From 3×4 to 3×3 Camera Matrix

A two-dimensional coordinate system can be defined in the three-dimensional space by an origin and two basis vectors $(\mathbf{V}_0, \mathbf{V}_x, \mathbf{V}_y)$. Given the two-dimensional coordinates of a point in this plane (X', Y') we can calculate the three-dimensional coordinates

$$\mathbf{X} = \mathbf{V}_0 + X'\mathbf{V}_x + Y'\mathbf{V}_y \tag{5.39}$$

To find the coordinate relationship between the plane in object space and the image plane, we write the projection equations in homogeneous coordinates

$$\begin{aligned} \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} &\sim \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} && (5.40) \\ &= \mathbf{P} \begin{bmatrix} \mathbf{V}_x & \mathbf{V}_y & \mathbf{V}_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}' \\ 1 \end{bmatrix} && (5.41) \\ &= \mathbf{P}' \begin{bmatrix} \mathbf{X}' \\ 1 \end{bmatrix} && (5.42) \end{aligned}$$

$\mathbf{P}'$ is now a 3×3 camera matrix working between the plane in object space and the image plane. Algebraically, the matrix has full rank if the null-space of $\mathbf{P}$ is not in the range of the $\mathbf{V}$-matrix above. Projectively, this means that the object space plane cannot pass through the focal center of the camera matrix, in which case the entire plane maps to a line in the image. In a structured industrial application, we can assure that this never happens.

Often, we consider the points in the usual coordinate system with a fixed

Geometric	World Space	Camera Space	2D Plane
$\mathcal{O}_{\mathbf{X}}\mathbf{V}_0$	$\mathbf{V}_0$	$\mathbf{R}\mathbf{V}_0$	$*$
$\mathcal{O}_{\mathbf{x}}\mathbf{V}_0$	$\mathbf{V}_0 + \mathbf{R}^T\mathbf{x}_0$	$\mathbf{R}\mathbf{V}_0 + \mathbf{x}_0$	$*$
$\mathbf{e}_{X'}$	$\mathbf{V}_x$	$\mathbf{R}\mathbf{V}_x$	$(1,0)^T$
$\mathbf{e}_{Y'}$	$\mathbf{V}_y$	$\mathbf{R}\mathbf{V}_y$	$(0,1)^T$

Table 5.2: Coordinate Representations.

value of e.g. $Z = Z_0$. This gives

$$\mathbf{P}' = \mathbf{P}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & Z_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.43}$$

$$= \begin{bmatrix} \mathbf{P}_{\cdot 1} & \mathbf{P}_{\cdot 2} & Z_0\mathbf{P}_{\cdot 3} + \mathbf{P}_{\cdot 4} \end{bmatrix} \tag{5.44}$$

This reduction is useful when solving the inverse problem of finding $\mathbf{X}'$ from $\mathbf{u}$.

Sometimes, it is more informative first to remove the effect of the intrinsic parameters from $\mathbf{u}$, and then set up a metric vector equation to determine $\mathbf{X}'$. In that case, the representations in table 5.1, page 35, and table 5.2 are helpful in setting up a consistent equation in either object space or camera space coordinates.

5.2.2 Interpretation of the 3×3 Camera Matrix

The interpretation of the 3×3 camera matrix follows the same reasoning as was used in section 5.1.3. It is therefore only described briefly.

The first row of the matrix defines the line that maps to the image line $u = 0$. Similarly, the second row is the line that maps to $v = 0$, and the third row is the line that maps to the line at infinity. Note, in equation 5.41, that if $\mathbf{V}_x$ and $\mathbf{V}_y$ are perpendicular to the principal axis, meaning that the object space plane is parallel to the image plane, the third row will be $(0, 0, 1)$ and the map is affine.

The first column describes the image of the vanishing point on the $\mathbf{X}'$-axis, the second column is the vanishing point on the $\mathbf{Y}'$-axis, and finally the third column is the image of the origin of the object plane. This is indeed also what is stated in equation 5.41.

As in section 5.1.3 the cross products of the columns can be interpreted. For instance, the line defined by $X' = 0$ has the image coordinates $\mathbf{P}'_{\cdot 2} \times \mathbf{P}'_{\cdot 3}$. Note, that the cross products of the columns play the role of the rows in the inverse collineation $\mathbf{P}'^{-1}$. This is only natural as the three cross products define the cofactor matrix, which is indeed the homogeneous inverse.

5.2.3 Direct Estimation

The 3×3 matrix can of course also be estimated directly from observed data. We build up a system of equations like 5.23, where the observed object space points are two-dimensional, such that the system matrix becomes $2n \times 8$. At least four points are needed. The problem is again solved by singular value decomposition.

5.3 Non-linear Lens Distortion

As stated earlier, the linear camera model is not sufficiently accurate for photogrammetric measurements. This is primarily due to non-linear image distortions caused by the camera optics. It is therefore necessary to include a few extra terms in the camera model to compensate for these effects.

5.3.1 Radial Distortion and Decentering

Typically, the non-linear correction is limited to the radial and tangential distortion of the lens. The terms used in this presentation are obtained from *Heikkilä and Silvén* [26]. The same model with several extensions can be found in *Melen* [49].

The first term models the pure radial distortion, which represents most of the non-linearities in normal lenses. The same model as is applied here is presented in most papers on non-linear lens calibration. See the literature in the beginning of the chapter. The second term describes a decentering distortion, which has both a radial and a tangential component. It is caused by non-collinearity of the centers of curvature of the lens surfaces [26].

Radial distortion is sometimes also called pin-cushion or barrel distortion with reference to the deformation of the image. It models by far most of the distortion caused by a typical lens. The coordinate corrections are modelled by the expression

$$\begin{bmatrix} \delta\tilde{u}_i^{(r)} \\ \delta\tilde{v}_i^{(r)} \end{bmatrix} = \begin{bmatrix} \tilde{u}_i(k_1 r_i^2 + k_2 r_i^4) \\ \tilde{v}_i(k_1 r_i^2 + k_2 r_i^4) \end{bmatrix}, \tag{5.45}$$

where k_1 and k_2 are the coefficients for radial distortion and $r_i = \sqrt{\tilde{u}_i^2 + \tilde{v}_i^2}$. More terms can be added in the polynomial, but the expression above typically gives satisfactory results.

The decentering distortion is modelled by

$$\begin{bmatrix} \delta\tilde{u}_i^{(t)} \\ \delta\tilde{v}_i^{(t)} \end{bmatrix} = \begin{bmatrix} 2p_1\tilde{u}_i\tilde{v}_i + p_2(r_i^2 + 2\tilde{u}_i^2) \\ p_1(r_i^2 + 2\tilde{v}_i^2) + 2p_2\tilde{u}_i\tilde{v}_i \end{bmatrix}, \tag{5.46}$$

with coefficients p_1 and p_2.

The effect of these two types of non-linear distortion is depicted in figure 5.2. The two images on the left show the positive and negative radial distortion, respectively, and the image on the right shows an example of positive tangential distortion. The effect has, of course, been exaggerated in these images.

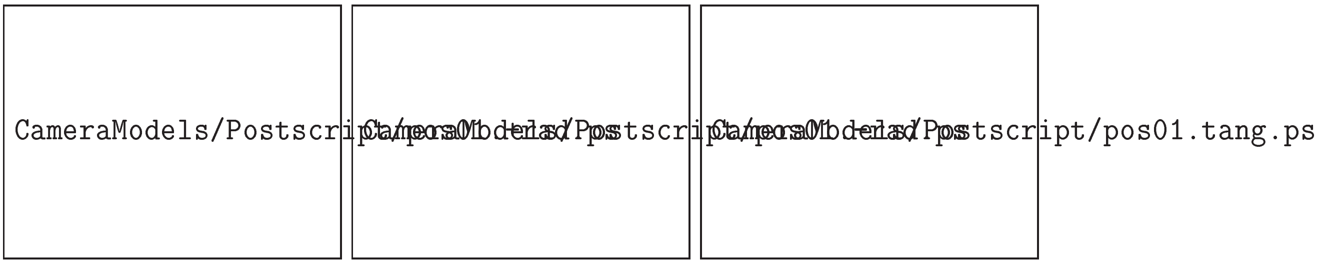

Figure 5.2: The modelled non-linear lens distortion. Left: pin-cushion (+) and barrel (-) radial distortion. Right: decentering (+) distortion. The signs refer to the dominating parameter.

Since these correction terms model distortions in the lens, they should be applied after the projection, but before the coordinates are converted from metric units to pixels. We therefore introduce

$$\tilde{\mathbf{u}}_i := \tilde{\mathbf{u}}_i + \delta\tilde{\mathbf{u}}_i^{(r)} + \delta\tilde{\mathbf{u}}_i^{(t)} \tag{5.47}$$

right before equation 5.8.

Note, that the introduction of the correction terms makes the camera model non-linear. This causes the algebra to be less simple, and projections and back-projections become computationally more expensive. One can therefore choose to perform an image rectification before doing any image analysis. This might also be necessary if the image processing e.g. relies on the straightness of projected lines or the elliptic shape of projected circles. If the image analysis algorithms give no motivation to rectify the image, one should be aware that many pixels, that will never be accessed, are rectified.

Even though equation 5.47 maps from undistorted coordinates to distorted coordinates, it is the relevant expression to use when rectifying an image. The rectification proceeds as follows. For each undistorted coordinate we calculate the corresponding distorted coordinate. At this non-integer pixel location we perform a bilinear or bicubic interpolation in the image to obtain the new intensity. The choice of interpolation depends on the application. Typically, I have used Catmull-Rom cubic interpolation described by *Mitchell and Netravali* [50] and *Dodgson* [14]. In their parametrization this corresponds to $(B, C) = (0, 0.5)$. Figure 5.3 shows an example of rectification. The image to the left is rectified using the parameters estimated in the next chapter, and the result is shown in the middle. The right-hand image is the difference between the two left images. The largest displacement in the image is approximately 60 *pixels*.

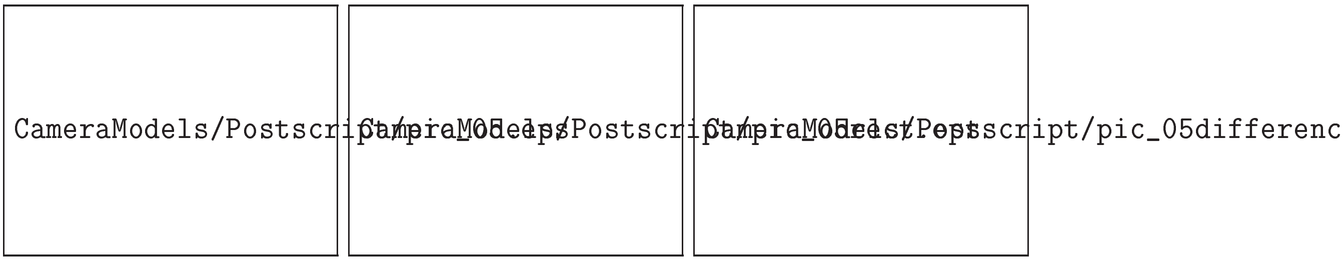
CameraModels/Postscript/pic_05.eps
CameraModels/Postscript/pic_05rect.eps
CameraModels/Postscript/pic_05differenc

Figure 5.3: Image rectification. Left: an image acquired with a JAI camera with an 8 *mm* lens. Middle: the rectified image. Right: subtraction of the two left images.

5.3.2 Inversion

If rectification is not applied, a method is required to un-distort coordinates to perform back-projection. In this case we need the real inverse. One approach that has been proposed is also to model the inverse as equation 5.45 and 5.46. The parameters are then estimated from a large number of generated correspondences. In this way, the inverse is calculated off-line. The drawback is that the residuals in the estimation add to the overall uncertainty of the camera system.

I have implemented a more direct solution based on the Newton-Raphson method. The rationale for this approach is that the distortion function has a very low curvature such that it is extremely well approximated by its first order expansion. Let us denote the distortion function in 5.47 by

$$\mathbf{d}(\mathbf{u}) = \mathbf{u} + \delta\mathbf{u}^{(r)} + \delta\mathbf{u}^{(t)} \tag{5.48}$$

If we are given a distorted coordinate $\mathbf{u}^{(d)}$ and want to calculate the undistorted coordinate $\mathbf{u} = \mathbf{d}^{-1}(\mathbf{u}^{(d)})$ it is equivalent to solving the equation

$$\begin{aligned} \mathbf{e}(\mathbf{u}) &= \mathbf{u}^{(d)} - \mathbf{d}(\mathbf{u}) && (5.49) \\ &= \mathbf{0} && (5.50) \end{aligned}$$

with respect to $\mathbf{u}$. It is therefore possible to determine the inverse through a number of steps in the Newton-Raphson method. If we let the subscript denote the iteration number, we have

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n - (\frac{\partial}{\partial\mathbf{u}}\mathbf{e}(\mathbf{u}_n))^{-1}\mathbf{e}(\mathbf{u}_n) && (5.51) \\ &= \mathbf{u}_n + (\frac{\partial}{\partial\mathbf{u}}\mathbf{d}(\mathbf{u}_n))^{-1}(\mathbf{u}^{(d)} - \mathbf{d}(\mathbf{u}_n)) && (5.52) \end{aligned}$$

A natural starting guess is

$$\mathbf{u}_0 = \mathbf{u}^{(d)} \tag{5.53}$$

The 2×2 partial derivative of the distortion function can be directly calculated from equation 5.45 and 5.46. Remember that r is a function of u and v

$$\begin{aligned} \left(\frac{\partial\mathbf{d}}{\partial\mathbf{u}}\right)_{11} &= 1 + k_1(r^2 + 2\tilde{u}^2) + k_2 r^2(r^2 + 4\tilde{u}^2) + 2p_1\tilde{v} + 6p_2\tilde{u} \\ \left(\frac{\partial\mathbf{d}}{\partial\mathbf{u}}\right)_{12} &= 2\tilde{u}\tilde{v}(k_1 + 2k_2r^2) + 2p_1\tilde{u} + 2p_2\tilde{v} \\ \left(\frac{\partial\mathbf{d}}{\partial\mathbf{u}}\right)_{21} &= 2\tilde{u}\tilde{v}(k_1 + 2k_2r^2) + 2p_1\tilde{u} + 2p_2\tilde{v} \\ \left(\frac{\partial\mathbf{d}}{\partial\mathbf{u}}\right)_{22} &= 1 + k_1(r^2 + 2\tilde{v}^2) + k_2 r^2(r^2 + 4\tilde{v}^2) + 6p_1\tilde{v} + 2p_2\tilde{u} && (5.54) \end{aligned}$$

To test the method's performance on average, every pixel in an image was distorted with this approach. It was decided to stop the Newton-Raphson

Method	1 iter.	2 iter.	3 iter.	4 iter.	Function Calls
(5.52)	15%	85%	0%	0%	185%
(5.55)	15%	52%	31%	2%	220%

Table 5.3: Computational cost of the inverse distortion calculation. The numbers are based on an inverse rectification of all the pixels in figure 5.3 and describe the fraction of pixels that need one, two, three, or four iterations to converge. The right column shows the number of calls to the distortion function compared to the number of calls required for a rectification.

iterations when $|\mathbf{u}^{(d)} - \mathbf{u}| < 10^{-2}$. The algorithm was tested on the image in figure 5.3, where the largest pixel displacement was approximately 60 *pixels*. The result was that the algorithm required 2 iterations on average. There are two reasons for this fast convergence. First, many pixels near the center of the image are almost not distorted, and therefore require only one iteration. Second, as stated above, the curvature of the distortion function is very small. Note, that the Hessian matrix 5.54 is very close to the identity $\mathbf{I}$. We can in fact use this approximation to speed up the calculations. Equation 5.52 then reduces to

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \mathbf{u}^{(d)} - \mathbf{d}(\mathbf{u}_n) \tag{5.55}$$

This method was also tested. The result of the two different approaches is summarized in table 5.3. The table shows the percentage of pixels that required one, two, three, and four iterations. No pixels required any more iterations. The last column shows the number of calls to the distortion function. Both algorithms use approximately twice as many calls as the direct distortion. Note however, that the simplified version does not need to calculate the Hessian. The two algorithms were equally fast in a non-optimized implementation. The test-run shows that if there are no immediate image processing or visualization reasons to rectify the image, and if only a fraction of the points need to be back-projected, it should be considered to apply the inverse function presented in this section.

Chapter 6

Calibration

6.1 Introduction

The previous chapter discussed the algebraic relationship between points in the scene and their image coordinates. The result was a camera model with six extrinsic parameters $(x_0, y_0, z_0, \omega, \phi, \kappa)$ representing the pose of the camera and eight intrinsic parameters describing the projection into the image. Four of these are the usual linear parameters, focal length f, aspect ratio s_u, and image center (u_0, v_0), where the skewness parameter β is assumed to have no effect, $\beta = 0$. The remaining four intrinsic parameters (k_1, k_2, p_1, p_2) model radially symmetric and decentering image distortion caused by the optics. All these parameters need to be estimated with high accuracy before the camera can be used for photogrammetry. Furthermore, it is desirable to know the uncertainty of the parameter estimates in order to be able to evaluate the quality of the vision measurements.

The standard approach to calibration is to acquire images of a known calibration object, thereby obtaining corresponding sets of object points and image points. Typically, the object points are modelled to be un-corrupted by noise, while the image points are assumed to have been detected with some uncertainty. All the camera parameters can be estimated from a single image if the points on the calibration object are not coplanar. If they *are* coplanar, more than one image is required.

The camera calibration is only valid as long as the lens settings are not changed. A consequence of this constraint is that the focus and the aperture of the camera must remain set for the application during calibration. At the shipyard, the camera is typically calibrated for large scale applications, so that only objects at some distance from the camera appear sharp in the image. Consequently, the calibration object must also be large to be of any use. To avoid spending too much time on maintaining a large three-dimensional calibration object, it has been decided to base the calibrations at the yard on a planar object. Figure 6.1 shows 51 images of the calibration plate, which measures $1400\,mm \times 2000\,mm$. The printed circular marks have been manually glued onto temperature insensitive graph paper, that is itself glued onto a wooden plate. The entire plate has been coated with a plastic film.

This chapter will present the computer program that has been developed to estimate the camera parameters from these images. The calibration quality will be evaluated with features in this program, and different pitfalls in the interpretation of the results will be discussed.

6.2 Literature

The calibration discussed in this chapter is based feature extraction, pose estimation, and camera model fitting. The camera models and different approaches to calibration are addressed by the authors cited on page 33. The feature extraction part is discussed in chapter 7. Section 6.3.3 on page 57 in this chapter contains some pointers to different pose estimation methods. It seems that the most common and useful guideline concerning accurate calibration is to perform a careful choice of noise model and optimize all parameters in a bundle adjustment, cf. *Kraus* [42] and *Kanatani* [37].

6.3 Calibration Software

Much effort has been put into camera calibration and its interpretation in this dissertation. The reason is that calibration demands a lot of time resources, and errors in this phase affect all future measurements with the camera. It is therefore worthwhile to invest some time in thorough analysis of the calibration procedure, to be able to answer questions like: How many images are needed? From which angles and distances should the images be acquired? What is the uncertainty in the parameters? And how does it effect the projection?

As stated above, the calibrations at the yard are based on a planar cal-

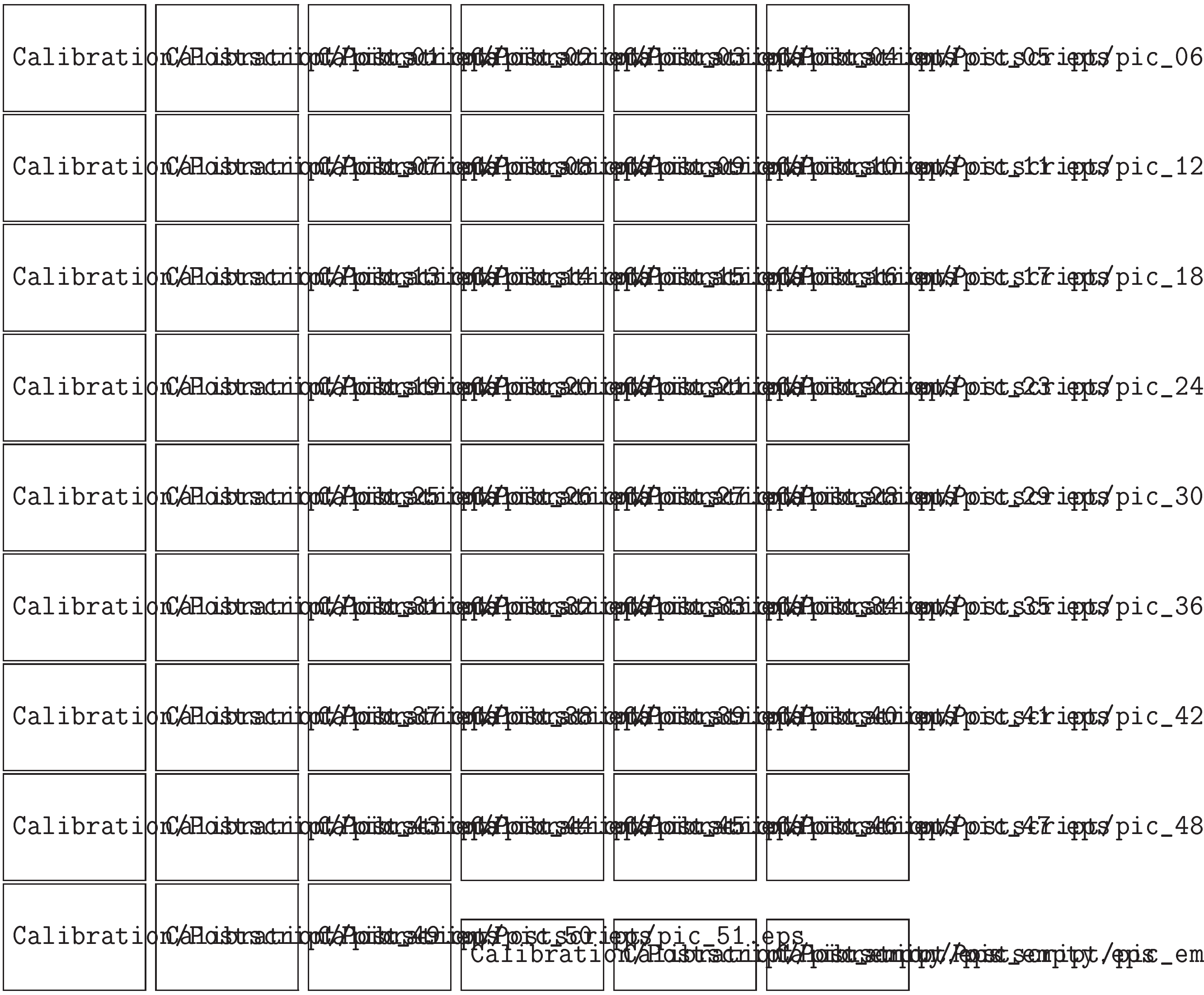

Figure 6.1: Calibration Images. The plate contains 60 coplanar calibration points and measures approximately $1400\,mm \times 2000\,mm$.

ibration object. As can be seen in figure 6.1, the object contains $p = 60$ calibration features, each with two coordinates. Since the 8 intrinsic camera parameters are common to all the images, only 6 additional extrinsic parameters are needed for each image included in the calibration. With v images, the problem is therefore over-constrained by $2pv - 6v - 8$ equations.

Some calibration methods rely on an automatic extraction of the calibration points in the image. An initial value for the parameters can then be obtained by use of the method in section 5.1.4, page 40. It is, however, unreasonable to proceed with the calibration without checking the quality of the acquired images, and it is likely that an automatic approach will establish some erroneous correspondences between image points and calibration points. Furthermore, the automatic feature extraction is liable to fail, as it has to model large scale differences and perspective effects owing to the unknown camera pose. The time spent in detecting such problems at a later stage easily exceeds the extra time required to operate a semi-automatic program. Bearing these considerations in mind, I have based the software on manual initialization. The images are loaded along with a CAD model of the calibration object. The user is then able to drag four points on the model to their respective positions in the image. As the user moves these handles, the pose is updated on-line, and the user can follow the parameter values in a dialog. The calibration procedure is

1. Load image(s)
2. Load model, cf. section 6.3.1 and figure 6.2
3. Load initial camera parameters, cf. section 6.3.2 and figure 6.3
4. Position handles, cf. section 6.3.3 and figure 6.2
5. Check point distribution, cf. section 6.4.3 and figure 6.5
6. Detect calibration features, cf. section 6.3.4 and figure 6.4
7. Check detection, cf. section 6.3.4 and figure 6.4
8. Fit camera parameters, cf. section 6.3.5 and figure 6.3
9. If parameters changed goto 6
10. Check residuals, cf. section 6.4.5 and figure 6.7
11. Check parameter dispersion, cf. section 6.4.4 and figure 6.6
12. Save camera parameters

The following sections will describe the different steps in the list above, with emphasis on the interpretation of the results.

6.3.1 Calibration Object Model

The calibration is based on a model of the calibration object. This model is loaded into the program and projected onto the image. By interaction the user can then adjust the position of the model in the image, and thereby indirectly initialize the extrinsic camera parameters, cf. section 6.3.3. Figure 6.2 shows an overlay of the model of the calibration object on the upper left-hand image in figure 6.1. The calibration features in this particular model consist of three concentric circles alternating in color between black and white. The type and dimension of the calibration feature is specified in the model file. This file also contains two basis vectors that span the plane of the feature. A typical entry in the file looks like this:

```
C3 6.0 30.0 60.0 67.0 -820.0 570.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0
```

The first field, `C3`, specifies the type of the feature, in this case being three concentric circles. A single circle has also been used in some applications. The next four entries specify the three radii in millimeters, and a radius limiting the region of interest. The last nine entries are the position of the feature in the calibration object coordinate system, and the two vectors that span the plane of the feature. The orientation of the feature is used to back-project the image to an isotropic coordinate system, cf. chapter 7. It is also possible to define a model with features on different planes, such that the program can do a three-dimensional calibration from a single image.

6.3.2 Camera Parameters

The initial camera parameters can be read from a file or be entered manually. Figure 6.3 shows the parameter dialog with typical initial values. These are used to project the model into the image. If the parameters deviate a lot from the optimal values, the projected features may be very dissimilar from the actual image. This will cause the feature detection in the first iteration to be biased and will decrease the quality of the fitted camera parameters. The initial guess therefore determines the goodness of the fit in the first iteration. However, the improvement of the parameters in the first iteration, typically leads to a very satisfactory feature detection in the second iteration. Nevertheless, it is always worthwhile to enter the nominal focal length of the lens. If the user wants to interpret the values of the parameters physically, it is also

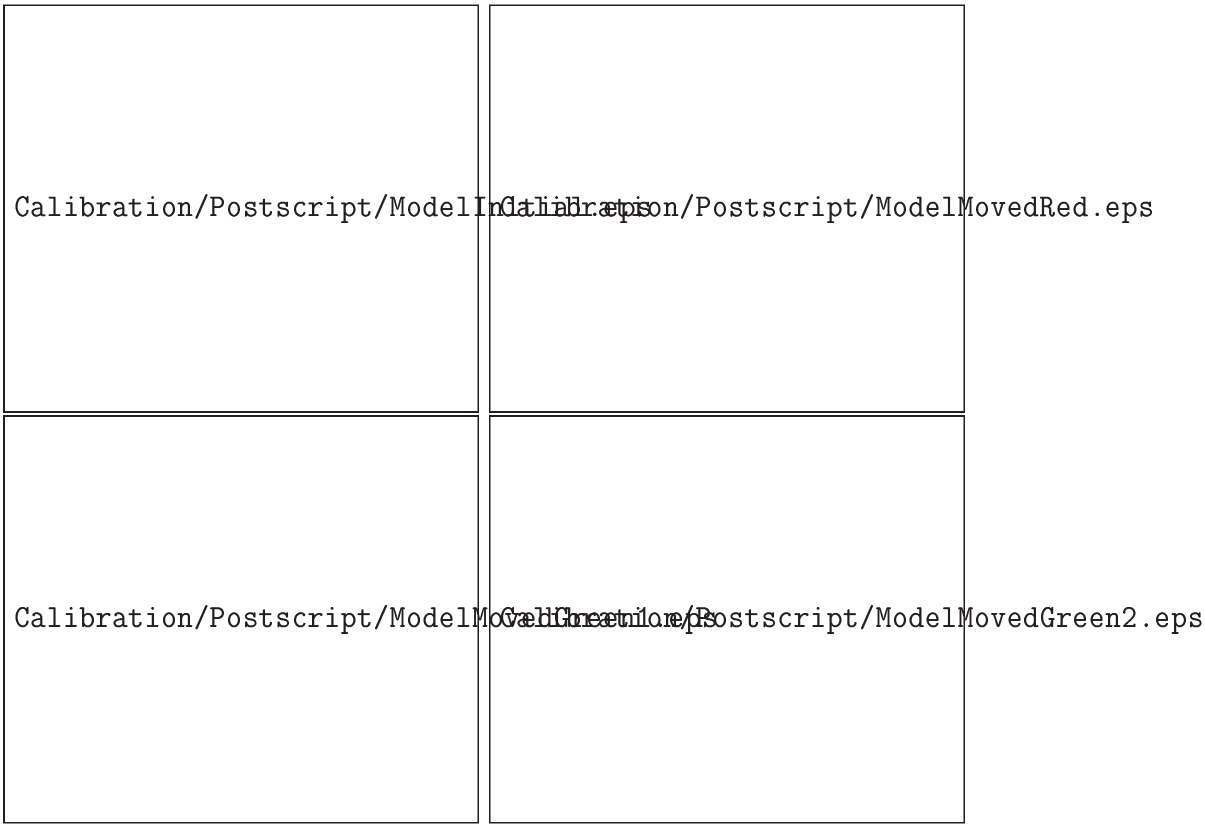
Calibration/Postscript/ModelInitial.eps
Calibration/Postscript/ModelMovedRed.eps
Calibration/Postscript/ModelMovedGreen1.eps
Calibration/Postscript/ModelMovedGreen2.eps

Figure 6.2: Software interface with manual initialization of the pose. The red handles can be moved around in the CAD model, and the green handles can be moved in the image.

necessary to specify the dimension of the CCD chip. All other parameters are automatically assigned a reasonable default value. The default value for the principal point is taken as the center of the loaded image(s). When a camera is re-calibrated or similar cameras are calibrated it can save time to initialize the calibration with a previously estimated set of values.

Figure 6.3 also shows a check-box in front of every field. By un-checking a field, the user can prevent the corresponding parameter from being estimated. The significance of the various parameters can be evaluated by fitting different models. Note, that un-checking all the non-linear lens distortion parameters leaves us with the pure linear model.

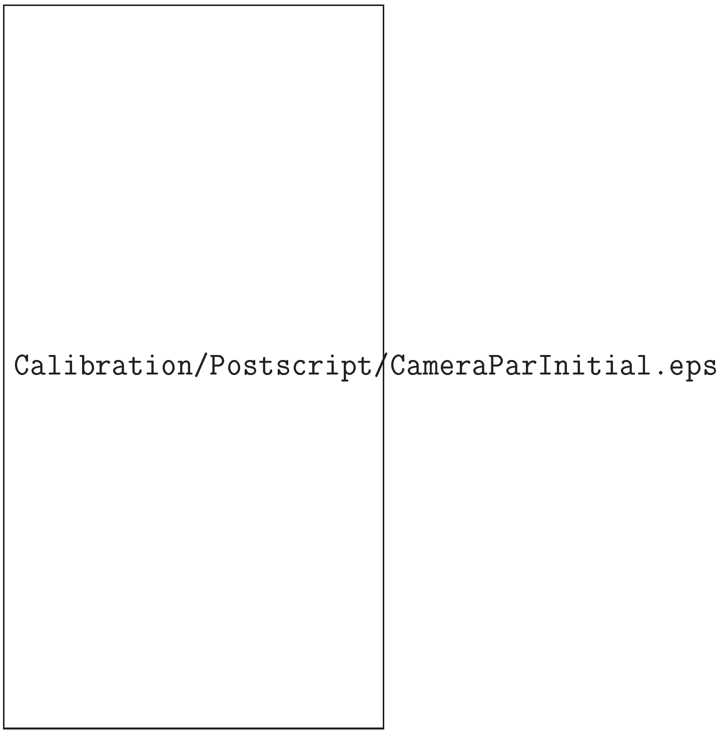

Figure 6.3: The camera parameters dialog with initial values. The user can check the parameters to be estimated.

6.3.3 Pose From Handles

There are two reasons why calibration feature detection can be very time-consuming. First, we have no knowledge of where in the image to perform the search. Second, the appearance of the feature varies a lot in terms of shape and size depending on the pose of the camera. Thereby the size of the search space is increased. Both problems are overcome by the specification of an initial pose. In the software program the pose is calculated from the image position of four handles, that are attached to the CAD model. These handles can be moved around by the operator.

Figure 6.2 shows how the handles are used to drag the model in place. The upper left-hand image shows the handles in the initial positions. In the top right and bottom left images the user is dragging the green handles to their corresponding positions in the image. The bottom right image shows that the handles can also be moved around on the CAD model, if desired.

As the user adjusts the handles, the pose is calculated on-line and can be followed in the parameter dialog. The values of the intrinsic camera parameters are all used when calculating the pose. In fact, the pose could be determined from only three points, but then there is the risk of having four different solutions, cf. *Haralick et al.* [20]. This ambiguity is removed by using four points. With four points the problem can be solved analytically, cf. *Horaud et al.* [29], or by the iterative scheme proposed by *Dementhon et al.* [11, 54], and refined by *Horaud et al.* [30, 8]. The latter schemes are based on iterated affine approximations of the full perspective model, and can be applied to any number of points greater than four. *Horaud et al.* improved the accuracy and convergence of the algorithm by replacing the weak perspective model by the para-perspective model, which is a higher order approximation to the perspective model. The papers also address the special case where the calibration points are coplanar. The immediate advantage of these algorithms is that they require no starting guess. The drawback is that they minimize another error measure than the image residual. Therefore, I always apply a non-linear technique to the output to obtain the true least squares solution. In the present program I have implemented the algorithms by *Dementhon et al.*

6.3.4 Calibration Feature Detection

The accuracy of the entire calibration depends on the quality of the feature detection. The underlying theory of the detection is described in chapter 7. This section will only show some results and demonstrate the value of doing iterations on the detection and the estimation of camera parameters. The detection of features is based on fitting a parametric model to the image intensities. The parameters model the position, the smoothing, the light and dark phase, and possibly a light trend across the feature. Figure 6.4 shows the situation in the first and second iteration of the calibration. In the left image in the top row the operator has positioned the handles attached to the four corner features. Thereby, the approximate position of all the other features is also given. Note, that the non-linear lens distortion, which has not yet been estimated, causes the shape of the model to differ from the underlying image. In the second row, the features have been modelled, and the models have been projected into the image. In the bottom row the model has been subtracted from the image to show the difference. Note, that the position has been fairly well approximated for all features. However, since the shape is not modelled, there is a significant difference between the image and the model. The right column shows the second iteration. Now, the intrinsic parameters have been estimated and the pose is based on all the points. The projected model is seen to be much more similar to the features in the image. The second and third row show that the parametric fit has improved significantly.

Even though the fit is based on least squares minimization, the correlation score has also been calculated. The score for each feature can be seen in the second row. Features that score higher than 0.8 are marked green and are included in the estimation of the camera parameters. The residual in the parametric fit can be used to estimate a theoretical dispersion of the parameters of the feature model. The theory is described in appendix B. Only the position is interesting here for which reason I have conditioned the dispersion of the position estimate on all the other parameters. The estimated dispersions of the upper left-hand feature in figure 6.4 is

$$\mathbf{D}_1(x_c, y_c) = \begin{bmatrix} 1.5e-02 & 7.1e-04 \\ 7.1e-04 & 1.7e-02 \end{bmatrix} mm^2 \approx \begin{bmatrix} 2.6e-03 & 1.2e-04 \\ 1.2e-04 & 2.9e-03 \end{bmatrix} pixel^2 \tag{6.1}$$

$$\mathbf{D}_2(x_c, y_c) = \begin{bmatrix} 3.4e-04 & 4.5e-08 \\ 4.5e-08 & 3.4e-04 \end{bmatrix} mm^2 \approx \begin{bmatrix} 5.8e-05 & 7.8e-09 \\ 7.8e-09 & 5.9e-05 \end{bmatrix} pixel^2, \tag{6.2}$$

where the subscripts refer to the iteration number, and where the position parameters are denoted (x_c, y_c). The dispersion has been mapped to image units by a pure scale factor $60\,mm/25\,pixels$. As can be seen, the uncertainty in the position estimate is reduced significantly by the improved estimate of the intrinsic parameters. In fact, the theoretical standard deviation of the position estimate is as small as $\sigma_{detection} = 0.01\, pixel$. The detection accuracy is of course worse in the images where the features are smaller. The worst case detections have a standard deviation about a tenth of a pixel.

6.3.5 Fitting Camera Parameters

It is important, when dealing with imperfect data, to model the noise where it occurs, cf. *Kanatani* [36]. In camera calibration it is natural to model the noise in the detection of the calibration features in the image. That is, it is assumed that the CAD model of the calibration object precisely describes the physical object, and that the camera model fully describes the projection of the features to some 'true' image locations. Then we try to estimate the camera parameters that minimize the distance between these 'true' positions and the observed positions. The cost function is

$$\mathcal{F}(\mathbf{a}_1, \ldots, \mathbf{a}_v) = \sum_{i=1}^{v} \sum_{j=1}^{p} |\mathbf{u}_{ij,obs} - \mathbf{u}_{ij}(\mathbf{a}_i, \mathbf{X}_p)|^2, \tag{6.3}$$

where the camera parameters $\mathbf{a}_i$ have the intrinsic part in common.

The actual estimation of the camera parameters is based on the Levenberg-Marquardt algorithm [61]. This algorithm uses a mixture of gradient descent and quasi-Newton steps to arrive at the minimum. As the starting guess of the optimization is very close to the optimal values, the update formula will very soon be dominated by the quasi-Newton step. The convergence will therefore be almost quadratic from an early point.

6.4 Calibration Example

The previous section described the different steps in the calibration procedure. This section will show an example of a calibration, which will serve as a good basis for a few interesting discussions.

The example concerns the calibration of a JAI M50 camera, cf. table 3.1, page 23, with an $8\, mm$ lens. Cameras with focal lengths in that range typically have a considerable amount of non-linear lens distortion, as can be seen in figure 6.4. The calibration has been performed with a subset of the images in figure 6.1, namely the images in column 1, 3, and 5. Numbering the images

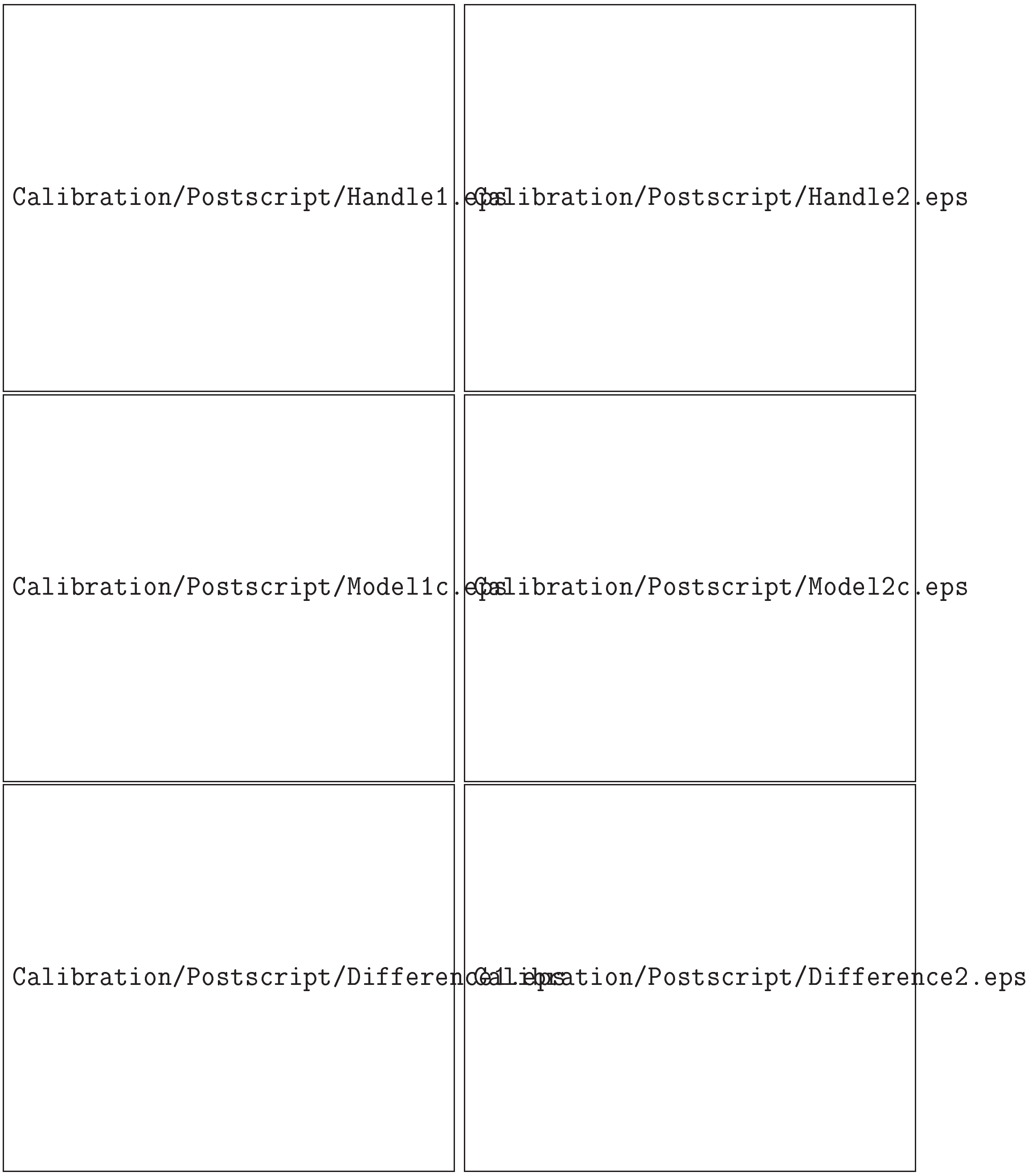
Calibration/Postscript/Handle1.eps
Calibration/Postscript/Handle2.eps
Calibration/Postscript/Model1c.eps
Calibration/Postscript/Model2c.eps
Calibration/Postscript/Difference1.eps
Calibration/Postscript/Difference2.eps

Figure 6.4: Calibration mark detection. Left: first iteration. Right: second iteration. Top: calibration mark shape predicted from handle positions and detected positions, respectively. Middle: fitted models with correlation scores. Bottom: difference between fit and image.

