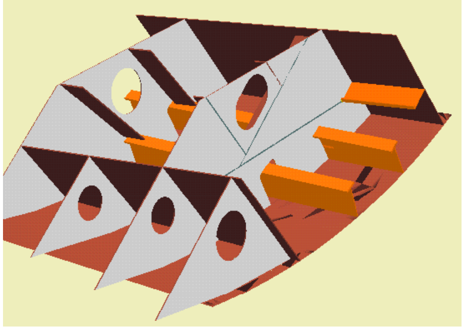
Industrial Vision

Ole Knudsen

PhD Thesis, December 1997

Dissertation from the Industrial Research EF 466 Industrial Vision



Institute of Mathematical Modelling, Technical University of Denmark

The Maersk Institute, Odense University

Odense Steel Shipyard

Danish Academy of Technical Sciences

Industrial Vision

Published by

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

Coverpage notes.

One of the most challenging tasks in today's automation is the building of the bridge from the real production world back to the ideal design (CAD) world. Today the bridge is one-way from CAD to production in the form of computer generated programs for controlling any kind of mechanical device (robot, cutting machine, etc.). The ultimate goal for Odense Steel Shipyard is to have a full two-way bridge between the worlds, where any physical part initially is described in CAD, then produced via CAD-based instructions and finally the result is checked against the CAD model, thereby producing the so-called Product State Model.

Today Odense Steel Shipyard has reached the stage of having the complete ship described in CAD, and the entire production line is furnished with numerical controlled equipment. The next step, to which this dissertation contributes, is to use vision technology to find, identify and measure the ship elements as they are produced.

The first image on the frontpage shows a typical block from a complex part of the ship. The block is a test block, originally produced for one of the ESPRIT projects in which OSS (with the author as local project leader) has participated. Its design is cut out of one of the real CAD models of a large supertanker and the block is therefore perfectly realistic as a subject for experiments. For the same reason, a complete CAD model of the block is available as well and is shown in the second image.

Preface

This dissertation presents the results of the project EF-466 with the title:

Industrial Vision

The project is under the Industrial Ph.D. fellowship programme administered by The Danish Academy of Technical Sciences. The participants are the Institute of Mathematical Modelling (IMM) at the Technical University of Denmark, The Maersk Institute at Odense University and Odense Steel Shipyard Ltd. The project has been carried out in the years 1993 to 1997, including breaks for implementation of subresults.

I thank The Danish Academy of Technical Sciences and Mr Thorkild Vorm for supporting the project and the committee members Dr. Horaud and Dr. Jørgensen for their time and effort spent on reviewing this thesis. Special thanks go to my supervisor Prof. Perram and to Prof. Conradsen who has inspired me in the course of project time and given me moral rearmament when needed. I am also grateful to the Shipyard, especially Torben Andersen and Carl Erik Skjølstrup who initially gave me the chance to do this dissertation. I thank my colleagues for their collaboration, especially Claus Gramkow who joined the company one year ago has been a great help during the writing of this thesis. Also special thanks to Helen MacLean who volunteered in proof-reading this thesis.

Last, but most important, I thank my dear wife Kirsten for her love and patience during the whole project. Without her inspiration and moral support, I would not have been able to complete the work.

Abstract

This dissertation is concerned with the introduction of vision-based applications in the ship building industry. The industrial research project is divided into a natural sequence of developments, from basic theoretical projective image generation via CAD and subpixel analysis to a description of an implementation in real production environments.

The theory for projection of world points into images is concentrated upon the direct linear transformation (DLT), also called the Extended Pinhole model, and the stability of this method. A complete list of formulas for calculating all parameters in the model is presented, and the variability of the parameters is examined and described.

The concept of using CAD together with vision information is based on the fact that all items processed at OSS have an associated complete 3D CAD model that is accessible at all production states. This concept gives numerous possibilities for using vision in applications which otherwise would be very difficult to automate.

The requirement for low tolerances in production is, despite the huge dimensions of the items involved, extreme. This fact makes great demands on the ability to do robust subpixel estimation. A new method based on cross correlation is presented.

Working with vision in harsh environments with few possibilities for controlling light, vibrations, electrical noise etc. requires knowledge about all factors and components in the vision system, which can possible influence the image generated. A description of the experience achieved during the project is provided.

The project is industrial oriented. An essential part of the project has been focused on the possibilities for immediate use of the results. A full implemented application doing vision based positioning is described.

It is concluded that vision-based applications in ship building are not only possible, but also holds great potential in the area of quality control and automation. The effort involved is not necessarily very great, at least not, if the vision-based information can be accompanied by relevant CAD information.

Dansk sammendrag (Danish abstract)

Denne afhandling omhandler introduktion af visionbaserede applikationer til skibsbygningsindustrien. Erhvervsforskerprojektet er inddelt i en naturlig rækkefølge af udviklinger fra en teoretisk gennemgang af projektiv billede-generering over CAD og subpixel analyse til en beskrivelse af en implementering i reelle produktionsomgivelser.

Teorien for projektion af verdenskoordinater ind på billedkoordinater er baseret på en direkte lineær transformation (DLT), også kaldet den udvidede “Pinhole” model. Endvidere afdækkes stabiliteten af denne metode. En komplet liste af formler for udregning af alle parametre i modellen bliver præsenteret, og variationen på de enkelte parametre bliver beskrevet.

Metoden med at anvende CAD sammen med visioninformation er baseret på det faktum, at ethvert emne, der bliver bearbejdet på OSS, har en tilhørende komplet 3D CAD model, som til enhver tid og ethvert sted er tilgængelig. Dette er et koncept, som giver uanede muligheder for at bruge vision i applikationer, hvor det ellers ville være noget nær umuligt at automatisere.

Kravet om lave tolerancer er, på trods af emnernes enorme dimensioner, ekstremt. Et faktum som stiller store krav til evnen til at bestemme billedpunkter med subpixel nøjagtighed. En ny metode baseret på kryds korrelation bliver præsenteret.

Vision i barske omgivelser med få muligheder for at kontrollere lys, vibrationer, elektrisk støj osv. kræver viden om alle de faktorer i visionsystemet, som på nogen måde kan have indflydelse på billedgenereringen. De opnåede erfaringer i løbet af projektet bliver præsenteret.

Projektet er et erhvervsforskerprojekt. En væsentlig del af projektet er derfor fokuseret på mulighederne for umiddelbar ibrugtagen af opnåede delresultater. En komplet implementeret applikation, som udfører visionbaseret positionsbestemmelse, er beskrevet.

Det konkluderes, at visionbaserede applikationer i skibsbygningsindustrien er ikke bare mulig men ydermere indeholder et stort potentiale med hensyn til kvalitetskontrol og automatisering. Indsatsen for at opnå disse gevinster er ikke nødvendigvis særlig høj, især ikke hvis den visionbaserede information kan suppleres med den korresponderende CAD information.

Summary

Introduction describes the background for the project. Reasons are given for OSS to move into a completely new area where no experience was present at the start of the project. The critical situation for shipyards in Europe is illustrated and the need for fast implementation of new technology is described.

Perspective transformations is the theoretical background for the rest of the work. The chapter starts with an introduction to the Direct Linear Transformation (DLT) of world points into image points, the so-called Extended Pinhole model. Secondly, a new method for finding the parameters in the extended pinhole model is described. The derivation of the formulas is based on the strong symbolic calculator from MathCad©, combined with extensive use of analogy to the simpler 2D-1D case. The results are finally applied to a typical transformation matrix. On basis of image points coming from fully calibrated cameras, the task of finding the inverse perspective transformation, i.e. the point(s) in 3D world is examined. Also these results are used on two real situations, one with 2 fixed cameras and another with one movable camera. Finally, as a consequence of the difficulties in doing good estimations of all pinhole parameters, the variability of them is estimated using two different methods. The results are leading to a new method for doing high accuracy calibration.