Parameter	1st Iteration	2nd Iteration
Focal length f $[mm]$	7.93	7.95
Principal point u_0 $[pix]$	388.17	389.61
Principal point v_0 $[pix]$	290.99	290.18
Aspect ratio s_u	1.021	1.021
Radial distortion k_1	-5.47e-03	-6.01e-03
Radial distortion k_2	1.69e-05	4.55e-05
Tangential distortion p_1	1.27e-04	9.59e-05
Tangential distortion p_2	-3.56e-05	-1.16e-06
RMS Residual image 1 σ_1 $[pix]$	0.20	0.10
RMS Residual all images σ $[pix]$	0.14	0.09

Table 6.1: Estimated camera parameters for a JAI M50 camera with an 8 mm lens.

from left to right, these are called the odd images. This series contains 26 different views. Table 6.1 shows the estimated parameters after first and second iteration. More iterations will not change the parameters significantly. First, we verify that the estimated parameters seem reasonable. The focal length is very close to the nominal value, the principal point is within a few pixels from the image center $(384, 288)$, and the aspect ratio is very close to 1.0. The non-linear lens distortion coefficients are rather small, and the dominating terms only change a little from the first to the second iteration. Second, we examine the calibration residual. The residual after the second iteration is $\sigma = 0.09\,pixel$. The calibration residual is often interpreted as the uncertainty of the camera model, i.e. the expected inaccuracy when using the camera. A more sophisticated interpretation of this measure will be attempted in the following sections.

6.4.1 Goodness Of Fit

Two different situations can be imagined when fitting the parameters in a model. If the scale of the observation noise is known, the terms in equation 6.3 can be normalized by their variance. Since the terms are also expected to have zero mean, we have a χ^2-distributed cost function. We can therefore perform a goodness of fit test of the cost function in a χ^2-distribution. If, on the other hand, the scale of the observation noise is not known, we can estimate the general noise level from the value of the cost function. This is how the residuals were estimated in table 6.1. However, we actually have estimates of the noise level from the feature detection, cf. equation 6.2. The estimated residual is seen to be 10 times greater than the standard deviation of the position estimate. Therefore, the model would have failed a goodness of fit test on any reasonable significance level. This indicates that our presumptions have been violated. Either the camera model is not able to describe the true physical projection from object space to image space, or the CAD model is itself corrupted by noise. The sum of contributions can be written

$$\sigma^2 = \sigma^2_{cad} + \sigma^2_{model} + \sigma^2_{detection} \tag{6.4}$$

where the first two terms on the right-hand side are the CAD model inaccuracy and the unknown camera model error. To investigate if the residual could be explained by the CAD model, we re-scale the residual to metric units $\sigma \approx 0.23\, mm$. The scaling factor is valid for the average distance to the calibration object. The calibration object was built manually from printed paper, remember, so it is very plausible, that most of this residual is explained by an inaccurate CAD model. It has been attempted to estimate a better CAD model from the images, by back-projecting the detected feature locations in all images to a fixed depth $Z = 0$ to construct an average CAD model. Calibration with this model did not improve the calibration residual. This could be due to the fact, that the different poses have compensated for the discrepancy between the image and the projected CAD model in different ways. It could also be due to the features not being really coplanar. To reveal this would require a bundle adjustment with all camera parameters and CAD data at the same time. This has not been attempted.

6.4.2 Verification On Test Data

Another way to analyze the quality of a fit is to test it against a set of data, that was not used in the optimization. This is a very important test, as it proves the model's ability to generalize. Especially, it can reveal if too many parameters are included in the model. In the calibration case, we have not yet used any of the even images in figure 6.1. We can perform a calibration with these images without estimating any of the intrinsic camera parameters. However, we do have to estimate the pose for every image. The overall residual becomes $\sigma_{even} = 0.10\,pixel$. Note, that this value is very close to the calibration residual, which suggests that the model is not over-fitted.

6.4.3 Spatial Calibration Feature Distribution

A small calibration residual does not itself prove a good calibration. It is also important that the calibration generalizes to the entire image field. Normally, it is stated that the calibration is only valid within the convex hull of the calibration points. It is therefore desirable that the calibration points are distributed evenly in the entire image to ensure that the camera model performs equally well in all pixel positions. To verify this, the calibration program can show the position of all features in one single image. Figure 6.5 shows such an image, where each yellow spot indicates a feature in some image. It is seen that the points span the image very well.

Figure 6.5: Calibration point distribution. The validity of the calibration depends on the local density of calibration points. The calibration should only be used within the convex hull of the points.

6.4.4 Parameter Dispersion

As stated earlier, the calibration residual is often interpreted as a lower bound on the quality of the camera model. This is only true in the sense that the residual is a measure of the detection accuracy, which will be unchanged for future measurements. However, it neglects the fact that more images cause better calibrations. Assuming a perfect camera model, a perfect CAD model, and a fair noise distribution, we are theoretically able to estimate the parameters of the camera to any precision regardless of the noise level. This fact is not reflected in the calibration residual, which is limited by the detection accuracy. It is more fair to study the theoretical dispersion of the camera parameters. This can be estimated from the calibration matrix and the Jacobian of the camera function, cf. appendix B.

Let us first see which information can be obtained from the dispersion matrix. Table 6.2 shows the dispersion of the camera parameters when the model is fitted only from the first image in figure 6.1. Despite the known singularities, a single image has been used in this example to limit the size of the dispersion matrix. This also gives us a chance to verify the strong correlations that arise from a single image of coplanar data. The correlations in the table are very much as expected. If we imagine an incrementation of the focal length, the image points will move towards the center of the image. Since the calibration points primarily project to the upper left-hand side of the image, we will see a similar behaviour by incrementing either x_0, y_0, or z_0. This is reflected in the correlation between those four variables. Further, we see that these parameters anti-correlate with the coordinates of the principal point. This is also what should be expected from the camera model. It is also interesting to observe the magnitude of the variances in the dispersion matrix. For instance, the standard deviation of u_0 is approximately $20\,pixels$ and the standard deviation of z_0 is $100\,mm$.

To obtain a more true picture of the accuracy in the calibration, it is necessary to remove some of these correlations. This is, of course, the objective of using more images. It can, however, also be modelled from the example above, by e.g. conditioning the intrinsic parameters on the extrinsic parameters, cf. appendix B. Table 6.3 shows the conditional dispersion of the intrinsic parameters. The variances have now been reduced significantly, and the intrinsic parameters are much less correlated with each other. Note, however, that the radial distortion parameters still correlate quite strongly with the focal length. This indicates, that the calibration images should be chosen to span the perspective distortion and the lens distortion independently. This is done by acquiring images from different angles to yield data with different amounts of perspective distortion and other images where the calibration features spread nicely in the image field to span the variation modelled by the lens distortion

	f	u_0	v_0	s_u	k_1	k_2	p_1	p_2	x_0	y_0	z_0	ω	φ	κ
f	1.1e-01	-6.0e 00	-1.3e 00	2.3e-03	-1.1e-05	1.6e-06	6.0e-05	2.6e-04	1.7e 01	3.2e 00	3.6e 01	-1.4e-04	9.8e-03	-3.2e-04
u_0	-6.0e 00	3.6e 02	6.4e 01	-9.6e-02	4.9e-04	-9.6e-05	-3.1e-03	-1.6e-02	-9.8e 02	-1.7e 02	-1.9e 03	1.0e-02	-5.2e-01	1.5e-02
v_0	-1.3e 00	6.4e 01	1.5e 01	-2.8e-02	1.2e-04	-1.7e-05	-6.8e-04	-2.9e-03	-1.8e 02	-4.0e 01	-4.0e 02	2.7e-03	-1.1e-01	4.6e-03
s_u	2.3e-03	-9.6e-02	-2.8e-02	6.5e-05	-2.5e-07	2.3e-08	1.3e-06	4.2e-06	2.9e-01	7.1e-02	7.4e-01	-6.4e-07	2.0e-04	-7.3e-06
k_1	-1.1e-05	4.9e-04	1.2e-04	-2.5e-07	7.6e-09	-2.6e-10	-3.2e-09	-1.5e-08	-1.5e-03	-3.1e-04	-3.5e-03	2.7e-08	-9.5e-07	3.8e-08
k_2	1.6e-06	-9.6e-05	-1.7e-05	2.3e-08	-2.6e-10	3.0e-11	7.5e-10	4.2e-09	2.6e-04	4.3e-05	5.0e-04	-3.4e-09	1.3e-07	-4.2e-09
p_1	6.0e-05	-3.1e-03	-6.8e-04	1.3e-06	-3.2e-09	7.5e-10	3.5e-08	1.4e-07	8.7e-03	1.8e-03	1.9e-02	-3.2e-08	5.2e-06	-1.6e-07
p_2	2.6e-04	-1.6e-02	-2.9e-03	4.2e-06	-1.5e-08	4.2e-09	1.4e-07	7.2e-07	4.3e-02	7.3e-03	8.4e-02	-4.6e-07	2.3e-05	-6.8e-07
x_0	1.7e 01	-9.8e 02	-1.8e 02	2.9e-01	-1.5e-03	2.6e-04	8.7e-03	4.3e-02	2.7e 03	4.7e 02	5.4e 03	-2.7e-02	1.5e 00	-4.4e-02
y_0	3.2e 00	-1.7e 02	-4.0e 01	7.1e-02	-3.1e-04	4.3e-05	1.8e-03	7.3e-03	4.7e 02	1.0e 02	1.0e 03	-7.3e-03	2.8e-01	-1.2e-02
z_0	3.6e 01	-1.9e 03	-4.0e 02	7.4e-01	-3.5e-03	5.0e-04	1.9e-02	8.4e-02	5.4e 03	1.0e 03	1.2e 04	-4.3e-02	3.1e 00	-1.0e-01
ω	-1.4e-04	1.0e-02	2.7e-03	-6.4e-07	2.7e-08	-3.4e-09	-3.2e-08	-4.6e-07	-2.7e-02	-7.3e-03	-4.3e-02	2.6e-06	-1.1e-05	1.6e-06
φ	9.8e-03	-5.2e-01	-1.1e-01	2.0e-04	-9.5e-07	1.3e-07	5.2e-06	2.3e-05	1.5e 00	2.8e-01	3.1e 00	-1.1e-05	8.5e-04	-2.7e-05
κ	-3.2e-04	1.5e-02	4.6e-03	-7.3e-06	3.8e-08	-4.2e-09	-1.6e-07	-6.8e-07	-4.4e-02	-1.2e-02	-1.0e-01	1.6e-06	-2.7e-05	1.8e-06

	f	u_0	v_0	s_u	k_1	k_2	p_1	p_2	x_0	y_0	z_0	ω	φ	κ
f	1.00	-0.94	-0.96	0.85	-0.38	0.85	0.95	0.93	0.97	0.95	1.00	-0.26	1.00	-0.71
u_0	-0.94	1.00	0.87	-0.63	0.30	-0.92	-0.86	-0.99	-1.00	-0.86	-0.94	0.34	-0.93	0.61
v_0	-0.96	0.87	1.00	-0.87	0.34	-0.78	-0.93	-0.86	-0.90	-1.00	-0.96	0.44	-0.96	0.87
s_u	0.85	-0.63	-0.87	1.00	-0.36	0.53	0.88	0.62	0.70	0.87	0.86	-0.05	0.87	-0.68
k_1	-0.38	0.30	0.34	-0.36	1.00	-0.55	-0.20	-0.20	-0.32	-0.35	-0.38	0.20	-0.38	0.33
k_2	0.85	-0.92	-0.78	0.53	-0.55	1.00	0.73	0.89	0.91	0.77	0.84	-0.39	0.83	-0.57
p_1	0.95	-0.86	-0.93	0.88	-0.20	0.73	1.00	0.87	0.89	0.92	0.95	-0.11	0.95	-0.65
p_2	0.93	-0.99	-0.86	0.62	-0.20	0.89	0.87	1.00	0.99	0.86	0.92	-0.34	0.92	-0.61
x_0	0.97	-1.00	-0.90	0.70	-0.32	0.91	0.89	0.99	1.00	0.89	0.96	-0.32	0.96	-0.64
y_0	0.95	-0.86	-1.00	0.87	-0.35	0.77	0.92	0.86	0.89	1.00	0.95	-0.45	0.95	-0.88
z_0	1.00	-0.94	-0.96	0.86	-0.38	0.84	0.95	0.92	0.96	0.95	1.00	-0.25	1.00	-0.71
ω	-0.26	0.34	0.44	-0.05	0.20	-0.39	-0.11	-0.34	-0.32	-0.45	-0.25	1.00	-0.24	0.75
φ	1.00	-0.93	-0.96	0.87	-0.38	0.83	0.95	0.92	0.96	0.95	1.00	-0.24	1.00	-0.71
κ	-0.71	0.61	0.87	-0.68	0.33	-0.57	-0.65	-0.61	-0.64	-0.88	-0.71	0.75	-0.71	1.00

Table 6.2: Dispersion of camera parameters when only estimated from the first image in figure 6.1. Top: covariance matrix. Bottom: correlation matrix.

	f	u_0	v_0	s_u	k_1	k_2	p_1	p_2
f	1.6e-06	2.4e-05	3.7e-06	-4.8e-10	-5.2e-08	2.1e-09	-2.5e-09	-1.6e-08
u_0	2.4e-05	1.3e-03	1.1e-04	6.2e-06	-5.2e-07	2.4e-08	-1.2e-07	3.4e-07
v_0	3.7e-06	1.1e-04	2.5e-04	-4.1e-07	-2.3e-07	4.8e-09	-1.6e-07	-2.0e-07
s_u	-4.8e-10	6.2e-06	-4.1e-07	1.1e-07	1.1e-08	-1.2e-10	3.9e-10	2.0e-08
k_1	-5.2e-08	-5.2e-07	-2.3e-07	1.1e-08	4.0e-09	-1.1e-10	3.3e-10	3.5e-09
k_2	2.1e-09	2.4e-08	4.8e-09	-1.2e-10	-1.1e-10	4.1e-12	-7.1e-12	-5.8e-11
p_1	-2.5e-09	-1.2e-07	-1.6e-07	3.9e-10	3.3e-10	-7.1e-12	1.6e-10	3.1e-10
p_2	-1.6e-08	3.4e-07	-2.0e-07	2.0e-08	3.5e-09	-5.8e-11	3.1e-10	4.8e-09

	f	u_0	v_0	s_u	k_1	k_2	p_1	p_2
f	1.00	0.53	0.18	-0.00	-0.66	0.82	-0.16	-0.18
u_0	0.53	1.00	0.19	0.53	-0.23	0.34	-0.26	0.14
v_0	0.18	0.19	1.00	-0.08	-0.23	0.15	-0.77	-0.18
s_u	-0.00	0.53	-0.08	1.00	0.54	-0.18	0.09	0.87
k_1	-0.66	-0.23	-0.23	0.54	1.00	-0.86	0.42	0.80
k_2	0.82	0.34	0.15	-0.18	-0.86	1.00	-0.28	-0.41
p_1	-0.16	-0.26	-0.77	0.09	0.42	-0.28	1.00	0.35
p_2	-0.18	0.14	-0.18	0.87	0.80	-0.41	0.35	1.00

Table 6.3: Conditional dispersion of the intrinsic camera parameters, when only estimated from the first image in figure 6.1. Top: covariance matrix. Bottom: correlation matrix.

parameters. The extrinsic parameters can also be conditioned on the intrinsic parameters. The result is shown in table 6.4. The position of the calibration object relative to the camera is now determined within 0.1–0.2 mm, which can be taken as a measure of the expected pose accuracy. Note, that x_0 as expected is very correlated with the rotation around the $\mathbf{y}$-axis φ.

Now, the conditioning example above, of course, is untrue to the data, since neither the extrinsic nor the intrinsic parameters can be assumed known. Furthermore, some of the variances observed in tables 6.2-6.4 are hard to interpret. What is, for instance, the significance of a given standard deviation on the focal length, a tangential distortion parameter, or an angle, in terms of measurement accuracy? And at which rate do these standard deviations decrease as the number of images is increased? To answer these questions, the total dispersion matrix, containing the covariance of the intrinsic parameters and all the poses, has been mapped into the image. The result is a dispersion matrix in u and v for each pixel in the image. This gives a visual impression of the accuracy of the camera model, and represents the uncertainties in a domain where they can be interpreted directly. Figure 6.6 shows various examples of the camera parameter dispersion mapped into the image. The ellipses represent one standard deviation, and have been exaggerated 1000 times. The numbers are the non-exaggerated standard deviation in the point $\sigma_{uv} = \sqrt{\sigma_u^2 + \sigma_v^2}$. The first row shows the dispersion resulting from the cali-

	x_0	y_0	z_0	ω	φ	κ
x_0	2.5e-03	5.1e-04	-6.5e-03	3.5e-09	2.3e-06	-1.1e-07
y_0	5.1e-04	7.1e-04	-1.7e-03	4.4e-07	6.3e-07	1.3e-07
z_0	-6.5e-03	-1.7e-03	2.3e-02	1.9e-06	-8.1e-06	2.6e-07
ω	3.5e-09	4.4e-07	1.9e-06	1.5e-08	2.0e-11	4.6e-09
φ	2.3e-06	6.3e-07	-8.1e-06	2.0e-11	7.2e-09	2.1e-10
κ	-1.1e-07	1.3e-07	2.6e-07	4.6e-09	2.1e-10	3.5e-09

	x_0	y_0	z_0	ω	φ	κ
x_0	1.00	0.38	-0.87	0.00	0.54	-0.04
y_0	0.38	1.00	-0.41	0.14	0.28	0.08
z_0	-0.87	-0.41	1.00	0.10	-0.63	0.03
ω	0.00	0.14	0.10	1.00	0.00	0.63
φ	0.54	0.28	-0.63	0.00	1.00	0.04
κ	-0.04	0.08	0.03	0.63	0.04	1.00

Table 6.4: Conditional dispersion of the extrinsic camera parameters, when only estimated from the first image in figure 6.1. Top: covariance matrix. Bottom: correlation matrix.

bration from the odd images after the first and second iteration. The standard deviation is seen to decrease by a third after the first iteration. This is expected, since the calibration residual decreases by that amount and the configuration remains unchanged. Naturally, the standard deviation is only a measure of the calibration accuracy if the presumptions hold, i.e. the CAD model is perfect and the feature detection noise is independent Gaussian. It is, however, still remarkable, that the camera can predict the projection of a point with such a small standard deviation under these presumptions. Note, how the uncertainty increases towards the border of the image field.

The uncertainty ellipses in the image can also give some information on the distribution of the calibration points in the images. The images in the second row and the first image in the third row in figure 6.6 show the dispersions caused by using only images 1–3, images 1–2, and image 1 in the calibration, respectively. Both image 1 and 2 have all the calibration points in the left-hand side of the image, but the points in image 1 are closer to the upper edge of the image. Image 3 has points in the right-hand side of the image. Note, how the uncertainty increases massively in the right-hand side of the image field as image 3 is excluded from the calibration. Omitting image 2, makes only a smaller difference, but some increase in the uncertainty is observed in the lower part of the image field.

The method can also be used to demonstrate the problem of overparametrization. The lower right image in figure 6.6 shows the result of a calibration based on image 1, where the tangential distortion has not been estimated. Remember, that the lower left image shows the same example where all parameters are estimated. Including the tangential distortion makes the calibration residual decrease from 0.11 to 0.06. Judging only from the calibration residual, this is a much better calibration. Looking instead at the image dispersion, it is seen that the standard deviation in the worst case increases by a factor of four. At the same time the best improvement is less than 70%. This emphasizes that, while improving the calibration residual, the ability to generalize from a model deteriorates as the number of parameters is increased - at least if the data does not span the added parameters.

6.4.5 Residual Analysis

It is also important to study the individual image residuals in the fitting of the camera parameters. They show the spatial error between the detected features and the optimally projected CAD model. The previous chapter gave an example, where the systematic behaviour of these residuals revealed a bias in the feature detection. Hence, the residual plot can be used to verify the validity of the noise model. Figure 6.7 shows three plots. The left plot contains the residuals in the first image after the first iteration in the calibration from the odd images. The right and bottom plots correspond to the second iteration, where the latter contains the residuals of all images simultaneously. The vectors are exaggerated by a factor 100, and the numbers show the non-exaggerated magnitude of the vector. Ideally, the plot with the residuals from all images should reveal if the the camera model is not sufficiently parametrized. However, it should be remembered that each image has six degrees of freedom in the pose to make the residuals sum to zero. This will tend to decrease the local correlation of the residuals.

Looking at the right plot, there actually seems to be some correlation between residuals. The direction of the error in the top row of features tend to follow a systematic pattern. Similar correlation can be seen in other parts of the image as well. A test statistic can better quantify the amount of correlation. The test can either be performed on a single image, or on all residuals simultaneously. The latter test is performed here to benefit from the larger amount of data, even though, as mentioned above, the pose estimations are expected to produce local anti-correlation and thereby reduce the quality of the test. Hotelling's T^2 test will be used. It is described in appendix B. The

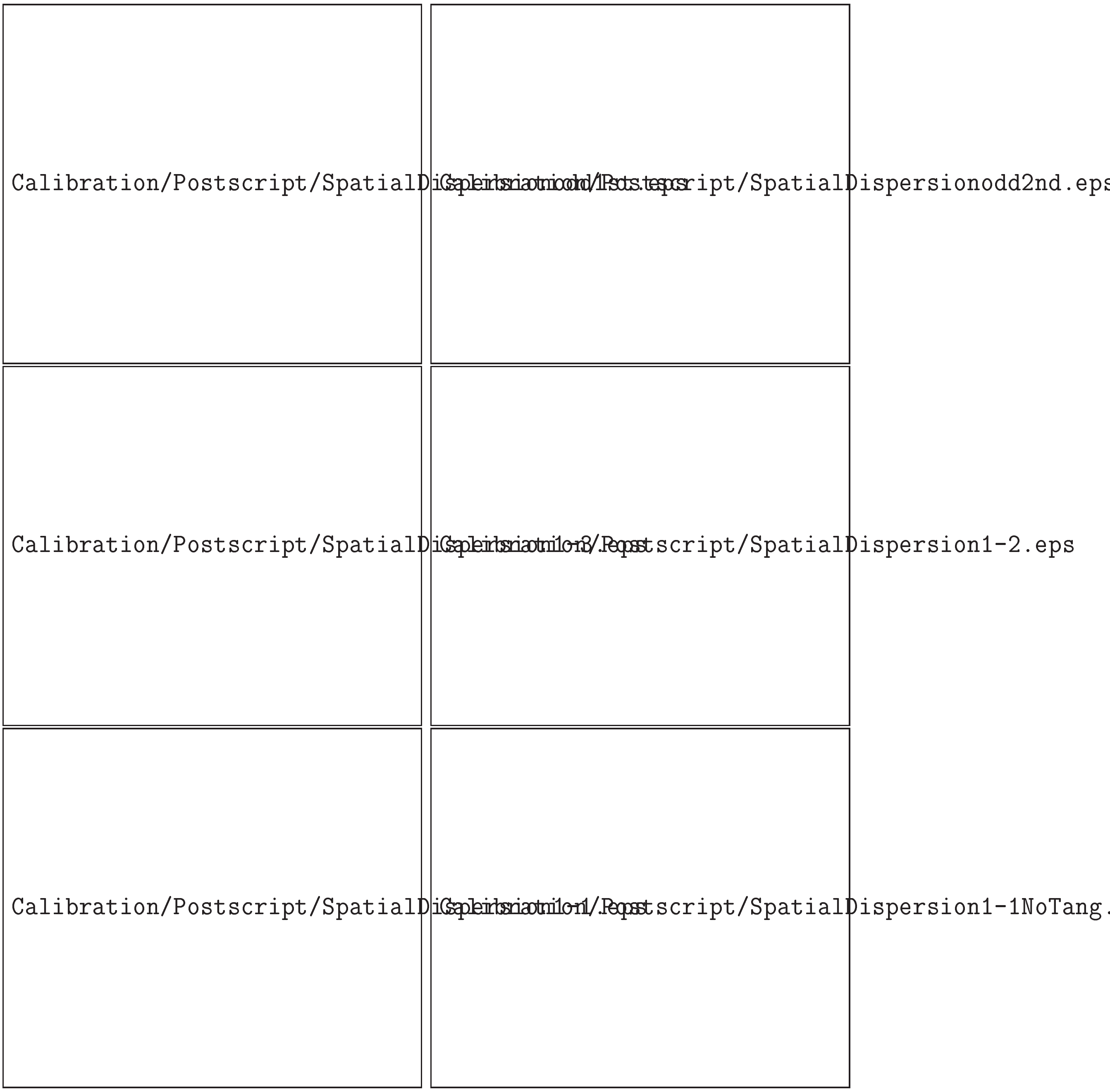
Calibration/Postscript/SpatialDispersionodd1st.eps
Calibration/Postscript/SpatialDispersionodd2nd.eps
Calibration/Postscript/SpatialDispersion1-3.eps
Calibration/Postscript/SpatialDispersion1-2.eps
Calibration/Postscript/SpatialDispersion1-1.eps
Calibration/Postscript/SpatialDispersion1-1NoTang.

Figure 6.6: The camera parameter uncertainty projected into the image plane. From the top, left to right: odd images after 1st iteration, odd images after 2nd iteration, images 1–3, images 1–2, image 1, and image 1 without tangential distortion.

Calibration/Postscript/Residual01odd1stZoom.eps

Calibration/Postscript/Residual01odd2ndZoom.eps

Calibration/Postscript/ResidualAllodd2ndZoom.eps

Figure 6.7: Calibration residuals with vectors exaggerated 100 times. Top: image 1 after 1st iteration. Middle: image 1 after 2nd iteration. Bottom: all images after 2nd iteration. In each image an area has been blown up.

Calibration/Postscript/HotellingT2OddText.eps

Calibration/Postscript/HotellingT2OddThresholdText.eps

Figure 6.8: Hotelling's T^2 test for locally correlated residuals. Top: the number of residuals in the test and the probability of the test statistic being greater than or equal to the observed value. Bottom: threshold of the upper image at significance level $\alpha = 5\%$.

test calculates the Mahalanobis distance from zero to the local mean, using the local empirical dispersion matrix. The test statistic is calculated in 15×21 rectangles and is based on all the residuals within a radius of $40\,pixels$. The upper image in figure 6.8 shows the probability of a test statistic being greater than or equal to the observed value. Each rectangle also shows a number specifying how many residuals were used to calculate the statistic. The grey level is equal to the probability, so that one can threshold in the image to see the rejection rate for different significance levels. The cyan areas contain too few points to perform the test. All in all the image contains 276 tests.

As argued in the appendix, the significance level should reflect the number of performed tests. It is not reasonable to expect no rejections of the null-hypothesis on a significance level of $\alpha = 5\%$ when doing 276 tests. One can then either calculate the appropriate significance level, $\alpha_{276} = 1 - 0.95^{1/276} = 2e - 04$, or compare the number of rejections for $\alpha = 5\%$ to the expected rate. With respect to the first test it is immediately seen, that no probability in the test comes near $2e - 04$, meaning that this test does not reject the null-hypothesis, that the local residual mean is zero. The latter test is preferred here as it also gives some information on the neighborhood sensitivity to the test. The bottom image in figure 6.8 shows a threshold at a significance level of $\alpha = 5\%$. It is seen that the null-hypothesis is rejected in 20 out of 276 tests. If the mean was indeed zero, we would expect a rejection rate close to the significance level, i.e. 14 rejections. The standard deviation would be $0.05 \cdot 0.95 \cdot 276 = 13$. The observed number of rejections is therefore within one standard deviation of the mean. In a calibration with the even images the result was 34 rejections in 273 tests, which is somewhat closer to a rejection of the null-hypothesis. Even though none of the tests rejects the null-hypothesis, it is noted that they give large test statistics. Especially because the pose estimates are expected to favour the null-hypothesis and because there seems to be some correlation in the right image in figure 6.7.

In relation to the discussion in section 6.4.1, it is interesting if there is some pattern in the residuals. If the calibration residual is mainly determined by the manual placement of the calibration circles, the residuals would be expected to be stochastic in the (x, y)-plane. The exception is the situation, where one or a few outliers cause a strong correlation of all the other residuals, which seems not to be the case here. It is therefore more likely that the calibration residual is caused by something else. Two possibilities are imaginable. Either the calibration object is slightly curved, so that the observed residuals are caused by parallax, or the lens distortion is more complex than what can be captured by the radial and decentering parameters. I believe that either possibility is equally probable. In relation to the former, the pose reveals that the plate is angled 35 *deg.* with respect to the image plane in image 1. That is, a parallax of 0.2 *mm* can be caused by a misplacement of the calibration feature of only 0.3 *mm* in the Z-direction. The latter possibility also seems likely, since the calibration in the example is based on a high distortion 8 *mm* lens. In fact, the magnitude of the distortion function is as large as 60 *pixels*, which is two orders of magnitude greater than the calibration residual.

6.5 Selection of Images

The calibration in the example above was performed on 26 images with 25 additional images in the test set. It is very time-consuming to acquire that many images. The camera is moved around on a tripod, and the camera and light settings sometimes need to be changed when moving from one view angle to another. It is therefore desirable to limit the number of images to an acceptable minimum. The tools discussed in this chapter give a basis for the selection of calibration images, in terms of their number and their nature. It is not possible to give explicit guidelines, since the choice depends on the needed accuracy and the specific camera. Generally, the images should span the variability in the camera model to limit the correlation among the intrinsic parameters. Otherwise, there is a chance that the correlated parameters are optimal for the calibration images, but give poor results with future images. This is why we have acquired the images from different angles and different distances. As stated earlier, it is also important that the calibration features are distributed evenly in the image field. This ensures that the optimized parameters are valid in all but the border of the image field. When calibrating the Basler cameras with low distortion lenses, we typically use about 15 images.

6.6 Calibration Constancy

When calibrating a camera, it is of course also of interest how long the calibration will remain valid. At the yard, we fix the focus and aperture settings with tape before we begin calibration. Furthermore, heated camera houses are used in the outdoor installations to limit the effect of temperature changes. No thorough tests have been performed to verify the constancy of the calibrations, but the parameters have been seen to perform well over six-month periods. There is no reason to believe that the calibrations should not be valid for even longer periods.

In most of the applications at the yard we perform an on-site pose calculation for every image, as the cameras are mounted on large structures that move as a result of variations in temperature and mechanical loads. Since the perspective effect is typically small in these images, the pose calculation can actually compensate for many small variations in the intrinsic parameters, cf. table 6.2. The parameter variations that are most difficult to compensate by the pose parameters are aspect ratio and non-linear lens distortions. However, these parameters are also the least likely to change over time. Consequently, the on-site calibration induced by the pose calculation, helps to prolong the validity of the camera model.

6.7 Conclusion

This chapter has presented a calibration method based on coplanar calibration data. More importantly, the calibration has been analyzed by use of a number of different tools. It has been shown that the image residuals or the total calibration residual are not sufficient measures to validate a calibration. The parameter dispersion and its projection into the image field gave additional information on the camera model uncertainty, and provided a tool to choose the number of parameters and the combination of images. A test calibration was performed resulting in a calibration residual of a tenth of a pixel. The majority of the residual is expected to originate either from the severe distortion in the lens or from an imperfect CAD model of the calibration object. Nevertheless, the calibration residual is by far small enough for the required accuracy in the future measurements with the camera.

The academic and industrial partners in the Esprit project **RobVision** have agreed to use the developed calibration software in the project.

Chapter 7

Feature Extraction

7.1 Introduction

The previous chapters have dealt with the understanding and the modelling of the geometric relationship between the observed world and the image projection. These results will naturally be used to derive information about the world from detected image features. This chapter will present the theories and methods that have been used for feature detection in the present project.

The first section will address the problem of accurately estimating the position of reference marks in the image. Reference marks are designed to be estimated with high precision in the image and are used for internal as well as for external calibration of the camera. The second section will discuss the problems of edge detection, which is used for the actual measurements in the scene.

7.2 Circular Features

Reference marks are placed in the scene to establish a link between the camera coordinate system and the local coordinates in the work cell. This way the measurements can be fed back to e.g. the manipulating tool. Another use is in the field of calibration, which was shown in the previous chapter.

Besides a possible limit on the size of the reference marks, they can be designed freely. It is therefore attempted to make the marks easy to estimate with a high accuracy. At the shipyard we have chosen to use binary circular marks, i.e. black and white marks with one or more concentric circles.

Rectangles or some other design could equally well have been chosen, but the isotropy of the circular marks make them easy to work with.

The position of a mark in an image can be estimated in various ways. For instance, the image can be correlated with a template that represents the mark. The problem is that the size and shape of the marks vary a lot as a function of the point of view, and that a difference between the image and the template will produce a bias in the position estimate. It is therefore necessary either to parametrize the template or to transform the image to make the appearance of all marks similar. The latter is possible because the approximate pose is always known in a stationary industrial installation. Using this *a priori* knowledge is very appealing as it increases the robustness of the estimation. However, neither solution is well suited for correlation. Parametrizing the template implies a very large search space that makes correlation very slow. Transforming the image deforms the raster, which also makes correlation difficult. Another method that does not suffer these problems, is to fit a parametric model to the data. This approach will work on the transformed image, and the parameters in the model give quantitative information about the estimation accuracy. I have therefore chosen this solution. Various parametric models have been described in the literature, cf. *Rohr* [64], *Valkenburg et al.* [74] and *Blaszka and Deriche* [4]. The latter paper presents an edge model that I have modified for the present application. The model is described in the next section.

7.2.1 Model

As stated above, the model is based on the assumption that the reference mark is isotropic with known size. This situation is obtained by back-projecting all the pixels to a metric coordinate system using the approximate pose of the camera and the known object space location and orientation of the reference mark. The involved equations are described in section 5.2.1, page 44. The size and shape of the mark does therefore not need to be modelled. The parameters that we do need to estimate are the center coordinates $\mathbf{x}_c$, the image intensities of the black and white phases (i_b, i_w), and a value α that models the image smoothing. The model presented here contain one white disc on a black background, but it is easily extended to several concentric circles. The image intensity $i(\mathbf{x})$ at position $\mathbf{x}$ is modelled as

$$i(\mathbf{x}) \quad = \quad i(r) \qquad (7.1)$$

$$= \quad (i_w - i_b)\phi(r) + i_b, \tag{7.2}$$

$$\phi(r) = \begin{cases} 1 - \frac{1}{4}e^{\alpha(r-r_0)}(2 - \alpha(r-r_0)) & r \leq r_0 \\ \frac{1}{4}e^{-\alpha(r-r_0)}(2 + \alpha(r-r_0)) & r > r_0 \end{cases} \tag{7.3}$$

$$r = |\mathbf{x} - \mathbf{x}_c|, \tag{7.4}$$

where r_0 is the radius of the white disc. The radial edge 7.3 follows the model described by *Blaszka and Deriche* [4]. The edge is approximately modelled as a Gaussian step edge and the model has a simple derivative

$$\frac{\partial}{\partial r}\phi(r) = -\frac{1}{4}\alpha e^{-\alpha|r-r_0|}(1 + \alpha|r - r_0|) \tag{7.5}$$

The value of α can be chosen to give the approximation the same energy as the Gaussian

$$\alpha = \frac{5}{2\sigma\sqrt{\pi}} \tag{7.6}$$

This relationship gives some intuition about the magnitude of α.

The derivatives of the model can be calculated in closed form and the model is easily fitted to the data. The Levenberg-Marquardt algorithm is used to minimize the least squares error in the fit. The initial values for the parameters can be chosen very robustly. A good guess for the maximum and minimum intensities can be obtained directly from the data, the center is approximately known from the approximate pose, and the image smoothing typically corresponds to $\sigma \approx 1.0$. Once an estimate of the position of the center is obtained, it can be projected back into the image to yield the estimate in image coordinates. The procedure can naturally be repeated, if the new pose is very different from the approximate pose, but this is seldom the case.

There is one objection that can be made against this approach. The smoothing induced by the camera system is typically uniform in the image. When the image is mapped to a metric coordinate system, the smoothing will no longer be isotropic. This is not considered in the model above. It should not, however, seriously affect the estimation since every mark is fitted individually and locally the anisotropy is only modest.

7.2.2 Trend in Illumination

In certain applications it was noticed that the reference marks were unevenly illuminated. This was typically the case during calibration, where a temporary light source was pointed towards the calibration object. This produces a strong illumination of the center of the image that falls off towards the borders. If this kind of illumination causes a bias in the estimation, the bias will probably have the same nature as the lens distortion. Consequently, the bias will be absorbed in the camera model without any warning in the calibration residual and the model will produce systematic errors in future measurements.

The effect of varying illumination has therefore been investigated. Twenty synthetic images have been constructed with different amounts of trend in the illumination across the reference mark, cf. figure 7.1. The trend has been modelled by a multiplicative factor inspired by empiric data. This implies a higher reflectance from lighter surfaces. The model in equations 7.1-7.4 is then extended with

$$i_{ill.}(\mathbf{x}) = (1 + \beta \cdot (\mathbf{x} - \mathbf{x}_c)))\ i(\mathbf{x}), \tag{7.7}$$

where β is a vector in the direction of the trend. The images in figure 7.1 all have a trend with $\angle\beta = -\pi/6$. As the v-axis points downwards, the direction of the trend is from bottom-left to top-right. The magnitude of the trend is $|\beta| = 0.0000, 0.0003, \ldots, 0.0057$. The sizes of the images are 160×160, so that $|\mathbf{x} - \mathbf{x}_c| < 120$. The radius of the reference marks is $50\,pixels$ and the black and white intensity levels are 50 and 200, respectively. The images have been smoothed with a Gaussian of width 2.0. The reference marks in the images have been detected without estimation of the trend. That is, the test shows the bias introduced in the estimation by neglecting the trend in the illumination. Figure 7.2 shows the displacement of the center caused by the trend in illumination. The direction of the bias is shown in figure 7.3 where the theoretical direction of the trend is also plotted. It is obvious that the detection is biased in the direction of the trend. In real images, the magnitude of the trend in illumination has been estimated to be as large as $|\beta| = 0.002$.

7.2.3 Detection Example

This section will show an example of a circle detection that uses the model above. As earlier seen the very fitting of the model yields information about the parameter accuracy. Furthermore, the interpretation of the estimated parameters gives a hint about the image quality. The reference mark to the left in figure 7.4 is detected in the present example. This mark is actually the third mark from the right in the bottom row of the fifth image in figure 6.1

Figure 7.1: Twenty images with different amounts of trend in the illumination across the reference mark.

Figure 7.2: The magnitude of the bias introduced by the trend in illumination.

Figure 7.3: The direction of the bias introduced by the trend in illumination. The direction of the actual trend is plotted as a dotted line.

on page 53. The radii of the three concentric circles are 6 *mm*, 30 *mm*, and 60 *mm*. The projection of the largest circle has a radius of approximately 30 *pixels*. As suggested above, the model parameters are estimated in the metric coordinate system, where the perspective effect and the lens distortions are removed. The direction of the horizontal x-axis and the vertical y-axis and the estimated center are indicated in the image by white lines. The fitted model and the residual image are shown to the right in figure 7.4. The residual shows that the model does not produce a perfect fit on the edges, but it is also seen that the error is well distributed around the mark. That is, a light shadow, indicating that the image is lighter than the model, is always matched with a light shadow on the opposite side of the mark. The same is true for the dark shadows.

The estimated parameter values are shown in table 7.1. The center coordinates of the mark are of course the most important parameters, but valuable information can also be derived from the other parameters. The dark and light phases of the mark have been estimated to have the intensities 10 and 143, respectively. This shows that the projected mark does not suffer from black level cut-off or saturation, cf. chapter 3. The image smoothing is estimated to

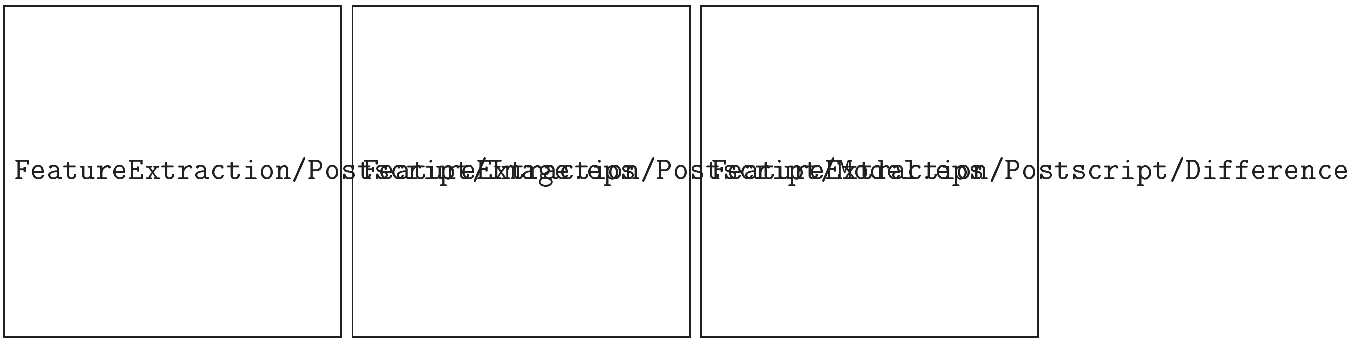

Figure 7.4: Example of circle detection. Left: original image with horizontal x-axis and vertical y-axis. Middle: the fitted model. Right: the difference between the model and the image, i.e. the fitting residual. The artifacts on the outer radius of the mark are the line segments that were used to position the mark on the graph paper.

be equivalent to a Gaussian smoothing with $\sigma = 1.3\,mm \approx 0.7\,pixel$. This is within the expected range, and shows that the camera is well focused. Finally, the magnitude of the trend in illumination is found to be small. Nevertheless, the direction of the trend is in good agreement with the light variation in the total image on page 53. The bottom row of the table shows the fitting residual, which is 4% of the difference between black and white. The estimated center is seen as the crossing of the axes in the original image in figure 7.4, and the result is in good agreement with the visual impression.

As in the calibration example, the dispersion of the estimated parameters can be derived from the Jacobian of the model. The covariance and correlation matrices for the parameters are shown in table 7.2. The way the model is formulated, the parameters are not expected to be particularly correlated. This is also seen in the correlation matrix. The largest correlations are found between the center position and the β-vector, reflecting the relationship that was also shown in figures 7.2 and 7.3. The position estimate is nicely uncorrelated with the other parameters, i.e. the intensity and smoothing factor estimates have very little influence on the position. Accordingly, the dispersion of the center estimate conditioned on the rest of the parameters, which is not presented here, is not very different from the un-conditioned dispersion.