Camera and CAD is, on the basis of the extended pinhole model derived in previous chapter, going through the work of creating a virtual camera with the same parameters as the real camera. The implementation is using OpenGL$^{\mathrm{TM}}$ language, which provides a lot of facilities for simulating realistic camera configurations and transformations. The only shortcomings are the lack of possibility for describing skewness and scale, parameters that in any case are nearly negligible. Real images are compared with synthetic images constructed on basis of the extended pinhole model and a few calibrating points. The results show good robustness and consistency. A method for finding the transformation from the calibration coordinate system to the robot coordinate system is described. The problem is addressed as the Hand-Eye calibration problem.

Method for subpixel estimation describes a new method for finding specific features in an image. The method is based on initial crosscorrelation of the image with the feature, followed by a model fitting to the cross correlation peaks. The method works if either the feature is rotation symmetric or the rotation is fixed between camera and object. The accuracy of the method is not theoretically supported, but it is shown that generally the accuracy can be expected to be in the area of 0.1 to 0.2 pixels.

Vision components goes through a variety of aspects and considerations encountered during the project. The chapter is describing undesirable effects in the camera like “Gamma correction” and “Edge enhancement”. A brief description of the principles in a lens system is followed by a diagram for finding the right lens for a given combination of object size, distance to camera and CCD size. Framegrabbers and computers are the other end of the vision chain. Some reflections about the importance of having exact synchronisation between the camera and the framegrabber are presented.

First implementation: B4 is the first vision based application ever installed in production environments at Odense Steel Shipyard. The task was, with a moving camera, to determine the position of two templates relative to the robot coordinate system. Despite the simplicity of the core vision task, the effort needed to make it completely integrated into the other production software, defined by the cell controller interface, has been great. This chapter describes most of the various issues that had to be dealt with during implementation. The vision problem was a special case of the general problem described in “Perspective transformations”. The cameras are moving in a plane parallel to the object plane and therefore the mapping could be modelled as a pure 2D-2D case. This assumption leads to the development of simpler but similar equations. The measurements were heavily influenced by the inaccuracy of the mechanical device (deflection, twisting and backlash) and a method for separating the contributions to the total error is presented. In relation to that, it is shown that the accuracy of the vision module is very high. The distance between the measured templates could be directly compared to the theoretical distance and a list of typical results is presented.

Conclusion concludes that shipbuilding is ready for extensive use of vision. The applicability of vision is heavily increased when vision can be accompanied with CAD. The areas for vision are numerous, covering quality control and tool positioning as well as process monitoring and safety surveillance. The possibilities are further enhanced by the exceptionally low prices for high quality equipment and the fast development of the cameras giving better and better resolution. This thesis has made the first step towards a wide introduction of vision applications at Odense Steel Shipyard. At the hour of writing many other vision applications are being specified with the intention of being implemented before the end of 1997. The most important contribution from this dissertation is the new awareness of all developers at Odense Steel Shipyard that vision is a tool, which must always be taken into consideration, and which very often offers the best solution to a specific problem.

Dansk resumé (Danish summary)

Introduction beskriver baggrunden for nærværende projekt. Årsagerne til OSS's ønske om at inddrage et helt nyt område, hvor absolut ingen erfaringer var tilstede ved projektstart, klarlægges. Den kritiske situation, som alle europæiske værfter befinder sig i, bliver beskrevet, og behovet for hurtig implementering af ny teknologi begrundes.

Perspective transformations danner teoretisk baggrund for resten af arbejdet i rapporten. Kapitlet indledes med en gennemgang af den Direkte Lineære Transformation (DLT) af verdenspunkter til billedpunkter, den såkaldte "Udvidede Pinhole" model. Derefter beskrives en ny metode til at finde de parametre, der er involveret i den udvidede Pinhole-model. Udledelsen af formlerne er baseret på den effektive symbolske kalkulator fra MathCad©, kombineret med omfattende brug af analogier til det simplere 2D-1D tilfælde. Resultaterne bliver afslutningsvis anvendt på en typisk 3x4 transformationsmatrix. Med fuldt kalibrerede kameraer forsøges det herefter på baggrund af punkter i et billede at finde den inverse perspektiviske transformation, dvs. det punkt i 3D, som billedpunkterne hidrører fra. Teorien anvendes på 2 forskellige fysiske situationer, en med to fikserede kameraer og en anden med ét bevægeligt kamera. Som en konsekvens af vanskelighederne med at finde gode estimater for alle Pinhole parametrene bliver variationen af parametrene analyseret med to forskellige metoder. Resultaterne herfra fører til udledning af en ny metode til nøjagtig kalibrering efter den udvidede Pinhole model.

Camera and CAD gennemgår, på basis af den udvidede Pinhole model fra forrige afsnit, udviklingen af et virtuelt kamera med samme parametre som det virkelige kamera. Implementeringen bygger på OpenGL^{TM}, som indeholder en række faciliteter til at lette beskrivelsen af realistiske kamera konfigurationer og transformationer. Eneste begrænsning er den manglende mulighed for at beskrive skævheden og skalaen mellem billedets koordinatakser. Dette er imidlertid parametre, som generelt er tæt på at være negligerbare. På basis af Pinhole modellen og nogle få kalibreringspunkter konstrueres syntetiske billeder, som efterfølgende sammenlignes med de virkelige. Resultaterne udviser stor robusthed og overensstemmelse. Afslutningsvis beskrives en metode til at finde transformationen fra kalibreringskoordinatsystemet til robotkoordinatsystemet. Dette problem findes i litteraturen under titlen "Hand-Eye" kalibreringsproblemet.

Method for subpixel estimation beskriver en ny metode til at finde et bestemt objekt i et billede. Metoden baserer sig på krydskorrelation af billedet med objektet, efterfulgt af en modeltilpasning til de fundne krydskorrelationstoppe. Metoden virker på rotationssymmetriske emner og i de tilfælde hvor kamera og emne har en fast rotation i forhold til hinanden. Nøjagtigheden på metoden er ikke matematisk underbygget, men det er eftervist, at nøjagtigheden generelt kan forventes at ligge i størrelsesordenen 0.1 til 0.2 pixels.

Vision components gennemgår en række aspekter og overvejelser, der opstod under projektet. Kapitlet beskriver uønskede effekter i kameraet så som "Gamma korrektion" og "Kant forstærkning". Der gives en kort beskrivelse af principperne i et linsesystem. Et diagram præsenteres til udvælgelse af den korrekte linsestørrelse som funktion af ønsket billedstørrelse og afstand til kamera. I den anden ende af visionkæden befinder framegrabber og computer sig. Her understreges vigtigheden af god synkronisering mellem kamera og framegrabber.

First implementation: B4 er den første vision baserede installation, der nogensinde er blevet indført i et produktionsled på Odense Stålskibsværft. Selve opgaven bestod i, med et bevægeligt kamera, at bestemme positionen af to kendte objekter. Positionen skulle opgives i forhold til robotkoordinatsystemet. På trods af dette forholdsvis simple koncept viste det sig at være en stor opgave at integrere vision delen fuldt ud i det øvrige produktionssoftware, som det er defineret af cell controller-snitfladen. Kapitlet beskriver de fleste af de forskellige overvejelser, som opstod under implementeringen. Visionproblemet udgjorde et specialtilfælde af det generelle problem, som blev behandlet i “Perspective transformations”. Kameraerne bevæger sig i et plan parallelt med objektplanet og mapningen kunne derfor beskrives som et rent 2D-2D tilfælde. Denne model ledte til lignende men enklere ligninger. Nøjagtigheden på målingerne viste sig generelt at være totalt domineret af usikkerheden på det mekaniske system (nedbøjning, vridning og slør). En metode til at adskille usikkerhedsbidragene bliver præsenteret, og det påvises i den forbindelse, at visionbidraget er forsvindende i forhold til den totale usikkerhed. Afstanden mellem de 2 målte objekter kunne direkte sammenlignes med den teoretiske afstand og en række typiske resultater bliver præsenteret.