When you look at the magnitude of the different variances it is seen that all parameters are estimated rather accurately. The black and white phases are

Parameter	Estimated Value
x_c $[mm]$	1.02
y_c $[mm]$	-2.23
u_c $[pix]$	32.06
v_c $[pix]$	31.50
i_b $[intensity]$	9.63
i_w $[intensity]$	143.49
α $[mm^{-1}]$	1.06
$\sigma(eqn.\ 7.6)$ $[mm]$	1.33
β_x	-3.69e-05
β_y	4.34e-04
Fitting Residual $[intensity]$	5.40

Table 7.1: Estimated circle model parameters.

estimated with a standard deviation of ±0.2 intensity levels and the standard deviation of the smoothing factor is a hundredth of a pixel. Most important, the position estimate has a standard deviation of $0.02\,mm \approx 0.01\,pixel$. The quality of the theoretically derived dispersion matrix compared to the actual dispersion is shown in the next section, where a few design criteria are given for reference marks.

7.2.4 Design

The obtainable accuracy in the estimation of a reference mark depends on the design, i.e. the intensity, size, and shape of the mark. It is interesting to investigate this dependency, as it allows us to optimize the design. The discussion in this section will be limited to binary circular marks. The question is how the accuracy depends on the number and size of the concentric circles. In appendix B it is shown that the dispersion of the parameters in a fitted parametric model can be estimated from the variance of the image noise. The link between the two is the Jacobian of the parametric model. If we only consider the accuracy of the position estimate $\hat{\mathbf{x}}_c$ and assume the image noise to be independent and identical with variance σ, we have

$$\mathbf{D}(\hat{\mathbf{x}}_c) = \sigma^2 \left(\sum_{\mathbf{x}} \nabla i(\mathbf{x}) \nabla i(\mathbf{x})^T \right)^{-1}, \tag{7.8}$$

	x_c	y_c	i_b	i_w	α	β_x	β_y
x_c	1.6e-04	1.4e-07	-1.3e-05	5.9e-05	-6.1e-07	1.6e-07	-6.1e-10
y_c	1.4e-07	1.6e-04	3.3e-05	-1.1e-04	2.0e-06	-5.2e-10	1.5e-07
i_b	-1.3e-05	3.3e-05	1.8e-02	-6.9e-03	4.9e-04	-1.3e-07	2.7e-07
i_w	5.9e-05	-1.1e-04	-6.9e-03	3.8e-02	-8.7e-04	7.5e-07	-1.4e-06
α	-6.1e-07	2.0e-06	4.9e-04	-8.7e-04	9.2e-05	-1.2e-08	2.6e-08
β_x	1.6e-07	-5.2e-10	-1.3e-07	7.5e-07	-1.2e-08	1.5e-09	-2.2e-11
β_y	-6.1e-10	1.5e-07	2.7e-07	-1.4e-06	2.6e-08	-2.2e-11	1.5e-09

	x_c	y_c	i_b	i_w	α	β_x	β_y
x_c	1.00	0.00	-0.01	0.02	-0.01	0.32	-0.00
y_c	0.00	1.00	0.02	-0.04	0.02	-0.00	0.31
i_b	-0.01	0.02	1.00	-0.26	0.37	-0.03	0.05
i_w	0.02	-0.04	-0.26	1.00	-0.46	0.10	-0.19
α	-0.01	0.02	0.37	-0.46	1.00	-0.03	0.07
β_x	0.32	-0.00	-0.03	0.10	-0.03	1.00	-0.02
β_y	-0.00	0.31	0.05	-0.19	0.07	-0.02	1.00

Table 7.2: Dispersion of reference mark parameters for the mark in figure 7.4. Top: covariance matrix (values are given in millimeters, $1\,pixel \approx 2\,mm$). Bottom: correlation matrix.

where ∇ is the gradient operator. The expression shows that reference marks are better estimated in the direction where they have more edges. For instance, an ellipse will be located with smaller variance in the direction of the shorter axis. If we assume that the mark is approximately isotropic, i.e. the diagonal elements of the matrix are equal in size, we can use $tr(\mathbf{D}^{-1})$ as a measure of the certainty of the estimation. The certainty is then expressed as

$$tr(\mathbf{D}^{-1}) = \frac{1}{\sigma^2} \sum_{\mathbf{x}} |\nabla i(\mathbf{x})|^2 \tag{7.9}$$

We see that the certainty is proportional to the sum of squared gradients in the model. In fact, the certainty is proportional to the edge length in the binary case, where all gradients have the same magnitude. It is therefore clear, that the reference mark should contain as much edge as possible. This also implies that the concentric circles should be separated to such a degree that the inevitable image smoothing does not weaken neighboring edges. Similarly, the edge gradient should be as large as possible, i.e. the difference in intensity between the black and white phases should be maximized. The result has been tested on nine reference marks of different sizes. Table 7.3 shows the radii of three concentric circles for each of these marks. The table also shows the total edge length. The actual reference marks are shown in figure 7.5. The position of each of the marks has been estimated and the theoretical dispersion of the estimate has been calculated. The certainty measure in equation 7.9 has then been determined from the dispersion matrices and is plotted as a function of the edge length in figure 7.6. A linear regression on the data gives a multiple correlation coefficient of $R^2 = 0.9866$. Equation 7.9 shows that the proportionality factor is linear in the additive image noise variance and the squared difference between the black and white intensities. This result agrees with *Åström's* [2] analysis of one-dimensional edge detection. The linear regression is of course particularly successful, because the theoretical dispersion has been used instead of an empiric dispersion. For comparison, an empiric dispersion has been calculated from fifteen images with different realizations of the additive noise. The empiric certainty is also shown in the figure. It is seen to be somewhat larger than the theoretical certainty. This might be a result of aliasing effects in the synthetic images. Furthermore, a linear regression will give a poor fit on the empirical data, but this is mainly because of the sensitivity of the inverse to small numbers. Plotting instead the total variance,

Mark	Radius 1	Radius 2	Radius 3	Total Circumference
1	4.5	10.5	16.5	197.9
2	8.5	14.5	20.5	273.3
3	12.5	18.5	24.5	348.7
4	16.5	22.5	28.5	424.1
5	20.5	26.5	32.5	499.5
6	24.5	30.5	36.5	574.9
7	28.5	34.5	40.5	650.3
8	32.5	38.5	44.5	725.7
9	36.5	42.5	48.5	801.1

Table 7.3: The radii of three concentric circles for each of nine reference marks. The right column shows the total edge length.

which is the trace of the dispersion matrix, we see very similar behavours for the theoretical and empirical data, cf. figure 7.7.

The conclusion is that the reference mark should contain as much edge as possible, but separated so much that edges do not cancel in the projected image. The analysis shows that a reference mark with e.g. three concentric circles with radii 1, 2, and 3 is located with the same certainty as a mark with one circle with radius 6, despite the fact that the area of the latter is four times greater.

`FeatureExtraction/Postscript/circles.eps`

Figure 7.5: Different size circles to test the relation between the total circumference and the detection accuracy.

7.3 Straight Edge Detection

All the objects that are to be measured at the shipyard have linear edges or can be modelled to be piece-wise linear. It is therefore important to be able to measure straight edges in images with high accuracy. Many algorithms

FeatureExtraction/Postscript/sizeregression.eps

Figure 7.6: The detection certainty as a function of the total circumference. The linear regression, $tr(\mathbf{D}^{-1}) = 290.75 \times Circumference$, gives a multiple correlation coefficient $R^2 = 0.9866$.

Figure 7.7: The total detection variance as a function of size.

already exist for this purpose. They can be divided into parametric and non-parametric methods. The parametric methods are typically based on a model of the image intensity variation across the edge. This model can then be fitted to the data globally, as was seen in the reference mark example above. The advantage of these methods is that all the image information is used simultaneously to obtain a globally optimal edge estimate. The disadvantage is that the observed data need to conform to the model globally in order for the estimate to be unbiased. The non-parametric methods typically estimate the local edge location from the magnitude of the image gradient. A global estimate is then obtained subsequently by fitting a straight line to the edge data using e.g. the Hough transform or regression techniques. The advantage of this method is that the presumed model only needs to be valid locally. For instance, the illumination and the background can vary along the edge. Furthermore, the user only needs to specify a small number of parameters, such as the size of the gradient filter and a threshold value to discard outliers. The disadvantage is that the local image structure, apart from the edge location and perhaps the edge strength, is ignored in the final estimate. Consequently, a local edge that is corrupted by structured noise, might be discarded in the global estimate or might even affect the estimate in a bad direction, even though part of the local image structure supports the global estimate.

The edges that are used for measurement at the shipyard are characterized by varying illumination and structured noise in the vicinity of the edge. The noise in the foreground is typically burned edges or rust on the steel plates. In the background it can be grids or spikes underneath the plates on the cutting tables. Since this structured noise cannot be modelled, it is impossible to apply conventional parametric models to these edges. On the other hand, it is also problematic not to perform a global estimate, as the available local edge information might be lost when using the non-parametric methods, because of the strong influence of the local noise. It has therefore been attempted to design an edge detection method, that is non-parametric and optimize a global measure. The different methods that have been investigated are described in section 7.3.2 below, and a comparison to more conventional methods is performed in section 7.3.3. The next section will introduce some literature on the subject of edge detection.

7.3.1 Literature

Edge detection has attracted much interest since the early days of computer vision, owing to the informational content of edges in images. Much of the early research focused on the problem of extracting some sort of structure from totally unknown images. That is, the methods were designed to give a strong, single, and robust response to edges without any a priori knowledge. The pioneering work was the paper by *Canny* [6], where a Gaussian-like edge detector was proved to satisfy a number of optimality criteria. *Deriche* [12] has devised a fast recursive implementation of the Gaussian convolution to make the approach efficient. The statistical properties of the edge detection has been further analyzed by *Åström et al.* [3, 2]. Their analysis covers both the one-dimensional and the two-dimensional edge. *Harris and Stephens* [21] have described a combined corner and edge detector, that is also based on the local structure of the image, using linear scale-space operators. *Devernay* [13] has addressed the practical problem of obtaining a high accuracy with the methods based on linear convolution. His paper describes non-maxima suppression on sub-pixel level. Other papers have introduced different parametric models for edge detection. An example is the model by *Blaszka and Deriche* [4] used in the beginning of this chapter. Methods for sub-pixel detection of corners and edges with parametric models have also been treated by *Rohr* [64] and *Valkenburg et al.* [74].

7.3.2 Methods

Different methods have been used for edge detection throughout the project. These methods are presented in this section with some comments on their speed and accuracy. The performance on synthetic data is presented in a comparison study in the next section. The different algorithms that were used have been named

1. Brute Force
2. Smoothed Hough Transform
3. Normalized Gaussian Convolution on the Projected Image
4. Edge Fitting with Linear Regression
5. Gaussian Convolution with Linear Regression

The algorithms are presented in the order in which they were implemented in the project. The first two methods were extremely slow, but that is only occasionally a problem at the yard. Typically, the throughput on an installation is one or two plates per hour, such that there is plenty of time to process the acquired images.

Brute Force

The brute force method is based on the magnitude of the image gradient. A well-defined measure for the gradient of the discrete image $i(\mathbf{u})$ is found by Gaussian convolution [12]

$$\begin{aligned} \nabla_g i(\mathbf{u}) &= \nabla(g(\mathbf{u},\sigma) * i(\mathbf{u})) && (7.10) \\ &= \nabla g(\mathbf{u},\sigma) * i(\mathbf{u}) && (7.11) \end{aligned}$$

where σ is the width of the Gaussian function $g(\mathbf{u},\sigma)$. The brute force method aims at finding the line l in the image that covers the most gradient

$$l = \arg\max_{l \in L} \int_{\mathbf{u} \in l} \mathbf{n}_\theta \cdot \nabla_g i(\mathbf{u}) d\mathbf{u}, \tag{7.12}$$

where $\mathbf{n}_\theta = (\cos\theta, \sin\theta)$ is the normal vector to the line l and L is some set of candidate lines. As in the case of the Hough transform, L is represented by an array indexed by the normal parameters of the line (ρ, θ), so that the line is defined by

$$\begin{aligned} \rho &= u\cos\theta + v\sin\theta && (7.13) \\ &= \mathbf{n}_\theta \cdot \mathbf{u} && (7.14) \end{aligned}$$

The approach is to sample and sum up the image gradients at some density along all the candidate lines. The edge location is defined by the cell in the array L, that gave the highest count. A super-resolution value for the maximum can be obtained by a parabolic fit in the array. This is, of course, a very primitive and slow approach. It was only used in the very beginning of the project. Nevertheless, the method is robust and finds the edge with a high precision if the image space and the line space is sampled sufficiently dense. The accumulated array is not shown here as it is visually similar to the smoothed Hough space presented in the next section.

Smoothed Hough Transform

The problem with the brute force approach is that each image pixel is sampled a large number of times. This is particularly severe because the method seeks a ridge in the gradient image, and therefore has to employ bicubic interpolations, that are based on sixteen pixels for each sample. The computationally cheaper bilinear interpolation cannot detect ridges with sub-pixel precision, because the maximum image intensity will always be found on the image raster. The remedy is to use the available pixels as starting point for the accumulation instead. This is what the usual Hough transform does. It is, however, normally based on a subset of the image pixels that have been detected to be local edge candidates.

The paper *"From Hough Transform to Radon Transform using Interpolation Kernels"* at the back of the dissertation shows how the grey-level Hough transform, based on all image pixels, can be used to detect edges in images. A typical problem concerning the Hough space is a severe aliasing caused by the discrete raster of the input image. This aliasing, which is worse for a super-resolution Hough space, makes the peak detection very difficult. The paper shows how the effect of continuous interpolation in image space can be modelled in Hough space to remove the aliasing. The result is shown for nearest neighbor, bilinear, and isotropic/anisotropic Gaussian interpolation. It is also shown that the image derivative can be modelled in Hough space as a one-dimensional filter, opposed to the two-dimensional version that must be used in image space.

The smoothed Hough space can actually be regarded as the Radon transform of a continuously interpolated image. The theory therefore provides a means to analyze one-dimensional orthogonal projections of images in high resolution. The comparison study below will show that other edge detection methods are superior in terms of speed and accuracy. Especially, in the presence of systematic noise, i.e. outliers, where the image structure is not really one-dimensional. Nonetheless, the theory provides a good understanding of the structure of the Hough space, and will most certainly be useful in other applications.

Normalized Gaussian Convolution on Projection

The two methods described above are both rather slow, as lines or curves are sampled densely in either image space or Hough space. Furthermore, the accuracy of the edge detection is limited by the chosen density of the accumulator array. It was therefore attempted to find a function of the line parameters that could be maximized by a normal iterated optimization method without sampling the image pixels too many times. This function should preferably be independent of the particular edge profile, i.e. it should *not* be a parametric model of the edge. It was decided to see if the normalized Gaussian convolution could be used for this purpose. The theory of this interpolation approach is described by *Knutsson and Westin* [41]. It is useful for interpolating irregularly sampled data or data with missing samples. In the one-dimensional case the interpolated value is calculated as

$$\begin{aligned} i(x) &= \frac{\sum_{i=1}^{n} i(x_i) g(x - x_i, \sigma)}{\sum_{i=1}^{n} g(x - x_i, \sigma)} && (7.15) \\ &= \frac{i(x) * g(x, \sigma)}{i^0(x) * g(x, \sigma)}, && (7.16) \end{aligned}$$

where $i(x_i)$, $i = 1, \ldots, n$, are the available data, $g(x, \sigma)$ is the Gaussian kernel of width σ, and $i^0(x_i)$ is the function with value 1 in all the available sample positions. The advantage of the normalized convolution compared to the normal convolution is that the interpolation does not drop in the vicinity of missing data. An example is shown in figure 7.8. It is seen that the interpolated value obtained by the normalized convolution is controlled by the closest available datum. This is a necessary requirement if a non-linear optimization scheme is used to detect an edge in the interpolated image. Otherwise, the optimizer would tend to converge towards the edge between available and non-available data instead of the edge inside the data.

FeatureExtraction/Postscript/NormalizedConvolution.eps

Figure 7.8: Normalized Gaussian convolution. The interpolations are based on the imperfect data. The normal Gaussian convolution falls off, while the normalized convolution is controlled by the closest available datum.

The normalized Gaussian convolution can be used to interpolate the orthogonal projection $i^p(\rho, \theta)$ of the image described by the edge parameters ρ and θ. That is, the projected image value is estimated as

$$\begin{aligned} i^p(\rho, \theta) &= \frac{\sum_{i=1}^{n} i(\mathbf{u}_i) g(\rho - \mathbf{n}_\theta \cdot \mathbf{u}_i, \sigma)}{\sum_{i=1}^{n} g(\rho - \mathbf{n}_\theta \cdot \mathbf{u}_i, \sigma)} && (7.17) \\ &= \frac{\mathbf{G}(\rho, \theta) \cdot \mathbf{I}}{\mathbf{G}(\rho, \theta) \cdot \mathbf{1}} && (7.18) \end{aligned}$$

To simplify the notation, the values of the image and the Gaussian function have been gathered in the vectors $\mathbf{G}(\rho, \theta)$ and $\mathbf{I}$, respectively. The vector $\mathbf{1}$

consists of ones. Now, the edge position is assumed to be the projection with the largest gradient with respect to ρ. That is, we seek to maximize

$$\begin{aligned} i^p_\rho(\rho, \theta) &= \frac{(\mathbf{G} \cdot \mathbf{1})(\mathbf{G}_\rho \cdot \mathbf{I}) - (\mathbf{G}_\rho \cdot \mathbf{1})(\mathbf{G} \cdot \mathbf{I})}{(\mathbf{G} \cdot \mathbf{1})^2} \qquad (7.19) \\ &= \frac{\mathbf{G}_\rho \cdot \mathbf{I}}{\mathbf{G} \cdot \mathbf{1}} - \frac{\mathbf{G}_\rho \cdot \mathbf{1}}{\mathbf{G} \cdot \mathbf{1}} \cdot \frac{\mathbf{G} \cdot \mathbf{I}}{\mathbf{G} \cdot \mathbf{1}} \qquad (7.20) \end{aligned}$$

It is emphasized that this method is based on an interpolation of the projected data, and not on a projection of the interpolated data. This circumstance makes it difficult to analyze the properties of the maximum analytically. The empirical properties are analyzed in the comparative study below. In terms of speed, the method is rather fast. The sums that need to be calculated in equation 7.20 have many common terms and the optimization converges in 10–20 iterations.

Edge Fitting with Linear Regression

The global edge detection methods described above have been compared to more conventional approaches, where a line is fitted to a number of locally detected edges. To avoid image interpolation and to speed up the calculations, the local edge detections are performed in the raster direction. That is, a one-dimensional edge detection is performed in each column for edges with a small slope. If the edge has a large slope, the edge detections are performed in each row. The optimal edge position can then be found by normal regression techniques, cf. section 7.3.2.

The first one-dimensional edge detector, that has been investigated, is based on the same model that was used for circle detection in section 7.2.1 above. That is, the edge is modelled by an approximated Gaussian step function. The parameters of the step function is the position of the step u_c, the degree of smoothing α, and the image intensities i_b and i_w of the background and the foreground. The illumination trend is not modelled here.

Gaussian Convolution with Linear Regression

The other one-dimensional edge detector that has been used is the normal Gaussian edge detector. That is, the image is convolved with the derivative of the Gaussian function, cf. equation 7.11. This is probably the most popular edge detector in the vision community. The convolution has been performed on pixel resolution and the sub-pixel edge location has been determined by a parabolic fit based on the pixel with the largest output and its two neighbors. If the width of the Gaussian is $\sigma = 1$, the filter can safely be truncated to $[-3; 3]$, as $g_u(4, 1) < 1.0e - 05 \times g_u(1, 1)$.

Regression and Outlier Detection

The results of the one-dimensional edge detections described above are used in a normal regression to find the optimal edge location

$$v_i = \alpha + \beta u_i + \epsilon_i \tag{7.21}$$

The edge is parametrized by v if its orientation is vertical. It is well-known that normal least squares estimation is sensitive to outliers. It is therefore advisable to do some sort of outlier detection. The different approaches that have been tried are

1. Least Squares
2. Least Squares ($\epsilon < \sigma$)
3. Least Squares ($\epsilon < \sigma$, 50%)
4. Re-weighted Least Squares

The first one is the normal least squares fit with no outlier detection. The second method is based on omitting the data that give rise to residuals ϵ that are greater than one standard deviation σ. As the standard deviation is a function of the least squares fit, the regression must be iterated a few times. In the tests below the outlier detection was performed twice. The third method is also based on omitting data that cause residuals greater than one standard deviation. However, the standard deviation is calculated from the 50% of the data, that have the smallest residuals. This is done in only one iteration. The final method implements one of the so-called M-estimators by re-weighted least squares, cf. *Zhang* [78]. The chosen M-estimator is denoted “Fair” and minimizes a function of the residuals that increases slower than the square

$$\rho(\epsilon) = c^2(\frac{|\epsilon|}{c} - \log(1 + \frac{|\epsilon|}{c})), \tag{7.22}$$

where $c = 1.3998$ is a tuning constant. It is implemented by iterative calculation of the weights

$$w(\epsilon) = \frac{1}{1 + \frac{|\epsilon|}{c}}, \tag{7.23}$$

that are used in the least squares estimation.

7.3.3 Comparison

The edge detection approaches described above have been tested on four synthetic images. The images are shown in figure 7.9. All images have dimension 200×200 and show an edge going from intensity level 50 to 150. The slope of the edge is $-1/10$ in the coordinate system of the images, and the edge passes through the point $\mathbf{u} = (150, 80)$. The edge is modelled as a Gaussian step edge with the width $\sigma = 1$. Images 1 and 3 have no additive noise, while images 2 and 4 have been corrupted by Gaussian noise with standard deviation 3. In images 3 and 4, the edge has been broken to model outliers near the edge. The two holes in the edge are each ten pixels long, such that the outliers represent 10% of the total length of the edge. The two outlier edges are 2 and 3 pixels away from the global edge, respectively.

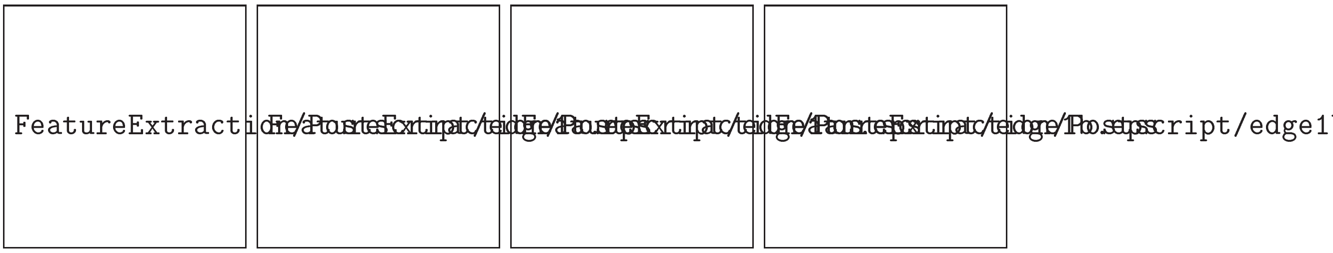

Figure 7.9: Synthetic images used to test different edge detection algorithms. From left to right: image 1, image 2, image 3, and image 4. Images 2 and 3 are corrupted by additive Gaussian noise. In images 3 and 4 the edge is corrupted by structured noise.

These edges are detected a hundred times with each of the methods described above. The brute force method is not included in the test. The search region is defined by an a priori knowledge of the edge position given by two end points $\mathbf{u}_1 = (50, 106)$ and $\mathbf{u}_2 = (180, 92)$. However, the coordinates of the two end points are perturbed by a uniform distribution in $[-3; 3]$ in each test, to model the uncertainty of the a priori knowledge. 20 pixels are sampled on each side of the line between the two end points, i.e. the edge detections are based on the data in a skew window of dimension 130×40. A new realization of the additive noise is used for each test. The width of the Gaussian has been set to $\sigma = 1$ in the methods where a Gaussian smoothing or a Gaussian edge detector is used.

To simplify the analysis, the result of each edge detection is evaluated as the signed distance d from the point $\mathbf{u} = (150, 80)$, which is positioned on the theoretical edge, to the detected edge $(\hat{\rho}, \hat{\theta})$

$$d = \hat{\rho} - \mathbf{n}_{\hat{\theta}} \cdot (150, 80) \tag{7.24}$$

The result of the test is summarized in tables 7.4 and 7.5. The tables show the mean and standard deviation of d and the interval defined by the extreme values.

First of all, the test shows that all methods are un-biased when there are no outliers, while the normal least squares result is clearly biased in the presence of outliers. The overall performance of the approaches based on the smoothed Hough space and the normalized Gaussian convolution is unsatisfactory. Both methods have unacceptable large extremum values, i.e. they are particularly sensitive to certain noise realizations or starting guesses. However, their average performance is not really bad.

It is clear from the test that the approaches based on regression are to be preferred. Even in the presence of additive noise, all the edge detections result in distances within a few hundredths of a pixel. Their response to outliers is a little different. As noted before the normal least squares result is clearly affected by the outliers. The method based on re-weighted least squares is a little less sensitive, and methods 2 and 3 are the most robust in this test. This was also expected, since the "Fair" weighting function does not completely cut off outliers, as the two other methods do. In practical use, the advantage of the M-estimator is its continuous response to the data, i.e. it probably has better convergence properties than methods 2 and 3, that implement a binary weighting of the data. The choice in a given situation must be based on the observed characteristics of the noise.

The methods based on convolution and on fitting a parametric model to the edge seem to perform equally well. However, convolution is linear and makes no strict assumptions about the edge profile other than symmetry. On the other hand, the parametric model is designed for a particular edge profile, and requires a non-linear method to fit the parameters. The assumptions about the appearance of the edge are not crucial in an industrial application, where the edges are typically close to being Gaussian step edges, but the finite

Method	Image 1 ($\sigma = 0$)	Image 2 ($\sigma = 3$)
Smoothed	0.0158 ± 0.0281	0.0154 ± 0.0290
Hough Space	$[-0.0218; 0.2235]$	$[-0.0328; 0.2235]$
Normalized	0.0129 ± 0.5936	0.0222 ± 0.5722
Gauss	$[-1.6376; 3.0443]$	$[-1.6328; 3.2632]$
Edge Fit	-0.0002 ± 0.0008	-0.0012 ± 0.0071
Reg. (1)	$[-0.0018; 0.0015]$	$[-0.0176; 0.0195]$
Edge Fit	-0.0017 ± 0.0010	-0.0002 ± 0.0110
Reg. (2)	$[-0.0038; 0.0003]$	$[-0.0255; 0.0298]$
Edge Fit	-0.0010 ± 0.0007	-0.0013 ± 0.0084
Reg. (3)	$[-0.0028; 0.0004]$	$[-0.0239; 0.0158]$
Edge Fit	-0.0002 ± 0.0008	-0.0010 ± 0.0071
Reg. (4)	$[-0.0019; 0.0015]$	$[-0.0189; 0.0144]$
Convolution	0.0012 ± 0.0011	0.0012 ± 0.0117
Reg. (1)	$[-0.0005; 0.0026]$	$[-0.0291; 0.0216]$
Convolution	0.0037 ± 0.0005	0.0013 ± 0.0147
Reg. (2)	$[0.0020; 0.0053]$	$[-0.0322; 0.0277]$
Convolution	0.0026 ± 0.0010	0.0023 ± 0.0139
Reg. (3)	$[0.0009; 0.0044]$	$[-0.0319; 0.0349]$
Convolution	0.0013 ± 0.0011	0.0003 ± 0.0105
Reg. (4)	$[-0.0004; 0.0026]$	$[-0.0281; 0.0291]$

Table 7.4: Edge detection results for the two left images in figure 7.9. Each method has been applied hundred times with varying starting guesses and different realizations of the noise. The signed distance from a point on the theoretical edge to the detected edge is evaluated each time. The table shows the mean distance, the standard deviation, and the minimum and maximum distances.

Method	Image 3 ($\sigma = 0$)	Image 4 ($\sigma = 3$)
Smoothed	0.0679 ± 0.0341	0.0634 ± 0.0361
Hough Space	$[0.0254; 0.2735]$	$[-0.0115; 0.2735]$
Normalized	0.1105 ± 0.5547	0.1280 ± 0.6812
Gauss	$[-1.2262; 3.4679]$	$[-1.2662; 3.8363]$
Edge Fit	0.1498 ± 0.0143	0.1488 ± 0.0159
Reg. (1)	$[0.1230; 0.1753]$	$[0.1155; 0.1845]$
Edge Fit	0.0103 ± 0.0140	0.0127 ± 0.0133
Reg. (2)	$[-0.0018; 0.0325]$	$[-0.0186; 0.0442]$
Edge Fit	0.0016 ± 0.0008	0.0219 ± 0.0093
Reg. (3)	$[0.0004; 0.0033]$	$[-0.0017; 0.0484]$
Edge Fit	0.0886 ± 0.0081	0.0889 ± 0.0113
Reg. (4)	$[0.0732; 0.1023]$	$[0.0607; 0.1161]$
Convolution	0.1566 ± 0.0146	0.1460 ± 0.0397
Reg. (1)	$[0.1280; 0.1822]$	$[0.0707; 0.2380]$
Convolution	-0.0100 ± 0.0019	-0.0145 ± 0.0147
Reg. (2)	$[-0.0135; -0.0078]$	$[-0.0511; 0.0165]$
Convolution	-0.0020 ± 0.0017	0.0399 ± 0.0239
Reg. (3)	$[-0.0044; 0.0010]$	$[-0.0191; 0.1153]$
Convolution	0.0825 ± 0.0073	0.0826 ± 0.0233
Reg. (4)	$[0.0677; 0.0941]$	$[0.0354; 0.1457]$

Table 7.5: Edge detection results for the two right images in figure 7.9. Each method has been applied hundred times with varying starting guesses and different realizations of the noise. The signed distance from a point on the theoretical edge to the detected edge is evaluated each time. The table shows the mean distance, the standard deviation, and the minimum and maximum distances.

Method	Time [1/100 s]
Smoothed Hough	700
Normalized Gaussian	19
Parametric Edge Fitting	10
Gaussian Convolution	3

Table 7.6: The average time required for one edge detection. The edge detections are based on the pixels in a skew window of dimension 130 × 40.

calculations associated with convolution is clearly preferred to the non-linear optimization required for the parametric model.

The test has also been performed on four images with an edge with a non-rational slope of $-\pi/4$. The results were very much the same, except for the smoothed Hough space, which gave very biased results. This was caused by the skewness of the sampled window. In general, the methods based on the smoothed Hough space and on normalized Gaussian convolution are affected by the shape of the sampled area. The performance of the smoothed Hough space has been seen to be rather insensitive to the slope of the edge, if the sampled region is almost symmetric with respect to the edge. Similarly, the normalized Gaussian convolution has been seen to be very consistent if the sampled window is closely aligned with the edge.

The test of the different edge detection approaches was also timed. The average time required for a single detection is shown in table 7.6. The given time includes the sampling of the window in the image. Clearly, convolution is superior to the other methods, while the calculation of the smoothed Hough space is extremely slow.

7.3.4 Conclusion

This section has presented different approaches to edge detection. The two new methods based on the smoothed Hough space and the normalized Gaussian convolution were designed to give a global estimate of the edge position. The empirical test showed, however, that they both are rather sensitive to the layout of the sampled window and produce estimates with a large spread. All the methods that used one-dimensional edge detection with subsequent regression gave very nice results and, furthermore, they performed a lot faster in the test. Different approaches to outlier detection were tested, and they all gave satisfactory results. The choice of method should be guided by the noise process in the actual data.

It is to be concluded clearly from the test that simple Gaussian convolution is the best approach with the given noise distribution. It is robust, fast, and makes only few assumptions about the edge profile. A better robustness can be achieved by using two-dimensional filters, but the cost is a decrease in the localization accuracy.

The smoothed Hough space is not suited for high precision edge detection. Nonetheless, the background theory provides a good understanding of the projection carried out in Hough space, and the results can probably be used in other applications.

The methods that were analyzed in this section have been implemented in the course of the Ph.D. project. That is, methods that have been proved to have inferior properties in the test, were used in some of the applications that are presented in the coming chapters. Fortunately, the smoothed Hough space and the normalized Gaussian convolution were only used in the installations where the sampled window was closely aligned with the edge. It is therefore not expected that the shortcomings of these methods have affected the presented results.

Chapter 8

Averaging Rotations

This chapter presents a paper with the title *"Averaging Rotations"*, that was printed in the proceedings of the 11th Scandinavian Conference on Image Analysis, SCIA'99, in Kangerlussuaq, Greenland.

The work was inspired by a technical report by *Pennec* [57] from INRIA, France, that addressed the problem of least squares optimization on non-linear manifolds. In this particular problem, the least squares cost function should be based on geodesic rather than vectorial distances. The difference is significant in the case where the manifold is very curved. *Pennec* used the group of three-dimensional rotations as an example in the report. The conventional approaches to averaging rotations are to calculate the vectorial average of the rotation matrices or the quaternions that represent the rotations. However, the barycentric means do not satisfy the constraints to be proper rotations, and need to be orthogonalized and normalized, respectively.

In my paper, I try to evaluate the difference between the least squares solutions based on the vectorial distances and the geodesic distances in the particular case of rotations. It is emphasized that the analysis does not challenge the interesting general results presented by Pennec. It turns out that if the squared distance is replaced by two different cosine approximations, the vectorial means are true least squares solutions, in which the orthogonalization and normalization are inherent parts. The cosine approximations are automatically developed around the mean rotation, and they make it possible to quantify the committed error by using the vectorial distances instead of the geodesic distances. It is shown that either linear approximation is very close to the method that considers the non-linearity of the manifold.

The results have not yet been applied to problems at the shipyard. It could, however, be used to fuse e.g. pose results obtained from different cameras, if, for some reason, a global optimization using all information simultaneously can not be performed.

The paper can be found at the back of this dissertation.

Chapter 9

Projective Reconstruction

For a period of six months in the summer of 1998, I visited INRIA Rhône-Alpes, Grenoble, France. During the stay I was studying projective geometry and obtained some preliminary results on reconstructing and matching an OSS mock-up. This work as well as some theory on the structure-from-motion problem is described in an internal report of the VIGOR project, that is enclosed at the end of this dissertation. The report is entitled *"Reconstruction and Matching of OSS Mock-Up"*.

The experiments were primarily based on software and algorithms developed at INRIA. The calculation of the reconstruction was entirely based on software developed by *S. Christy* at INRIA.

The objective of the experiments was to learn how well a real industrial object can be reconstructed using a fully automatic approach. In short, the reconstruction is based on a series of point correspondences obtained by automatic extraction of interest points using the Harris corner detector [21] and automatic tracking of these points through an image sequence.

The conclusion was that the automatic extraction of points with no use of a priori knowledge was much too unreliable to reconstruct any real structure of the object. Many extracted points had no physical interpretation in the object. They either represented the projection of skew lines or highlights that drifted with the motion of the camera. Other points were lost in the tracking. If they were re-extracted, they were not identified with the lost points. Despite these problems, it was possible to identify a few points in the reconstruction that could be matched with the corresponding points in a CAD model and, using the reconstructed motion, this model could be projected into the images to yield a good visual match. The many point ambiguities made it impossible to automate the manual matching.

A number of points were extracted and tracked manually to test the quality of the actual reconstruction algorithm. The agreement between the images and the re-projected CAD model was very satisfactory in this test. However, the three-dimensional deviations between the model and the reconstruction was still rather large.

It is clear that the chosen approach was much too simple to be of any use industrially. Nevertheless, it disclosed the complexity and some of the difficulties associated with the structure-from-motion problem. Much more structure and robustness could be obtained if the reconstruction was instead based on a series of extracted lines. The experiments were, however, not continued as other projects were given higher priority.

Chapter 10

Check-In and Quality Control on Plasma Cutter

10.1 Introduction

The first site where vision technology was installed within this project was on a plasma cutting station in the production. An image of the cutting berth is shown in figure 10.1. The plasma cutter is used to cut flat steel plates that are assembled to large ship blocks later in the production. The aim of the installation was to perform automatic check-in and quality control. The check-in procedure serves two purposes, namely to ensure that the plate is in the right position before cutting and that the dimension of the plate satisfies the minimum requirements. The quality control after cutting can supply information to the subsequent processes, but is equally important as a means to reveal a damaged tool. It happens frequently that the tool is bent by the heavy steel plates and a number of mis-cut plates are passed on in the production. This is rather costly as the logistics at the yard are not geared to transport plates backwards in the production line. An automatic quality control installation would have a substantial savings potential as the yard has 18 plasma cutting berths.

Unfortunately, this vision installation was given a lower priority before the running-in phase was finished. Consequently, the testing of the entire set-up was incomplete. For instance, the probable origin of a bias in the measurements was never verified. Nevertheless, the installation demonstrates some of the accuracy that can be obtained with vision measurements. The other project that was given a higher priority is described in the next chapter. The two projects are conceptually similar, and the results in the next chapter are also relevant for the present installation.

CheckInAndQualityControlOnPlasmaCutter/Postscript/Berth_light.eps

Figure 10.1: Plasma Cutting Berth.

10.2 Set-up

The plasma cutting berth measures approximately $4\,m \times 18\,m$, but the field of view of the cameras must be a bit larger to include reference marks in the scene. There is not much freedom in the placement of the cameras. The crane that transports the steel plates needs to move in and out of the area, forcing the cameras to be mounted just below the ceiling. This gives a distance of approximately $9600\,mm$ from the cameras to the berth. Using four JAI 576×768 cameras with $8\,mm$ lenses the area can be covered with a resolution of approximately $8\,mm/pix$. The cameras were installed through four holes in the roof to allow easy access without intervention with the production. The holes and the crane are seen in figure 10.2 showing an image of the ceiling just above the berth. The requirement was to measure the plates in the berth with an accuracy of $\pm 1\,mm$. Consequently, the edge detection in the images should be performed with sub-pixel accuracy.

CheckInAndQualityControlOnPlasmaCutter/Postscript/Ceiling.eps

Figure 10.2: Ceiling with three of the four holes for the cameras marked with circles. The fourth hole is covered by the crane.

A complete three-dimensional camera model is employed even though the measurement task is basically two-dimensional. This is necessary to accommodate for the varying thicknesses of the plates. At the same time the physical interpretation of the external camera parameters can be used to monitor the stability of the set-up. The constraints on the placement of the cameras make stereo vision impossible. It is therefore necessary to assume that the cutting berth is planar, i.e. the measurements are based on a plane to plane back-projection. The position of the object space plane is determined by the thickness of the current plate and the average height of the berth. The pose calculation for each camera is based on a number of reference marks in the scene. The marks are placed to give the best pose estimates, but are constrained to the edges of the berth. The main dimensions of the set-up and the placements of the reference marks are summarized in figures 10.3 and 10.4.

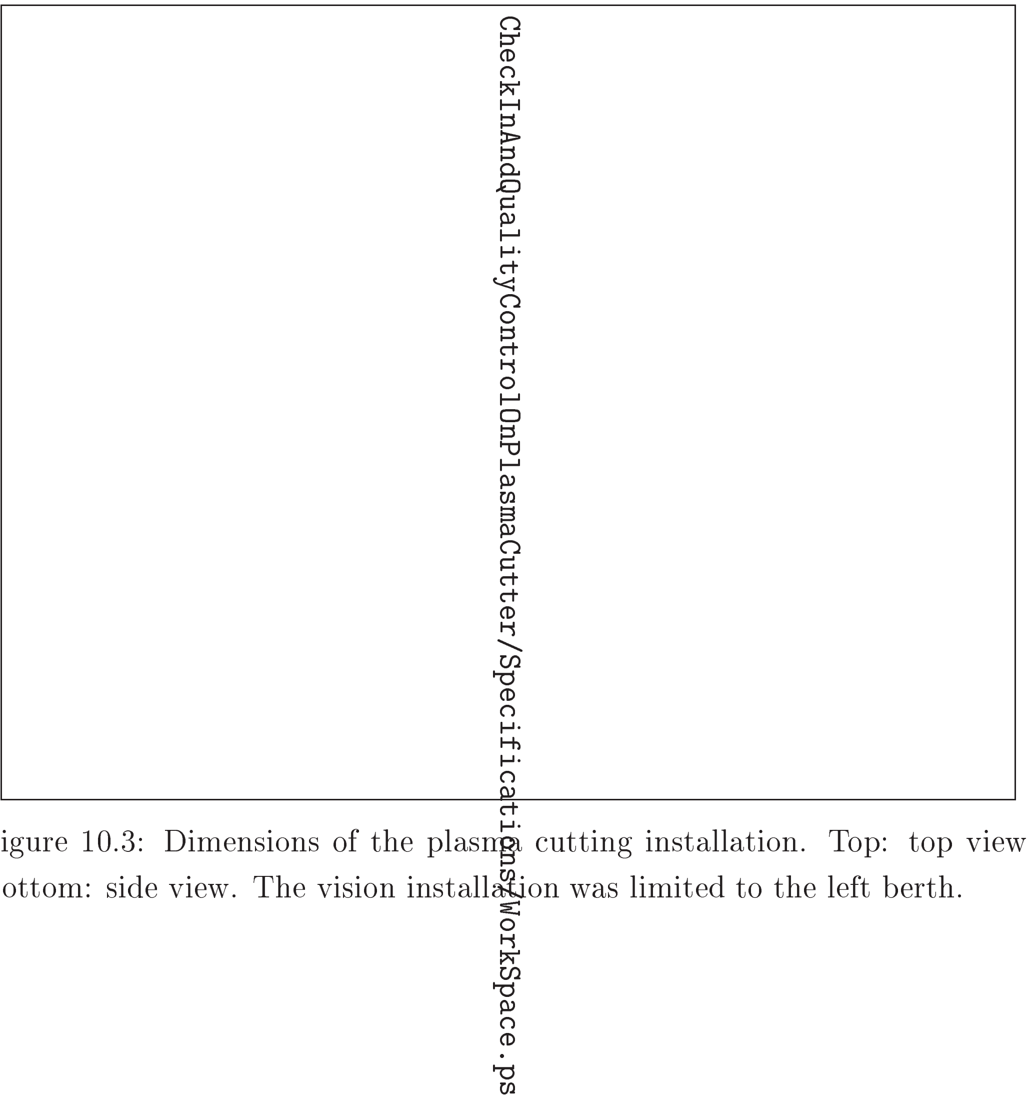
CheckInAndQualityControlOnPlasmaCutter/SpecificationsWorkSpace.ps

Figure 10.3: Dimensions of the plasma cutting installation. Top: top view. Bottom: side view. The vision installation was limited to the left berth.

CheckInAndQualityControlOnPlasmaCutter/Specifications/PlacementsWidth.ps
CheckInAndQualityControlOnPlasmaCutter/Specifications/PlacementsLength.ps

Figure 10.4: The placement of the reference marks on the plasma cutting berth (black dots). Left: longitudinal placements. Right: transversal placements.