Conclusion konkluderer at skibsbygning er moden til omfattende brug af vision. Visionteknologiens muligheder øges drastisk, hvis vision kan suppleres med CAD information. Anvendelsesmulighederne er talrige dækkende lige fra kvalitetskontrol og værktøjspositionering til procesmonitorering og sikkerhedsovervågning. Mulighederne bliver yderligere forstærket af de meget lave priser på højkvalitetsudstyr og af den hurtige udvikling af kameraerne. Nærværende projekt har taget første skridt mod en bred introduktion af vision på Odense Stålskibsværft, og i skrivende stund er adskillige vision applikationer ved at blive specificeret med henblik på ibrugtagen inden udgangen af 1997. Projektets vigtigste bidrag er dog utvivlsomt den bevidsthed, der nu er hos alle udviklere om, at en visionbaseret løsning altid skal overvejes og meget ofte er den bedste til et specifikt problem.

Table of contents

1. Introduction

The shipbuilding industry has during the last decades experienced increasing competition, and the struggle for survival has become more and more intense. Europe especially has seen many venerable old Yards go down because of the competition from mainly the Far East shipyards. The shipbuilding industry in Europe has to improve its competitiveness in order to survive, otherwise the Far East will capture the small European market share that is left. Fig 1-1 shows how the European shipbuilding industry has lost market shares from 1983 to 1991. Things have turned even worse since then.

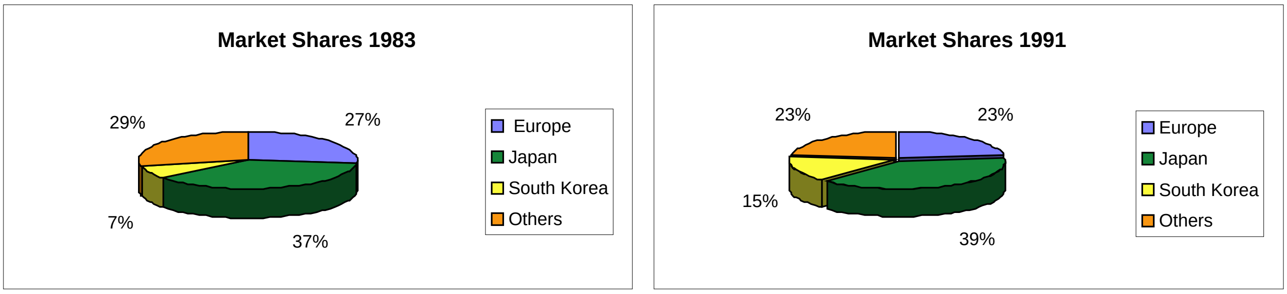


Fig 1-1 The Far East captures the European market share [source KPMG 1992]

The main reason why the European yards have lost market shares is that the total production costs are too high. First of all European yards must increase their productivity to survive, since the average European productivity level is only 63% of the average Japanese level. Odense Steel Shipyard is one of the few European yards that have done well in the competition battle. This is mainly because OSS has managed to keep control over the production expenses through improvements in productivity. The productivity can only be increased by introducing more technology into the shipbuilding industry; OSS has perceived that, and has completely adopted the philosophy of surviving through technology investments.

One of the new areas of technology, in which Odense Steel Shipyard has decided to invest, is vision-based applications. Vision has many advantages which are extremely well suited to heavy industry[1]:

⇒ **It is passive**. There is no time for marking the objects before measuring, and the availability of the CAD model strongly advocates for passive vision based methods.

⇒ **High angular resolution**. Shipblocks are big. A high angular resolution is crucial when measuring such enormous objects.

⇒ **Long working range**. The distance to the object will often and for many reasons be in the range of 2 to 20 metres.

⇒ **Cost competitive**. Vision technology is cheap compared to any other kind of sensors that the cameras have to compete with.

⇒ **High reliability**. Cameras have a high “mean time between failure” and they can stand up to big shifts in temperature.

⇒ **Low operating costs**. The only expenditure is the 12V-power supply.

[1] The list of advantages is taken from an ESPRIT (vision) proposal (VIGOR) in which OSS is participating. The creator of the list is Radu Horaud, INRIA.

The advantages mentioned above provide the shipbuilding industry with numerous possibilities:

- **Quality Control**. The most time- and money-saving single objective is reduction of the tolerances in the production lines. Vision can provide dimensional control of huge as well as small parts of the ship. The control can be 2D or 3D depending on the type of production. Very often, cutting is a matter of 2D quality control while welding is 3D.
- **Positioning**. The time it takes to put equipment into the right position relative to the object is a crucial competition factor. Vision can provide an elegant, fast and passive solution.
- **Gap control**. Today, the welding process is controlled by the so-called arcsensor, which measures various values during welding, and changes the welding parameters according to these values. A camera mounted in front of the welding gun would provide the possibility for changing the welding parameters at exactly the right time.
- **Process monitoring**. A camera behind the welding gun can check the quality of the seam.
- **Safety surveillance**. If one robot in a team breaks down then all other robots with overlapping working areas have to be stopped during repair. This is clearly an expensive problem that can be solved if each robot were surveyed by cameras, thereby reducing the actual safety zone per robot to a minimum.

Another important issue, to which the Yard gives high priority and which is going to be developed in a newly accepted Esprit project, is the possibility of measuring the actual state of the product, the so-called Product State Model. The idea is to transfer immediately the knowledge obtained at one production site, via the cell controller interface, to next production site and use the information there to compensate for any deviations from the ideal CAD model. This concept is far better than letting the machines work directly after the CAD model. At the same time the measurements from each production site are sent back to the offices and stored for later use, if any questions should come up during the lifetime of the ship.

The aim of this project is to make the initial steps in the direction of achieving all the goals mentioned above. The first thing to do is to ***make vision visible*** in practice and in people's minds. By far the most effective way of achieving that is by making visible results. On the other hand, in the beginning of such a project it is very important not to make failures that could jeopardise the sympathy.

More specifically, the intention of the project is to obtain know-how about taking good images in rough environments. Special focus is put on ***camera calibration*** and ***experience*** in light settings etc. Working with vision in harsh environments with few possibilities for controlling light, vibrations, electrical noise etc., requires knowledge about all factors and components in the vision system which possibly can influence the image generated. Another important issue is the concept of using ***CAD together with vision*** information. This objective is based on the fact that all items processed at OSS have an associated complete 3D CAD model, which is accessible at all production states: a concept which gives numerous possibilities for using vision in applications, which otherwise would be very difficult to automate. The requirement for low tolerances in production is, despite the huge dimensions of the items involved, extreme. This fact makes great demands on the ability to do robust ***subpixel estimation***.

2. Perspective transformations

The aim of this thesis is in general terms to use the information lying on a flat CCD-chip to depict as much as possible about the world surrounding the camera. For this purpose, 2 transformation formulas have to be derived: *The Camera Model* or *Direct Perspective Transformation* mapping three-dimensional world points to two-dimensional image points and *The Inverse Perspective Transformation*, which is used to identify the three-dimensional world points corresponding to a particular two-dimensional image point. In the following and throughout the thesis, these transformations are assumed to be linear. In practice it has been proved that this assumption is sufficiently accurate for many applications at Odense Steel Shipyard, at least for lenses with a focal length above 16 mm.

The simplest and most common used linear Camera Model is the *Pinhole Model.* All light beams go through a common point called *Focal point* or *Projection Centre* and form an image on the *image plane* a distance f behind the focal point.

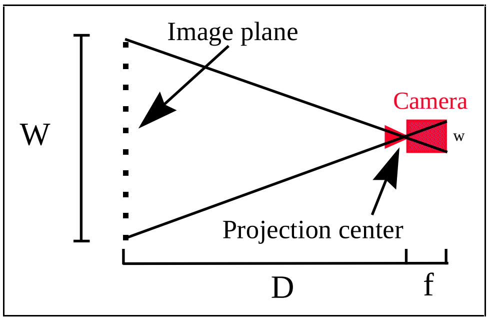


Fig 2-1. The Pinhole model

A more detailed description still based on the pinhole model but introducing the lens:

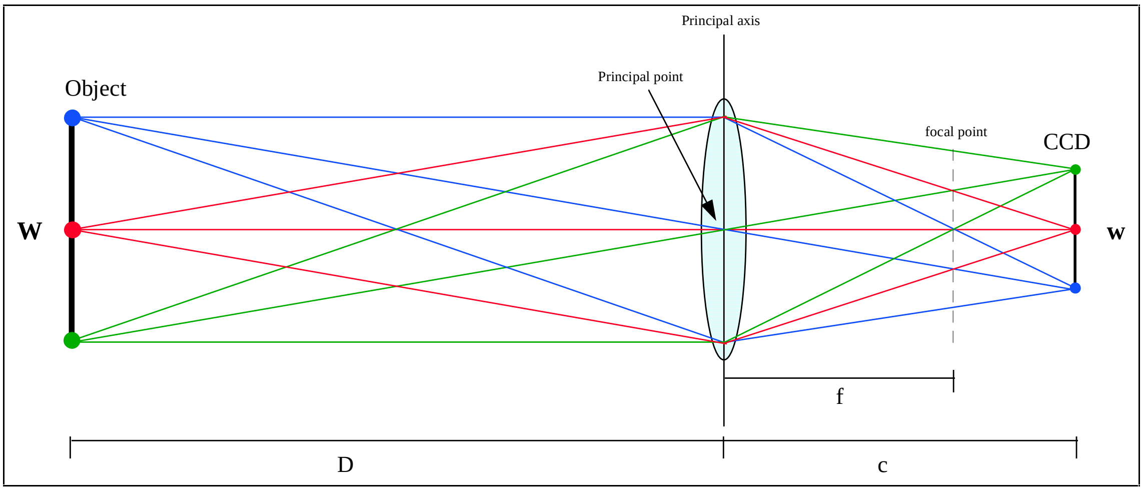


Fig 2-2. An optical lens.

In Fig 2-1 and Fig 2-2, the notation from classical photogrammetry (Ghosh[i], Lauridsen[ii],Jacobi[iii]) is used:

D: distance from the principal axis of the lens to the object

c: distance from principal axis to CCD chip (image plane) ≡ camera constant

f: distance from principal axis to the focal point of the lens ≡focal length (mm lens)

W: width of object

w: width of imaged object on CCD

From Fig 2-1 and Fig 2-2, we have by use of similar triangles:

EQ. 2.1
$$\frac{W}{D}=\frac{w}{c}\Rightarrow W=D\frac{w}{c}$$

The camera constant is generally unknown but from the *lens formula*:

EQ. 2.2
$$\frac{1}{D}+\frac{1}{c}=\frac{1}{f}$$

EQ. 2.1 and EQ. 2.2 gives

EQ. 2.3
$$W=Dw(\frac{1}{f}-\frac{1}{D})=w(\frac{D}{f}-1)\Rightarrow$$

EQ. 2.4
$$f=D\frac{w}{W+w}\approx D\frac{w}{W}=c$$

EQ. 2.4 shows that the camera constant and the focal length in practice are nearly identical, and they can be calculated if the size of the object *w* is known in the same units as W (typically the size of the CCD chip is specified).

The pinhole model is extremely valuable when trying to understand the nature of image mapping, and also all non-linear contributions are normally added to the pinhole model as perturbations, leaving the pinhole model as the basic transformation model. Also, when generating synthetic images from CAD (see next chapter) a complete and in-depth understanding of the pinhole model is crucial.

The linear model DLT (*Direct Linear Transformation*) was first presented in 1963 in the classical paper of Roberts[iv], which covers most of today's important subjects in computer vision (perspective transformations, line extraction, CAD, matching and display. In 1973, Duda[v] gives a different interpretation where the 3rd image coordinate (y_p) is used as a free variable. The DLT takes the form of a 4x3 matrix and is geometrically described by an Extended Pinhole Model. The extension introduces 4 additional internal parameters: *a shift of the image origin* (U_0,V_0) and a linear distortion in the image plane described by a *scale* difference and a lack of orthogonality (*skewness*) between the image axes.

The biggest non-linear error arises from distortions in the lens. Such distortions are only partly linear and generally not fully absorbed by the back projection matrix. Most lens distortion can be described as radial symmetric distortion, which means that the correction dr for a given distance r from Principal point is constant and can be approximated by a polynomial in uneven powers of r up to maximum 7:

EQ. 2.5
$$dr=a_1r+a_3r^3+a_5r^5+a_7r^7$$

The linear term is absorbed by the transformation matrix while the rest of the terms in many practical cases are negligible due to other and bigger error sources.

Many attempts have been made to create a more accurate model than DLT without introducing many extra parameters to estimate. Tsai[vi] proposed in 1987 a model that combined parts of the linear model with some correction for radial distortion. This model did not consider skew, and it assumed lens distortion to be radial and centred in Focal Point. Furthermore, the model often runs into singularity problems when trying to calibrate the parameters.

On basis of the theory taken from the book of Dueholm[vii], this section analyses the full 11 parameters version of the pinhole model completely. With a new approach based on the MathCad© symbolic calculator, a full description is given of what each of the elements in the 4x3 transformation matrix contains. As a consequence of that, all formulas for calculating the pinhole parameters in terms of the matrix elements are also presented. Several others have done work in this area but many of them have not been able to extract all 11 parameters. Ganapathy[viii] worked without the skewness parameter integrated into the equations, and with a pure algebraic approach primarily based on the properties of a rotation matrix, he was able to extract the remaining 10 parameters. Strat[ix] did the same in 1984, but in a more elegant way, based on geometrical considerations and so did Faugeras[x] in a less legible paper in 1986. A complete decomposition was first given in 1987 in an article by Shih[xi], and independently in a follow-up article by Faugeras[xii] . A different and more stable method, based on a better normalisation criteria, was presented by Melen[xiii] in 1994. The method presented in this dissertation is a pure algebraic approach, which only uses trigonometric relations, and the complete set of 11 formulas has, to my knowledge, not been presented in this form before.

One of the crucial points is to find the image centre and that is a difficult task. Puget & Skordas[xiv] give a geometrical description of the relations between the well-defined external parameters and the ill-behaving position of the piercing point (where focal line intersects image plane). The problem is analysed in this thesis in the subchapter “Variability of parameters”. Li & Lavest have done similar work in the field[xv] although their work primarily was considering zoom lenses. Wilson & Schafer[xvi] have listed no less than 16 different definitions of the image centre and 16 descriptions of how to measure them. Wang & Tsai[xvii] have in an elegant way, used vanishing-line information from a rectangular parallelepiped to determine the position of the image centre (as well as focal length). Seetharaman[xviii] presents similar ideas but in a less readable form. One problem that emerge from their work, is that the image centre is not constant for different distances of calibration and measurement. Although the loss in accuracy on that account is limited in our applications, where the measuring distance generally is well known, the point has to be considered.