10.3 Lighting

To achieve a good contrast on the plate edges in the cutting berth, it was necessary to put up additional lighting in the work space. It turned out to be a difficult task to obtain even illumination in such a large area. The positions of the light sources are again constrained by the layout of the work cell. Additionally, strong light must not be directed or reflected towards the cameras or the people that work in the area.

A professional lighting company was hired to design an even illumination. The suggested design actually gave an illumination that seemed even to the human eye, but the linear cameras revealed that certain areas were illuminated twice as much as other areas. The lighting was then modified by shielding some of the projectors to make the lighting more diffuse. A truly even illumination was, however, never achieved.

10.4 Measurement

Although the examples in this chapter all show plates in a fixed position in the berth, the objective of the installation is to measure the position as well as the dimensions of the plates. The most probable position of the plate is therefore determined by relating the CAD model to the measured edges. The local deviations that are reported to the controller are thus specified in relation to the optimal position estimate. The steps in a measurement are as follows

1. grab image(s)
2. detect reference marks
3. estimate camera pose(s)
4. down-load the CAD model and the approximate plate position from the controller
5. superpose the CAD model on the image(s)
6. detect edges
7. calculate the corner positions
8. apply the best rigid transformation to match the measured corners with the CAD model corners
9. report results to the controller

The image acquisition in this installation is very simple as the illumination is constant. Hence, the aperture and the shutter time are always the same. The detection of reference marks in step 2 is performed as described in chapter 7. The pose estimates in step 3 are based on a least squares minimization of the image residuals.

The edge detection approach employed in this installation, and in that of the next chapter, is a little unconventional. To avoid rectifying the entire image(s) and to make optimal use of the approximate knowledge of the edge locations, it was chosen to re-sample the edges in a CAD based, i.e. metric, coordinate system. This was obtained by direct use of the camera model. In this particular installation, time was not an issue, and the edge detection was based on the brute force method described in chapter 7. This was implemented by applying a Gaussian edge detector [12] to the acquired images. For different line parameter values in CAD coordinates, the line was then sampled in the images to calculate the total edge strength along the edge. The line that scored the highest in this global cost function was taken as the plate edge. In this approach, it is very important to sample the images with bicubic interpolation, since the ridge of the differentiated edge might be located between pixels.

The corners of the plate are located by the crossings of the edges. The examples in this chapter only show rectangular plates, but in a real situation many cuts will be made in the plate to yield a large number of corners. These detected corners are matched with the corners in the CAD model. In step 8 the two-dimensional position and orientation of the plate are determined by the method described in appendix A. This is necessary even if the plates are initially laid in a fixed position, as the release of residual stress in the plates may cause them to move during cutting.

The most important feedback to the controller in step 9 is the position and orientation of the plate and the main dimensions. In the long term, it is interesting to extend the feedback with the local deviations of the plate. To determine those, the global edge detection method described above has been applied locally. In the examples given below, the deviations of these local estimates have been exaggerated relative to the global estimates to show the variations along the edge.

10.5 Data Analysis

Unfortunately, the data material from the installation is a little sparse. As noted earlier, the reason is that another project was given a higher priority, so that the very resource demanding data collection was not completed. Nevertheless, some tests that were performed show the accuracy of the installation. The tests investigate the quality of the plate measurements as well as the consistency in the pose estimates.

Figure 10.5 shows three images of a plate in the cutting berth. At the time of the tests the fourth camera was not yet calibrated. This was not a problem, as the majority of the plates that pass through the cutting station are covered by the field of view of the first three cameras. Note, the serious lens distortion in the images. Also note the reference marks on the edges of the berth and the image overlap. The reference marks and the average height of the berth have been measured with a theodolite system to establish a world space coordinate system.

CheckInAndQualityControlOnPlasmaCutter/Postscript/cam1.eps
CheckInAndQualityControlOnPlasmaCutter/Postscript/cam2.eps
CheckInAndQualityControlOnPlasmaCutter/Postscript/cam3.eps

Figure 10.5: Three images from the cameras 10 m above the cutting berth. Plate **pb090**. From left to right: image 1, image 2, and image 3.

The camera poses can be determined, once the reference marks have been detected in the images. Remember, that the internal parameters of the cameras have been estimated in the off-line calibration. Hence, the entire geometry of the camera set-up is now established. It is therefore possible to re-sample the image intensities in a plane in object space, that coincides with the surface of the plate in the berth. This is, in fact, what is done in the edge detection phase. To show the principle in the re-sampling of the image, the entire work space has been re-sampled in figure 10.6. The sampling rate is $10\,mm/pix$ and the image measures $550\,pixels \times 1200\,pixels$. Several observations should be made in this image. First, the lens distortion and the perspective effects are no longer visible. Second, the overall intensity levels of the images are different. This could be corrected to improve the visualization, but it is not reasonable in a measurement situation where the edge strength carries information about the detection variance, cf. chapter 7. Third, the reconstruction of the reference marks is imperfect. This is because they are positioned in another height than the plate, giving rise to parallax effects. This explanation has been verified by performing a re-sampling in the height of the reference marks. Note, in the junction between the first and the second image that it is only the slight difference in image intensity that reveals the different origin of the pixels. There is no spatial evidence that the image is merged from two cameras.

Figure 10.6: The images in figure 10.5 re-sampled in a CAD based coordinate system. Plate **pb090**.

10.5.1 Pose Estimation

Before proceeding to study the measurement accuracy, it is worthwhile to check the consistency of the pose estimation. This is done empirically by acquiring nine images from each of the three cameras and calculating the poses. The

Parameter	Camera 1	Camera 2	Camera 3
$x_0[mm]$	-3816.1 ± 0.16	-3534.6 ± 0.17	-4086.1 ± 0.27
$y_0[mm]$	-2538.4 ± 0.11	-7073.6 ± 0.07	-11409.3 ± 0.09
$z_0[mm]$	9637.6 ± 0.54	9743.3 ± 0.49	8837.0 ± 0.96
$\omega[rad]$	3.13757 ± 0.00014	3.15912 ± 0.00011	3.11438 ± 0.00010
$\varphi[rad]$	0.00914 ± 0.00007	0.00219 ± 0.00004	-0.06236 ± 0.00008
$\kappa[rad]$	1.57184 ± 0.00001	1.58579 ± 0.00001	1.57371 ± 0.00002
$X_c[mm]$	2622.4	7041.1	10824.5
$Y_c[mm]$	3780.0	3810.2	3858.5
$Z_c[mm]$	9629.3	9662.6	9638.7

Table 10.1: Empirical pose statistics based on nine images from each camera. The bottom three rows are the camera positions in object space calculated from the top six rows.

result is shown in table 10.1. The only physical dependency between the images is a rather short time delay between the acquisitions. It is therefore very reassuring to see that the standard deviation of the position estimate is less than 1 *mm*. The camera placements in object space are calculated in the bottom three rows of the table. Note, that the cameras are placed on a straight line within 100 *mm* along the direction of the berth, i.e. the X-axis. The longitudinal positions of the cameras do not match the specifications in figure 10.4, because the camera placements were adjusted to avoid the girders in the ceiling, cf. figure 10.2.

The empirical statistics only prove the consistency of the pose estimations. The quality of the individual pose estimates can be evaluated from the reference mark residuals. Therefore, a residual analysis was performed on the reference marks in the images to show if the differences between the projected and the detected marks were random or systematic. For a perfectly calibrated camera and exactly known reference mark positions, the residual would naturally be random. Figure 10.7 shows the residuals resulting from the pose calculation in seven images from each of the three cameras. The residual vectors are plotted in image coordinates with an exaggeration factor of 1000. It is clear that the independent images have very dependent residuals. For instance, the total variance of the residuals in the left camera in figure 10.7 is $1.4e-02$. If the systematic part is removed, i.e. the average over the seven images, the variance is reduced to $1.8e-04$. In other words, the stochastic part only represents 10% of the standard deviation. The observed systematic deviation corresponds to approximately 1–2 *mm* in object space. Unfortunately, later experience showed that the accuracy of the theodolite measurements were very dependent on the people who performed them. To benefit from the accuracy of the system it is required to place targets very precisely and patiently wait for the system to iterate for a short while. It is therefore likely that the observed deviations stem from poor reference mark measurements and not from a poor camera model.

There is also reason to believe that the systematic error is random from mark to mark. Therefore, the redundancy in the number of marks should lead to a satisfactory pose after all. It should also be considered that the present installation is based on relative measurements, so that a modest pose variation only has small influence on the measurements. This is naturally only true as long as the measurements are performed close to the plane spanned by the reference marks.

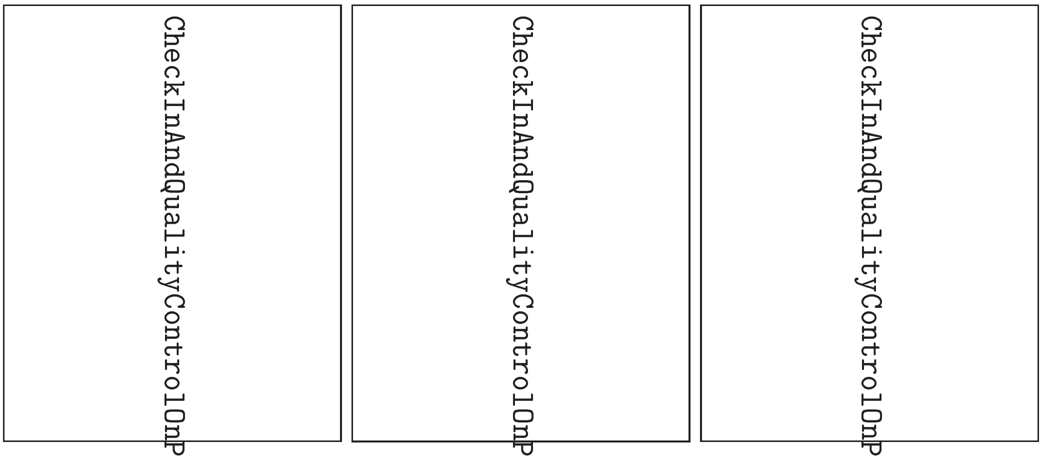
CheckInAndQualityControlOnPlasmaCutter/Postscript/residuals.cam1.pp480.ps
CheckInAndQualityControlOnPlasmaCutter/Postscript/residuals.cam2.pp480.ps
CheckInAndQualityControlOnPlasmaCutter/Postscript/residuals.cam3.pp480.ps

Figure 10.7: Pose residual plots for seven images from each camera. The residual vector in each point has been exaggerated 1000 times. Coordinates are in pixels.

10.5.2 Plate Measurements

The ability to measure accurately with the system is naturally also tested. The plate in figure 10.6 has been measured nine times with independent images. The results are shown in table 10.2. Unfortunately, outlier detection was not

	Manual [mm]	Measured [mm]	Min. [mm]	Max. [mm]
$length_1$	10145	10142.2 ± 0.09	10142.17	10142.47
$length_2$	10156	10166.4 ± 0.29	10165.92	10166.82
$width_1$	2808	2807.2 ± 0.09	2806.91	2807.21
$width_2$	2807	2804.9 ± 0.09	2804.85	2805.16

Table 10.2: Measurement statistics for plate **pb090** based on nine images from each camera. The lamp influences the length measurements.

	Manual [mm]	Measured [mm]	Min. [mm]	Max. [mm]
$length_1$	9753	9753.7 ± 0.5	9753.1	9754.3
$length_2$	9762	9761.6 ± 0.3	9761.2	9761.8
$width_1$	1808	1805.9 ± 0.1	1805.8	1806.1
$width_2$	1808	1808.5 ± 0.0	1808.5	1808.5

Table 10.3: Measurement statistics for plate **nn** based on three images from each camera.

implemented in the software at the time of the test. The two length measurements are therefore biased by the saturating lamp near one of the edges. Another test was performed with three measurements on the second plate in figure 10.8, where there are no reflections. These results are reported in table 10.3. Both tests show that the reproducibility of the system is satisfactory. The observed stochastic variation is well below the required measurement accuracy of ±1 *mm*. Unfortunately, the measurements are a few millimeters biased even when there are no external factors to influence the result. The probable origin of this bias will be discussed in the next section, where the local deviations of the measurements are analyzed. Despite the fact that the requirements are not met, it is convincing that the plate is automatically measured within 2–3 *mm* at a distance of 10 *m* without any use of targets.

10.5.3 Local Deviations

In this section it is investigated how the edge position varies along the edge. In a real measurement, this variation should reflect the deformation of the plate as a result of heat injection or the release of residual stress. In the present example, where the plate has not yet been cut, it serves to show if the variation is random as would be expected. Especially, it should show if the edge detection behaves continuously across image junctions.

The local edge position is determined by splitting up the edges in 330 *mm* long pieces. These pieces are then detected in the same manner as the global edges. The distance from the end points of the segments to the globally detected edge is exaggerated 50 times to visualize the deviation. The saturating lamps are excluded from the edge detections in that example. Consequently, the local edge detections are sometimes unstable owing to the sparsity of edge information. These deviations should be ignored. The residual plots for four different plates are shown in figure 10.8. The larger deviations measure approximately 3 *mm*. It is seen that there are no particular discontinuities at the image junctions. However, the magnitude and the correlation of the deviations are concerning. Comparing the four images, it is noticed that the deviations follow a similar pattern. As discussed earlier, this can either be a result of a poor camera model or a violation of the assumptions about the observed world. The latter seems to be the case here. Subsequent measurements of the height variations in the cutting berth revealed that it was far from planar. The variation amounted to more than 10 *mm*, which actually produces a parallax error in the order of magnitude of the observed deviations.

It is hard to see how this problem can be solved. The spikes that support the plates in the berth are constantly damaged by the plasma cutter, and are frequently replaced. It is therefore not straightforward to maintain a planar support for the plates. It is very unfortunate that this problem was not realized at an earlier stage. It emphasizes the importance of making proper specifications early in a project.

10.6 Conclusion

The installation on the plasma cutter is seen as a success even though it is not presently used in the production. The project has shown that a camera's pose can be found accurately and consistently by the use of reference marks, and that the images from multiple cameras can be employed simultaneously in a measurement. Repeated measurements on a plate showed that the system possessed a very convincing reproducibility. Despite the non-planar berth, the system achieved a measurement accuracy of 2–3 *mm* at a distance of 10 *m*. That is, the installation came very close to meeting the specifications

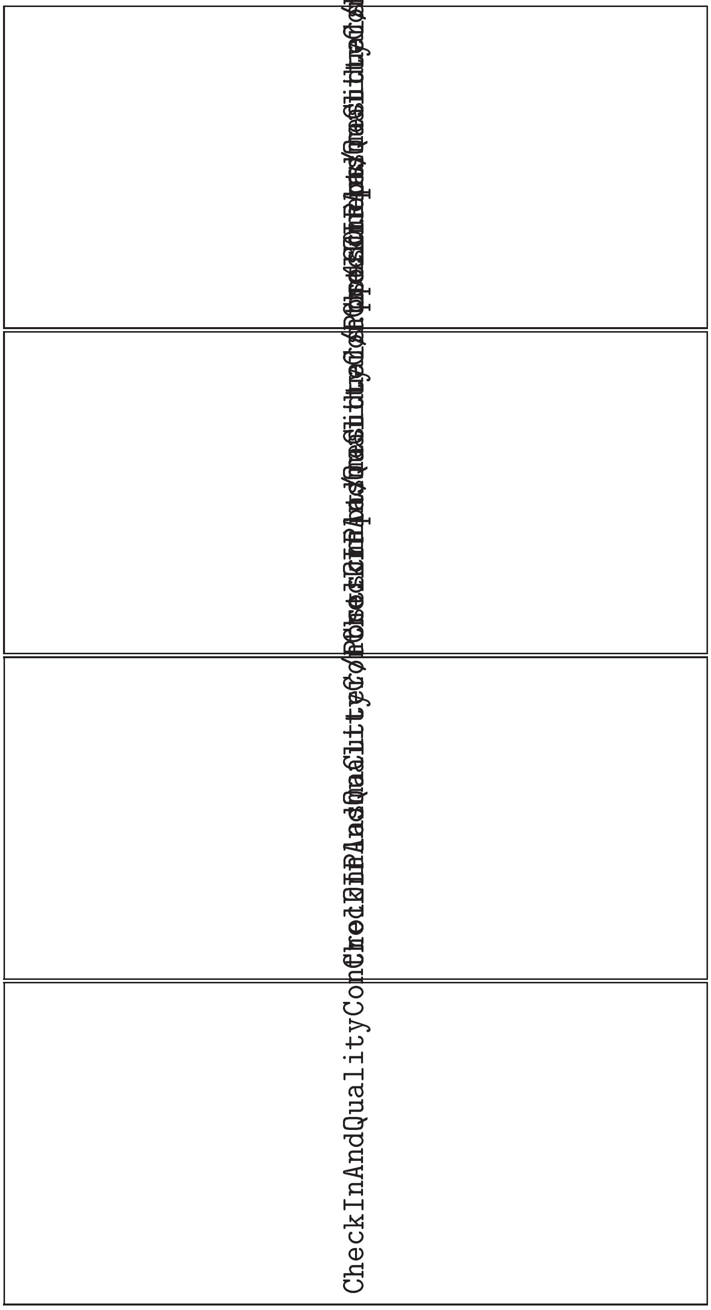

Figure 10.8: Residual plot for plates **pb090**, **nn**, **164pp390**, and **pp480**. The difference between the locally and globally detected edges are exaggerated 50 times. Ignore artifacts near lamps.

of ±1 mm. It would therefore be worthwhile to try to solve the remaining problem. It could e.g. be attempted to level out the berth, or to introduce additional cameras or lasers to obtain three-dimensional information about the edge.

One additional circumstance would need to be considered if the installation should work in production. During cutting the plates are submerged in water, that become increasingly dirty. The water will probably pose a serious problem to the edge detection. This problem has not been examined.

Chapter 11

Check-In of Steel Plates on the Quay

11.1 Introduction

The vision installation on the plasma cutting station was postponed to release resources for a check-in system on the quay, where the steel plates are received and organized in a storage area. The images in figures 11.1 and 11.2 give an impression of the storage, with the large cranes, that arrange the plates.

All plates that arrive at the yard are ordered for a specific position in a ship. On arrival they are identified in a database, marked with dimensions etc., and placed in stock for later retrieval. It is the intention to automate this process on the yard. This also implies that the cranes, or at least the storage control, will be automatic. Hopefully, an optimal storage control can reduce the number of times each plate is handled before it enters the production line. The aim of the vision installation is to measure the dimensions of the incoming plates to determine the corresponding entry in the database. In some cases, the measurement can be used to reject plates that do not satisfy the ordered dimensions. This allows the yard to return plates as early as possible in the production line. The dimension and other identification is printed on the plate right after measurement. The printer is hidden in the white box in the images in figure 11.2. The installation is intended to function day and night all year round, but the first efforts were focused on making satisfactory measurements under normal daytime conditions. The plate thickness is estimated mechanically as the plate rolls into the measurement area.

Although this project attracted more interest in the management than the project in the previous chapter, it was only accepted reluctantly in the vision group. The two projects are almost similar in their set-up. Since a number of uncertainties on the plasma cutting station were still not fully understood, it was considered premature to build an outdoor installation with the varying illumination being yet another unknown factor. This concern was later justified, and the set-up of the installation had to be changed in the course of the project. This is described in section 11.2 below.

CheckInOfSteelPlatesOnTheQuay/Postscript/be
CheckInOfSteelPlatesOnTheQuay/Postscri

Figure 11.1: The check-in station on the quay. The cameras are mounted on the beam

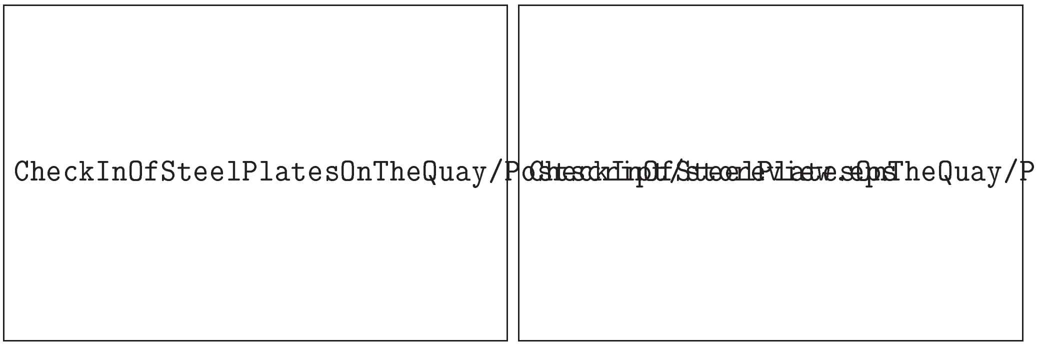
CheckInOfSteelPlatesOnTheQuay/P
CheckInOfSteelPlatesOnTheQuay/Postscript/reference

Figure 11.2: The check-in station on the quay. The large and small reference marks have a diameter of 330 mm and 110 mm, respectively.

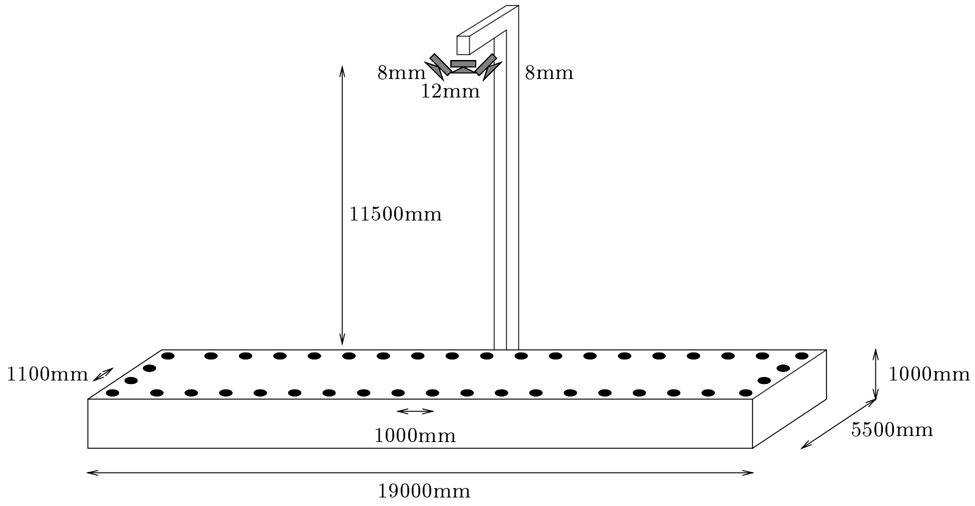

Figure 11.3: The dimensions of the measurement station on the quay.

11.2 Set-up

As mentioned above, the installation set-up was changed during the project. In the very beginning, vision was supposed to measure the width as well as the length of the plates. The cameras therefore needed to cover the entire work area, i.e. approximately $20\,m \times 6\,m$. Again, the cameras had to be placed out of reach of the large crane that handles the plates. This was achieved by raising a $12\,m$ tall beam reaching into the work area from outside the crane tracks. The beam is seen from two views in figure 11.1. It is possible to turn the beam 90 degrees away from the work area to yield access to the cameras without interrupting the working cranes. The two cameras that point to the far ends of the roller table are equipped with $8\,mm$ lenses to give a sufficient field of view. The center region of the work area can then be covered by a camera with a $12\,mm$ lens. The latter camera is turned 90 degrees to cover the width of the table. Three JAI cameras where used, cf. chapter 3. 42 reference marks were placed on the edge of the table for pose determination. Each reference mark has a diameter of $330\,mm$. A white ruler was placed in the middle of the table to aid the coarse detection of the plates, cf. figure 11.4. The main dimensions of the installation are shown in figure 11.3. To avoid the problems experienced on the plasma cutter, the height variation of the rollers was measured. All measurements were found to be within 1–2 mm.

Even though the placement of the cameras is different from that in the plasma cutting installation, the software could actually be applied directly. It was simply a question of specifying a new set of reference marks and three new approximate camera poses. This shows the clear advantage of utilizing the available a priori knowledge. Acquiring three images and calculating the poses, it is again possible to re-sample an image in metric coordinates. This is seen in figure 11.4. The images are re-sampled at 12 *mm/pix*. In the upper example the merged image is based on the images from cameras 1 and 3 only, showing that these cameras actually cover the entire work area. The images from all three cameras were used in the lower example. Of course, the middle image is normally used, as it has a 50% better resolution on the width measurement. The fitted reference mark models have been subtracted in the upper merged image to verify the quality of the detection.

The two examples in figure 11.4 have some artifacts that should be ignored. In the upper example, circular marks have been placed on the plate to make a test of the calibration, that is independent of the edge detection. In the lower example, all but the reference marks have been gamma corrected to test the measurement's sensitivity to non-linear intensity changes. The trouble shooting on the installation is described in section 11.5 below.

Unfortunately, it was not possible to obtain satisfactory length measurements with the set-up described above. The error on the length measurements were often 4–5 times greater than the error on the width measurements. The dominating source of error was the reflecting light from the sun. Adjusting the shutter to obtain a satisfactory contrast on the edges and the nearby reference marks in sunshine, often caused the reference marks in the direction of the sun to be saturated. In fact, this problem was even visible on overcast days. The upper images in figure 11.4 are acquired on a cloudy day, but the reference marks in the left-hand side of the left image are 30% lighter than reference marks in the right-hand side of the same image. In consequence, a lot of reference marks far from the camera had to be omitted from the pose estimation in heavy sunlight. However, these marks were also the most important in terms of pose accuracy. At the same time the measurements with the angled cameras are the most sensitive to a correct pose. On top of these problems, the plate edge was often also corrupted by reflected sunlight. Remember, that sub-pixel edge detection only makes sense, if we know a good model of the image creation process. This is not the case when the sunlight causes specular reflections on a curved edge.

CheckInOfSteelPlatesOnTheQuay/Postscript/merge.eps
CheckInOfSteelPlatesOnTheQuay/Postscript/Mergeb.eps

Figure 11.4: Merging image information on the check-in station. Top: three original images and an image merged from camera 1 and 3. Bottom: three original images and an image merged from all cameras. The work space was re-sampled at 12 mm/pix in both examples.

Instead of pursuing these problems, it was decided to solve the length measurement by a mechanical solution. This was easily implemented by placing a tachometer on the roller table. This solution was not chosen immediately, as the outdoor environment with dirty plates was anticipated to cause difficulties for the mechanical device. It has also turned out that a tolerance of 5–10 mm has to be accepted.

It was then attempted to refine the vision installation's width measurement. That is, the installation was reduced to the middle camera. This camera was replaced by a digital Basler camera with a 1296×1030 resolution, cf. chapter 3. The camera was changed for two reasons. First, the better resolution and the better lens of the Basler camera could improve the measurement accuracy. Second, the video signal had been seen to be very affected by electrical noise. The origin was the electrical fields induced by the electrical roller table and the cranes and their electromagnets. If a digital signal is affected by noise it is easily seen as speckles in the image. In comparison, an analog signal has a much more continuous response to noise, making the noise harder to see.

In the redesign of the installation it was chosen to add light to the scene to make the plate appear dark on a light background. However, this would also make the reference marks dark and impossible to see. To make the contrast on the edge and the reference marks similar, it was decided to base a number of new reference marks on the same light sources that were used for the plate. This was achieved by suspending thin steel plates with holes above the light sources. The holes would then appear as reference marks. Each of four light boxes was equipped with a plate with a configuration of five reference marks with diameters of 110 mm. These marks can be seen in figures 11.2 and 11.5. The next section will describe the artificial light that was added to the scene.

11.3 Lighting

As expected the light turned out to be the hardest problem to solve in the outdoor installation. At first it was planned to measure the edge between the lighter plate and the darker background. In the design phase, it was considered to dig a 1 m deep pit underneath the roller table. Covering the pit with a dark grid would prevent light to be reflected even on days with snow. The idea turned out to be too expensive and the concrete base was painted black instead. This solution turned out to be unsatisfactory. The painted background was only black for a short while and produced strong reflections in sunlight. Furthermore, the plates often have dark rusty edges, making subpixel edge detection very difficult.

Consequently, it was decided to change approach when the installation was reduced to a single camera. Since the installation is too large for the light to be shielded, it was chosen to add light to the scene instead. It is impossible to improve the contrast by illuminating the plate, as the sun produces one or two factors more light than we could possibly add. Instead, four light boxes where placed under the roller table and pointing directly towards the camera. This way the plate appears as a black edge in the image. Figure 11.5 shows the four light boxes that were placed underneath the plate. This solution has the advantage that rust and dirt on the plate no longer affect the measurements. Only the light boxes will need to be kept clean. Each light box consists of 15 60 *watt* neon tubes with a high frequency supply, which is necessary owing to the short shutter time and the fast response of the neon tubes. At first, the light boxes were covered by mat Plexiglas to diffuse the light, but this was soon replaced by transparent glass to improve the daytime performance. The produced contrast was still not sufficient in heavy sunlight, but the installation was now functional under normal conditions as well as at night.

11.4 Measurement

Even though two different set-ups have been described above, the measurement situation is very much the same. The algorithm listed below is therefore common to the two set-ups, but a few steps are not relevant or not necessary when only the width measurement is performed. The main difference in the measurement between this installation and the installation on the plasma cutter is the limited knowledge of the position of the plate. It is attempted to center the plates on the roller table, but that is the only available a priori knowledge. It is therefore natural to separate the measurement into a coarse localization and an accurate edge detection. The rough position of the plate is obtained by image projections, i.e. the average image intensity is calculated along the rows and the columns of the image, respectively. Such projections can be seen in continuation of the edges in figures 11.4 and 11.5. The projections are drawn

CheckInOfSteelPlatesOnTheQuay/Postscript/modtagest.eps
CheckInOfSteelPlatesOnTheQuay/Postscript/Merge1_Zoom.eps
CheckInOfSteelPlatesOnTheQuay/Postscript/Merge2_Zo

Figure 11.5: A width measurement based on back lighting. Top: the original image. Left: re-sampled image with coarse detection result shown with white crosses. Right: re-sampled image with accurate detection result shown with white line. Two regions in each of the two lower images have been blown up to show the details.

with a width of fifteen pixels for better visualization, but they are of course one-dimensional arrays. To obtain an estimate of the plate orientation, the projections are performed bilaterally in the longitudinal direction. The rollers in the table and the white ruler are avoided in the projections. Now, the edge positions can be found from the projections by convolving with a Gaussian edge detector, cf. *Deriche* [12]. The result of the convolution is also drawn in the images and the edges show up as clear maxima and minima. Before the results are accepted they are tested against the assumption of symmetry around the center and the position of the edge on the white ruler. In fact, the accuracy of the coarse detection is about 1–2 *pixels* ≈ 10–20 *mm*. In figures 11.4 and 11.5 the result is displayed with small white crosses on the edge of the plate in the merged image. The detected edge is naturally associated with the center row/column of the projected region.

The coarsely detected edges are then refined with a more accurate edge detector. The method based on the smoothed Hough transform was used in the length and width measurements. To reduce the processing time, this method was later replaced by the approach based on normalized Gaussian convolution. The bias, that is discussed in chapter 7, is not a problem in this installation because the edge is re-sampled on a closely aligned raster.

The different steps in a measurement are

1. receive estimated plate thickness
2. grab image(s)
3. check image intensities in background and foreground regions
4. if *not* ok, adjust shutter and offset and goto 2
5. if necessary mask images
6. perform horizontal and vertical image projections
7. calculate coarse plate position
8. perform accurate edge detection
9. report results to the controller
10. locate the plate in the database

A few steps in this list have not yet been mentioned. The image grabbing has to be enclosed in a loop that adjusts the shutter and offset of the camera(s). This is absolutely necessary in outdoor operations. If the old setting does not produce satisfactory images, the offset parameter is reset to a default value, and the grabbing procedure is iterated until a good shutter value is found. Finally, the offset parameter is adjusted to prevent black level cut-off. An image masking has also been introduced. This step tries to limit the effect of local reflections that make the image saturate and is done by thresholding the image at some high intensity level to isolate the saturating pixels, which can then be discarded from the edge detection. The neighboring pixels, that may also be affected, are removed as well by performing a few dilations on the thresholded image. A simple pixel count can reveal whether the image is acceptable or if the amount of reflection is too severe.

11.5 Trouble Shooting

The previous sections have touched on the difficulties that were encountered on the initial installation. It was observed that length measurements were biased 5–10 *mm* or varied 2–5 *mm* between measurements. The problem is that the origin of these deviations is not visually obvious in the images owing to the poor resolution. It is therefore impossible immediately to conclude from an image if the problem is caused by reflections from the sun, a hanging or bulgy plate, or a poor pose estimation. A series of experiments were therefore designed to isolate the influence of various factors. However, it is very difficult, if not impossible, to eliminate individual factors in an outdoor experiment and the conclusions are therefore rather vague. Nevertheless, the indoor experiments have provided an indirect proof that the dominating problem on the installation is the sunlight. The light affects the measurement through the reflection on the edge as well as on the pose estimation. In either case, it can cause an error in the detection of the edge of as much as 1 *pixel*, which is approximately 10 *mm*. Another problem could be that the theodolite measurements of the reference marks were inaccurate resulting in poor pose estimates. A measurement of the marks was ordered three times, and the results were seen to vary 2–3 *mm*. Despite the redundancy in the number of reference marks, this could easily have a serious effect on the measurements.

It was attempted to analyse the effect of three factors independently in the experiment, namely the light, the position of the plate, and the size of the plate. A special 4 *m* plate was constructed to eliminate the effect of reflections on the edge. The plate was actually based on a larger plate with black background painted around the border to make a virtual 4 m plate in the center. This also ensured a perfect background for the edge detector.

In the light experiment, the plate was kept in the same position to eliminate the variations caused by the geometry of the installation. The plate was positioned right below camera 2, where it could be seen by all three cameras. The plate was measured 210 times with each camera in two minute intervals. The mean and standard deviations of the measurements from camera 1, 2, and 3 were $4002.1 \pm 2.2\,mm$, $4004.9 \pm 1.3\,mm$, and $4002.8 \pm 3.4\,mm$, respectively. It is seen that the standard deviations are a bit too large to measure the plate within the specified requirement of $\pm 2\,mm$. Especially, the measurements from camera 3, which points towards the sun, is affected by the light changes. It is also seen that the bias in the estimate is different from camera to camera. This could be an indication that the pose estimates are biased, which might be explained by the different reflections from reference marks at different positions, or the poor theodolite measurements of the marks.

The dependency on the position of the plate was tested by moving the 4 m plate to different positions on the roller table. It is, however, impossible to eliminate the serious effect of light in this experiment. For instance, a biased pose caused by the light will have a larger influence on the measurement when the plate is positioned at the ends of the table. The length of the plate was measured ten times in ten different positions on the table. The standard deviation in the same position was typically 0.5–1.0 mm, but the mean values varied as much as 10 mm. The variation of the mean was not systematic. Even though this experiment was meant to isolate the geometric effect, it is very hard to conclude from the data whether the light or the geometry is the origin of the variation.

In the final experiment, the size of the plate was varied from 1 m to 4 m by applying a fake background with black tape to one of the ends of the plate. Twenty measurements were performed on each of nine different plate lengths. The standard deviations of the measurements were again 1–2 mm, with the bias varying randomly up to 10 mm.

The two geometric experiments above show that the bias in the dimension measurement varies with the position and the size of the plate. This can be caused by purely geometric errors, such as the plate not lying totally flat on the table or poor theodolite measurements of the reference marks. It can,

Parameter	Mean	Std. Dev.	Min	Max
$X_c[mm]$	8063.4	± 8.3	8044.6	8088.5
$Y_c[mm]$	2175.7	± 4.2	2162.3	2187.6
$Z_c[mm]$	11360.1	± 4.3	11344.8	11368.7

Table 11.1: The pose variation of the camera mounted on the beam

however, also be caused by the sunlight's indirect influence on the pose through the reflections on the reference marks. The length measurement was given up before the exact origin of the variations was determined, but the indoor experiments show that the desired measurement accuracy can be obtained if there is no sunlight and the geometry of the set-up is carefully measured. This statement is also supported by the width measurements discussed below, which were performed under much better lighting conditions.

11.6 Data Analysis

The accuracy of the width measurement was tested in 32 repeated measurements of a plate. A measurement was conducted every half hour from 3:00 *p.m.* July 30 to 6:30 *a.m.* June 1, 1999, so that the test period contained night as well as day. The plate was also measured manually in five different positions giving the results 3080 *mm*, 3082 *mm*, 3082 *mm*, 3081 *mm*, and 3080 *mm*. This gives a mean and standard deviation of 3081 ± 1.5 *mm*. The thickness of the plate was 10 *mm*.

The automatically adjusted shutter time of the camera did not vary much, as the scene is dominated by the artificial light. All shutter times were between 530 μs and 690 μs.

The pose of the camera was not constant. However, this was expected as the beam is affected by the wind. The variation in the calculated position of the camera can be seen in table 11.1. It is seen that the estimated position of the camera varies as much as 40 *mm*. The results below, will show that the movement of the camera has no effect on the measurements.

The width of the plate is derived from the edge detections described in section 11.4 above. There are two edge segments on each side of the plate. They are drawn in the right image of figure 11.5. The shortest distance to the opposite segment is calculated for each of the eight end points. Distances that correspond to opposite points are averaged, such that we are left with four width measurements. These are shown in the upper plot in figure 11.6. The four widths are reduced to one by a consensus scheme. If a measurement is found to be an outlier it is not included in the average. If the measurements agree two by two, the average of the two smallest widths is used. This approach is chosen because the false edges always occur on the light side of the edge. The width, that is returned to the controller, is displayed in the lower plot of figure 11.6.

Figure 11.6: Width measurements as a function of time. Top: the width at four different positions. Bottom: the consensus result.

The plots show that the measurement accuracy is clearly better at night. All the large fluctuations occur in the first four hours, while the sun is still very bright. Despite these variations, all the measurements lie within $2\,mm$ and have a standard deviation of only $0.3\,mm$. This is a very nice result, considering that the camera is placed $11.4\,m$ away from the edge and that the camera has moved between images.

The behaviour seen in the first four hours gives an indication of the problems that are experienced when the installation is exposed to heavy sunlight. The day the test was performed was actually mildly overcast. The conclusion is that the installation is not able to measure reliably on bright summer days.

11.7 Conclusion

This chapter has described the measurement station on the quay. From the beginning, it was meant to measure the length as well as the width of the steel plates that arrive at the yard. Soon the installation was reduced to the width measurement. It has been shown that accurate measurements can be obtained with the given set-up. It is, however, only possible when the outdoor light is diffuse. When the sun shines on the plate edges, it causes reflections, that makes sub-pixel edge detection impossible. The installation might work day and night if the light boxes were replaced by some more powerful light sources that could compete with the strong reflections from the sunlight. This could possibly be strobed light, and the source would not have to have the physical size of the light boxes.

The installation in its current state has not been accepted in the production. It was decided to see what could be obtained by a mechanical solution on the width measurement as well.

Chapter 12

Check-In and Quality Control on Laser Cutter

12.1 Introduction

The most recent vision installation on the yard is designed to do accurate measurements on a new gantry station for laser cutting and welding. The high energy laser tool is expected to give more accurate cuts and welds as a result of better focusing and less heat injection. The alternative processes employed at the yard put large amounts of energy into the material during processing, causing a deformation of the assemblies. The laser tool is mounted on a very accurate gantry system. According to the specifications, the gantry can be positioned within 0.1 *mm*. The gantry and the laser cutter can be seen in figure 12.1. It is planned to use vision to measure the performance of the laser cutter.

The vision system serves several purposes. It is also providing results for the Brite-EuRam project **QualiGlobe**. This project focuses on the use of product state information in processing. That is, a product state model (PSM) is maintained as the product travels through the different steps in the production. Specifically, it is planned to estimate the width of the gap to be welded between two parts from the actual shape of the parts. It is therefore necessary to be able to measure in-plane deformations as well as out-of-plane deformations. Only the former task has been addressed so far, but it is planned to investigate what accuracy can be obtained on the out-of-plane measurements by extending the vision installation with an extra camera or a

CheckInAndQualityControlOnLaserCutter/Postscript/laserworkarea.eps

Figure 12.1: The laser cutting and welding installation. The table measures approximately $4500\,mm \times 18000\,mm$.

laser. The size of the gap is an important piece of information in terms of optimal time planning and optimal choice of process parameters.

The installation is at the same time a test bench where the results in the previous installations can be verified without too many unknown factors. In this project, the camera is much closer to the object to be measured and the lighting conditions are much more stable. Furthermore, the laser cutting tool performs clean cuts on the primed plates without burning the edges, so that the condition of the edges is well suited for vision measurement. It is therefore a perfect set-up to prove the potential of vision technology in shipbuilding.

A fast check-in system has also been implemented on the laser cutter to determine the position and orientation of plates on the table. The operator only has to jog the gantry camera to a corner of the plate, and the system will automatically detect the other corners and calculate the rotation and translation of the plate. This allows an arbitrary placement of the plates on the table. This system is already running in production.

12.2 Set-up

It was chosen to benefit from the accuracy of the gantry system by placing the camera on the gantry in this installation. This makes it possible to mount the camera much closer to the edge to be measured and thereby increase the image resolution significantly. The placement of the camera can be seen in figure 12.2 on the exterior of the hood on the vertical Z-axis. The camera is a digital 1296×1030 Basler A113, cf. chapter 3.

The work area measures approximately $4500\,mm \times 18000\,mm$, i.e. the camera must be moved around by the gantry to measure the plates. Apart from an initial pose calculation, the external calibration of the camera therefore relies on the position read-out on the gantry controller. It might be argued that this set-up cannot be transported to a gantry station, that is less accurate. However, the deviations from the ideal cut on the other cutting stations on the yard typically originate from damaged tools, poor process parameters, or plate deformations caused by the process and *not* the accuracy of the gantry. It is therefore normally fair to base a measurement on the precision of the gantry, which is in any case setting the lower limit on the performance of the process.

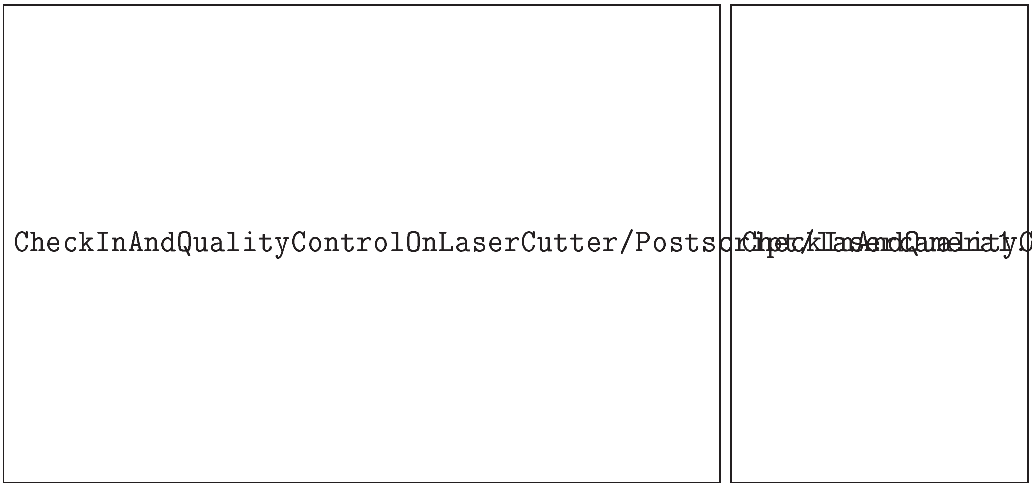
CheckInAndQualityControlOnLaserCutter/Postscript/LaserCamera
CheckInAndQualityControlOnLaserCutter

Figure 12.2: The camera placement on the gantry.

On the laser gantry system we operate with three different coordinate systems. First, we have the coordinate system of the cutting table $(\mathcal{O}, \mathbf{i}, \mathbf{j}, \mathbf{k})$. No subscript is used for this global coordinate system. Second, we have the coordinate system centered on the TCP $(\mathcal{O}_T, \mathbf{i}_T, \mathbf{j}_T, \mathbf{k}_T)$ with the subscript T.

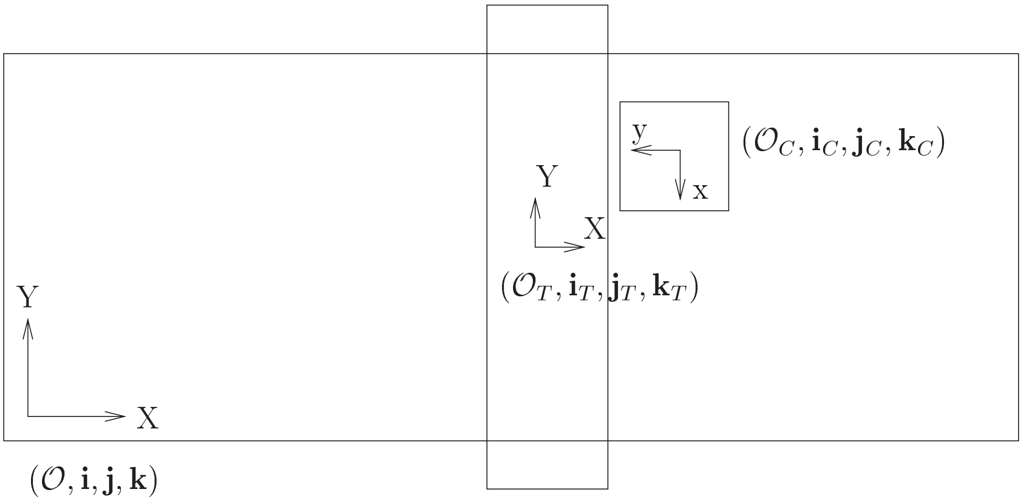
y
x
$(\mathcal{O}_C, \mathbf{i}_C, \mathbf{j}_C, \mathbf{k}_C)$
Y
X
$(\mathcal{O}_T, \mathbf{i}_T, \mathbf{j}_T, \mathbf{k}_T)$
Y
X
$(\mathcal{O}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Figure 12.3: The three coordinate systems at the laser station. From left to right: the gantry frame, the TCP frame, and the camera frame.

The TCP is located at the origin of this frame. Finally, we have the coordinate system of the camera $(\mathcal{O}_C, \mathbf{i}_C, \mathbf{j}_C, \mathbf{k}_C)$ with the subscript C. It is from this frame that the central projection to the image plane is performed. That is, the focal center is placed at the origin of this frame and the z-axis is the principal axis of the camera. In the camera model that we use, there is no skewness parameter, meaning that the x-axis is aligned with the horizontal u-axis of the image and the y-axis is aligned with the vertical v-axis. The three coordinate frames are depicted in figure 12.3.

The relation between the global coordinate system and the TCP-centered coordinate system is defined by a pure translation. In fact, this translation is directly related to the read-out on the gantry controller $\mathbf{X}_G(\tau)$ at time τ

$$\mathbf{X}_T(\tau) = \mathbf{X} - \mathbf{X}_G(\tau) \tag{12.1}$$

We assume that the TCP is positioned at the origin of the table when the read-out on the controller is zero, $\mathbf{X}_G(0) = (0, 0, 0)$. If, however, these two frames are not aligned, an offset can be introduced in the equation above. In the following we assume that the arbitrary origin of the coordinate system of the table is placed so that the equation holds, cf. figure 12.4.

The mapping between the TCP frame and the camera is described by the

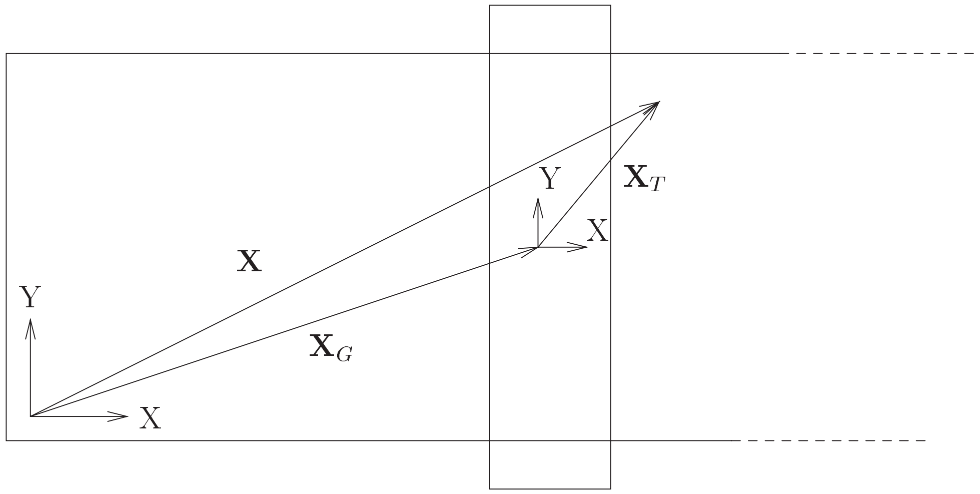

Figure 12.4: The relation between the global frame and the frame centered on the TCP.

camera's pose, i.e. rotation $\mathbf{R}$ and translation $\mathbf{t}$. This gives

$$\mathbf{X}_C(\tau) = \mathbf{R}\mathbf{X}_T(\tau) + \mathbf{t} \tag{12.2}$$

Note, that this relation holds independent of the gantry position and describes the fixed position of the camera in relation to the TCP, cf. figure 12.5. Naturally, we can combine equations 12.1 and 12.2 to get the direct mapping from the global coordinate system to the camera frame

$$\mathbf{X}_C(\tau) = \mathbf{R}(\mathbf{X} - \mathbf{X}_G(\tau)) + \mathbf{t} \tag{12.3}$$

The external calibration of the camera consists of finding $\mathbf{R}$ and $\mathbf{t}$, which are both independent of the gantry position. This is done by cutting e.g. a rectangle with the TCP and moving the camera to a gantry position $\mathbf{X}_G(\tau_{CAL})$ that makes the rectangle visible in the image. The coordinates of the gantry position are then subtracted from the coordinates of the rectangle to obtain the rectangle in the TCP coordinate system. The pose of the camera is then easily calculated from the projection of the rectangle. The relation between the global coordinates of the rectangle and its image projection is then described by the calibration matrix

$$\mathbf{P}(\tau_{CAL}) = \mathbf{K}\left[\mathbf{R}\ \ \mathbf{t}\right]\mathbf{T}_G^{-1}(\tau_{CAL}), \tag{12.4}$$

where

$$\mathbf{T}_G(\tau_{CAL}) = \begin{bmatrix} \mathbf{I} & \mathbf{X}_G(\tau_{CAL}) \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{12.5}$$

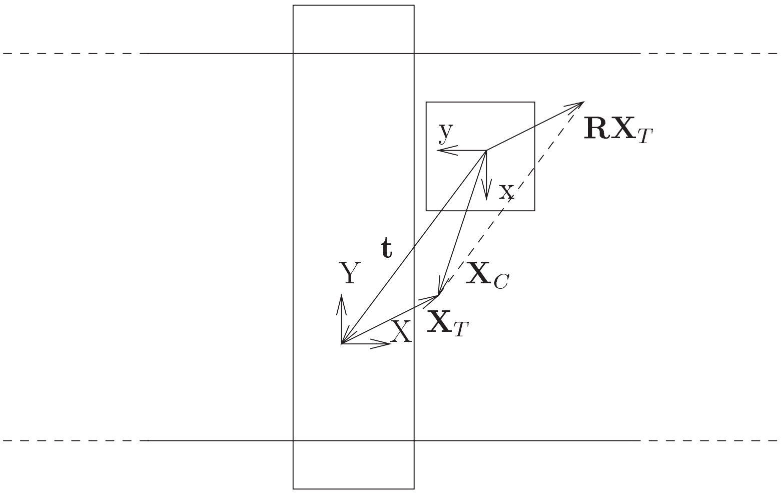

Figure 12.5: The relation between the frame centered on the TCP and the camera frame.

is the translation represented by a homogeneous matrix, and $\mathbf{K}$ contains the internal calibration, that is determined off-line. This formulation is used to calculate the appropriate calibration matrix every time an image is acquired from a new gantry position $\mathbf{X}_G(\tau)$

$$\begin{aligned} \mathbf{P}(\tau) &= \mathbf{K}\,[\mathbf{R}\;\;\mathbf{t}]\,\mathbf{T}_G^{-1}(\tau) && (12.6)\\ &= \mathbf{P}(0)\mathbf{T}_G^{-1}(\tau), && (12.7) \end{aligned}$$

where $\mathbf{P}(0)$ is the gantry-independent camera matrix that maps from the TCP frame into the image.

12.3 Gantry and Measurement Accuracy

A number of experiments have been conducted to verify the absolute as well as the relative accuracy of the gantry system. The absolute accuracy was tested by a theodolite system by moving the various axes of the gantry and comparing the theodolite measurements and the gantry controller read-out. At this point in time, the measurement data from the gantry are very sparse, but it seems that the absolute accuracy of the gantry is far from what is promised in the specifications. In an experiment, the X-axis of the gantry, that moves in the longitudinal direction of the cutting table, was asked to move 8500 mm. The distance that was actually moved, was measured by

Coordinate		Mean ± Std.Dev.	[Min;Max]	Max-Min
Corner 1	X	9090.96 ± 0.03	[9090.93;9091.02]	0.09
	Y	2204.10 ± 0.03	[2204.04;2204.16]	0.12
Corner 2	X	11697.23 ± 0.03	[11697.16;11697.30]	0.14
	Y	3761.97 ± 0.03	[3761.92;3762.03]	0.11

Table 12.1: The relative gantry accuracy on the laser in two times 40 measurements. The test also shows the reproducibility of the vision system. All numbers are in millimeters.

theodolite to be 8498.6 *mm*, i.e. the gantry lost 1.4 *mm*. At the same time the Z-coordinate was seen to increase by 1.7 *mm*. Discrepancies between the theodolite measurements and the gantry controller were also observed when moving the other axes. It is necessary to perform more measurements to give definite conclusions on the accuracy of the gantry, but if the first indications are verified, the gantry needs to be better calibrated. This is, of course, important for the vision system, but it is even more crucial for the cutting and welding on the installation. The gantry is assumed to be properly calibrated in the sequel.

The relative accuracy was tested with the vision camera by repeated measurements of two diagonal corners of a plate. The gantry was programmed to move back and forth between the two corners. The corners were each time approached from a different direction to best analyse the behaviour of the gantry. The position of the two corners were each measured 40 times with the method described in section 12.5 below. The results of the measurements are summarized in table 12.1. Note, that this test also verifies the reproducibility of the vision measurement. It is seen that all measurements lie within 0.14 *mm* with a standard deviation of 0.03 *mm*, even though the gantry moves 10500 *mm* between each measurement. The image resolution is approximately 2 *pixels/mm*. The test lasted an hour and was performed around noon.

Finally, the performance of the vision system was tested as a function of time. This was done by leaving the gantry in a fixed position for 20 hours, while measuring the same corner every half hour. The experiment tests the sensitivity to changes in illumination. The result is shown in figure 12.6. The upper plot shows the variation of the two coordinates around their mean, and the lower plot shows the variation of the shutter setting. Measurements started at 1:40 *p.m.* It is seen that the position varies smoothly as a function of time. The change in illumination cause the estimate to drift 0.3 *mm* from the minimum to the maximum value.

12.4 Calibration

The internal calibration of the camera is naturally the same as in the other installations. With regard to the external calibration, the camera needs to be calibrated with respect to the tool center point (TCP), since this installation is meant to produce absolute measurements. The external calibration is obtained by letting the TCP cut e.g. a rectangle and using the corners to make a pose calculation. This procedure can be performed in any place in the work area, since the calibration is invariant to the gantry position and only depends on the relation between the camera and the TCP. In practise, we obtained calibration data from each end of the table and optimized a single pose estimate using the data simultaneously. This way, the calibration is equally valid at both ends of the table, even if the pose of the camera is slightly affected by the different gantry positions. The residual of the pose calculation was 0.56 *pixel*, which is way too high. This is known to be caused by a varying Z-coordinate of the corners of the rectangles in the calibration images. This will be corrected in the future.

To benefit from the simplicity of projective algebra, it was chosen to work with rectified images in this installation. Consequently, all images are rectified right after acquisition. The calibration is therefore described by a 3×4 camera matrix derived from the individual intrinsic and extrinsic parameters. That is, the entries of the matrix were *not* estimated by the linear method. The result of the calibration and the separation into intrinsic and extrinsic parameters is

$$\mathbf{P}(0) \sim \tag{12.8}$$

$$\begin{bmatrix} -1.766e-02 & -4.803e+00 & -1.173e+00 & 5.552e+02 \\ -4.828e+00 & -3.274e-02 & -7.317e-01 & 3.195e+03 \\ -6.925e-05 & -7.838e-06 & -1.753e-03 & 1.000e+00 \end{bmatrix} \sim \tag{12.9}$$

$$\begin{bmatrix} 2734.78 & 0.00 & 680.50 \\ 0.00 & 2733.94 & 525.53 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \begin{bmatrix} 0.006 & -1.000 & 0.004 & -26.11 \\ -0.999 & -0.006 & 0.040 & 556.57 \\ -0.039 & -0.004 & -0.999 & 570.06 \end{bmatrix} \tag{12.10}$$

CheckInAndQualityControlOnLaserCutter/Postscript/XYVsTime.eps

CheckInAndQualityControlOnLaserCutter/Postscript/ShutterVsTime.eps

Figure 12.6: Corner position vs. time on the laser cutter. Top: the variation of the X and Y position around the mean. Bottom: the shutter value variation. Time is measured from 1:40 $p.m.$

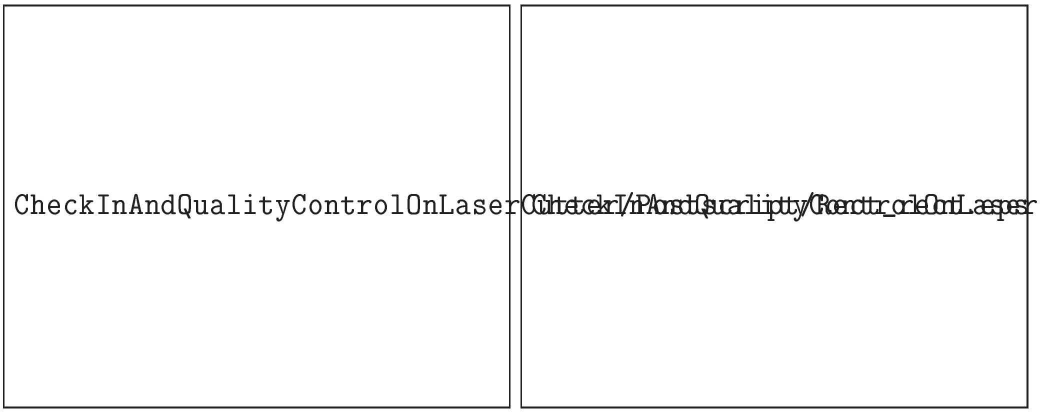

Figure 12.7: The two rectified images that are used for the external camera calibration on the laser cutting installation.

The entries of these matrices can easily be interpreted as described in chapter 5. The right-hand matrix in the last equation contains the rotation and the translation of the camera pose. The last entry in the translation vector shows that the TCP is 570 mm below the camera. During measurement, the camera is lifted to the gantry position $Z_G = 890\,mm$, so that the total distance to the plate is approximately $570 + 890 = 1460\,mm$. It can therefore be seen from the intrinsic camera parameters that the image resolution is $2734\,pixels/1460\,mm = 1.9\,pixels/mm$. The rotation matrix reveals the orientation of the camera. The object space X-axis points in the direction of the negative y-axis of the camera, the Y-axis points in the direction of the negative x-axis, and the Z-axis points opposite the camera's z-axis. The placement of the TCP in relation to the focal point is $(-26.1\,mm, 556.6\,mm, 570.1\,mm)$ in the camera coordinate system. This is in good agreement with the physical set-up, cf. figure 12.2. That is, the TCP is in front of the camera but translated 556.6 mm in the direction of the camera's y-axis. Some information can also be extracted directly from the camera matrix $\mathbf{P}$. The projection of $\mathbf{X}_T = (0, 0, 0)$ gives the (virtual) image coordinates of the TCP $(555.2\,pixels, 3194.8\,pixels)$. That is, the TCP is centered in the u-direction and is two image heights beyond the bottom of the image. The direction of the principal axis is given by the first three entries in the bottom row of the camera matrix or the bottom row of the rotation matrix $(-0.039, -0.004, -0.999)$. That means that an error in the plate's height of 10 mm will result in a parallax error in the measurement of $(0.39\,mm, 0.04\,mm)$ in the principal point $(681\,pixels, 526\,pixels)$. Experiments have shown that the cutting table is not even, and that the height variation amounts to as much as 10–15 *mm*. Measurements are therefore always performed in the point where there is no parallax, namely the point right below the focal point. We call this point nadir and find it as the vanishing point on the Z-axis in object space. The image coordinate is determined by the third column of the camera matrix and is $(-1.17, -0.73)/-0.00175 = (669\,pixels, 417\,pixels)$. The image center, the principal point, and nadir are specified in figure 12.8. Remembering the sensitivity to height variations in the principal point, it is remarkable how fast the precision degrades away from nadir.

Figure 12.8: Image center, principal point, and nadir in an image from the laser station.

12.5 Check-In

As mentioned above, a check-in system has been implemented on the laser cutter. This system is used to determine the position and orientation of plates on the cutting table, so that the cutting programs can be properly transformed. This gives the freedom to place objects to be cut or welded arbitrarily on the table, allowing more flexibility in the process planning. The system only requires the operator to jog the camera to a predefined corner on the plate. The rest of the check-in is then performed automatically. The steps are

1. receive approximate dimensions from the controller
2. prompt the operator to jog the camera to the first corner
3. measure the corner position and orientation
4. calculate the next corner position
5. move the gantry automatically
6. measure the corner position and orientation
7. if *not* last corner, goto 4
8. calculate translation and rotation
9. report result to controller

The user interface can be seen in figure 12.9. In the left-hand side of the image, the operator selects the plate to be checked in from the database of cutting program files. He then clicks the corners that he wants to measure in the graphic display to the right, and indicates whether the plate is lying in one of two fixed positions or is placed arbitrarily. Finally, he starts the program, that will ask him to jog the camera if necessary.

The actual detection of the corner in the image is based on a number of image analysis algorithms. The task is different from the earlier edge detections in the limited amount of a priori knowledge. The only information that is given to the vision system is the opening angle of the corner and the approximate orientation. This information is contributing to an error rate that is practically zero. The difficulty in the task lies in the grid below the plate, which produces many false edges and highlights.

The different steps in the image analysis are shown from left to right in figure 12.10. First, the image is median filtered with a large 39×39 filter. The size of this filter is determined to remove the underlying grid. The plate can

CheckInAndQualityControlOnLaserCutter/Postscript/LaserScrdump.eps

Figure 12.9: User interface for plate check-in on the laser station.

now be separated from the background by a simple threshold. The threshold value is determined automatically to best separate the intensity distributions in the foreground and the background. Next, the edge pixels are found as the difference between the thresholded image and an eroded version of the thresholded image. This image is then Hough transformed to find candidate edges. Only the eight strongest edges are considered. Each time an edge is found, it is drawn in the image and the contribution to the Hough space from the edge pixels nearby is removed. This way the multiple representation of the same line, that is often the result of aliasing in the Hough space, is avoided. These eight lines are then paired two by two to form corners. Among the corners that satisfy the a priori knowledge, i.e. corners that have a given opening angle and orientation, a measure of fit is calculated. This measure is based on the number of edge pixels that is actually covered by the corner and the total edge length of the corner. The detected edges are shown in dark grey in the fifth image of figure 12.10 and the corner with the highest score is drawn in light grey. The best candidate is then refined by a more accurate edge detection method. On this installation, the accurate edge detection is based on the maxima of one-dimensional Gaussian convolutions with subsequent linear regression with outlier detection. The result is shown in the last image in the figure. Actually, the method just described has recently been modified to perform the coarse corner detection on an image, that is down-sampled twice. That is, the reduced image contains sixteen times fewer pixels. At the same time the size of the median filter can be reduced correspondingly. The final accurate detection is, of course, still performed on the original image, so that the improved computational efficiency does not cause a decrease in the precision.

The accuracy and reproducibility of the check-in procedure was already described in section 12.3.

12.6 In-Plane Measurement

Once a plate has been cut, it is measured along the edge. The density and the organization of the measurements are determined by the desired detail of the PSM, which is managed by another group at the yard. The software, that is developed in the vision group, is ready to be interfaced with the PSM, but it also works as a stand-alone system, so that vision specific results can be tested off-line.

The interface to the vision system is very simple. The controller simply provides the two end points $(\mathbf{X}_1, \mathbf{X}_2)$ of an edge and the offset α from one of these points to the position $\mathbf{X}$ where the measurement is to be made. The offset is specified as a fraction of the total edge length, so that

$$\mathbf{X} = \mathbf{X}_1(1-\alpha) + \mathbf{X}_2\alpha \tag{12.11}$$

The vision system then calculates the gantry coordinates that will place the camera so that the point $\mathbf{X}$ is in the nadir position in the image. When the gantry is moved to this position, it is an easy task to detect the edge position. The orientation of the line in the image is approximately known from the specified end points, and these can also be used to restrict the edge detection near the corners. The edge detection is based on the available edge information in the image, i.e. the edge is detected as a line and not as a point. This line in the image is represented by the homogeneous coordinate $\mathbf{L}_{edge}$. In object space, the line and the focal point of the camera defines a plane $\mathbf{P}_{edge}$. Obviously, the plate edge is lying in this plane. It is calculated directly from

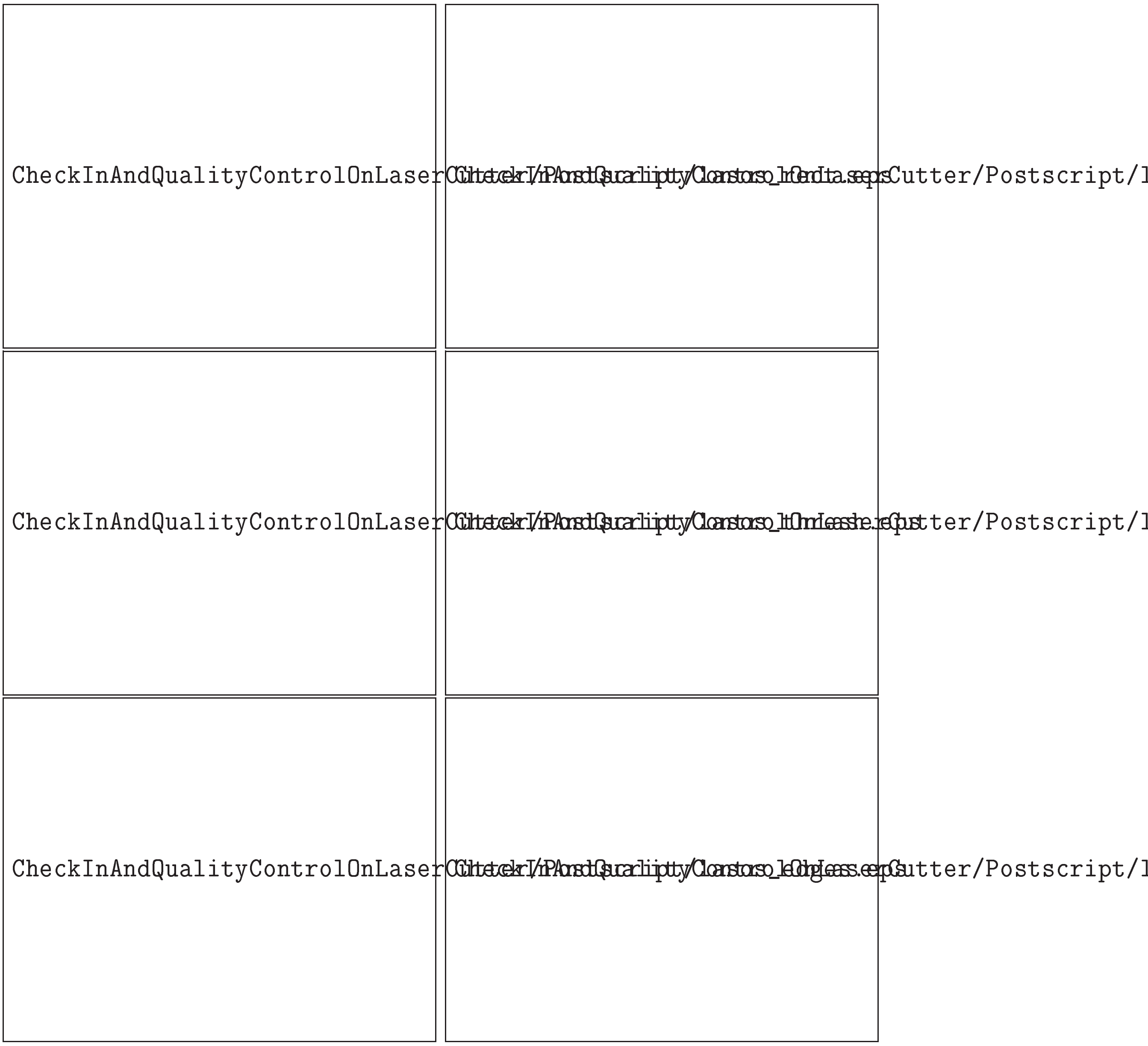

Figure 12.10: The different steps in the corner detection. From left to right: rectified image, median filtered image (39 × 39), thresholded image, eroded image (3 × 3), detected edges, and superposed corner. In the latter image a region has been blown up to show the sub-pixel position of the detected line.

the camera matrix $\mathbf{P}(\tau)$

$$\mathbf{P}_{edge} = \mathbf{L}_{edge}\mathbf{P}(\tau) \tag{12.12}$$

We have to use two more constraints to find the edge position corresponding to the point $\mathbf{X}$. The first constraint is the height h of the edge determined by the thickness of the plate, which defines the vertical location of the observed edge. That is, the edge is also constrained to the plane with the homogeneous coordinate

$$\mathbf{P}_{Z=h} = (0, 0, 1, -h) \tag{12.13}$$

Finally, it is required to find the edge location exactly corresponding to the point $\mathbf{X}$. We therefore construct a plane that intersects the edge in precisely that point. This plane is denoted $\mathbf{P}_{norm}$, since it is normal to the edge

$$\mathbf{P}_{norm} = (\mathbf{X}_2 - \mathbf{X}_1, -\mathbf{X}_2 \cdot \mathbf{X} + \mathbf{X}_1 \cdot \mathbf{X}), \tag{12.14}$$

where '$\cdot$' denotes the usual dot product. The desired edge location is now the intersection of these three planes, which is well-defined owing to the construction above. The homogeneous coordinate of the edge location may be found as the null-space of the matrix

$$\begin{bmatrix} \mathbf{P}_{edge} \\ \mathbf{P}_{Z=h} \\ \mathbf{P}_{norm} \end{bmatrix} \tag{12.15}$$

In the visualization of the results, we also use the orientation of the edge to draw a cubic spline between measurements. The orientation can be found by omitting the constraint defined by the plane $\mathbf{P}_{norm}$ above. That is, the edge is spanned by the two points that define the null-space of

$$\begin{bmatrix} \mathbf{P}_{edge} \\ \mathbf{P}_{Z=h} \end{bmatrix}. \tag{12.16}$$

The angle is straightforwardly calculated from these two points.

Figure 12.11 shows the result of a measurement. The edge position is measured relative to an ad hoc CAD model based on the four check-in corners. The bars, that are perpendicular to the CAD model, represent the individual measurements. They are interpolated by a cubic spline, which is based on the local edge direction estimates. The maximum deviation from the "CAD" model is seen to be 0.3 *mm*. In five repetitions of the plate measurement, the maximum deviation on the same point was 0.03 mm. The plate has not yet been measured accurately with the theodolite system, why it is not possible to evaluate the absolute accuracy of the measurements. Note, the continuity of the independent measurements along each edge.

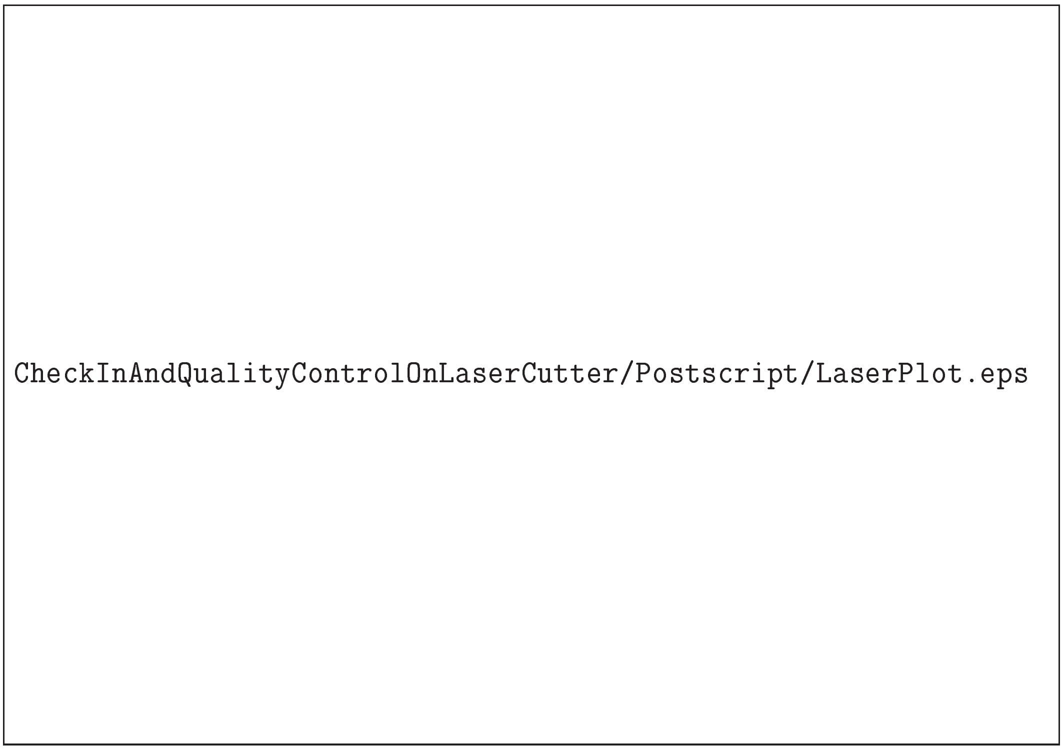
CheckInAndQualityControlOnLaserCutter/Postscript/LaserPlot.eps

Figure 12.11: Measurements along the plate edge on the laser cutting station. The bars show the measurements in relation to the rectangle based on the four check-in corners, which is denoted CAD. The spline is based on the local direction estimates. The graphic representation is exaggerated 100 times, but the numbers are the actual deviation in millimeters.

12.7 Conclusion

The laser installation has been the most successful project in the Ph.D. period. The implemented check-in system is already used in production, and the measurement system for the product state model (PSM) has shown promising results.

The check-in system is able to measure the corners within 0.1 mm when the corners are approached from varying directions. Edge measurements, where the direction of approach is always the same, have been reproduced within 0.03 *mm* in five tests. The absolute accuracy of the whole system remains to be shown, but preliminary theodolite measurements indicate that the absolute positioning of the gantry is not satisfactory. This is primarily a problem with respect to cutting and should be solved to benefit from the advantages of the laser tool. The tests show that the accuracy of the vision system should be comparable to that of the gantry system.

Chapter 13

Conclusion

The present dissertation has described the theory and practise of the work that has been done on large-scale object measurement with vision at Odense Steel Shipyard in the course of the project. The emphasis has been put on accurate dimension measurement and uncertainty evaluation. All the examples that have been given have concerned the measurement of flat steel plates, but the experience gained in the project in terms of geometry, calibration, lighting conditions etc., is also applicable more generally in the industry. Two immediate examples are the use of vision to position large three-dimensional assemblies and to provide a visual input to welding robots.

The dissertation has presented the basic algebra describing projective geometry and its relation to typical camera models. The factorization and interpretation of the linear models have been described. A complete non-linear model with radial and decentering lens distortion has been employed to obtain a highly accurate description of the relation between object points and image points. As a possible alternative to a complete rectification of the image, Newton iterations have been proposed as a means to find the inverse of the lens distortion function.

A chapter has described the problem of camera calibration with focus on a proper evaluation of the calibration quality. It is argued that the mere calibration residual is often an insufficient measure for the quality. It is shown how the theoretic dispersion of the fitted camera parameters can be propagated to uncertainty ellipses in the image. For instance, this shows that an extra decentering distortion parameter that gives a marginal improvement of the calibration residual results in a significant increase in the uncertainty of the model outside the convex hull of the calibration points. It is also proposed to analyze the local correlation structure of the residuals, which indicates if the camera model is under-parametrized. The various tests are performed in a calibration program, that has been developed in the course of the project. The cameras used at the yard have been successfully calibrated to an accuracy of 0.05–0.10 *pixel*.

Accurate detection of two kinds of features has been discussed. Circular reference marks have been detected by the use of a parametric model and it has been shown that the theoretical and empirical dispersions are comparable in magnitude. It was proposed to apply the a priori knowledge on the approximate pose of the camera to back-project the reference marks to a metric coordinate system, where the marks are isotropic. This adds to the stability of the system and seems more fair than to estimate a separate perspective distortion for each mark. The effect of a trend in the illumination has been analyzed leading to an appropriate extension of the model. To guide the design of reference marks, the relation between size and detection certainty was analyzed. It was shown that the certainty was proportional to the edge length for radial symmetric binary marks.

Two new methods based on simultaneous use of all image information were proposed for straight edge detection. The first method is based on smoothing the Hough space obtained by a grey level Hough transformation of the raw image. The involved theory gives valuable insight on the relationship between the Hough and the Radon transforms and shows that two-dimensional anisotropic image interpolation kernels can be applied as one-dimensional filters in Hough space. Bilinear and nearest neighbor interpolations were shown as examples. The second method is based on normalized Gaussian convolution of the projected image. The two new methods were compared to conventional sequential approaches where local edge detection and line fitting are performed in two separate steps. The comparative study showed that Gaussian convolution with subsequent linear regression with outlier detection was superior to the other methods in the test in both speed and accuracy.

Three approaches to averaging rotations that have been proposed in the literature have been compared theoretically. The two most popular methods are to represent the rotations by matrices or quaternions and simply calculate the barycentric mean. Both methods require some sort of subsequent correction to obtain proper rotations. The third method considers that rotations are represented on a non-linear manifold and minimizes a least squares cost function based on geodesic distances. It is shown that the approaches are theoretically very similar, and that the subsequent corrections of the two linear methods are justified by the formulation of an approximated least squares cost function.

A preliminary attempt to reconstruct a mock-up from the shipyard from a sequence of images has been performed. The test showed that feature extraction with no a priori knowledge is very difficult on a real industrial object. Nevertheless, the tracking of points lead to a fair reconstruction of the motion. The Euclidean reconstruction algorithm was shown to produce nice results on manually tracked points.

Finally, three installations that have been implemented at the yard were presented. In the two first applications, it was necessary to base the measurement on several cameras mounted approximately $10\,m$ away from the object. It was shown that those measurements were very sensitive to the varying lighting conditions and to the geometric stability of the installations. It was found to be very difficult to ensure or maintain a satisfactory geometric description of the reference marks and the plate tables. Nonetheless, the reproducibility studies showed that measurements could be performed within $1\,mm$ under favourable conditions. In the third installation the camera was mounted on the gantry 1–2 m from the plate. At that distance, the plates were measured with an accuracy better than $0.3\,mm$. The check-in system is currently used in the production on the laser cutting station.

The overall conclusion is that accurate measurements can actually be performed on the raw steel plates without reflectors or markers. But the measurements are very sensitive to the illumination of the scene, to the general state of the plates, and on the quality of the geometric model of the work area. Subpixel accuracy measurements can therefore only be achieved in very controlled set-ups. While this requires special precautions in large-scale installations, it is more easily obtained when the camera is placed close to the object. It is concluded that vision technology does have a potential in shipbuilding both as a tool for measurement and as a robot sensor.

Chapter 14

From Hough Transform to Radon Transform

From Hough Transform to Radon Transform using Interpolation Kernels

C. Gramkow

Odense Steel Shipyard Ltd. Lindø, Denmark

Department of Mathematical Modelling, Technical University of Denmark

E-mail: cgr@imm.dtu.dk

Accurate straight edge detection in images applies to dimension measurement, camera calibration, Euclidean reconstruction etc. Among the non-parametric robust approaches the Hough transform is probably the most popular. However, the interpretation of the Hough space tends to be qualitative more than quantitative. In this paper, a link is established between image interpolation, the Hough transform, and the Radon transform allowing for a quantitative interpretation of the projection space. This generalizes the notion of influence functions introduced in the context of the Hough transform. To support the theory it is shown that edge detection can be performed on the projected image. Furthermore, it is shown that lens blur and additive pixel noise parameters may be extracted in Hough space.

Key Words: Hough transform, Radon transform, high resolution, influence function, interpolation, sinc, nearest neighbor, bilinear, edge detection.

14.1 Introduction

It is often of great importance to be able to estimate features to a high sub-pixel accuracy in images. This paper will concentrate on straight edge detection, which is relevant for photogrammetric measurements, calibration of camera parameters, Euclidean reconstruction from stereo images and so forth.

A usual non-parametric scheme for straight line detection is to apply an edge detector proceeded by a threshold algorithm to extract the relevant edge points. The edge points are then used to perform a robust estimation of the edge parameters. The Hough algorithm (Hough [33], Duda and Hart [15]) has been widely used due to its robustness. An alternative method is to fit the parameters of an edge model defined by the edge location, the edge strength, and the edge width. However, this usually requires that the image is approximately uniform on the two sides of the edge to avoid a bias in the estimation.

Unfortunately, the discrete nature of the Hough space makes the peak detection very difficult and the problem becomes even worse when the resolution of the Hough space is increased as severe aliasing occurs. Several researchers have addressed the problems of obtainable precision and interpretability of the Hough transform. Accuracy has e.g. been analyzed by Niblack and Petkovic [52] and Shapiro and Iannino [66], who also investigated the peak spread by a geometrical construction. Peak bias due to side-lobes and the background distribution has been addressed by Brown [5], Leavers and Boyce [44], and Maitre [47]. The use of influence functions to regularize the Hough space was introduced by Thrift and Dunn [68]. Influence functions has been applied in a hypothesis testing framework by Palmer et al. [55, 56] and Princen et al. [62]. In Kiryati and Bruckstein [38, 39, 40] the influence function was optimized with respect to the frequency properties of the Hough space. A probabilistic Hough transform was also defined by Hunt et al. [34] and Stephens [67], who was inspired by the work of Weiss [75]. Illingworth and Kittler [35] produced a thorough survey of the Hough transform research before 1988.

Deans [10] emphasized already in 1981 the strong relation between the Hough transform and the Radon transform. Some theory on the Radon transform is given in Rosenfeld and Kak [65] and an implementation is proposed in Hinkle et al. [27]. Much research has been carried out on the Radon transform due to its application to medical imaging, radioastronomy etc.

In this paper, the link between the Hough transform, the Radon transform, and the use of influence functions as mentioned above is studied. It is shown that performing a Radon transform on an interpolated image is equivalent to applying an influence function to the Hough transform of the image. This leads to the design of influence functions that facilitate the interpretation of the Hough space. In addition, it is shown that important statistical measures on the edge can be estimated from the projection.

To best illustrate the implications of the theory, the Hough transform is applied to the raw image data in the sequel. Although, only a subregion of the images is processed, this is naturally computationally expensive. However, the theory is also applicable in the normal set-up, where the Hough transform is based on the edge pixels of a pre-processed image.

An advantage of working on the raw image is that the edge detection is performed globally without any intermediate thresholds. In conventional edge detection all higher order information about the edge structure is disregarded through the thresholding performed prior to the Hough transform. Actually, the presented theory can be applied to analyze any straight structure in an image.

14.2 Notation

To make the presentation easier to read the following notation is introduced. Image intensity functions and filters have lower case names, e.g. $f(x, y)$. The image coordinates are denoted (x, y) or sometimes (X, Y) to emphasize discrete coordinates. Subscript x and y are used to distinguish the two factors of a separable function. The polar line parameters are (ρ, θ) and the line normal is denoted $\mathbf{n}_\theta = (\cos\theta, \sin\theta)$. The Hough transform is denoted by $\mathcal{H}\{\}$ or superscript h. It is considered a function of ρ and the dependency on θ is expressed by subscript θ. Similarly, the Radon transform is $\mathcal{R}\{\}$ or superscript r. $\mathcal{F}\{\}$ is the Fourier transform and the resulting function is denoted by a capital letter $F(u, v)$. It has rectangular coordinates (u, v) and polar coordinates (ω, θ). Where the Gaussian function is used the scale parameter is denoted by σ. Dirac's δ-function is widely used. It is zero everywhere except at zero where it goes to infinity. Across zero it integrates to one. When nothing else is specified, integrals have the limits $]-\infty; \infty[$. The operator '$*$' denotes spatial convolution. Superscripts 0 and 2 show that the image intensities are replaced by the value one or has been squared, respectively.