2.1 The 2D-1D transformation

Usually, geometry is easier to understand and much easier to draw in 2D. I will therefore start with the simpler 2D-1D example illustrated in Fig 1.

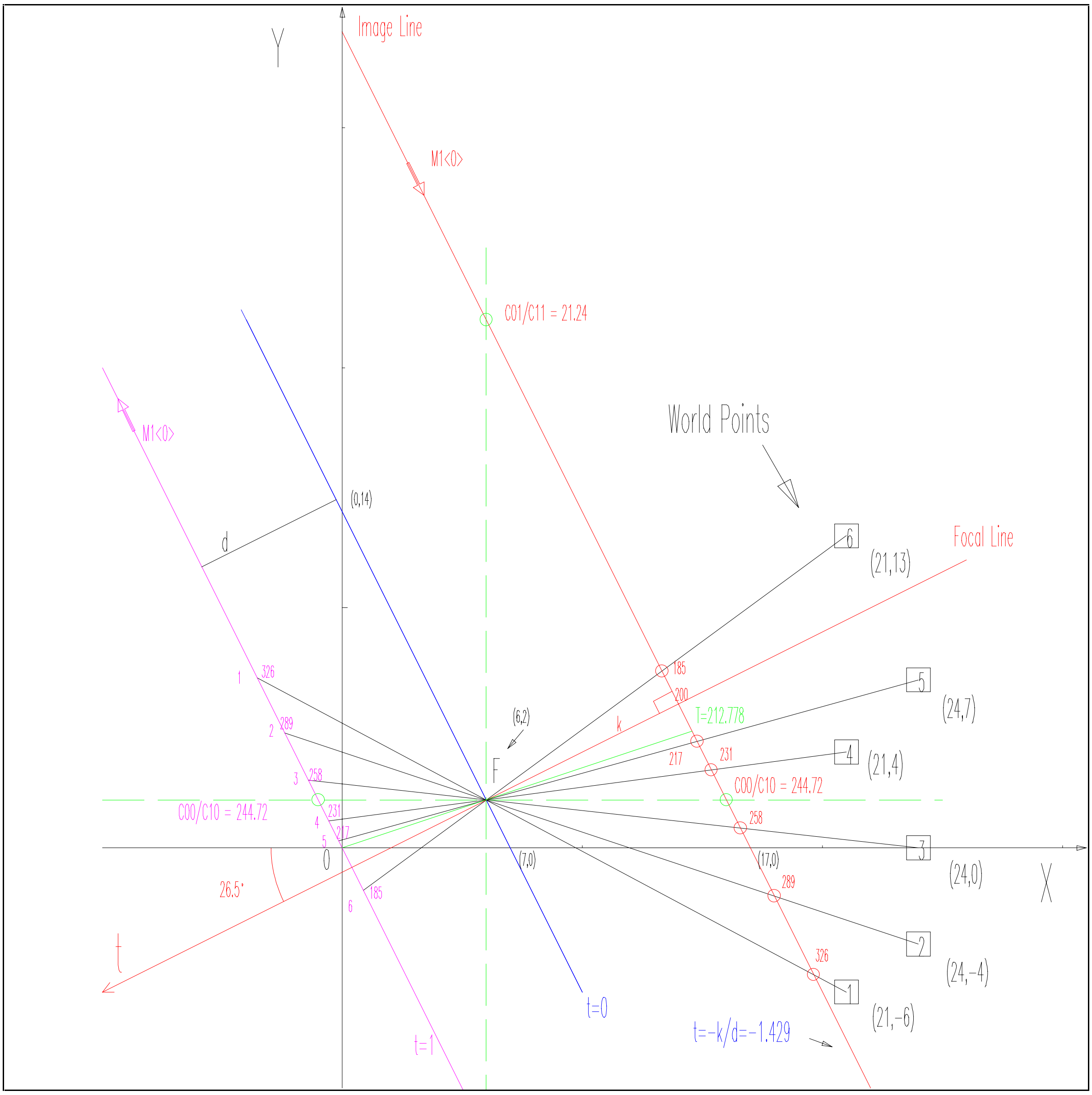


Fig 2-3 2D-1D perspective transformation

Our calibration data are 6 points in 2D space imaged through the focal centre F onto the virtual 1D-image line which for convenience has been put in front of the lens instead of the physically correct position behind the lens.

The focal centre is placed at point (6,2) in world co-ordinates with the physical image line placed $k = \sqrt{80}$ [world units] away. The focal line, which is perpendicular to the image line and intersecting at U_0=200 [camera units], makes an angle of $\tan^{-1}(0.5)$=26.5° with the x-axis. The scale between the world units and the camera units is in this case 10. All these values are from the beginning assumed unknown as well as all other camera parameters. The only data known are the 6 world points (in world coordinates) and their corresponding images (in camera coordinates) on the image line. So we have the data:

X	Y	U
21	-6	326.03292237
24	-4	289.44271910
24	0	257.87470059
21	4	230.74593469
24	7	217.45223787
21	13	184.72929186

Table 2.1 2D-1D calibration data

2.1.1 Construction of 2D-1D transformation matrix

The basic theory is taken from Vernon[xix], Ballard & Brown[xx] and the master thesis of John Immerkaer[xxi]. The reduction to the 2D-1D case is more or less straightforward and is described below. First we want to find the homogeneous transformation matrix **C** which brings world co-ordinates into image coordinates:

EQ. 2.1 $$\textit{world point:} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad , \quad \textit{image point:} \begin{pmatrix} u \\ t \end{pmatrix}$$

The transformation matrix **C**, mapping two-dimensional world points to corresponding one-dimensional image points satisfies the equation:

EQ. 2.2 $$\mathbf{C}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ t \end{pmatrix} \quad , \quad \mathbf{C} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & 1 \end{bmatrix}$$ *Perspective projection*

We note that this is a system of equations containing 5 unknowns. Expanding EQ. 2.2 gives:

EQ. 2.3 $$\left.\begin{array}{l} C_{00}x + C_{01}y + C_{02} = u = Ut \\ C_{10}x + C_{11}y + 1 = t \end{array}\right\} \Rightarrow$$

EQ. 2.4 $$C_{00}x + C_{01}y + C_{02} - UC_{10}x - UC_{11}y - U = 0$$

We see that each calibration point gives only one equation. The minimum number of calibration points is therefore as high as five for the 2D-1D case of image mapping. We shall later see that the minimum number required for the 3D-2D case is only 5½ points. Decreasing the 3D problem by one dimension only reduces the number of calibration points needed with half a point (half a point could for instance be a line where only the x coordinate is known).

After having determined the transformation matrix **C**, it is possible to find the corresponding set of points in the real world on basis of an imaged position. EQ. 2.4 can be rewritten to

EQ. 2.5 $$y = ax + b$$

where

EQ. 2.6

$$a = \frac{UC_{10} - C_{00}}{C_{01} - UC_{11}}$$
$$b = \frac{U - C_{02}}{C_{01} - UC_{11}}$$

As expected, the corresponding set of world points is described by a line.