14.3 Image Material

The present paper will focus on the problem of finding accurate parameters for straight edges with almost known location. This is a typical set-up in industrial applications, where approximate line parameters can be obtained

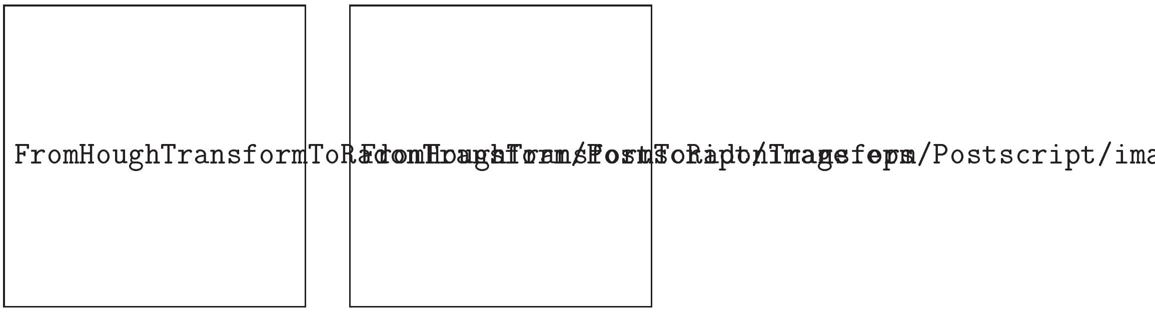

Figure 14.1: Images of an edge defined by $4 = 0.8x - 0.6y$. The edge in the left image has a width defined by $\sigma^2 = 0.5$ and no added noise. The right image was produced with $\sigma^2 = 2.0$ and additive independent Gaussian white noise with standard deviation 5.0.

from the ideal CAD model of the scene. Consequently, the problems sometimes encountered with multiple close lines will not be addressed in this paper.

The derived algorithms will be demonstrated on two very simple images. They both show an edge given by the equation $4 = 0.8x - 0.6y$. x is positive to the right and y is positive down. The intensity variation over the edge follows the accumulated Gaussian from 50 on the dark side to 200 on the light side. In one image the edge has a width of $\sigma^2 = 0.5$ and in the other image the width is $\sigma^2 = 2.0$. In the latter, independent Gaussian white noise with standard deviation 5.0 has been added. The edge images are shown in figure 14.1.

14.4 The Radon and Hough Transforms

In the following the very close relationship between the Radon and the Hough transforms will be explained.

14.4.1 The Hough Transform

We assume the normal representation of a line in the Hough transform as proposed by Duda and Hart [15]

$$\rho = x \cos\theta + y \sin\theta \tag{14.1}$$

In the Hough transform every image point (x, y) votes for possible line parameter values (ρ, θ) in parameter space through the relation (14.1). That is, the

Hough space consists of the superposition of all the sinusoids formed by the image points. Consequently, the cell associated with the line parameters (ρ, θ) holds a count of all image points along that line. If the digital image $f(X, Y)$ is defined on a discrete raster indexed by the integer coordinates (X, Y) on the domain $\{0, \ldots, D_X - 1\} \times \{0, \ldots, D_Y - 1\}$ the Hough transform is defined by

$$\begin{aligned} f_\theta^h(\rho) &= \mathcal{H}\{f(x, y)\} && (14.2) \\ &= \sum_{X=0}^{D_X-1} \sum_{Y=0}^{D_Y-1} f(X, Y)\, \delta(\rho - X\cos\theta - Y\sin\theta) && (14.3) \end{aligned}$$

The δ-function effectively restricts the summation to the pixels that satisfies (14.1). In practice, the Hough transform is represented by a two-dimensional accumulator array indexed by ρ and θ.

The continuous equivalent to (14.3) would be to integrate the image intensity along a certain line. That is what the Radon transform does.

14.4.2 The Radon Transform

Formally, the Radon transform is given by

$$\begin{aligned} f_\theta^r(\rho) &= \mathcal{R}\{f(x, y)\} && (14.4) \\ &= \iint f(x, y)\, \delta(\rho - x\cos\theta - y\sin\theta)\, dx\, dy && (14.5) \end{aligned}$$

The transform is defined as an integral over the entire two-dimensional domain, but the δ-function picks out the integral along the line satisfying (14.1). The two-dimensional Fourier transform of the image function

$$\begin{aligned} F(u, v) &= \mathcal{F}\{f(x, y)\} && (14.6) \\ &= \iint f(x, y)\, e^{-j2\pi(ux+vy)}\, dx\, dy && (14.7) \end{aligned}$$

is closely related to the the one-dimensional Fourier transform of the Radon transform

$$\begin{aligned} F_\theta^r(\omega) &= \mathcal{F}\{f_\theta^r(\rho)\} && (14.8) \\ &= \int f_\theta^r(\rho)\, e^{-j2\pi\omega\rho}\, d\rho && (14.9) \end{aligned}$$

If we denote the polar representation of $F(u, v)$ by $F(\omega, \theta)$, introducing $u = \omega\cos\theta$ and $v = \omega\sin\theta$, the Fourier slice theorem, a.k.a. the projection slice theorem, states that

$$F(\omega, \theta) = F_\theta^r(\omega) \tag{14.10}$$

The theorem follows from straightforward calculation. The Fourier slice theorem will be be applied extensively in the following. But first the relation to the Hough transform is established.

14.4.3 From Hough Transform to Radon Transform

We take the Radon transform of the discrete image function $f(X,Y)$ and show that we arrive at the Hough transform. To do so we need a representation of the image on the continuous domain defined by x and y. This is obtained by the use of δ-functions

$$f(x,y) = \sum_{X=0}^{D_X-1} \sum_{Y=0}^{D_Y-1} f(X,Y)\, \delta(x-X)\, \delta(y-Y) \tag{14.11}$$

We can now take the Radon transform of the image by inserting this expression in (14.5). The Radon transform evaluates to

$$\begin{aligned}
&\mathcal{R}\{f(x,y)\} \\
&= \iint \sum_{X=0}^{D_X-1} \sum_{Y=0}^{D_Y-1} f(X,Y)\, \delta(x-X)\, \delta(y-Y)\, \delta(\rho-x\cos\theta-y\sin\theta)\, dx\, dy && (14.12)\\
&= \sum_{X=0}^{D_X-1} \sum_{Y=0}^{D_Y-1} f(X,Y) \iint \delta(x-X)\, \delta(y-Y)\, \delta(\rho-x\cos\theta-y\sin\theta)\, dx\, dy && (14.13)\\
&= \sum_{X=0}^{D_X-1} \sum_{Y=0}^{D_Y-1} f(X,Y)\delta(\rho-X\cos\theta-Y\sin\theta) && (14.14)\\
&= \mathcal{H}\{f(X,Y)\} && (14.15)
\end{aligned}$$

In fact, the two transforms are identical in the generalized representation. In consequence, the results derived for the Radon transform may be readily applied to the Hough transform as well. This is used in the following section.

For later reference the Hough transform of the left image in figure 14.1 is shown to the left in figure 14.2. The Hough space does not resemble the usual Hough spaces, as the transform is based on the raw image and not on the edge pixels of a processed image. Vertically $\Delta\theta \in [-5;5]$ is discretized in 201 cells and horizontally $\Delta\rho \in [-7.5;7.5]$ is discretized in 301 cells around the true line parameters. The resulting resolution for ρ is 0.05 *pixel/cell*. The image points contributing to the transformation are extracted from a twenty pixel wide rectangular band from $(x,y) = (20,20)$ to $(x,y) = (80,100)$ giving an integration across a length of about 100 pixels. Note, the strong aliasing in the

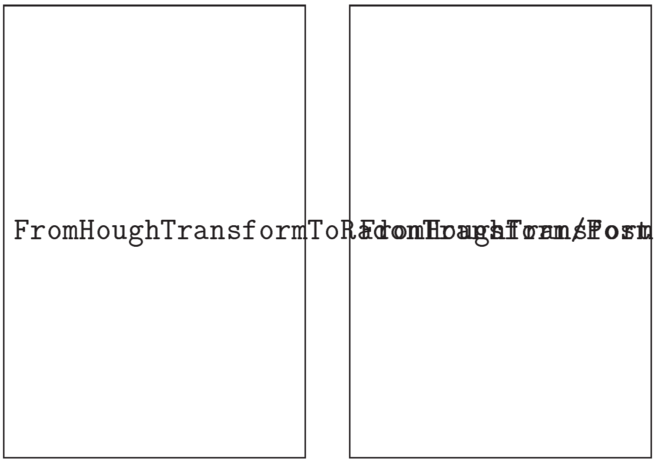

Figure 14.2: Hough transform and normalization image for the left image in figure 14.1. Vertically $\theta \in [-5; 5]$ is discretized in 201 cells and horizontally $\rho \in [-7.5; 7.5]$ is discretized in 301 cells around the true line parameters. The resulting resolution for ρ is 0.05 *pixel/cell*. Note, the strong aliasing caused by the image raster.

parameter space caused by the image raster. In fact, columns corresponding to lines with rational slopes are easily identified. In the center of the Hough space the true edge with the slope 4/3 is seen and further to the right the line given by 3/2 gives rise to even worse aliasing. This clearly shows that the Hough transform without careful peak detection will favour lines with a rational slope. If all pixels in the image are assigned the value 1 prior to the transformation, we obtain the right image in the figure. As explained later this image is used for normalization.

14.5 Linearity of the Radon transform

It follows directly from the Fourier slice theorem that the Radon transform is linear. Since the linearity will be used to derive the algorithms in the rest of the paper, the proof is given here. Assume $h(x, y) = f(x, y) * g(x, y)$. The Radon transform of $h(x, y)$ is then given by

$$\begin{aligned} h^r_\theta(\rho) &= \mathcal{F}^{-1}\{H^r_\theta(\omega)\} && (14.16) \\ &= \mathcal{F}^{-1}\{H(\omega, \theta)\} && (14.17) \\ &= \mathcal{F}^{-1}\{F(\omega, \theta)\ G(\omega, \theta)\} && (14.18) \end{aligned}$$

$$\begin{aligned} &= \mathcal{F}^{-1}\{F^r_\theta(\omega)\ G^r_\theta(\omega)\} && (14.19)\\ &= f^r_\theta(\rho) * g^r_\theta(\rho), && (14.20)\end{aligned}$$

where $\mathcal{F}^{-1}\{\}$ in all equations refer to the one-dimensional inverse Fourier transform with respect to ω. A consequence of the linearity is that any convolution performed in image space may just as well be applied in Radon space by convolving with the Radon transform of the two-dimensional kernel. Remembering that the Hough transform is merely the Radon transform of a discrete function, it is now clear that the Radon transform of a two-dimensional convolution kernel can be identified with the influence functions introduced by Thrift and Dunn [68] and later addressed by Kiryati and Bruckstein [38, 39, 40], Palmer et al. [56, 55], and Princen et al. [62].

In relation to edge detection, or rather edge analysis, it should be noted that once we have a non-aliased high resolution projected representation of the edge, we can apply any of the non-parametric or parametric one-dimensional edge detection theories that have been suggested in the literature, see e.g. Canny [6], Petrou and Kittler [59], Deriche [12], or Åström [2].

14.6 Separable Convolution Kernels

The linearity of the Radon transform can be used to find the transform of an interpolated discrete image, since interpolation is simply obtained by convolution. One may expect that the Radon transformation of the bilinear, the bicubic or the sinc interpolation kernels gives rise to complicated integrals. This is actually not so, as the Radon transform becomes particularly simple for separable functions. A separable function $h(x, y)$ can be written as

$$\begin{aligned} h(x,y) &= h_x(x)\ h_y(y) && (14.21)\\ &= (h_x(x)\ \delta(y)) * (\delta(x)\ h_y(y)), && (14.22)\end{aligned}$$

such that the linearity of the Radon transform can be applied. The Radon transform of each of the two terms in (14.22) has already been given by Deans [10]. For the first term the result is

$$\begin{aligned} \mathcal{R}\{h_x(x)\ \delta(y)\} &= \iint h_x(x)\ \delta(y)\ \delta(\rho - x\cos\theta - y\sin\theta)\ dx\ dy && (14.23)\\ &= \int h_x(x)\ \delta(\rho - x\cos\theta)\ dx && (14.24)\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{|\cos\theta|}\int h_x(x)\,\delta(\frac{\rho}{\cos\theta}-x)\,dx && (14.25)\\
&= \frac{1}{|\cos\theta|}h_x(\frac{\rho}{\cos\theta}) && (14.26)
\end{aligned}
$$

It is obvious from the above integrals that when $\cos\theta$ becomes zero the Radon transform will only be non-zero for $\rho = 0$ and the value will be the area under $h_x(x)$. For interpolation kernels this area will usually be 1. Similar to (14.26), the scale change of the second term in (14.22) is related to the angle of projection

$$\mathcal{R}\{\delta(x)\,h_y(y)\} = \frac{1}{|\sin\theta|}h_y(\frac{\rho}{\sin\theta}) \tag{14.27}$$

As a result the Radon transform of the image signal $f(x,y)$ interpolated by the separable kernel becomes

$$\mathcal{R}\{f(x,y) * h(x,y)\} = f_\theta^h(\rho) * \frac{1}{|\cos\theta|}h_x(\frac{\rho}{\cos\theta}) * \frac{1}{|\sin\theta|}h_y(\frac{\rho}{\sin\theta}) \tag{14.28}$$

Note, that the convolutions are performed directly on the usual Hough transform. Also note, that the kernel only changes scale, such that the scheme immediately applies to any separable kernel. Furthermore, most of the convolution kernels investigated in the following are easily optimized.

14.6.1 Sinc Interpolation

For theoretical reasons it is interesting to investigate the sinc interpolation kernel. The discrete image is a sampled version of the continuous image that passes through the camera lens. Assuming that the imaging system has smoothed the continuous signal so that there are no frequency components above the Nyquist frequency associated with the image raster, no loss of information will occur in the sampling process. That is, it is possible to reconstruct the continuous signal from the sampled signal by ideal low-pass filtering. The ideal low-pass filter is given by the sinc function. The sinc interpolation kernel is

$$
\begin{aligned}
\mathrm{sinc}\,(x,y) &= \mathrm{sinc}\,(x)\,\mathrm{sinc}\,(y) && (14.29)\\
&= \frac{\sin\pi x}{\pi x}\cdot\frac{\sin\pi y}{\pi y} && (14.30)
\end{aligned}
$$

It is seen that the two-dimensional kernel is separable. The filter in Hough space, that corresponds to this interpolation is obtained directly from (14.28)

$$\begin{aligned} \mathrm{sinc}\,_\theta^r(\rho) &= \frac{1}{|\cos\theta|}\mathrm{sinc}\,(\frac{\rho}{\cos\theta}) * \frac{1}{|\sin\theta|}\mathrm{sinc}\,(\frac{\rho}{\sin\theta}) & (14.31) \\ &= \frac{1}{w}\mathrm{sinc}\,\frac{\rho}{w}, & (14.32) \end{aligned}$$

where $w = max\{|\cos\theta|, |\sin\theta|\}$. The wider kernel completely cancels the narrower kernel due to the ideal nature of the sinc function. Consequently, the effect in Hough space of a two-dimensional sinc interpolation of the image can be obtained by a single convolution in the ρ-direction of the Hough space. Note, that the width of the sinc kernel varies with the angle θ. Actually, the Hough space is ideally low-pass filtered at the effective frequency of the image raster, namely the frequency of the raster seen under the angle θ. The result also shows that even though we need a high resolution Hough space when drawing the sinusoids to reduce the effect of rounding errors, we can immediately after the sinc reconstruction re-sample the Hough space at the effective frequency. This could be of interest if the representation was to be stored.

The Radon transform of the edge images in figure 14.1 interpolated by the sinc function, implemented as a convolution in Hough space, are shown in the top row of figure 14.5. The figure to the left corresponds to the image without noise whereas the figure to the right corresponds to the noisy image. The Radon space value is displayed as a height in the three-dimensional figures. The data has been normalized using the right image in figure 14.2 as described in a later section. As expected, the edge is steepest when integrated along the true direction of the edge corresponding to $\theta_{idx} = 101$. The edge profiles corresponding to this θ-value are displayed in the top row of figure 14.6. These plots show some rippling caused by the truncation of the very wide sinc function. For the same reason, there is a bias on the estimate of the intensity on the two sides of the edge.

14.6.2 Nearest Neighbor Interpolation

For practical purposes the sinc function is often found to be too badly localized, i.e. we prefer filters that are not too wide in the spatial domain to reduce the computational cost of the convolution. The most simple interpolation

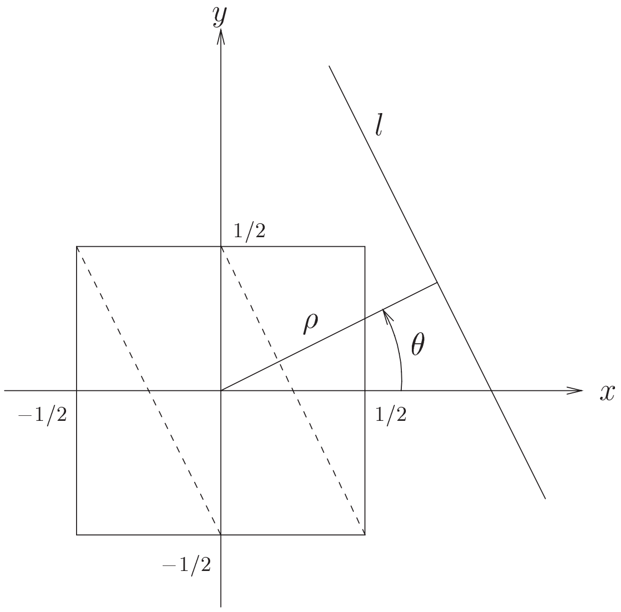

Figure 14.3: The Radon transform is obtained by the integral over l of the uniform function.

scheme is the nearest neighbor interpolation where the image value at non-integer positions are simply obtained from the closest integer position. This is equivalent to convolving the image with the uniform kernel

$$\mathrm{uni}\,(x,y) = \begin{cases} 1\,, & (x,y) \in [-\frac{1}{2};\frac{1}{2}] \times [-\frac{1}{2};\frac{1}{2}] \\ 0\,, & else \end{cases} \tag{14.33}$$

Deans [10] has already derived the Radon transformation of this kernel, however shifted to the first quadrant. The Radon transform of the uniform kernel can be obtained both by use of (14.28) and by geometric interpretation. The latter is included here to verify the analytic result. The set-up is shown in figure 14.3 for $\theta \in [-\pi/4;\pi/4]$. It follows directly from the drawing that as the value of ρ increases $\mathrm{uni}\,^r_\theta(\rho)$ will increase linearly from zero, then remain at a constant value, ending with a linear decrease back to zero. The size of the respective regions and the magnitude of the integral follows from simple trigonometry. The function is drawn in figure 14.4, where the dimensions are also given. The figure suggests that the function may be separated into two uniform kernels of width $|\cos\theta|$ and $|\sin\theta|$. Since the separation is symmetric in $|\cos\theta|$ and $|\sin\theta|$ the derived integral is valid for general values of θ. That

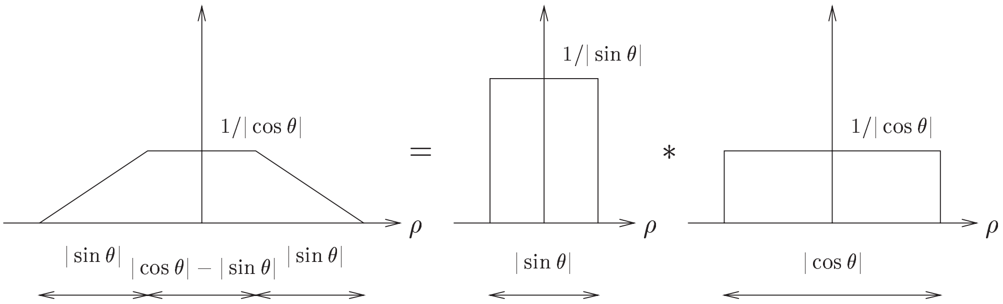

Figure 14.4: The integral over l as a function of θ and the separation into two uniform kernels.

is, the geometric result is in agreement with the analytic result

$$\mathrm{uni}\,^r_\theta(\rho) = \frac{1}{|\cos\theta|}\mathrm{uni}\,(\frac{\rho}{\cos\theta}) * \frac{1}{|\sin\theta|}\mathrm{uni}\,(\frac{\rho}{\sin\theta}) \tag{14.34}$$

Note, that convolution by the uniform kernel is exceptionally fast as it has equal weights for all entries. The problem of non-integer widths of the filters is solved by assigning fractional weights on the boundary elements of the convolution kernel.

The middle row of figure 14.5 shows the Radon transform of the edge images interpolated with nearest neighbor interpolation. The comments in section 14.6.1 applies here as well. The interpolation looks smooth because the edge runs through the images at an angle. Had the edge been oriented in one of the raster directions, the normal blocky nature of the nearest neighbor interpolation would have been revealed. The profiles corresponding to the true angle are shown in the top row of figure 14.6. Note, the fine estimation of the intensity levels in the two intervals where the function is constant.

14.6.3 Bilinear Interpolation

Nearest neighbor interpolation is very crude and the uniform kernel has very poor characteristics in the frequency domain. Instead, it is often chosen to interpolate discrete images by bilinear or bicubic interpolation kernels. Still, it is only a matter of applying (14.28). The result is given for the bilinear interpolation below, as the linearity makes this case special. The bilinear interpolation kernel is given by

$$\mathrm{bil}\,(x,y) = \begin{cases} 1-|x|-|y|-|xy|\,, & (x,y) \in [-1;1]\times[-1;1] \\ 0\,, & else, \end{cases} \tag{14.35}$$

i.e. the kernel is linear in the raster directions. The bilinear interpolation kernel is the convolution of two uniform kernels

$$
\begin{aligned}
\mathrm{bil}\,(x,y) &= \mathrm{uni}\,(x,y) * \mathrm{uni}\,(x,y) && (14.36)\\
&= \iint \mathrm{uni}\,(x-x', y-y')\,\mathrm{uni}\,(x',y')dx'dy' && (14.37)\\
&= \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{uni}\,(x-x', y-y')\; dx'\; dy' && (14.38)\\
&= (1-|x|)\,(1-|y|) && (14.39)\\
& && (14.40)
\end{aligned}
$$

These remarks lead to

$$\mathrm{bil}_\theta^r(\rho) = \mathrm{uni}_\theta^r(\rho) * \mathrm{uni}_\theta^r(\rho), \qquad (14.41)$$

where each of the two terms on the right is implemented as two convolutions in Hough space, such that the bilinear interpolation can be obtained by four convolutions. Remember, as noted earlier, that these convolutions may be implemented using fast algorithms.

The bottom row of figure 14.5 shows the Radon transform of the edge images interpolated with bilinear interpolation. Bilinear interpolation seems like a reasonable choice when the edge detection is postponed to a later processing step. If, however, we were to estimate the position of the maximum of a ridge from the projection we would need a non-linear interpolation scheme. Otherwise maxima of ridges in the raster direction are always detected on the raster. The profiles of the bilinear projection are shown in the top row of figure 14.6. Note again, the un-biased estimate of the intensities.

14.6.4 Gaussian Regularization

It is worthwhile also to analyze the Radon transform of the Gaussian as it is often reasonable to assume a spatial Gaussian distribution for the features in the image. The two-dimensional Gaussian with different widths in the two axis directions can also be separated and applied in Hough space

$$g(x,\sigma_x^2,y,\sigma_y^2) = g(x,\sigma_x^2)\; g(y,\sigma_y^2), \qquad (14.42)$$

where σ_x and σ_y are the widths of the respective Gaussians. Application of (14.28) gives

$$g_\theta^r(\rho) \;=\; \frac{1}{|\cos\theta|} g(\frac{\rho}{\cos\theta},\sigma_x^2) * \frac{1}{|\sin\theta|} g(\frac{\rho}{\sin\theta},\sigma_y^2) \qquad (14.43)$$

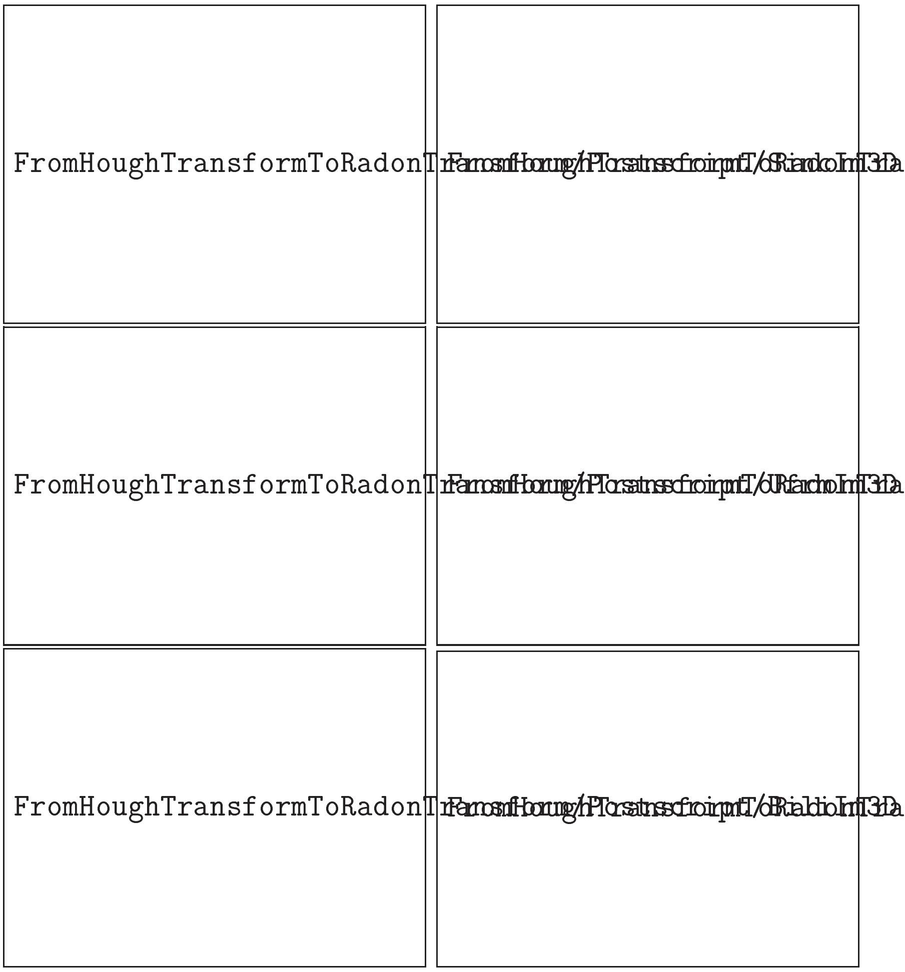

Figure 14.5: Radon transform of the interpolated edge images. The left column corresponds to the image without noise and the right column corresponds to the noisy image. The rows are sinc, nearest neighbor, and bilinear interpolation, respectively.

$$
\begin{aligned}
&= g(\rho,\sigma_x^2\cos^2\theta) * g(\rho,\sigma_y^2\sin^2\theta) && (14.44)\\
&= g(\rho,\sigma_x^2\cos^2\theta+\sigma_y^2\sin^2\theta), && (14.45)
\end{aligned}
$$

which, naturally, reduces to $g(\rho,\sigma^2)$ for $\sigma=\sigma_x=\sigma_y$. That is, the one-dimensional Gaussian influence function is just the Radon transform of the two-dimensional Gaussian influence. When the covariance structure of the Gaussian is not diagonal the theory is still applicable. Let us denote the principal variances σ^2_{min} and σ^2_{max} and the angle with the x-axis of the direction of the largest variance ψ. We get

$$
\begin{aligned}
g^r_\theta(\rho) &= g(\rho,\sigma^2_{max}\cos^2(\theta+\psi)) * g(\rho,\sigma^2_{min}\sin^2(\theta+\psi)) && (14.46)\\
&= g(\rho,\sigma^2_{max}\cos^2(\theta+\psi)+\sigma^2_{min}\sin^2(\theta+\psi)) && (14.47)
\end{aligned}
$$

It has now been shown that the Radon transform of the image interpolated by a separable kernel is easily derived from the traditional Hough transform applied to the raw image. To perform edge analysis on this Radon transform we need to know how to implement image differentiation. This is the subject of the next section.

14.7 Differentiation

In a situation of edge detection we would like to take the image derivative orthogonal to the direction of projection. First, we write the two-dimensional Fourier transform of the derived image

$$
\begin{aligned}
\mathcal{F}\{\frac{d}{d\mathbf{n}_\theta}f(x,y)\} &= \mathcal{F}\{\mathbf{n}_\theta\cdot\nabla f(x,y)\} && (14.48)\\
&= \mathbf{n}_\theta\cdot(j2\pi u, j2\pi v)F(u,v) && (14.49)\\
&= j2\pi(\omega\cos^2\theta+\omega\sin^2\theta)F(\omega,\theta) && (14.50)\\
&= j2\pi\omega F(\omega,\theta) && (14.51)\\
&= j2\pi\omega F^r_\theta(\omega) && (14.52)
\end{aligned}
$$

From the Fourier slice theorem we can now obtain the Radon transform of the derived image by the one-dimensional inverse Fourier transform

$$
\begin{aligned}
\mathcal{R}\{\frac{d}{d\mathbf{n}_\theta}f(x,y)\} &= \mathcal{F}^{-1}\{j2\pi\omega F^r_\theta(\omega)\} && (14.53)\\
&= \frac{d}{d\rho}f^r_\theta(\rho) && (14.54)
\end{aligned}
$$

Eq. (14.54) shows that taking the ρ-derivative in Radon space is exactly equivalent to projecting the two-dimensional gradients. In consequence, we can calculate the one-dimensional Radon space derivative, at the cost of one convolution, instead of calculating the two-dimensional image derivative, at the cost of two convolutions. Note, that edges that are aligned with the direction of projection give a high contribution whereas orthogonal edges give no contribution. In this sense the result is closely related to the methods that has been presented in the literature, where the estimated edge directions are used as weights when drawing the sinusoids in the Hough space.

Note, that if we search for the projection with the maximum derivative, and time is an important parameter, we can design a search strategy where we only interpolate and differentiate a few θ-columns in the Radon space.

In the bottom row of figure 14.6 the Radon spaces for the different interpolation strategies are shown. Only the profiles corresponding to the true edge direction are displayed. Note, that the sinc reconstruction gives the strongest derivative on the true edge location. These derivatives was obtained by finite difference and are therefore sensitive to noise.

14.8 Edge Profile Parameters

In the present framework, the projection of the image is based on an assumption that the image structure is basically one-dimensional. Once the projected image is calculated the direction θ of the one-dimensional structure being projected, e.g. an edge, can be determined. This could, for example, be carried out by searching for the maximal gradient or by correlation with an expected profile. The outcome of this process is the mean profile of the edge as a function of ρ. Now, all sorts of analysis may be performed on this profile. Firstly, the mean profile can be compared to an expected profile to reveal errors in the image. Secondly, the intensity variation across e.g. an edge, will give an idea of the precision to be expected from an edge detection. Thirdly, the blurring of the optical system may be estimated from the width of the derivative of the edge. The two latter variables appear in the expression by Åström [2] putting a lower bound on the edge detection variance. This is just a few suggestions to the information that can be extracted from the projection.

The mean and variance of the derivative of the edges shown in figure 14.6

FromHoughTransformToRadonTransform/Postscript/EdgeInt.eps
FromHoughTransformToRadonTransform/Postscript/Edge
FromHoughTransformToRadonTransform/Postscript/DiffInt.eps
FromHoughTransformToRadonTransform/Postscript/Diff

Figure 14.6: The top row shows the profiles of the Radon space that corresponds to the true edge angle, i.e. the mean edge appearance. The rippling of the sinc curve is due to truncation of the sinc filter. The bottom row shows the derivatives of these profiles. Note, that sinc has the steepest edge reconstruction. The theoretical derivative of the edge is added for reference. Theoretical (+), Sinc (solid), Uniform (dashed), and Bilinear (dotted).

	edge without noise		edge with noise	
	($\sigma^2 = 0.5, std.dev. = 0$)		($\sigma^2 = 2.0, std.dev. = 5$)	
	mean	variance	mean	variance
sinc	-3.3e-4	0.46	4.0e-2	1.87
uniform	0.0e0	0.59	-1.7e-2	1.96
bilinear	0.0e0	0.67	-2.1e-2	2.09

Table 14.1: Estimated Mean and Variance of the Edge obtained from the Derivative of the Radon Transform.

have been estimated and the results are given in table 14.1. The mean is given relative to the known location of the edge. The bias on the sinc estimates is assumed to originate from the truncation of the sinc filter. The nearest neighbor and the bilinear interpolation methods seem to give un-biased estimates. Hence, the edge location in the noisy image is determined with an accuracy of 2% of a pixel. The variance estimates are also close to the theoretical values of $\sigma^2 = 0.5$ and $\sigma^2 = 2.0$.

14.9 Radon Transform of other Image Moments

14.9.1 Normalization

It has long been known that the Hough space should be normalized to account for the shape of the retina. In the Radon transform framework it is of interest to determine a central estimate of the mean intensity along the direction of projection, i.e. the Radon integral should be divided by the length of the integration interval. When using interpolation kernels the effective length of the interval is dependent on the shape of the kernel. It then follows that the relevant normalization coefficients are obtained by calculating the Radon transform of the interpolated unit image

$$f^{0r}_{\ \theta}(\rho) = \mathcal{H}\{f^0(x,y)\} * \mathcal{R}\{h(x,y)\}, \tag{14.55}$$

where $f^{0r}_{\ \theta}(\rho)$ are the normalization coefficients, $f^0(x,y)$ is the unit image and $h(x,y)$ is the interpolation kernel.

The variation of the normalization coefficient as a function of ρ for the true value of θ is shown in figure 14.7. The left plot corresponds to nearest neighbor

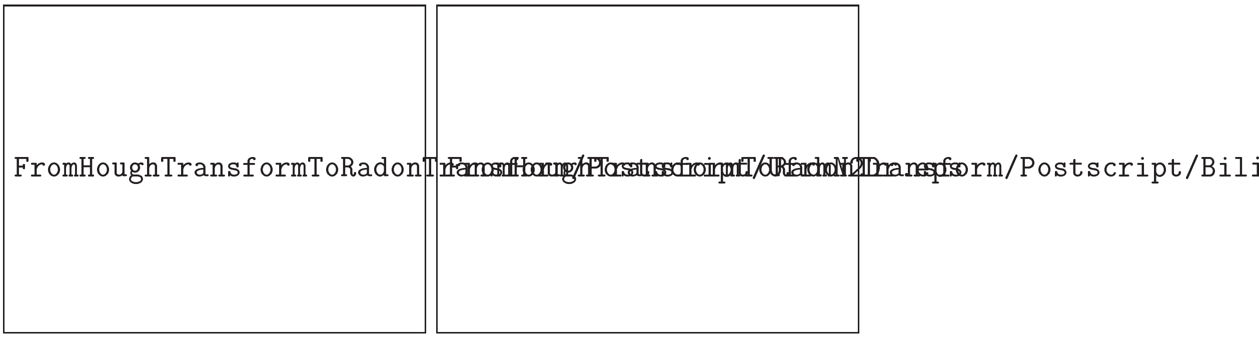

Figure 14.7: The normalization coefficients as a function of ρ for the angle corresponding to the true edge direction. The left plot shows the coefficients for nearest neighbor interpolation and the right plot shows the coefficients for bilinear interpolation.

interpolation whereas the right plot corresponds to bilinear interpolation. The former is an integral over a blocky binary domain and is piece-wise linear and the latter is an integral over a blocky parabolic surface and is piece-wise cubic. Obviously, they are different.

14.9.2 Variance

In line with the above, we can also determine other moments in the projection. The image variation along the direction of projection may be estimated from the projection of the second moment of the image

$$ {f^2}^r_\theta(\rho) = \mathcal{H}\{f^2(x,y)\} * \mathcal{R}\{h(x,y)\}, \tag{14.56} $$

where ${f^2}^r_\theta(\rho)$ is the second moment of the projection, $f^2(x,y)$ is the squared image and $h(x,y)$ is the interpolation kernel. Again, the result is dependent on the smoothing properties of the interpolation kernel. Additive noise in the image is associated with the individual pixels and to estimate the variance of this noise we should therefore use nearest neighbor interpolation. The estimate becomes

$$ V_\theta(\rho) = {f^2}^r_\theta(\rho) - f^{r\,2}_\theta(\rho), \tag{14.57} $$

where proper normalization according to (14.55) is assumed for both terms on the right. The result of this transform is shown in figure 14.8. The left column is obtained from the noiseless image and the right column corresponds to the noisy image. It is clear that the steeper edge gives rise to a higher variance on

Figure 14.8: The estimated variance along the projection. The left column corresponds to the edge image without noise and the right column to the noisy edge image. The bottom plots show the variation across the edge corresponding to the true edge direction.

integrals across the edge. The bottom plots show the variation across the edge corresponding to the true edge direction. Ideally, this plot should be constant across the edge, but due to the discretization there is a variance increase on the edge. Note, however, that the estimated variance on the two sides of the edge are in agreement with the added noise of $\sigma^2 = 0.0$ and $\sigma^2 = 25.0$, respectively.

The above result is mainly of theoretical interest owing to the time complexity. However, one might use the second moment to determine the value of θ being the direction of least variation. This could be the case when the only available information is that the image is basically one-dimensional. A possible scheme could be to take the average variation over all values of ρ. This has been done in figure 14.9. It is seen that the mean variation has a well-defined minimum corresponding to the true direction of the edge.

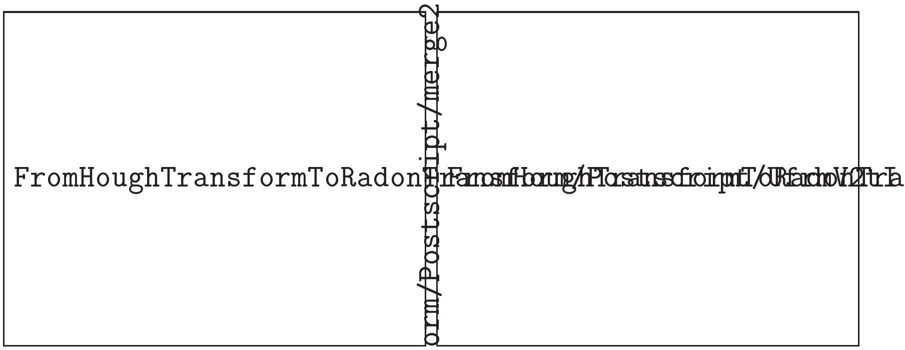
FromHoughTransformToRadonTransform/Postscript/UfrmV2.eps
FromHoughTransformToRadonTransform/Postscript/Ufrm
FromHoughTransformToRadonTransform/Postscript/merge2.ps

Figure 14.9: Average variation over all values of ρ. The variation shows a clear minimum corresponding to the true direction of the edge.

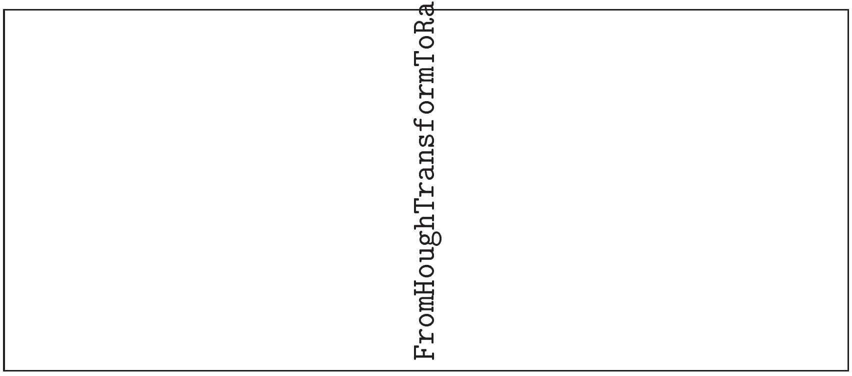

Figure 14.10: Image of a steel plate positioned in a plasma cutting birth.

14.10 Real Image Example

This section gives an example of the described theory applied to edge detection in a real image. Figure 14.10 shows an edge of a steel plate positioned in a plasma cutting birth. The edge is corrupted by a reflecting lamp. This will not cause a problem as we employ the normalization scheme and may omit saturated pixels from the Hough transformation without introducing a bias. The processing chain is shown in figure 14.11. In the shown plots $\theta \in [-12; 12]$ and $\rho \in [-5; 5]$ have both been discretized to 401 levels. The first column shows the general Hough transform of the image and the Hough transform of the unit image. In the second column these images have been interpolated using the bilinear interpolation scheme as explained in the previous sections, giving the Radon transform of the interpolated image. Note, the influence from the shape of the retina. The upper right image shows the normalized Radon transform.

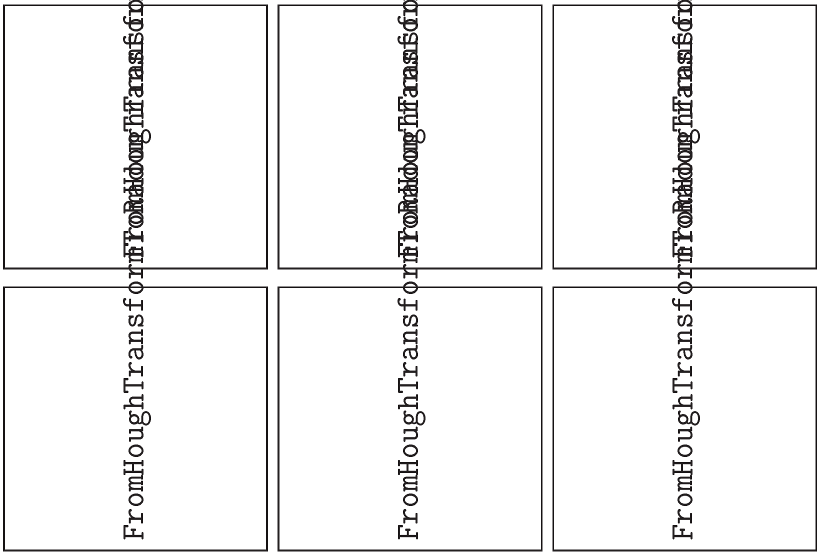

Figure 14.11: For the first two columns the upper row is the transformation on the image data and the lower row is the transformation of the unit image. The first column shows the usual Hough transform of the pixels in the vicinity of the edge. The second column shows interpolation by the bilinear kernel. Note, the strong influence from the shape of the retina. The top right plot is the normalized Radon transform and the bottom right plot is the ρ-derivative of the normalized Radon transform.

This is obtained by division of the two images in the middle column. The lower right image is the derivative with respect to ρ of the upper right image, using a Canny-Deriche filter with $\sigma = 1.0$. The edge parameters are determined by the maximum in this image. The normalized Radon transform and the related derivative are also displayed as surface plots in figure 14.12. It is seen that the maximum is very well-defined. In the lower plots the edge profile is shown for the value of θ with the highest gradient. The profiles reveal that the edge is approximately Gaussian, although part of this shape is inherited from the Canny-Deriche filter.

14.11 Conclusion

The close relationship between the Hough transform and the Radon transform has been investigated. It was shown that the Radon transform of interpolated discrete images can easily be obtained by convolutions in Hough space. In

Figure 14.12: The normalized Radon transform and the derivative with respect to ρ. The top images show the entire Radon space, while the bottom row only shows the profile corresponding to the highest gradient. It is seen that the maximum is well-defined, and that the average edge profile is approximately Gaussian.

particular, the adequate kernels for sinc, nearest neighbor, and bilinear interpolation were derived. Similarly, the Gaussian influence function was derived from the corresponding two-dimensional kernel. It was also shown how the image derivative can be derived in Radon space, such that edge detection can be performed on the projected image. Finally, it was suggested how to extract qualitative information from the interpolated Radon space such as edge strength and lens blur.

The presented theory can be useful in any analysis of straight one-dimensional structures in images. To illustrate the information available in the Hough space, a simple example of edge detection was given.

Chapter 15

On Averaging Rotations

On Averaging Rotations

C. Gramkow

Odense Steel Shipyard Ltd.
P.O. Box 176
DK-5100 Odense C., Denmark

Department of Mathematical Modelling,
Technical University of Denmark, Building 321
DK-2800 Lyngby, Denmark

In this article two common approaches to averaging rotations are compared to a more advanced approach based on a Riemannian metric. Very often the barycenter of the quaternions or matrices that represent the rotations are used as an estimate of the mean. These methods neglect that rotations belong to a non-linear manifold and re-normalization or orthogonalization must be applied to obtain proper rotations. These latter steps have been viewed as ad hoc corrections for the errors introduced by assuming a vector space. The article shows that the two approximative methods can be derived from natural approximations to the Riemannian metric, and that the subsequent corrections are inherent in the least squares estimation. Keywords: averaging rotations, Riemannian metric, matrix, quaternion.

15.1 Introduction

When we measure the pose of an object with different devices or different approaches we would like to integrate these measurements into a single mean pose. This could be the case in a robot vision system, where we measure the orientation of an object with multiple cameras, or it could occur in the registration of medical images, where several poses are computed from images of different modalities [58]. Figure 15.2 shows a typical ship block where robot manufacturing requires a high quality pose estimation with respect to a fixed frame. This can be obtained by the fusion of information from multiple cameras and/or sensors. In this article we only consider taking the mean of a set of three-dimensional rotations, and do not treat problems where there is a translational component.

The problem with calculating the mean of a set of rotations lies in the fact that rotations do not belong to a vector space, but lie on a non-linear manifold. If we choose to represent the rotation by a quaternion $\dot{\mathbf{q}} = (q, \mathbf{q})$ the proper rotations are constrained to the three-dimensional unit sphere $|\dot{\mathbf{q}}| = 1$ of a four-dimensional vector space. Another representation is obtained using a 3×3 matrix $\mathbf{R}$ constrained to be orthogonal, $\mathbf{R}^T = \mathbf{R}^{-1}$, i.e. a three-dimensional manifold in a nine-dimensional space. Despite the non-linearity of rotations it has become customary to calculate the average by the barycenter in the larger vector spaces. If we let underscore denote the mean, we have $\underline{\mathbf{R}} = \frac{1}{n}\sum_i \mathbf{R}_i$ or $\underline{\dot{\mathbf{q}}} = \frac{1}{n}\sum_i \dot{\mathbf{q}}_i$. There is, however, no guarantee that these means lie on the non-linear manifolds of allowable rotations. To ensure this, the rotation matrix must be orthogonalized, finding the closest orthogonal matrix to the calculated mean, or the mean quaternion must be re-normalized to lie on the unit sphere.

Recently, Pennec et al. [57, 58] have addressed the problem of taking the mean of geometric features in a very general framework. Their approach is to consider the geodesic distance between points on the non-linear manifold instead of the vector distance. The geodesic distance is the length of the shortest curve lying on the manifold and connecting the two points. Their thorough work applies to all geometric features on non-linear manifolds and also considers Mahalanobis distances in place of the simple least squares with scalar weights. They use the particularly simple example of rotations to demonstrate their theory. Without questioning the importance of the theory described above this article will try to answer the following question: *how close does the barycentric means of rotations approximate the mean based on the Riemannian metric.*

The question if other linearizations are superior to the two barycentric means is not addressed. The comparison is based on the equally weighted average, but applies just as well to the weighted case.

15.2 Notation

In the above section, different notations were introduced. Scalars are written in lower case, vectors are lower case bold, matrices are bold capitals, quaternions are lower case bold with dots with the scalar part q first and the vectorial part $\mathbf{q}$ last, and mean values are expressed with an underscore. The abbreviation *ort.* denotes the orthogonality constraint $\mathbf{R}^{-1} = \mathbf{R}^T$ on a matrix. The distance between two features is denoted $d(\cdot, \cdot)$, and $\theta(\cdot)$ is the angle of rotation induced by the quaternion or matrix argument. The cross product and dot product operators of vectors are '×' and '$\cdot$', respectively. The determinant of a matrix is denoted $det()$ and $diag(\ldots)$ is a diagonal matrix with the arguments on the diagonal.

15.3 The Invariant Metric

We have already argued that the vectorial distance $||\dot{\mathbf{q}}_1 - \dot{\mathbf{q}}_2||_2$ or $||\mathbf{R}_1 - \mathbf{R}_2||_2$ between two rotations ignores the curvature of the respective manifolds. We need a distance measure $d(\mathbf{R}_1, \mathbf{R}_2)$ that expresses the distance along a curve *on* the manifold. Rotations belong to a multiplicative group, i.e. rotations act on rotations by multiplication and form new rotations. It is therefore natural to require that the metric is invariant to rotations $d(\mathbf{R}_1, \mathbf{R}_2) = d(\mathbf{I}, \mathbf{R}_1^{-1}\mathbf{R}_2)$, such that the distance only depends on a single rotational argument. Invariance to orientation leads to the metric $d(\mathbf{R}_1, \mathbf{R}_2) = \theta(\mathbf{R}_1^{-1}\mathbf{R}_2)$, which is the metric used by Pennec [57].

In the following sections we will investigate how this metric is related to the barycentric mean approaches.

15.4 Quaternion-Based Mean

Rotations can be expressed using unit quaternions. A quaternion is a four-dimensional vector, that has a scalar part and a three dimensional vector part. We can also view it as a sort of complex number with a real part and three imaginary parts. A rotation through an angle θ around the unit vector $\mathbf{n}$ is represented by the quaternion $\dot{\mathbf{q}} = (\cos\theta/2, \mathbf{n}\sin\theta/2)$. It is easily verified that $\dot{\mathbf{q}}$ and $-\dot{\mathbf{q}}$ give rise to the same rotation. Having a set of rotations, we may choose consistent representatives by requiring that the dot product with the first quaternion is positive. For all reasonable sets of rotations this approach will work. It will occasionally fail if the quaternions scatter all over the unit sphere, but in that case the mean does not make much sense anyway. The inverse of a rotation $\dot{\mathbf{q}}^{-1}$ is obtained by the conjugate quaternion $\dot{\mathbf{q}}^* = (q, -\mathbf{q})$.

The composition of rotations represented by quaternions is achieved by the quaternion product defined by

$$\dot{\mathbf{q}}_1 \dot{\mathbf{q}}_2 = (q_1 \cdot q_2 - \mathbf{q}_1 \cdot \mathbf{q}_2, q_1 \mathbf{q}_2 + q_2 \mathbf{q}_1 + \mathbf{q}_1 \times \mathbf{q}_2) \tag{15.1}$$

We see immediately that $\dot{\mathbf{q}}\dot{\mathbf{q}}^* = 1$, verifying that multiplication with the conjugate rotation yields the identity. According to the metric derived above the distance between two rotations is

$$\begin{aligned} d(\dot{\mathbf{q}}_1, \dot{\mathbf{q}}_2) &= \theta(\dot{\mathbf{q}}_1^{-1} \dot{\mathbf{q}}_2) && (15.2) \\ &= \theta(\dot{\mathbf{q}}_1^* \dot{\mathbf{q}}_2) && (15.3) \\ &= \theta((\dot{\mathbf{q}}_1 \cdot \dot{\mathbf{q}}_2, q_1 \mathbf{q}_2 - q_2 \mathbf{q}_1 - \mathbf{q}_1 \times \mathbf{q}_2)) && (15.4) \\ &= 2 \arccos(\dot{\mathbf{q}}_1 \cdot \dot{\mathbf{q}}_2), && (15.5) \end{aligned}$$

where it was used in the last equality that the real part of a quaternion is the cosine of half the angle of rotation. It is not surprising that the distance along a geodesic takes this form. We know from classical geometry that the geodesics on a sphere are formed by the great circles and the distance between two points on the unit sphere is therefore the angle between the corresponding vectors. Neglecting the difference in scale, this is exactly what is derived in equation 15.5.

As expected the difference between two rotations is a non-linear function of the vectorial representations. To arrive at a closed form approximation to the mean, we need to linearize this expression. Consider the Taylor expansion of cosine

$$\cos\frac{\theta}{2} = 1 - \frac{1}{8}\theta^2 + o(\theta^4). \tag{15.6}$$

Rearranging this expression gives

$$\begin{aligned} \theta^2(\dot{\mathbf{q}}_1^* \dot{\mathbf{q}}_2) &= 8(1 - \cos\frac{\theta(\dot{\mathbf{q}}_1^* \dot{\mathbf{q}}_2)}{2}) + o(\theta^4) && (15.7) \\ &\approx 8(1 - \dot{\mathbf{q}}_1 \cdot \dot{\mathbf{q}}_2) && (15.8) \end{aligned}$$

Now, the mean rotation is the rotation $\dot{\mathbf{q}}$ that minimizes the quadratic cost function

$$\chi^2 = \sum_i \theta^2(\dot{\mathbf{q}}^* \dot{\mathbf{q}}_i) \qquad (15.9)$$

We have not considered weights w_i in this expression, but they are straightforwardly included. Using the approximation above, the mean rotation is determined as

$$\begin{aligned}
\underline{\dot{\mathbf{q}}} &= \arg\min_{|\dot{\mathbf{q}}|=1} \sum_i \theta^2(\dot{\mathbf{q}}^* \dot{\mathbf{q}}_i) && (15.10)\\
&\approx \arg\min_{|\dot{q}|=1} \sum_i 8(1 - \dot{\mathbf{q}} \cdot \dot{\mathbf{q}}_i) && (15.11)\\
&= \arg\max_{|\dot{q}|=1} \sum_i \dot{\mathbf{q}} \cdot \dot{\mathbf{q}}_i && (15.12)\\
&= \arg\max_{|\dot{q}|=1} \dot{\mathbf{q}} \cdot \sum_i \dot{\mathbf{q}}_i && (15.13)\\
&= \sum_i \dot{\mathbf{q}}_i / |\sum_i \dot{\mathbf{q}}_i| && (15.14)
\end{aligned}$$

i.e. the optimal quaternion is exactly the barycentric mean with re-normalization. It is interesting to note, that in this formulation the re-normalization is part of the true optimization of an approximated cost function and not an ad hoc repair of an approximated optimization.

15.5 Matrix-Based Mean

Let us now look to another representation of rotations, namely the rotation matrix. This is a 3 × 3 orthogonal matrix with the rotated basis vectors as columns. Naturally, the composition of two rotations is obtained by usual matrix multiplication and the inverse rotation is consequently the matrix inverse. The eigenvalues of an orthogonal matrix are $(1, e^{i\theta}, e^{-i\theta})$, where the eigenvector corresponding to the unit eigenvalue represent the rotation axis. The trace of a square matrix is the sum of the eigenvalues, such that

$$\begin{aligned}
tr(\mathbf{R}) &= 1 + e^{i\theta} + e^{-i\theta} && (15.15)\\
&= 1 + 2\cos\theta && (15.16)
\end{aligned}$$

Using this relation to evaluate the distance between two rotations represented by matrices we get

$$d(\mathbf{R}_1, \mathbf{R}_2) = \theta(\mathbf{R}_1^{-1}\mathbf{R}_2) \qquad (15.17)$$

$$= \arccos(\frac{tr(\mathbf{R}_1^{-1}\mathbf{R}_2) - 1}{2}) \tag{15.18}$$

This time we use the usual Taylor expansion of cosine

$$\cos\theta = 1 - \frac{1}{2}\theta^2 + o(\theta^4) \tag{15.19}$$

which gives the expression

$$\begin{aligned} \theta^2(\mathbf{R}_1^{-1}\mathbf{R}_2) &= 2(1 - \cos\theta(\mathbf{R}_1^{-1}\mathbf{R}_2)) + o(\theta^4) & (15.20) \\ &\approx 2(1 - \frac{tr(\mathbf{R}_1^{-1}\mathbf{R}_2) - 1}{2}) & (15.21) \\ &= 3 - tr(\mathbf{R}_1^{-1}\mathbf{R}_2) & (15.22) \end{aligned}$$

As in the case of quaternions, the mean rotation is defined by

$$\begin{aligned} \underline{\mathbf{R}} &= \arg\min_{\mathbf{R}\ ort.} \sum_i \theta^2(\mathbf{R}^{-1}\mathbf{R}_i) & (15.23) \\ &\approx \arg\min_{\mathbf{R}\ ort.} \sum_i 3 - tr(\mathbf{R}^{-1}\mathbf{R}_i) & (15.24) \\ &= \arg\max_{\mathbf{R}\ ort.} \sum_i tr(\mathbf{R}^{-1}\mathbf{R}_i) & (15.25) \\ &= \arg\max_{\mathbf{R}\ ort.} tr(\mathbf{R}^{-1}\sum_i \mathbf{R}_i). & (15.26) \end{aligned}$$

The latter optimization problem is actually the core of the 3D-3D pose problem, which has been solved among others by Horn [32], Arun et al. [1], and Umeyama [73]. The solution can be seen as the rotation that best rotates the identity matrix into the matrix consisting of the sum of rotations. In this light the solution is obviously minimizing the sum of squared distances of the individual elements of the matrix, and is what we understand of a best orthogonal approximation to the sum of rotations. Since the solution is scale invariant, we can also regard it as an orthogonalization of the barycentric mean. Just as for quaternions the orthogonalization is a natural part of the optimization, and is thus more than an ad hoc subsequent correction.

The solution to the above problem is most easily obtained by Singular Value Decomposition (SVD), $\sum_i \mathbf{R}_i = \mathbf{UDV}$, where the singular values are ordered in descending order. Introducing the matrix

$$\mathbf{S} = diag(1, 1, det(\mathbf{U})det(\mathbf{V})) \tag{15.27}$$

the mean rotation in the above problem is simply given by

$$\underline{\mathbf{R}} = \mathbf{USV} \tag{15.28}$$

Again, individual weights w_i on the rotation matrices were omitted, but they are easy to add.

Both approximations to the Riemannian metric are based on a linearization of the non-linear manifold around the identity rotation. Consequently, the contact point of the linear sub-space is effectively chosen as the estimated mean. This is immediately verified by noticing that any rotation applied to our set of rotations before estimating the mean will, due to linearity, cause a similar rotation of the mean. Thus, the linear approximations have the same invariancy to rotations as the non-linear metric does.

15.6 Comparison

As derived, both averaging quaternions and rotation matrices can be seen as approximations to the true least squares problem. They are based on the same order of approximation, namely the second[1] order Taylor expansion of cosine. Comparing the two representations we see, however, that they cannot give rise to the same optimum. The rotation matrix that may be derived from a quaternion contains the elements of the quaternion quadratically. This difference is also reflected in the quality of the approximation. The quaternion solution is based on an approximation of $\cos\theta/2$, whereas the rotation matrix solution is based on an approximation of $\cos\theta$. For the sake of comparison we preserve the scale in the $o()$ notation

$$\begin{aligned} quaternion &: \theta^2 \approx 8 - 8\cos\frac{\theta}{2} + o(\frac{1}{48}\theta^4) && (15.29) \\ matrix &: \theta^2 \approx 2 - 2\cos\theta + o(\frac{1}{12}\theta^4) && (15.30) \end{aligned}$$

For small values of θ, we see that the error caused by the latter approximation is four times larger than for the former approximation. It should be noted, though, that for any reasonable variation in the rotations either one of the approximations is very good. Figure 15.1 shows the errors related to the two different approximations. When using the quaternion-based method the relative error committed is less than 1% for angles smaller than 40 *deg*. For the matrix-based method the 1% limit is violated for angles greater than 20 *deg*.

[1] actually the third order as cosine has no odd terms

OnAveragingRotations/combi.ps
OnAveragingRotations/absolute.ps
OnAveragingRotations/relative.ps

Figure 15.1: Evaluation of the cosine approximation. Top: the true squared angle and the two approximations. Middle: the absolute error of the approximations. Bottom: the relative error of the approximations. For the quaternion and matrix approximations the angle corresponding to 1% relative error is approximately (40 *deg.*) and (20 *deg.*), respectively.

15.7 Test

A series of tests has been carried out to verify the similarity of the three approaches to taking the mean of rotations: Riemannian metric-based, quaternion-based, and matrix-based. Synthetic samples of rotations have been constructed to compare the different approaches. The rotation space is sampled by sampling the direction and the angle independently. The angle is sampled from a Gaussian distribution. The direction is sampled uniformly over all directions. In all experiments the rotations are sampled with the identity as mean. The comments at the end of section 15.5 justifies that the tests are invariant to this choice. In the first series of experiments we take the mean of only 10 rotations to investigate the robustness for a small number of samples. The experiment is repeated 50 times for each of three different choices of the standard deviation of the angle of rotation: 2 *deg.*, 10 *deg.*, and 40 *deg*. The quality of the estimation of the mean is measured by the angle of rotation required to bring the mean into the identity. The results are seen in figure 15.3. For the small value of the standard deviation there are no difference to be seen in the estimation. The plots lie on top of each other. As the standard deviation is increased the difference becomes visible. In the experiments the quaternion-based and matrix-based approximations actually seem to be superior to the method based on the Riemannian metric. This could be an effect of the more robust weighting function caused by the approximation, see figure 15.1, but the difference is too insignificant to make final conclusions. In another series of experiments the sample size has been increased to 100 to test if the true Riemannian metric is superior for large sample sizes. The result is displayed in figure 15.4. The three methods seem equally good in this experiment as well.

For all tests the theoretical standard deviation of the mean was calculated as $\sigma_{\underline{x}} = \sigma_x/\sqrt{n}$, where n is the number of samples. This value is plotted as a vertical line in figure 15.3 and 15.4. As expected around 70% of the mean values lie within one standard deviation.

In one of the Riemannian metric experiments with a standard deviation of 40 *deg.* in the sample of rotations, the *MATLAB*-routine **fmins** failed to converge in 800 iterations. The starting point of the optimization was in all cases the output from the linear quaternion method. When the identity rotation was used as starting point the test showed 34 cases of divergence. Of

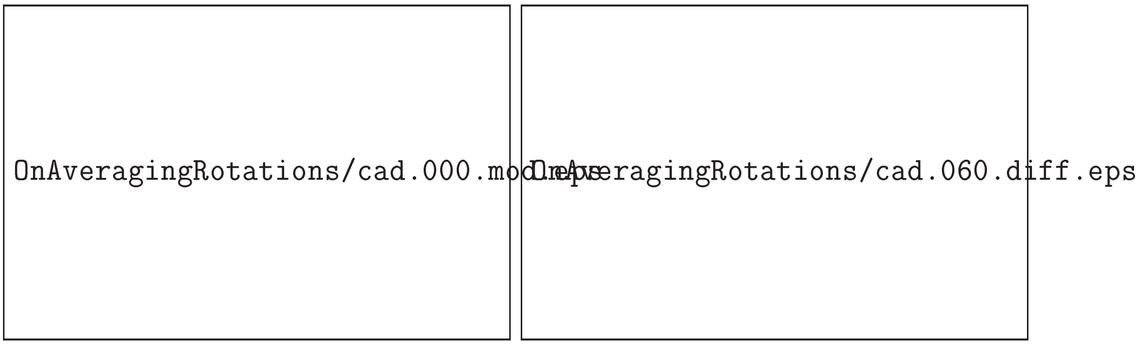
OnAveragingRotations/cad.000.mod.eps
OnAveragingRotations/cad.060.diff.eps

Figure 15.2: Top: CAD generated image of a ship block. Bottom: difference between the top image and the image of the ship block rotated 0.6 degrees horizontally around its center.

course, there is no convergence issue in the closed form solutions.

To give an idea of the quantities discussed above the effect of rotation on a ship block is displayed in figure 15.2. The figure shows a ship block with an almost quadratic base plate. This block has been rotated horizontally around its center through an angle of 0.6 degrees. This is by far greater than any angular deviation between the three methods observed when the standard deviation of the individual rotations is less than 10 degrees. The image of the rotated ship block has been subtracted from the image of the original block and the difference is displayed in the bottom of the figure. Note, that we in general expect a standard deviation on the individual pose estimates much less than 10 degrees.

15.8 Conclusion

In this article the problem of averaging rotations have been investigated. The customary approaches based on the barycenter of either the quaternion representations or the matrix representations have been compared to the 'exact' least squares method based on the Riemannian metric. Theoretical and empirical analysis have shown that both linear approaches give very good approximations to the non-linear average.

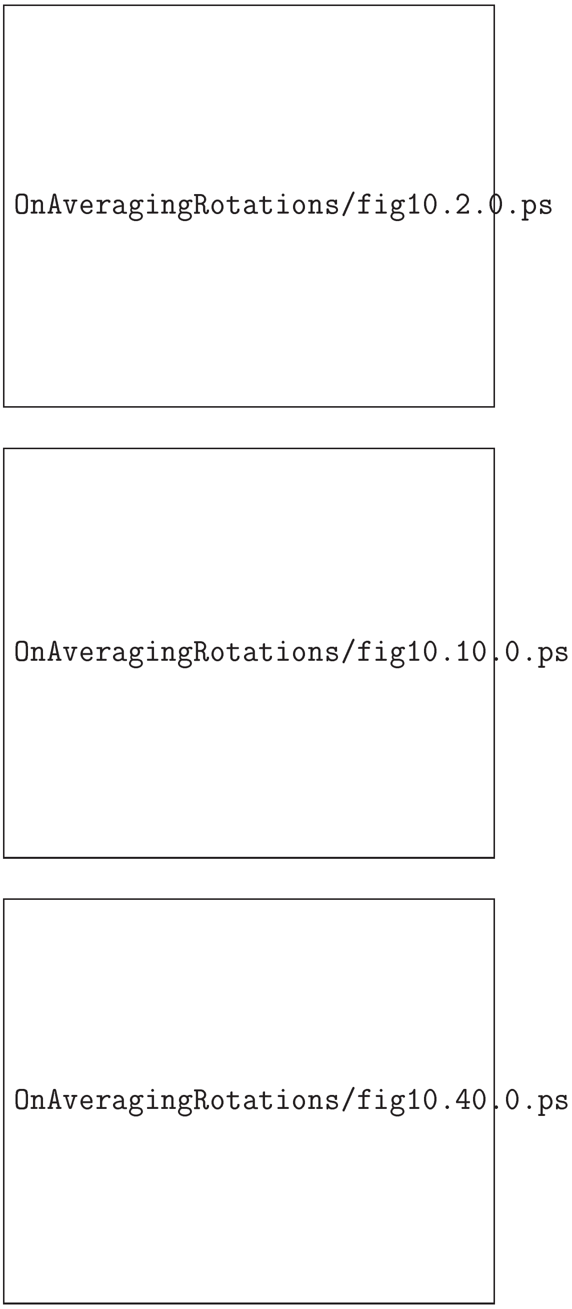
OnAveragingRotations/fig10.2.0.ps
OnAveragingRotations/fig10.10.0.ps
OnAveragingRotations/fig10.40.0.ps

Figure 15.3: The absolute angular deviation from the identity of the average of 10 rotations for three different values of the standard deviation of the angle of rotation: 2 *deg.*, 10 *deg.*, and 40 *deg.* In each case the experiment has been repeated 50 times.

OnAveragingRotations/fig100.2.0.ps
OnAveragingRotations/fig100.10.0.ps
OnAveragingRotations/fig100.40.0.ps

Figure 15.4: The absolute angular deviation from the identity of the average of 100 rotations for three different values of the standard deviation of the angle of rotation: 2 *deg.*, 10 *deg.*, and 40 *deg.* In each case the experiment has been repeated 50 times.

15.9 Acknowledgements

This work was carried out during a visit to the MOVI group at INRIA, Grenoble.

Chapter 16

Reconstruction and Matching of OSS Mock-Up

VIGOR, Esprit-IV reactive LTR project, number 26247

Reconstruction and Matching of OSS Mock-Up

Internal Report

Claus Gramkow
Odense Steel Shipyard, Ltd. (OSS)
e-mail: cg@oss.dk

September 25, 1998

VIGOR inquiries:
e-mail: Vigor@inrialpes.fr
WWW: http//www.inrialpes.fr/VIGOR

16.1 Introduction

The work described in this report was performed within the VIGOR project during a visit to INRIA, Rhône-Alpes. The time in Grenoble was mainly used for experimenting with three-dimensional reconstruction of a ship block mock-up supplied by Odense Steel Shipyard. The visit also served to align the industrial and academic needs of the partners.

The first three sections of this report will cover the underlying theory for the method applied for the three-dimensional reconstruction of the OSS mock-up. The description closely follows that of Christy and Horaud [7]. The idea of iteratively refining an estimate obtained with the weak perspective camera towards a solution for the full perspective camera was first described in Dementhon et al. [11, 54] and was later improved using the para-perspective camera model by Horaud et al. [28, 30]. The factorization of motion and structure for the various camera models was proposed by Tomasi, Kanade, and Poelman [69, 60].

16.2 Camera Models

A camera model describes the relation between the coordinates of a point in the observed world and the corresponding coordinates in the projected image. A widely used perspective camera model takes the form

$$\begin{pmatrix} su_i \\ sv_i \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{i}^T & t_x \\ \mathbf{j}^T & t_y \\ \mathbf{k}^T & t_z \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix}, \qquad (16.1)$$

where i is a point index. This model assumes a CCD chip with no skew and takes no account of any possible lens distortions. The latter is normally modelled independently. The left side of the equation is the homogeneous coordinates of the image point. The matrices on the right hand side are from left to right the internal camera calibration parameters, the central projection matrix, the rigid displacement between the camera coordinate system and the world coordinate system, and finally the points homogeneous world coordinates $(\mathbf{P}_i, 1) = (X_i, Y_i, Z_i, 1)$. We assume that the origin of the object coordinate system is $\mathbf{P}_0 = (0, 0, 0)$. If we have calibrated the camera we can eliminate the effect of the internal camera parameters by applying the equations

$$u_i = \alpha_u x_i + u_0 \tag{16.2}$$

$$v_i = \alpha_v y_i + v_0 \tag{16.3}$$

If we proceed with the metric image coordinates (x_i, y_i), we can write the camera model more compact as

$$x_i = \frac{\mathbf{i} \cdot \mathbf{P}_i + t_x}{\mathbf{k} \cdot \mathbf{P}_i + t_z} \tag{16.4}$$

$$y_i = \frac{\mathbf{j} \cdot \mathbf{P}_i + t_y}{\mathbf{k} \cdot \mathbf{P}_i + t_z} \tag{16.5}$$

Dividing both numerator and denominator by t_z and introducing the new variables $\mathbf{I} = \mathbf{i}/t_z$, $\mathbf{J} = \mathbf{j}/t_z$, and $\epsilon_i = \mathbf{k} \cdot \mathbf{P}_i/t_z$ we obtain

$$x_i = \frac{\mathbf{I} \cdot \mathbf{P}_i + x_0}{1 + \epsilon_i} \tag{16.6}$$

$$y_i = \frac{\mathbf{J} \cdot \mathbf{P}_i + y_0}{1 + \epsilon_i} \tag{16.7}$$

Note, that $x_0 = t_x/t_z$ and $y_0 = t_y/t_z$, where (x_0, y_0) are the image coordinates of the point P_0. The variable ϵ_i is denoted the perspective effect.

It is immediately noticed that if the perspective effect is zero the relation between the world coordinate and the image coordinate is just an affine function. We can obtain an affine approximation to the camera both by setting $1/(1 + \epsilon_i)$ equal to 1, leading to the weak perspective camera model, and by setting $1/(1+\epsilon_i)$ to $1-\epsilon_i$, leading to the para-perspective camera model. Only the former will be presented here.

16.3 Affine Reconstruction

The reconstruction problem consists of determining the geometry of the viewed three-dimensional world and the different positions of the camera simultaneously from the image coordinates of the points. It is naturally only possible to determine these sizes in an arbitrary coordinate system, e.g. the coordinate system of the first camera. It is assumed that the points in the different images have been matched, such that we know if two points in two different images actually correspond to the same physical three-dimensional point. Using the weak perspective camera model above, the problem may be solved in a very nice and compact manner. With this approximation equations 16.6 and 16.7 become

$$x_i = \mathbf{I} \cdot \mathbf{P}_i + x_0 \tag{16.8}$$

$$y_i = \mathbf{J} \cdot \mathbf{P}_i + y_0 \tag{16.9}$$

We may also move the origin of the image coordinate system to the point (x_0, y_0)[1] to obtain

$$\mathbf{s}_i = \begin{pmatrix} x_i - x_0 \\ y_i - y_0 \end{pmatrix} = \begin{pmatrix} \mathbf{I}^T \\ \mathbf{J}^T \end{pmatrix} \mathbf{P}_i \tag{16.10}$$

The above equation describes the relation between the world object and the image coordinates for a single camera. If the same world is imaged from different view points we get a series of these equations

$$\mathbf{s}_{ij} = \begin{pmatrix} x_{ij} - x_{0j} \\ y_{ij} - y_{0j} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_j^T \\ \mathbf{J}_j^T \end{pmatrix} \mathbf{P}_i = \mathbf{A}_j \, \mathbf{P}_i, \tag{16.11}$$

where subscript j is the image index and $\mathbf{A}_j$ is the affine projection matrix built from $\mathbf{I}_j$ and $\mathbf{J}_j$. If we observe n points from k different views we can set up the matrix equation

$$\mathbf{S} = \begin{pmatrix} \mathbf{s}_{11} & \cdots & \mathbf{s}_{n1} \\ \vdots & \ddots & \vdots \\ \mathbf{s}_{1k} & \cdots & \mathbf{s}_{nk} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_k \end{pmatrix} \begin{pmatrix} \mathbf{P}_1 & \cdots & \mathbf{P}_n \end{pmatrix} = \mathbf{AP} \tag{16.12}$$

This equation shows that the three-dimensional object being viewed and the motion of the camera may be separated into two independent matrices. The matrix **P** contains the coordinates of the world points and the matrix **A** contains the projection matrices associated with the different poses of the camera. That is, the *structure and motion* problem amounts to the factorization of the measurement matrix **S**, containing the measured image points.

It will now be described how we can obtain one such factorization. The key observation is that the right hand side of 16.12 is of rank three owing to the dimensions of **A** and **P**. The rank of **S** may be more than three due

[1] The reference point does not need to be a point of our data set, but can be the barycenter of the available points. In the affine set-up, the barycenter of the three-dimensional points projects to the barycenter of the image points.

to measurement noise, but the best rank three approximation to **S** can be obtained from the Singular Value Decomposition (SVD)

$$\mathbf{S} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{16.13}$$

or

$$s_{kl} = \sum_m u_{km} d_m v_{lm} \tag{16.14}$$

If we limit the summation in the latter equation to the three largest singular values d_1, d_2, and d_3[2], we obtain an optimal rank three factorization of **S** and we have the following estimates for the structure and motion

$$\hat{\mathbf{P}} = \begin{pmatrix} \mathbf{d}_1 & 0 & 0 \\ 0 & \mathbf{d}_2 & 0 \\ 0 & 0 & \mathbf{d}_3 \end{pmatrix}^{\frac{1}{2}} \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{pmatrix}^T \tag{16.15}$$

and

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{pmatrix} \begin{pmatrix} \mathbf{d}_1 & 0 & 0 \\ 0 & \mathbf{d}_2 & 0 \\ 0 & 0 & \mathbf{d}_3 \end{pmatrix}^{\frac{1}{2}}, \tag{16.16}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ denote columns of the matrices **U** and **V**.

The factorization above is however not the only one. Consider the matrices $\mathbf{A}' = \mathbf{A}\mathbf{T}$ and $\mathbf{P}' = \mathbf{T}^{-1}\mathbf{P}$, which are also possible factors of **S**

$$\mathbf{A}'\mathbf{P}' = \mathbf{A}\mathbf{T}\mathbf{T}^{-1}\mathbf{P} = \mathbf{S} \tag{16.17}$$

There are however some constraints on our motion matrix **A** that allows the estimation of the best possible **T**

$$|\mathbf{I}| = |\mathbf{J}| \qquad \textit{and} \qquad \mathbf{I}\cdot\mathbf{J} = 0 \tag{16.18}$$

The constraints arise from the fact that the camera performs a rigid motion, i.e. pure rotation and translation. Once we have estimated the matrix **T** that makes the motion rigid we also achieve that our reconstruction is Euclidean, and not just affine. In fact, **T** may only be determined up to an unknown sign, but this ambiguity can be resolved in a later processing step. The determination of **T** will not be described here.

[2] assuming that the singular values are ordered in decreasing order.

We can now estimate the Euclidean motion by

$$\mathbf{i}_j = \mathbf{I}_j/||\mathbf{I}_j|| \tag{16.19}$$

$$\mathbf{j}_j = \mathbf{J}_j/||\mathbf{J}_j|| \tag{16.20}$$

$$\mathbf{k}_j = \mathbf{i}_j \times \mathbf{j}_j \tag{16.21}$$

$$t_{z_j} = \frac{1}{2}(1/||\mathbf{I}_j|| + 1/||\mathbf{J}_j||) \tag{16.22}$$

$$t_{x_j} = x_{0j}t_{z_j} \tag{16.23}$$

$$t_{y_j} = y_{0j}t_{z_j} \tag{16.24}$$

Even with the estimation of $\mathbf{T}$ we cannot be sure that the vectors $\mathbf{i}$ and $\mathbf{j}$ are orthogonal. We should therefore orthogonalize the rotation matrix $(\mathbf{i}, \mathbf{j}, \mathbf{k})$. Christy et al. propose to solve the problem using a 3D-3D pose algorithm. However, the solution may be calculated directly in the particularly simple case where $\mathbf{k}$ is orthogonal to $\mathbf{i}$ and $\mathbf{j}$. The orthogonalized matrix $(\mathbf{i}', \mathbf{j}', \mathbf{k}')$ is

$$\mathbf{k}' = \mathbf{k} \tag{16.25}$$

$$\mathbf{j}' = \mathbf{i} + \mathbf{j} + \mathbf{k} \times (\mathbf{i} + \mathbf{j})/||\mathbf{i} + \mathbf{j} + \mathbf{k} \times (\mathbf{i} + \mathbf{j})|| \tag{16.26}$$

$$\mathbf{i}' = \mathbf{j}' \times \mathbf{k}' \tag{16.27}$$

16.4 Perspective Reconstruction

The above section have shown how the structure and motion problem may be solved using a simplified camera model, namely the weak perspective camera. The result is obtained by using linear algorithms and the approach can therefore be used without any initial estimates of the motion or the structure. The result can be improved by non-linear optimization using the complete camera model. Christy and Horaud [7] have proposed a method that takes advantage of the simple relation between the weak perspective camera model and the true perspective camera model.

The algorithm proceeds by iteratively performing affine reconstructions. At each step the perspective effect is estimated and subtracted from the data, such that the affine model becomes more and more valid. Given an Euclidean motion the perspective effect is estimated as

$$\epsilon_{ij} = \mathbf{k}_j \cdot \mathbf{P}_i/t_{z_j} \tag{16.28}$$

From equations 16.6 and 16.7 we have the relation between the observed perspective point coordinates (x_{ij}, y_{ij}) and the weak perspective point coordinates (x^w_{ij}, y^w_{ij})

$$x^w_{ij} = x_{ij}(1 + \epsilon_{ij}) \tag{16.29}$$

$$y^w_{ij} = y_{ij}(1 + \epsilon_{ij}) \tag{16.30}$$

The left hand side of the above equations is then used as input for the factorization algorithm in the next step.

It was mentioned earlier that the matrix $\mathbf{T}$ can only be determined up to a possible reflection of the reconstruction. This problem is resolved as the algorithm converges as the perspective effect will remove the ambiguity.

16.5 Results

16.5.1 The OSS Mock-Up

At OSS a small model of a ship block has been built to test vision algorithms. The mock-up is not part of an actual ship, but contains many of the features encountered on a ship block. At OSS it is interesting to be able to measure such a ship block as well as to move a robot to one of its weld lines. Assuming that the mock-up is placed arbitrarily in the camera's field of view, a first step towards one of these tasks is to estimate the location of the mock-up. We do this by relating CAD features to features observed in the images, i.e. we need to reconstruct the observed object.

The reconstruction is based on the described theory. We need to observe the same physical features from various view points to apply the algorithm. The easiest and most robust way to obtain this is by tracking the features through a sequence of images. Since the inter-image distance is small, the appearance of the feature will only change slightly from image to image, and the feature may be tracked by e.g. image correlation. Figure 16.1 shows five images of a sequence acquired of the mock-up. The sequence has in total eighty images. Unfortunately, the mock-up is very dark and has very little contrast compared to the surrounding world. To enhance the features of the mock-up, the grey levels of the images have been stretched, giving more contrast to the darker regions. At first glance, the images reveal some obvious pitfalls. The first is the shadows that introduce false edges, and consequently false features,

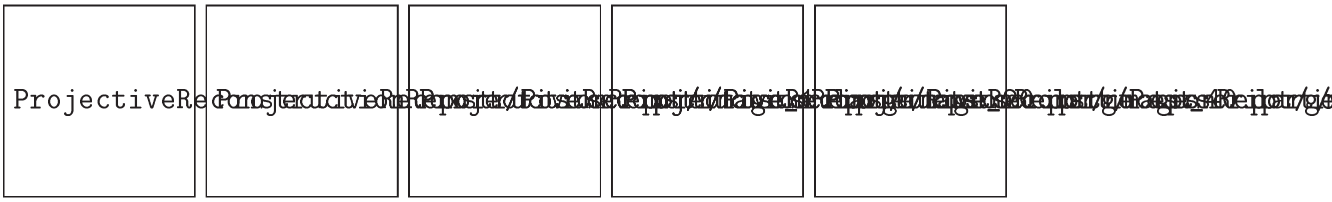

Figure 16.1: Images 1, 20, 40, 60, and 80 of the image sequence of the mock-up.

on the mock-up. The second is all the structure in the surroundings, that does not belong to the mock-up. These false features does not pose a problem for the reconstruction as they represent fixed points in the three-dimensional world, but they will make automatic matching with the CAD model very difficult. The third problem is the many lines that appear to intersect in the two-dimensional projection, but are actually skew lines in three-dimensional space. These points do not correspond to a fixed three-dimensional points as the camera moves and will corrupt the reconstruction algorithm.

16.5.2 Corner Detection

The first task in the tracking algorithm is to detect some points in the first image, that are likely to be corners on the mock-up. An algorithm called the Harris corner detector [21] is applied to do this. This algorithm computes the local auto-correlation of the image and selects points where the curvature is large in both image directions. These points are likely to be corners and are well localized. The result of such a detection is shown in figure 16.2. It is seen that the algorithm has detected many of the true corners of the mock-up. As expected, it has unfortunately also detected some false features as described above. It is possible to detect more points by lowering the tolerance of the Harris corner detector, but the small gain in true features is vastly overshadowed by the number of additional false features.

16.5.3 Tracking

The points that have been detected in the first image are now tracked through the sequence. The trajectory of part of the tracking is displayed in figure 16.3. The tracking is performed by local correlation. Unfortunately, some points are impossible to track from one image to another. This may be due to occlusion

ProjectiveReconstructionReport/Postscript/extract.ps

Figure 16.2: Points extracted by the Harris corner detector.

or to a change of appearance as the camera moves. It is therefore necessary to perform an occasional re-detection of features, when important CAD corners have been lost. This means that the tracking algorithm is only semi-automatic. Since the tracking is slow and an operator needs to survey the process, the presented approach is not suited for practical purposes. Furthermore, the points that are re-detected are not identified with the lost points, and the reconstruction will contain multiple candidates for the corresponding three-dimensional corners. The problem is shown in figure 16.4. The points that have been tracked, and where some have been re-detected, have been entered into the reconstruction algorithm. The reconstructed three-dimensional structure has been re-projected into the images. Note, how the upper corners in the foreground are represented by multiple points. Whereas it may be possible to fuse some of the spurious points in the foreground, it is obviously not so easy in the complex region in the left part of the image.

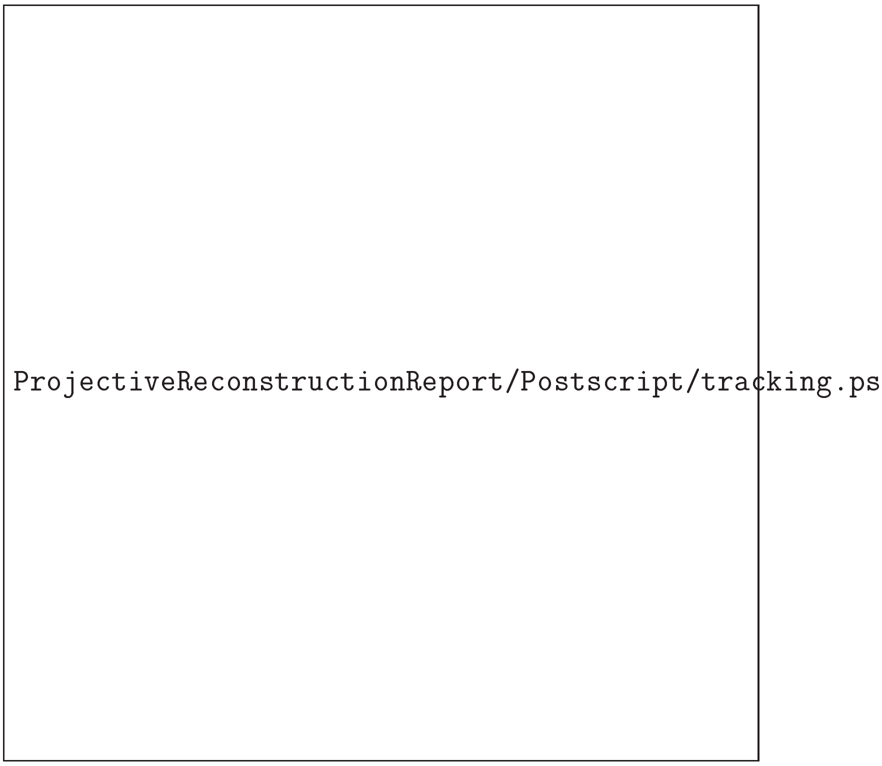

Figure 16.3: Trajectories of tracked points.