Let

EQ. 2.7

$$\mathbf{A} = \begin{bmatrix} x^1 & y^1 & 1 & -U^1x^1 & -U^1y^1 \\ x^2 & y^2 & 1 & -U^2x^2 & -U^2y^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x^N & x^N & 1 & -U^Nx^N & -U^Ny^N \end{bmatrix}, \mathbf{C} = \begin{pmatrix} C_{00} \\ C_{01} \\ C_{02} \\ C_{10} \\ C_{11} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} U^1 \\ U^2 \\ \vdots \\ U^N \end{pmatrix}$$

giving

EQ. 2.8

$$\mathbf{A\ C} = \mathbf{B}$$

where each superscript in EQ. 2.7 denotes an observation (calibration) point, and **C** is the unknown vector we want to find. Using the Least Squares method (Vernon p. 71), the problem can be solved by:

EQ. 2.9

$$\mathbf{C} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}$$

which can be solved in a traditional way. Since the number of unknowns and equations always are reasonable low, a method like the robust Householder[xxii] algorithm can be applied with good results. For the data in Table 2.1 the following transformation matrix is produced:

EQ. 2.10

$$\mathbf{C} = \begin{bmatrix} -34.96 & -1.505 & 212.778 \\ -0.143 & -0.071 & 1 \end{bmatrix}$$

This fundamental matrix contains *all* (linear) information about the “camera”. In the following I will step by step try to put this information in a more intuitively understandable form and give it a geometrical interpretation with reference to Fig 2-3.

Let us start with checking the validity of **C**:

$\begin{pmatrix} x \\ y \end{pmatrix}$	$\mathbf{C}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$	Divided by 2nd coordinate giving (t=1)	Measured value
$\begin{pmatrix} 21 \\ -6 \end{pmatrix}$	$\begin{pmatrix} -512.337 \\ -1.571 \end{pmatrix}$	$\begin{pmatrix} 326.033 \\ 1 \end{pmatrix}$	326.033
$\begin{pmatrix} 24 \\ -4 \end{pmatrix}$	$\begin{pmatrix} -620.234 \\ -2.143 \end{pmatrix}$	$\begin{pmatrix} 289.443 \\ 1 \end{pmatrix}$	289.443
$\begin{pmatrix} 24 \\ 0 \end{pmatrix}$	$\begin{pmatrix} -626.267 \\ -2.429 \end{pmatrix}$	$\begin{pmatrix} 257.875 \\ 1 \end{pmatrix}$	257.875
$\begin{pmatrix} 21 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -527.419 \\ -2.286 \end{pmatrix}$	$\begin{pmatrix} 230.746 \\ 1 \end{pmatrix}$	230.746
$\begin{pmatrix} 24 \\ 7 \end{pmatrix}$	$\begin{pmatrix} -636.824 \\ -2.929 \end{pmatrix}$	$\begin{pmatrix} 217.452 \\ 1 \end{pmatrix}$	217.452
$\begin{pmatrix} 21 \\ 13 \end{pmatrix}$	$\begin{pmatrix} -540.993 \\ -2.929 \end{pmatrix}$	$\begin{pmatrix} 184.729 \\ 1 \end{pmatrix}$	184.729

Table 2.2 Checking data

So the camera model fits perfectly for these synthetic data and we now know that since a transformation by a 2x3 matrix is equivalent to applying a camera model in form of a 2D pinhole model, it is possible from the transformation matrix to find the parameters involved in such a 2D pinhole model. In order to find the inverse perspective transformation let us first notice that we from Fig 2-3 and EQ. 2.5 & EQ. 2.6 can derive:

- The horizontal line through **F** is intersecting the image line at C_{00}/C_{10} (camera units).
- The vertical line through **F** is intersecting the image line at C_{01}/C_{11} (camera units).
- The line through origo (world system) and **F** is intersecting the image line at C_{02}.

2.1.2 The 2D Pinhole Model

For the 2D pinhole model the transformation can be described by successive use of appropriate transformation matrices[xxiii]:

EQ. 2.11
$$\mathbf{D} = \begin{bmatrix} 1 & 0 & -dx \\ 0 & 1 & -dy \\ 0 & 0 & 1 \end{bmatrix}$$
Translation

EQ. 2.12
$$\mathbf{R} = \begin{bmatrix} cos(\alpha) & sin(\alpha) & 0 \\ -sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
Rotation

EQ. 2.13
$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -f & 0 \end{bmatrix}$$
Perspective transformation (f=focal^{-1})

EQ. 2.14

$$\mathbf{U} = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix}$$

Shift of U_0

First we note that these matrices all together contain 5 unknowns like the transformation matrix **C**. The task now is to find analytical formulas for each of the parameters in the pinhole model as a function of the variables in **C**. Multiplying the 4 matrices and using the short notation sα for sin(α) and cα for cos(α):

EQ. 2.15

$$\mathrm{K} = \mathbf{UPRD}$$

EQ. 2.16

$$\mathbf{K} = \begin{bmatrix} c\alpha + \delta \cdot f \cdot s\alpha & s\alpha - \delta \cdot f \cdot c\alpha & -dx \cdot \delta \cdot f \cdot s\alpha - dy \cdot s\alpha + dy \cdot \delta \cdot f \cdot c\alpha \\ f \cdot s\alpha & -f \cdot c\alpha & -dx \cdot f \cdot s\alpha + dy \cdot f \cdot c\alpha \end{bmatrix}$$

Compared to the transformation matrix in EQ. 2.2 we first note that **K** is unnormalised. The normalisation of **C** is actually the operation that really mixes up things. The normalisation of **K** is achieved by division with K_{12}; a nicer normalisation would be obtained by division with K_{10} (which of course can be done at any time).

2.1.2.1 Relations between C and K

From EQ. 2.16 we can derive following relations:

EQ. 2.17

$$K_{10}^2 + K_{11}^2 = f^2 \Rightarrow$$

EQ. 2.18

$$\frac{K_{10}^2}{K_{12}^2} + \frac{K_{11}^2}{K_{12}^2} = \frac{f^2}{K_{12}^2} \Rightarrow$$

EQ. 2.19

$$C_{10}^2 + C_{11}^2 = \left(\frac{f}{K_{12}}\right)^2 \Rightarrow$$

EQ. 2.20

$$\frac{f}{K_{12}} = \sqrt{C_{10}^2 + C_{11}^2}$$

We now have the general relationship:

EQ. 2.21

$$C_{ij} = \frac{K_{ij}}{K_{12}} = \frac{\sqrt{C_{10}^2 + C_{11}^2}}{f} K_{ij}$$

EQ. 2.22

$$\frac{K_{ij}}{f} = \frac{C_{ij}}{\sqrt{C_{10}^2 + C_{11}^2}}$$

EQ. 2.23

$$K_{ij} = \frac{f}{\sqrt{C_{10}^2 + C_{11}^2}} C_{ij}$$

Where C_{ij} are normalised coefficients and f=1/focal length.

2.1.2.2 Focal centre

Let us rearrange the matrix in following way:

EQ. 2.24
$$\mathbf{C}\mathbf{x}_{hom} = \mathbf{u} \quad \textit{is equivalent to} \quad \mathbf{M}\mathbf{x} + \mathbf{T} = \mathbf{u}$$

or

EQ. 2.25
$$\begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} C_{02} \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ t \end{pmatrix}$$

where

EQ. 2.26
$$\mathbf{M} = \begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix}$$

and

EQ. 2.27
$$\mathbf{T} = \begin{pmatrix} C_{02} \\ 1 \end{pmatrix}$$
Translation

Now **M** is quadratic and we obtain:

EQ. 2.28
$$\mathbf{x} = \mathbf{M}^{-1}\mathbf{u} - \mathbf{M}^{-1}\mathbf{T}$$
Inverse perspective projection

or

EQ. 2.29
$$\begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} U \\ 1 \end{pmatrix} t - \mathbf{M}^{-1} \begin{pmatrix} C_{02} \\ t \end{pmatrix}$$

EQ. 2.29 is just a parametric line description and we get:

EQ. 2.30
$$\mathbf{V} = \begin{pmatrix} a \\ b \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} U \\ 1 \end{pmatrix}$$
Viewing direction

EQ. 2.31
$$\mathbf{F} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = -\mathbf{M}^{-1} \begin{pmatrix} C_{02} \\ 1 \end{pmatrix}$$
Focal centre F

If we try to perform the same operation on **K** as on **C** when we derived the world position of **F**, we first note that the normalisation procedure is factored out. The question now is whether the 2D pinhole model leads to the same result. In the following, a mark (´) on a matrix or a vector denotes that we are dealing with the unnormalised version.

EQ. 2.32
$$\mathbf{M}' = \begin{bmatrix} c\alpha + \delta \cdot f \cdot s\alpha & s\alpha - \delta \cdot f \cdot c\alpha \\ f \cdot s\alpha & -f \cdot c\alpha \end{bmatrix} \quad ; \quad \mathbf{M}' = -f$$

EQ. 2.33
$$\mathbf{M}'^{-1} = \begin{bmatrix} c\alpha & \frac{s\alpha}{f} - \delta \cdot c\alpha \\ s\alpha & \frac{c\alpha}{f} - \delta \cdot s\alpha \end{bmatrix} \quad ; \quad \left|\mathbf{M}'^{-1}\right| = \frac{-1}{f}$$

EQ. 2.34
$$\mathbf{T'}=\begin{pmatrix} -dx\cdot\delta\cdot f\cdot s\alpha - dy\cdot s\alpha + dy\cdot\delta\cdot f\cdot c\alpha \\ -dx\cdot f\cdot s\alpha + dy\cdot f\cdot c\alpha \end{pmatrix}$$

EQ. 2.35
$$\mathbf{F}:\quad -\mathbf{M'}^{-1}\,\mathbf{T'} = -\mathbf{M}^{-1}\mathbf{T} = \begin{pmatrix} dx \\ dy \end{pmatrix}$$

So fortunately (x_0,y_0) in EQ. 2.31 are the same as the zero offset described by the matrix in EQ. 2.11. Another way of writing the expression for (dx,dy) is:

EQ. 2.36
$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \frac{1}{\begin{vmatrix} K_{00} & K_{01} \\ K_{10} & K_{11} \end{vmatrix}} \begin{bmatrix} \begin{vmatrix} K_{00} & K_{02} \\ K_{11} & K_{12} \end{vmatrix} \\ \begin{vmatrix} K_{02} & K_{00} \\ K_{12} & K_{10} \end{vmatrix} \end{bmatrix} = \frac{1}{\begin{vmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{vmatrix}} \begin{bmatrix} \begin{vmatrix} C_{01} & C_{02} \\ C_{11} & C_{12} \end{vmatrix} \\ \begin{vmatrix} C_{02} & C_{00} \\ C_{12} & C_{10} \end{vmatrix} \end{bmatrix}$$

This means that finding the position of the focal centre in practice is the same as treating the calibration matrix **C** as a set of equations and solving it by applying Cramers rule to it.

2.1.2.3 Interior offset $\mathbf{U_0}$

The inverse of matrix **M**:

EQ. 2.37
$$\mathrm{M}^{-1} = \frac{1}{|\mathrm{M}|}\begin{bmatrix} -C_{11} & C_{01} \\ C_{10} & -C_{00} \end{bmatrix}$$

First column in $\mathbf{M}^{-1}$ ($\mathbf{M}^{-1<0>}$) is expanding the image line and the second column $\mathbf{M}^{-1<1>}$ + **F** is a point on that line (see Fig 2-3). U_0 is defined as the point where the focal line through the focal point is intersecting the image line with an angle of 90 degrees. Knowing the normal to the projection line U (in world coordinates) and the transformation **[MT]** which maps the world points to the image line, U_0 can be calculated as follows:

- $\begin{pmatrix} C_{10} \\ C_{11} \end{pmatrix}$ is perpendicular to the projection line U (another and more elegant way of seeing this, is presented by Strat[ix]. He sees that world points, lying in a line parallel to the image line and passing through focal centre, must have the homogeneous image coordinates (u,0). Putting that into EQ. 2.25 gives the statement above).
- **F** + $\begin{pmatrix} C_{10} \\ C_{11} \end{pmatrix}$ is a point lying on the focal line

EQ. 2.38
$$U_0 = \mathbf{M}(\mathbf{F} + \begin{pmatrix} C_{10} \\ C_{11} \end{pmatrix}) + \mathbf{T} \Rightarrow$$

EQ. 2.39
$$U_0 = \mathbf{M}\begin{pmatrix} C_{10} \\ C_{11} \end{pmatrix}$$

or

EQ. 2.40
$$U_0 = \frac{C_{00}C_{10} + C_{01}C_{11}}{C_{10}^2 + C_{11}^2}$$

or

EQ. 2.41

$$U_0 = \frac{\mathbf{M}^{T^{<0>}}\mathbf{M}^{T^{<1>}}}{\left|\mathbf{M}^{T^{<1>}}\right|^2}$$

Let us try to perform similar operations on the unnormalised matrix **K´**, again noting that the overall normalisation is factored out:

EQ. 2.42

$$\mathbf{M}'^{T} = \begin{bmatrix} c\alpha + \delta \cdot f \cdot s\alpha & f \cdot s\alpha \\ s\alpha - \delta \cdot f \cdot c\alpha & -f \cdot c\alpha \end{bmatrix}$$

EQ. 2.43

$$\mathbf{M}'^{T^{<0>}} = \begin{bmatrix} c\alpha + \delta \cdot f \cdot s\alpha \\ s\alpha - \delta \cdot f \cdot c\alpha \end{bmatrix}$$

EQ. 2.44

$$\mathbf{M}'^{T^{<1>}} = \begin{bmatrix} f \cdot s\alpha \\ -f \cdot c\alpha \end{bmatrix}$$

EQ. 2.45

$$U_0 = \frac{\mathbf{M}'^{T^{<0>}}\mathbf{M}'^{T^{<1>}}}{\left|\mathbf{M}'^{T^{<1>}}\right|^2} = \frac{\mathbf{M}^{T^{<0>}}\mathbf{M}^{T^{<1>}}}{\left|\mathbf{M}^{T^{<1>}}\right|^2} = \frac{C_{00}C_{10} + C_{01}C_{11}}{C_{10}^2 + C_{11}^2} = \delta$$

Again we see the resemblance between EQ. 2.14, EQ. 2.41 and EQ. 2.45

2.1.2.4 Exterior angle

From the relation

EQ. 2.46

$$\frac{K_{10}}{K_{11}} = \frac{C_{10}}{C_{11}} = -tan(\alpha)$$

we get

EQ. 2.47

$$\alpha = -a\,tan(\frac{C_{10}}{C_{11}})$$

here we note that since tangent is periodic with π, the equation has 2 solutions depending on which direction is positive on the image line.

2.1.2.5 Focal length

From the relation

EQ. 2.48

$$K_{00}^2 + K_{01}^2 = \delta^2 f^2 + 1$$

and EQ. 2.23 we get

EQ. 2.49

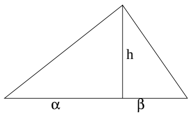
$$\left(\frac{f}{\sqrt{C_{10}^2 + C_{11}^2}}\right)^2 \left(C_{00}^2 + C_{01}^2\right) = \delta^2 f^2 + 1 \Rightarrow$$

EQ. 2.50
$$\frac{1}{f} = \sqrt{\frac{C_{00}^2 + C_{01}^2}{C_{10}^2 + C_{11}^2} - \delta^2}$$

where C_{ij} are normalised coefficients and f=1/focal length. In the 2D-1D case, EQ. 2.50 reduces to

EQ. 2.51
$$\frac{1}{f} = \frac{\begin{vmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{vmatrix}}{C_{10}^2 + C_{11}^2}$$

That result could also be achieved by looking at Fig 2-3 and using the well-known formula for a rectangular triangle: $\alpha \cdot \beta = h^2$



where h: focal length

α: C_{00}/C_{10} - U_0

β: C_{01}/C_{11} - U_0

we get

EQ. 2.52
$$focal = \sqrt{\left(\frac{C_{00}}{C_{10}} - U_0\right)\left(\frac{C_{01}}{C_{11}} - U_0\right)}$$

which can be reduced to:

EQ. 2.53
$$focal = \frac{|\mathbf{M}|}{C_{10}^2 + C_{11}^2}$$

Note that the exterior values (x_0,y_0,α) are measured in world coordinates, while the interior values (U_0 and focal length) are measured in camera coordinates. In order to get the relation between these two coordinate systems, the overall scaling factor has to be known (typically by knowing the size of the pixels).