Figure 16.4: Re-projection of reconstructed points. Notice how points that have been lost and re-extracted appear multiple times.

16.5.4 Reconstruction and Matching

The tracked points are used as input to the reconstruction algorithm. This algorithm will give a three-dimensional interpretation of the points seen in the image sequence. The back-projection of the reconstruction was shown in figure 16.4, but the quality is easier to evaluate when a CAD model is matched to the reconstructed points. The matching is performed by manually specifying corresponding points and subsequently calculating the three-dimensional rigid motion between the reconstruction and the CAD model. Once the CAD model has been matched to the reconstructed structure, it can be projected into the images by the reconstructed motion. The result is shown in figure 16.5. The result shows that a lot of information about the three-dimensional space has been recovered by the reconstruction. However, the quality of the result is far from suitable for robot manipulation or for measurement. It should be noticed, though, that the general motion of the camera has been recovered rather well, even in this noisy set-up.

The drawback of the approach lies in the unreliable point tracking. To demonstrate the capability of the pure reconstruction, a series of 17 points were tracked by hand in the five images shown in figure 16.1. In the same manner as before a reconstruction was performed, the CAD model was matched, and the model was projected into the images. The result is shown in figure 16.6. Clearly, the position of the CAD corners and the motion of the camera has

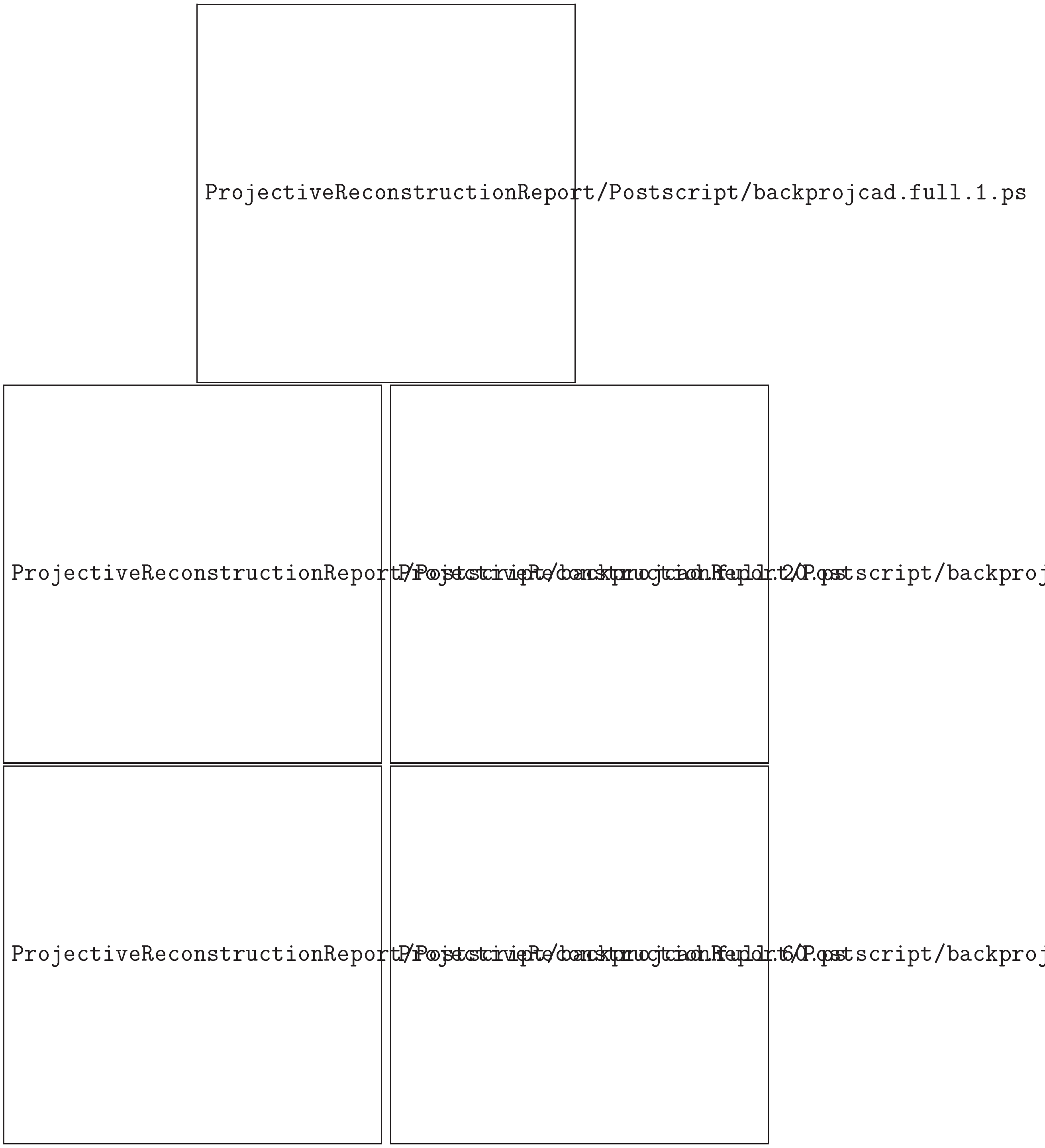
ProjectiveReconstructionReport/Postscript/backprojcad.full.1.ps

Figure 16.5: The CAD model has been matched to the reconstructed three-dimensional points and is projected into the images. Points have been tracked automatically.

been recovered with high accuracy. If a reconstruction of this quality could be obtained automatically it would be straightforward to proceed with sub-pixel localization of features, and calculate accurate positions and dimensions on the observed mock-up.

16.6 Conclusion

This report has described the basic theory behind the solution of the *structure and motion* problem. The algorithm has been tested on a mock-up from OSS. The tracking of points seems not to be a feasible way to obtain points to enter the reconstruction algorithm. Too many points are lost during tracking and the ratio of CAD related points to non-CAD related points seems to be too small to implement automatic matching reliably. Since the majority of points are tracked correctly from image to image, the reconstructed motion is in good agreement with the actual motion.

One way to solve the sketched problems could be to track lines instead of points. Lines can be detected more accurately, are easier to track, and will most often be true features of the observed object. With this reduced set of more reliable features, it should be possible to match the reconstruction and the CAD model automatically.

16.7 Acknowledgements

The author acknowledge support from the European Commission through the Esprit IV programme, reactive LTR project number 26247 VIGOR.

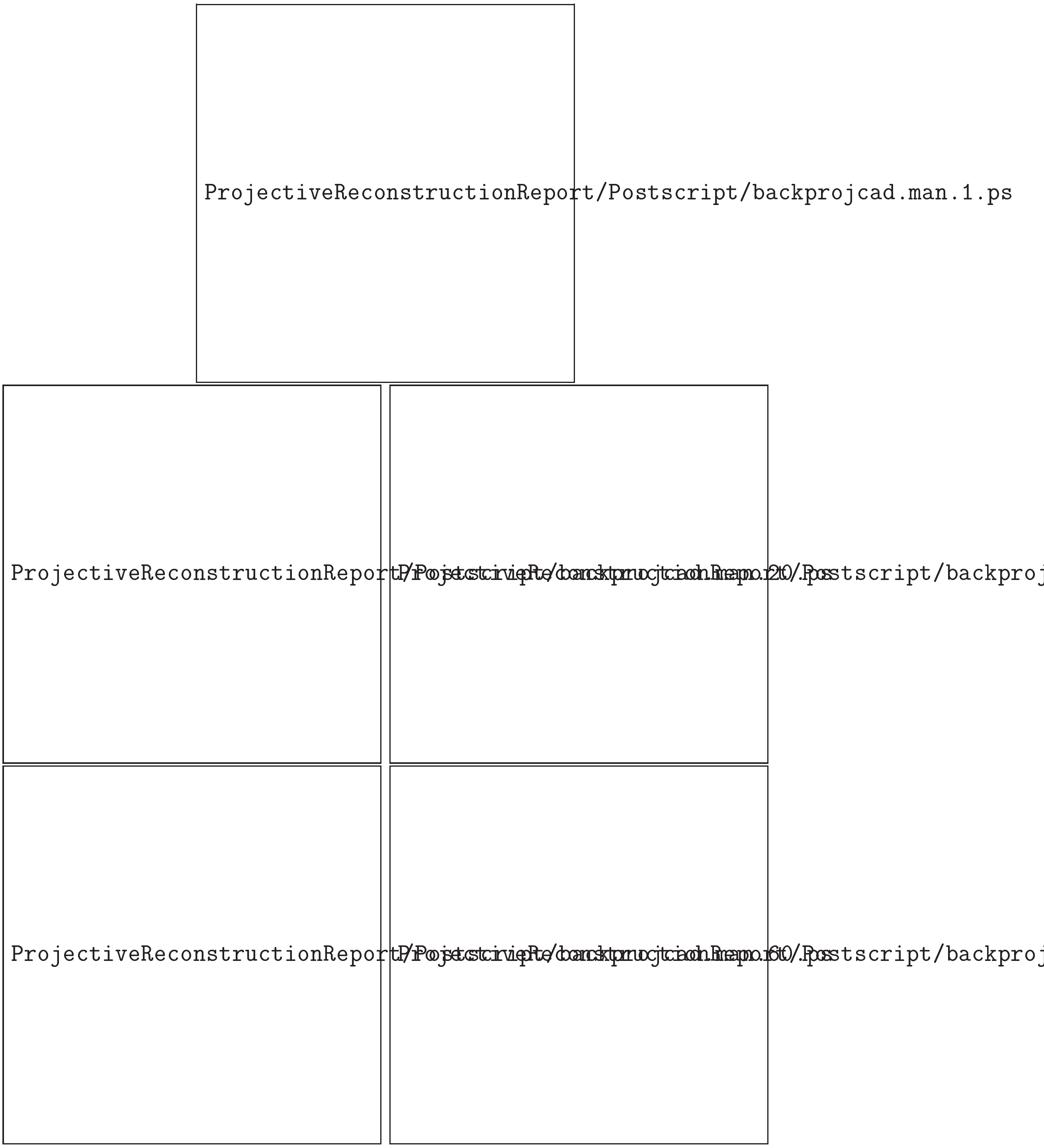
ProjectiveReconstructionReport/Postscript/backprojcad.man.1.ps
ProjectiveReconstructionReport/Postscript/backprojcad.man.20.ps
ProjectiveReconstructionReport/Postscript/backproj
ProjectiveReconstructionReport/Postscript/backprojcad.man.60.ps
ProjectiveReconstructionReport/Postscript/backproj

Figure 16.6: The CAD model has been matched to the reconstructed three-dimensional points and is projected into the images. Points have been tracked manually.

Appendix A

Point-Matching in the Plane

In the course of the project the need to relate two-dimensional measured positions to a CAD model has occurred in different forms. Mostly, the model is known up to a rigid motion, described by a rotation and a translation. In other situations, only the general shape of the model is known, e.g. a rectangle with unknown dimensions. In either case, we need to estimate an affine mapping from the CAD model to the measured data satisfying a set of constraints.

Formally, we can assume a set of point pairs, $(\mathbf{x}_i, \mathbf{y}_i),\ i = 1, \ldots, N$, where $\mathbf{x}_i$ are points in the CAD model and $\mathbf{y}_i$ are measured points. We want to estimate an affine map that minimizes the least squares distance between the two set of points. The affine model takes the form

$$\mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{t} + \epsilon_i, \qquad i = 1, \ldots, N, \tag{A.1}$$

where $(\mathbf{A}, \mathbf{t})$ are the affine parameters and ϵ_i is the residual related to measurement number i. In the least squares optimization we minimize the sum of squared norms of the residuals. The cost function becomes

$$\begin{aligned} \mathcal{F} &= \frac{1}{N}\sum_i |\epsilon_i|^2 && \text{(A.2)} \\ &= \frac{1}{N}\sum_i \epsilon_i^T \epsilon_i && \text{(A.3)} \\ &= \frac{1}{N}\sum_i (\mathbf{y}_i - \mathbf{A}\mathbf{x}_i - \mathbf{t})^T(\mathbf{y}_i - \mathbf{A}\mathbf{x}_i - \mathbf{t}) && \text{(A.4)} \end{aligned}$$

A.1 Translation

The translation vector $\mathbf{t}$ can easily be determined by taking the partial derivative of the cost function, and setting equal to zero

$$\begin{aligned}\frac{\partial \mathcal{F}}{\partial \mathbf{t}} &= -\frac{1}{N}\sum_i 2(\mathbf{y}_i - \mathbf{A}\mathbf{x}_i - \mathbf{t}) && \text{(A.5)}\\ &= -2(\bar{\mathbf{y}} - \mathbf{A}\bar{\mathbf{x}} - \mathbf{t}) && \text{(A.6)}\\ &= \mathbf{0}, && \text{(A.7)}\end{aligned}$$

where the bar denotes averages over N. Solving for $\mathbf{t}$ yields

$$\mathbf{t} = \bar{\mathbf{y}} - \mathbf{A}\bar{\mathbf{x}} \qquad \text{(A.8)}$$

which shows that the translation merely matches the center of mass of the two data sets. If the data are centered, so that $\bar{\mathbf{x}} = \bar{\mathbf{y}} = 0$, we see that $\mathbf{t}$ becomes independent of $\mathbf{A}$ owing to the linearity. This holds no matter what constraints we put on $\mathbf{A}$. We can therefore just as well consider the model

$$\mathbf{y}_i - \bar{\mathbf{y}} = \mathbf{A}(\mathbf{x}_i - \bar{\mathbf{x}}) + \epsilon_i, \qquad i = 1, \ldots, N \qquad \text{(A.9)}$$

In the sequel we assume the data to be centered. The model for the un-centered data A.1 is then obtained subsequently by the use of equation A.8.

A.2 Rotation

Most often we have an exact CAD model of the measured part. We only need to determine the position and rotation relative to our measurements. This corresponds to the case where the affine matrix is a proper rotation $\mathbf{A} = \mathbf{R}$. The cost function becomes

$$\begin{aligned}\mathcal{F}_{\mathbf{R}} &= \frac{1}{N}\sum_i (\mathbf{y}_i - \mathbf{R}\mathbf{x}_i)^T(\mathbf{y}_i - \mathbf{R}\mathbf{x}_i) && \text{(A.10)}\\ &= \frac{1}{N}\sum_i \mathbf{y}_i^T\mathbf{y}_i - 2\mathbf{y}_i^T\mathbf{R}\mathbf{x}_i + \mathbf{x}_i^T\mathbf{x}_i, && \text{(A.11)}\end{aligned}$$

where the orthogonality of $\mathbf{R}$ was applied in the last term. To bring the unknown matrix $\mathbf{R}$ into vector form we introduce the equivalence

$$\mathbf{R}\mathbf{x} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \text{(A.12)}$$

$$
\begin{aligned}
&= \begin{bmatrix} x_1 & -x_2 \\ x_2 & x_1 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} && \text{(A.13)} \\
&= \mathbf{Xr}, \quad |\mathbf{r}| = 1 && \text{(A.14)}
\end{aligned}
$$

This way we indirectly force the rotation to be proper, i.e. $det(\mathbf{R}) = 1$. The unity constraint on $\mathbf{r}$ is introduced by the Lagrange multiplier λ, and the total cost function becomes

$$\mathcal{F}'_{\mathbf{R}} = \overline{\mathbf{x}^T\mathbf{x}} + \overline{\mathbf{y}^T\mathbf{y}} - 2\overline{\mathbf{y}^T\mathbf{X}}\mathbf{r} + \lambda(\mathbf{r}^T\mathbf{r} - 1) \tag{A.15}$$

Taking the partial derivative with respect to $\mathbf{r}$ and setting equal to zero yields

$$
\begin{aligned}
\frac{\partial \mathcal{F}'_{\mathbf{R}}}{\partial \mathbf{r}} &= -2\overline{\mathbf{y}^T\mathbf{X}} + 2\lambda\mathbf{r} && \text{(A.16)} \\
&= \mathbf{0} && \text{(A.17)}
\end{aligned}
$$

This gives a closed-form solution for $\mathbf{r}$

$$\mathbf{r} = \frac{1}{\lambda}\overline{\mathbf{y}^T\mathbf{X}}, \tag{A.18}$$

where the unity constraint naturally gives

$$\lambda = |\overline{\mathbf{y}^T\mathbf{X}}| \tag{A.19}$$

By direct insertion, this gives the least squares error in optimum

$$\mathcal{F}'_{\mathbf{R}} = \overline{\mathbf{x}^T\mathbf{x}} + \overline{\mathbf{y}^T\mathbf{y}} - 2|\overline{\mathbf{y}^T\mathbf{X}}| \tag{A.20}$$

A.3 Rotation with Independent Scales

In a certain application we were faced with the problem of estimating the best rectangle that approximate four measured corners, in order to construct an artificial CAD model. This problem may be identified with the problem of optimizing the map between a unit square and the four corners, where the map allows for rotation, translation, and independent scaling of the width and length of the square. The considered affine model becomes

$$\mathbf{y}_i = \mathbf{RDx}_i + \epsilon_i, \qquad i = 1, \ldots, N \tag{A.21}$$

where the data are still assumed to be centered. The diagonal matrix $\mathbf{D}$ contains the two independent scales of the model. We write the cost function as

$$\mathcal{F}_{\mathbf{R},\mathbf{D}} = \frac{1}{N}\sum_i (\mathbf{y}_i - \mathbf{R}\mathbf{D}\mathbf{x}_i)^T(\mathbf{y}_i - \mathbf{R}\mathbf{D}\mathbf{x}_i) \qquad (A.22)$$

$$= \frac{1}{N}\sum_i \mathbf{y}_i^T\mathbf{y}_i - 2{\mathbf{y}_i}^T\mathbf{R}\mathbf{D}\mathbf{x}_i + \mathbf{x}_i^T\mathbf{D}^T\mathbf{D}\mathbf{x}_i, \qquad (A.23)$$

where the orthogonality of $\mathbf{R}$ was applied in the last term. To bring the unknown matrices $\mathbf{R}$ and $\mathbf{D}$ into vector form we introduce the following equivalences

$$\mathbf{R}^T\mathbf{y} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \qquad (A.24)$$

$$= \begin{bmatrix} y_1 & y_2 \\ y_2 & -y_1 \end{bmatrix}\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \qquad (A.25)$$

$$= \mathbf{Y}\mathbf{r}, \quad |\mathbf{r}| = 1 \qquad (A.26)$$

and

$$\mathbf{D}\mathbf{x} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad (A.27)$$

$$= \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix}\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \qquad (A.28)$$

$$= \mathbf{X}\mathbf{d} \qquad (A.29)$$

The unity constraint on $\mathbf{r}$ is introduced by the Lagrange multiplier λ, and the total cost function becomes

$$\mathcal{F}'_{\mathbf{R},\mathbf{D}} = \overline{\mathbf{y}^T\mathbf{y}} - 2\mathbf{r}^T\overline{\mathbf{Y}^T\mathbf{X}}\mathbf{d} + \mathbf{d}^T\overline{\mathbf{X}^T\mathbf{X}}\mathbf{d} + \lambda(\mathbf{r}^T\mathbf{r} - 1) \qquad (A.30)$$

Taking the partial derivative with respect to $\mathbf{r}$ and $\mathbf{d}$ and setting equal to zero yields

$$\frac{\partial\mathcal{F}'_{\mathbf{R},\mathbf{D}}}{\partial\mathbf{r}} = -2\overline{\mathbf{Y}^T\mathbf{X}}\mathbf{d} + 2\lambda\mathbf{r} \qquad (A.31)$$

$$= \mathbf{0} \qquad (A.32)$$

and

$$\frac{\partial\mathcal{F}'_{\mathbf{R},\mathbf{D}}}{\partial\mathbf{d}} = -2\overline{\mathbf{X}^T\mathbf{Y}}\mathbf{r} + 2\overline{\mathbf{X}^T\mathbf{X}}\mathbf{d} \qquad (A.33)$$

$$= \mathbf{0} \qquad (A.34)$$

Solving the latter equation for $\mathbf{d}$ gives

$$\mathbf{d} = \overline{\mathbf{X}^T\mathbf{X}}^{-1}\ \overline{\mathbf{X}^T\mathbf{Y}}\mathbf{r}, \tag{A.35}$$

which inserted in the former equation leads to

$$\overline{\mathbf{Y}^T\mathbf{X}}\ \overline{\mathbf{X}^T\mathbf{X}}^{-1}\ \overline{\mathbf{X}^T\mathbf{Y}}\mathbf{r} = \lambda\mathbf{r} \tag{A.36}$$

That is, $(\lambda, \mathbf{r})$ is an eigensolution to the matrix $\overline{\mathbf{Y}^T\mathbf{X}}\ \overline{\mathbf{X}^T\mathbf{X}}^{-1}\ \overline{\mathbf{X}^T\mathbf{Y}}$. This result can be inserted in the cost function to find

$$\mathcal{F}'_{\mathbf{R},\mathbf{D}} = \overline{\mathbf{y}^T\mathbf{y}} - \lambda \tag{A.37}$$

which means that $\mathbf{r}$ is the eigenvector corresponding to the greatest eigenvalue. The scale vector $\mathbf{d}$ can now be determined from equation A.35.

Appendix B

Analysis of Variance

Complex systems consist of a large number of components, each contributing to the total uncertainty of the system. A vision system, for instance, contain off-line calibration, on site pose estimation, measurement, and statistical validation against *ground truth*. The different components have different noise processes and we need to analyze these to estimate the accuracy of the vision output. If we assume a normal distribution for our errors, which is fair for the given processes, it is easy to propagate uncertainty through the individual parts of the system.

Another benefit of the analysis is the detection of singularities in e.g. the view-points chosen in the calibration of the camera or the lay-out of reference marks in an application.

The appendix will describe how the propagation of uncertainty through non-linear functions can be approximated, how to calculate conditional variance in the normal distribution, and the basic theory used for residual analysis in image space.

B.1 Propagation of Uncertainty

This section is inspired by *Conradsen* [9] and *Press et al.* [61]. The implementation of the Levenberg-Marquardt algorithm is based on *Press et al.* [61]. When we fit a model to some observed data, the uncertainty of the observations is propagated to the model parameters. This is the case when we estimate the parameters in a camera model from a set of world points and the corresponding image points. It also occurs when a model is fitted to the image intensities of a reference mark. Typically, the model is non-linear in the parameters $\mathbf{a}$. If we assume that the model $\mathbf{f}(\mathbf{x}, \mathbf{a})$ fully explains the deterministic part of the relation between the observations $\mathbf{y}$ and the regression variables $\mathbf{x}$, we can write

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \mu_a) + \epsilon, \tag{B.1}$$

where ϵ is the observation noise. It is modelled as a normal distribution with mean zero, $\epsilon \in N(\mathbf{0}, \mathbf{\Sigma})$. Normally, the observations are assumed to be independent and identically distributed, so that $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$. The assumed underlying 'true' value of the model parameters is denoted μ_a. If we observe p two-dimensional image points, the dimension of $\mathbf{y}$ will be $n = 2p$. The values of the parameters are unknown and must be estimated from the observations. We minimize the least squares cost function

$$\begin{aligned} \mathcal{F}(\mathbf{a}) &= \epsilon^T \mathbf{\Sigma}^{-1} \epsilon & \text{(B.2)} \\ &= (\mathbf{y} - \mathbf{f}(\mathbf{x}, \mathbf{a}))^T \mathbf{\Sigma}^{-1} (\mathbf{y} - \mathbf{f}(\mathbf{x}, \mathbf{a})) & \text{(B.3)} \end{aligned}$$

We see that $\mathcal{F}(\mu_a) \in \chi^2(n)$. The function is minimized by the Levenberg-Marquardt algorithm. Close to the optimal parameter value this algorithm uses Newton steps to arrive at the minimum. We therefore need the gradient vector and the Hessian matrix of the cost function. Taking the partial derivative with respect to $\mathbf{a}$ yields

$$\frac{\partial \mathcal{F}(\mathbf{a})}{\partial \mathbf{a}} = -2 \frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} (\mathbf{y} - \mathbf{f}(\mathbf{x}, \mathbf{a})) \tag{B.4}$$

The second derivative can be approximated by

$$\frac{\partial^2 \mathcal{F}(\mathbf{a})}{\partial \mathbf{a} \partial \mathbf{a}^T} \approx 2 \frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}}, \tag{B.5}$$

where the second derivative term of $\mathbf{f}$ has been omitted. This approximated expression is used for several reasons. First of all, this expression is guaranteed to be positive semi-definite. Secondly, the second derivatives of $\mathbf{f}$ are small for a reasonable model, and these terms are weighted by the residuals, which are also small. Note, that equation B.5 could also have been obtained by using a first order expansion of equation B.1. We see that equation B.4 as well as equation B.5 is based on the first derivative of $\mathbf{f}$, which can be calculated analytically.

After a number of iterations in the Levenberg-Marquardt algorithm we obtain a value of $\mathbf{a}$ very close to μ_a. The final estimate is given by the Newton step

$$\begin{aligned}\hat{\mu}_a &= \mathbf{a} - \left(\frac{\partial^2 \mathcal{F}(\mathbf{a})}{\partial \mathbf{a} \partial \mathbf{a}^T}\right)^{-1} \frac{\partial \mathcal{F}(\mathbf{a})}{\partial \mathbf{a}} && \text{(B.6)}\\ &= \mathbf{a} + \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right)^{-1} \frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} (\mathbf{y} - \mathbf{f}(\mathbf{x}, \mathbf{a})) && \text{(B.7)}\end{aligned}$$

To evaluate the statistics of this estimate, we apply the approximation

$$\mathbf{f}(\mathbf{x}, \mathbf{a}) = \mathbf{f}(\mathbf{x}, \mu_a) + \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{a})}{\partial \mathbf{a}} (\mathbf{a} - \mu_a), \tag{B.8}$$

which leads to

$$\begin{aligned}E(\hat{\mu}_a) &= \mathbf{a} + \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right)^{-1} \frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} E(\epsilon - \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{a})}{\partial \mathbf{a}} (\mathbf{a} - \mu_a)) && \text{(B.9)}\\ &= \mu_a && \text{(B.10)}\end{aligned}$$

$\hat{\mu}_a$ is therefore a central estimate for μ_a. By straightforward calculations we also obtain the dispersion of $\hat{\mu}_a$

$$\begin{aligned}D(\hat{\mu}_a) &= D(\left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right)^{-1} \frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \epsilon) && \text{(B.11)}\\ &= \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right)^{-1} \frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} D(\epsilon) \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}} \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right)^{-1} && \text{(B.12)}\\ &= \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right)^{-1} \frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}} \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right)^{-1} && \text{(B.13)}\\ &= \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{a}} \mathbf{\Sigma}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right)^{-1} && \text{(B.14)}\end{aligned}$$

Thus, the dispersion of $\hat{\mu}_a$ is immediately available as the Hessian matrix of the cost function. Note, that the expressions above are in agreement with the corresponding expressions in the linear regression. As expected a large gradient of $\mathbf{f}$ will decrease the uncertainty in the estimation of μ_a.

Assuming that we know the value of $\mathbf{\Sigma}$ we can perform a test of fit to validate our model. Under the hypothesis that the model is valid, the cost function follows a χ^2-distribution

$$\mathcal{F}(\hat{\mu}_a) \in \chi^2(n - m), \tag{B.15}$$

where m is the dimension of $\mathbf{a}$. However, $\boldsymbol{\Sigma}$ is usually unknown and we assume that $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, so that the the observations are independent and identically distributed. In that case, we proceed with $\sigma = 1$, so that $\mathcal{F}(\hat{\mu}_a) \in \sigma^2\chi^2(n-m)$, and we can finally estimate the value of σ to be

$$\sigma = \sqrt{\mathcal{F}(\hat{\mu}_a)/(n-m)} \tag{B.16}$$

B.2 Conditional Variance in the Normal Distribution

Very often, the parameter dispersion matrices are very large, making them difficult to interpret. For instance, a calibration with v views and p internal parameters gives rise to a parameter vector of dimension $m = p + 6v$. In addition, some of these parameters are known to be extremely correlated, thus making the overall uncertainty seem greater than it is. In fact, we are more interested in knowing the dispersion of a subset of the parameters when we assume all other parameters fixed. We do that by calculating the conditional dispersion. If we consider a normally distributed variable

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \in N(\begin{bmatrix} \mu_{a_1} \\ \mu_{a_2} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}) \tag{B.17}$$

we can find the distribution of $\mathbf{a}_1$ conditional on $\mathbf{a}_2$

$$\mathbf{a}_1|\mathbf{a}_2 \in N(\mu_{a_1} + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{a}_2 - \mu_{a_2}), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}) \tag{B.18}$$

Note, that the conditional dispersion of $\mathbf{a}_1$ is reduced by the knowledge of $\mathbf{a}_2$. The reduction is determined by the magnitude of the dispersion of $\mathbf{a}_2$ and the covariance between the two vectors. To ease the calculations, we take advantage of the fact that we can condition on the parameters one by one. We can write

$$\mathbf{a}_1|\mathbf{a}_2 \sim (\ldots((\mathbf{a}_1, a_{2(1)}, \ldots, a_{2(m-1)})|a_{2(m)})|\ldots)|a_{2(1)}, \tag{B.19}$$

where $a_{2(\cdot)}$ denote the individual entries of $\mathbf{a}_2$. Doing this we avoid the matrix inversion, as Σ_{22} in the conditioned distribution B.18 becomes scalar. The proof is based on induction and follows from straightforward calculations.

B.3 Two-dimensional Residual Analysis

There are several pitfalls in fitting models to observed data. First, one has to find a reasonable model. Next, the dimension of the model must be determined. If the number of parameters is chosen too large we risk to incorporate the stochastic noise in the deterministic model, and this noise will then affect the predictions served by the model in the future. If the number of parameters is too small, the model is not able to explain the deterministic behaviour of the observed data. This section is concerned with the latter situation in the case of camera calibration.

Now, how do we reveal an insufficient model? If the model fail to fit the data, the calibration residuals will locally have the same sign or in the two-dimensional case the same orientation. It is therefore appropriate to test whether the residuals locally can be assumed to have a trend in some direction or if they can be assumed to follow a distribution with mean zero. Such a test is given by Hotelling's T^2-test [9]. We want to test the hypothesis that the local mean of the residuals is zero, i.e.

$$H_0 \ : \ \mu_\epsilon = \mu_0 \qquad \text{against} \qquad H_1 \ : \ \mu_\epsilon \neq \mu_0, \tag{B.20}$$

where $\mu_0 = \mathbf{0}$. In order to calculate the test statistic we need the local statistics of the residuals. Let

$$\overline{\epsilon} = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \tag{B.21}$$

be the local mean of the residuals and let

$$\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\epsilon - \overline{\epsilon})(\epsilon - \overline{\epsilon})^T \tag{B.22}$$

be the local empirical dispersion of the residuals. The test statistic is then

$$T^2 = n(\overline{\epsilon} - \mu_0)^T \mathbf{S}^{-1}(\overline{\epsilon} - \mu_0) \tag{B.23}$$

We now have

$$\frac{n-p}{(n-1)p}T^2 \in F(p, n-p) \tag{B.24}$$

under H_0, where p is the dimension of the residual. For the residuals in the image plane, we naturally have $p = 2$. Consequently, the hypothesis can be tested in the F-distribution and H_0 is rejected for large values of the test statistic.

In the present application it is obviously not fair to require that H_0 is *not* rejected in all local tests on a significance level of say $\alpha = 5\%$. We should either choose a significance level that reflect the number of tests t, $\alpha_t = 1 - \sqrt[t]{1-\alpha}$, or we should compare the observed rejection rate to the theoretical rejection rate of $\alpha = 5\%$.

Another issue concerning the interpretation of the test result is the spatial layout of the calibration marks in the images. Remember, that the calibration is based on a number of images of the calibration plate acquired from different points of view. From these images the internal parameters of the camera are determined along with the individual pose associated with each point of view. Hence, each image has six degrees of freedom dedicated to minimize its own residuals. If the extent of the calibration marks in the image is limited to a small region, this set-up will induce local anti-correlation of the residuals. This will of course reduce the rejection rate of the proposed test. The quality of the test is therefore increased if the calibration marks exhibit a global nature in all images.

Bibliography

[1] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, september 1987.

[2] K. Åström. *Invariancy Methods for Points, Curves and Surfaces in Computational Vision.* Doctoral dissertation, Department of Mathematics, Lund Institute of Technology, Lund, Sweden, may 1996.

[3] K. Åström and A. Heyden. Stochastic modelling and analysis of image acquisition and sub-pixel edge detection. In *ICPR96*, 1996.

[4] T. Blaszka and R. Deriche. Recovering and characterizing image features using an efficient model based approach. Rapport de Recherche 2422, Institut National de Recherche en Informatique et en Automatique, november 1994.

[5] C. M. Brown. Inherent bias and noise in the hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(5):493–504, september 1983.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, november 1986.

[7] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. Rapport de Recherche 2421, Institut National de Recherche en Informatique et en Automatique, december 1994.

[8] S. Christy and R. Horaud. Fast and reliable object pose estimation fron line correspondences. In *7th International Conference on Computer Anal-*

ysis of Images and Patterns, pages 432–439, Kiel, Germany, September 1997.

[9] K. Conradsen. *En Introduktion til Statistik, Bind 2A-B*. IMSOR, DTH, Lyngby, Denmark, 4. edition, 1984.

[10] S. R. Deans. Hough transform from the radon transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(2):185–188, march 1981.

[11] D.F. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.

[12] R. Deriche. Recursively implementing the gaussian and its derivatives. Rapport de Recherche 1893, Institut National de Recherche en Informatique et en Automatique, april 1993.

[13] F. Devernay. A non-maxima suppression method for edge detection with sub-pixel accuracy. Rapport de Recherche 2724, Institut National de Recherche en Informatique et en Automatique, november 1995.

[14] N. A. Dodgson. Quadratic interpolation for image resampling. *IEEE Transactions on Image Processing*, 6(9):1322–1326, september 1997.

[15] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, january 1972.

[16] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In *European Conference on Computer Vision*, Santa Margherita, 1992.

[17] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. Rapport de Recherche 2665, Institut National de Recherche en Informatique et en Automatique, october 1995.

[18] O. Faugeras and L. Robert. What can two images tell us about a third one? Rapport de Recherche 2018, Institut National de Recherche en Informatique et en Automatique, july 1993.

[19] P. Gros, R. Hartley, R. Mohr, and L. Quan. How useful is projective geometry? *Computer Vision and Image Understanding*, 65(3):442–446, 1997.

[20] R.M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 1994.

[21] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[22] R. Hartley. In defense of the 8-point algorithm. In *Fifth International Conference on Computer Vision*, pages 1064–1070, Cambridge, Massachusetts, june 1988.

[23] R. Hartley and P. Sturm. Triangulation. In *Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns*, pages 190–197, Prague, Czech Republic, september 1995.

[24] R.I. Hartley. Self-calibration of stationary cameras. *Int. Journal of Computer Vision*, 22(1):5–23, 1997.

[25] R.I. Hartley and R. Gupta. Computing matched-epipolar projections. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 549–555, 1993.

[26] J. Heikkilä and O. Silvén. A four-step camera calibration procedure with implicit image correction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 1106–1112, San Juan, Puerto Rico, 1997.

[27] E. B. Hinkle, J. L. C. Sanz, A. K. Jain, and D. Petkovic. P3e: New life for projection-based image processing. *Journal of Parallel and Distributed Computing*, 4:45–78, 1987.

[28] R. Horaud, S. Christy, and F. Dornaika. Object pose: The link between weak perspective, para perspective, and full perspective. Rapport de Recherche 2356, Institut National de Recherche en Informatique et en Automatique, september 1994.

[29] R. Horaud, B. Conio, and O. Lebouilleux. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics, and Image Processing*, 47:33–44, 1989.

[30] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. Object pose: The link between weak perspective, paraperspective, and full perspective. *Int. Journal of Computer Vision*, 22(2):173–189, 1997.

[31] B.K.P. Horn. *Robot Vision.* MIT Press, 1986. 9th printing, 1993.

[32] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, april 1987.

[33] P. V. C. Hough. Method and means for recognizing complex patterns. US Patent 3 069 654, 1962.

[34] D. J. Hunt, L. W. Nolte, and W. H. Ruedger. Performance of the hough transform and its relationship to statistical signal detection theory. *Computer Vision, Graphics, and Image Processing*, 43:221–238, 1988.

[35] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–118, 1988.

[36] K. Kanatani. Computational projective geometry. *CVGIP: Image Understanding*, 54(3):333–348, 1991.

[37] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice.* Artificial Intelligence Laboratory, Department of Computer Science, Gunma University, Japan, 2 edition, 1995.

[38] N. Kiryati and A. M. Bruckstein. Antialiasing the hough transform. In *The 6th Scandinavian Conference on Image Analysis*, pages 621–628, Finalnd, june 1989.

[39] N. Kiryati and A. M. Bruckstein. Antialiasing the hough transform. *CVGIP: Graphical Models and Image Processing*, 53(3):213–222, may 1991.

[40] N. Kiryati and A. M. Bruckstein. On navigating between friends and foes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):602–606, june 1991.

[41] H. Knutsson and C.F. Westin. Normalized and differential convolution-methods for interpolation and filtering of incomplete and uncertain data. *CVPR'93*, pages 515–523, 1993.

[42] K. Kraus. *Photogrammetry - Fundamentals and Standard Processes*, volume 1. Dümmler / Bonn, 4 edition, 1993.

[43] J. Lawn and R. Cipolla. Epipole estimation using affine motion parallax. In *Proc. British Machine Vision Conference*, pages 379–388, Guildford, september 1993.

[44] V. F. Leavers and J. F. Boyce. The radon transform and its application to shape parameterization in machine vision. *Image and Vision Computing*, 5:161–165, 1987.

[45] M. Li and J.-M. Lavest. Some aspects of zoom-lens camera calibration. *IEEE Trans. Pattern Analysis and Machine Intell.*, 18:1105–1110, november 1996.

[46] Q.-T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: Analysis of different methods and experimental results. Rapport de Recherche 1894, Institut National de Recherche en Informatique et en Automatique, april 1993.

[47] H. Maitre. Contribution to the prediction of performances of the hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):669–674, september 1986.

[48] Matrox. Camera interface guide. October 1999.
http://www.matrox.com/imgweb/camera/camera_guide.htm.

[49] T. Melen. *Geometrical Modelling and Calibration of Video Cameras for Underwater Navigation.* Dissertation, Institutt for teknisk kybernetikk, Norges tekniske hgskole, Trondheim, Norway, november 1994. ITK Technical Report 94-103-W.

[50] D. P. Mitchell and A. N. Netravali. Reconstruction filters in computer graphics. *Computer Graphics*, 22(4):221–228, 1988.

[51] R. Mohr and B. Triggs. Projective geometry for image analysis. A Tutorial given at ISPRS, july 1996.

[52] W. Niblack and D. Petkovic. On improving the accuracy of the hough transform. *Machine Vision Applications*, 3:87–106, 1990.

[53] Y. Nomura, M. Sagara, H. Naruse, and A. Ide. Simple calibration algorithm for high-distortion-lens camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1095–1099, november 1992.

[54] D. Oberkampf, D.F. Dementhon, and L.S. Davis. Iterative pose estimation using coplanar feature points. *Computer Vision and Image Understanding*, 63(3):495–511, 1996.

[55] P. L. Palmer, J. Kittler, and M. Petrou. An optimizing line finder using a hough transform algorithm. *CVGIP: Graphical Models and Image Processing*, 67(1):1–23, july 1997.

[56] P. L. Palmer, M. Petrou, and J. Kittler. A hough transform algorithm with a 2d hypothesis testing kernel. *CVGIP: Image Understanding*, 58(2):221–234, september 1993.

[57] X. Pennec. Computing the mean of geometric features - application to the mean rotation. Rapport de Recherche 3371, Institut National de Recherche en Informatique et en Automatique, march 1998.

[58] X. Pennec, C.R.G. Guttmann, and J.-P. Thirion. Feature-based registration of medical images: Estimation and validation of the pose accuracy. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'98)*, pages 1107–1114, Cambridge, MA, USA, october 1998.

[59] M. Petrou and J. Kittler. Optimal edge detectors for ramp edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):483–491, may 1991.

[60] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In *J.O. Eklundh, ed., Computer Vision-ECCV '94 (Third European Conference on Computer Vision)*, pages 97–108, Stockholm, Sweden, 1994.

[61] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery. *Numerical Recipes In C, The Art of Scientific Computing.* Cambridge University Press, 2 edition, 1992.

[62] J. Princen, J. Illingworth, and J. Kittler. Hypothesis testing: A framework for analyzing and optimizing hough transform performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):329–341, april 1994.

[63] L. Quan. Algebraic relations among matching constraints of multiple images. Rapport de Recherche 3345, Institut National de Recherche en Informatique et en Automatique, 1998.

[64] K. Rohr. On the precision in estimating the location of edges and corners. *Journal of Mathematical Imaging and Vision*, 7:7–22, 1997.

[65] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 1, pages 365–375. Academic Press Inc., 2 edition, 1982.

[66] S. D. Shapiro and A. Iannino. Geometric constructions for predicting hough transform performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(3):310–317, july 1979.

[67] R. S. Stephens. Probabilistic approach to the hough transform. *Image and Vision Computing*, 9(1):66–71, february 1991.

[68] P. R. Thrift and S. M. Dunn. Approximating point-set images by line segments using a variation of the hough transform. *Computer Vision, Graphics, and Image Processing*, 21:383–394, 1983.

[69] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, november 1992.

[70] B. Triggs. Matching constraints and the joint image. In *Int. Conference on Computer Vision (ICCV'95)*, 1995.

[71] B. Triggs. Autocalibration from planar scenes. In *European Conference for Computer Vision (ECCV'98)*, Freiburg, Germany, july 1998.

[72] R.Y. Tsai. An efficient and accurate camera calibration technique for 3-D machine vision. In *IEEE Proc. Conference on Computer Vision and Pattern Recognition*, pages 364–374, 1986.

[73] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, april 1991.

[74] R. J. Valkenburg, A. M. McIvor, and P. W. Power. An evaluation of subpixel feature localisation methods for precision measurement. *International Symposium on Photonic Sensors and Control for Commercial Applications: Videometrics III. SPIE*, 16(4):329–341, april 1994.

[75] I. Weiss. Straight line fitting in a noisy image. In *IEEE Comp. Soc. Conf. CVPR'88*, pages 647–652, Ann Arbor, june 1988.

[76] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, october 1992.

[77] R.G. Willson and S.A. Shafer. What is the center of the image? pages 670–671, New York, june 1993.

[78] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. Rapport de Recherche 2676, Institut National de Recherche en Informatique et en Automatique, october 1995.

[79] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.

[80] A. Zisserman, P.A. Beardsley, and I. Reid. Metric calibration of a stereo rig. In *Workshop on Representation of Visual Scenes*, 1995.