2.1.3 Applying results

For the example described in Fig 2-3 we get:

EQ. 2.54
$$\mathbf{M} \quad = \begin{bmatrix} -34.96 & -1.508 \\ -0.143 & -0.071 \end{bmatrix}$$

EQ. 2.55
$$\mathbf{T} \quad = \begin{pmatrix} 212.778 \\ 1 \end{pmatrix}$$

EQ. 2.56
$$\mathbf{M}^{-1} = \begin{bmatrix} -0.031 & 0.661 \\ 0.063 & -15.322 \end{bmatrix}$$

We are now ready to extract information by use of the appropriate formulas:

(EQ. 2.35)
$$\mathbf{F} = -\mathbf{M}^{-1}\mathbf{T} = -\begin{bmatrix} -0.031 & 0.661 \\ 0.063 & -15.322 \end{bmatrix}\begin{pmatrix} 212.778 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \end{pmatrix}$$

(EQ. 2.20)
$$\sqrt{C_{10}^2+C_{11}^2}=\sqrt{(-0.143)^2+(-0.071)^2}=0.160=(\sqrt{5}/14)$$

(EQ. 2.45)
$$U_0=\frac{34.96\cdot 0.143+1.508\cdot 0.071}{0.143^2+0.071^2}=200$$

(EQ. 2.47)
$$\alpha=-a\,tan(\frac{-0.143}{-0.071})=-a\,tan(2)=-63.4^{\circ}$$

(EQ. 2.51)
$$\frac{1}{f}=\frac{\begin{vmatrix}-34.96 & -1.508\\ -0.143 & -0.071\end{vmatrix}}{0.143^2+0.071^2}=89.443=(10\sqrt{80})$$

All results are in exact correspondence with the values indicated on Fig 2-3.

2.2 The 3D-2D transformation

Now we are ready to do the full job on a physically realistic 4x3 transformation matrix mapping 3D world points on a 2D CCD image plane. The derivation will follow the scheme for the 2D-1D case where possible. Many derivations will be made by use of analogy arguments to the 2D-1D case.

2.2.1 Construction of 3D-2D transformation matrix

Again we want to find the homogeneous transformation matrix **C** which brings world coordinates into image coordinates:

EQ. 2.57
$$\text{world point: }\begin{pmatrix}x\\y\\z\\1\end{pmatrix}\quad,\quad\text{image point: }\begin{pmatrix}u\\v\\t\end{pmatrix}$$

The transformation matrix **C**, mapping 3-dimensional world points to corresponding 2-dimensional image points satisfies the equation:

EQ. 2.58
$$\mathbf{C}\begin{pmatrix}x\\y\\z\\1\end{pmatrix}=\begin{pmatrix}u\\v\\t\end{pmatrix}\quad,\quad\mathbf{C}=\begin{bmatrix}C_{00} & C_{01} & C_{02} & C_{03}\\ C_{10} & C_{11} & C_{12} & C_{13}\\ C_{20} & C_{21} & C_{22} & 1\end{bmatrix}$$

We note that this system of equations is containing 11 unknowns. Expanding EQ. 2.58 gives:

EQ. 2.59
$$\left.\begin{array}{l}C_{00}x+C_{01}y+C_{02}z+C_{03}=u=Ut\\ C_{10}x+C_{11}y+C_{12}z+C_{13}=v=Vt\\ C_{20}x+C_{21}y+C_{22}+1=t\end{array}\right\}\Rightarrow$$

EQ. 2.60
$$\begin{array}{l}C_{00}x+C_{01}y+C_{02}z+C_{03}-UC_{20}x-UC_{21}y-UC_{22}z-U=0\\ C_{10}x+C_{11}y+C_{12}z+C_{13}-VC_{20}x-VC_{21}y-VC_{22}z-V=0\end{array}$$

We see that each calibration point gives two equations. The minimum number of calibration points is therefore 5½ for the 3D-2D case of image mapping (half a point could for instance be a line where only the x coordinate is known). This should be compared with the minimum number required for the 2D-1D case, which is 5. So increasing the 2D problem with one dimension only enlarges the number of calibration points needed by half a point.

When having determined the transformation matrix **C**, it is, on basis of an imaged position, possible to find the corresponding set of points in the real world, which could have caused the image spot (it is of course only one of these points which is physically correct).

EQ. 2.60 can be rewritten to

EQ. 2.61

$$\begin{pmatrix} C_{00}-UC_{20} \\ C_{01}-UC_{21} \\ C_{02}-UC_{22} \end{pmatrix} \begin{pmatrix} x & y & z \end{pmatrix} = U \quad ; \quad \begin{pmatrix} C_{10}-VC_{20} \\ C_{11}-VC_{21} \\ C_{12}-VC_{22} \end{pmatrix} \begin{pmatrix} x & y & z \end{pmatrix} = V$$

The equations in EQ. 2.61 describe 2 planes and the corresponding set of world points is a line described by the intersection of the 2 planes.

Let

EQ. 2.62

$$\mathbf{A} = \begin{bmatrix} x^1 & y^1 & z^1 & 1 & 0 & 0 & 0 & 0 & -U^1x^1 & -U^1y^1 & -U^1z^1 \\ 0 & 0 & 0 & 0 & x^1 & y^1 & z^1 & 1 & -V^1x^1 & -V^1y^1 & -V^1z^1 \\ x^2 & y^2 & z^2 & 1 & 0 & 0 & 0 & 0 & -U^2x^2 & -U^2y^2 & -U^2z^2 \\ 0 & 0 & 0 & 0 & x^2 & y^2 & z^2 & 1 & -V^2x^2 & -V^2y^2 & -V^2z^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x^N & y^N & z^N & 1 & 0 & 0 & 0 & 0 & -U^Nx^N & -U^Ny^N & -U^Nz^N \\ 0 & 0 & 0 & 0 & x^N & y^N & z^N & 1 & -V^Nx^N & -V^Ny^N & -V^Nz^N \end{bmatrix}$$

and

EQ. 2.63

$$\mathbf{B} = \begin{pmatrix} U^1 \\ V^1 \\ U^2 \\ V^2 \\ \vdots \\ U^N \\ V^N \end{pmatrix} ; \mathbf{C} = \begin{pmatrix} C_{00} \\ C_{01} \\ C_{02} \\ C_{03} \\ C_{10} \\ C_{11} \\ C_{12} \\ C_{13} \\ C_{20} \\ C_{21} \\ C_{22} \end{pmatrix}$$

giving

EQ. 2.64

$$\mathbf{A\,C} = \mathbf{B}$$

where each superscript in EQ. 2.62 denotes an observation (calibration) point and **C** is the unknown vector we want to find. Again the problem can be solved by a traditional least squares method:

EQ. 2.65

$$\mathbf{C} = (\,\mathbf{A}^T\mathbf{A}\,)^{-1}\,\mathbf{A}^T\mathbf{B}$$

A typical set of data is shown in Table 2.3.

X	Y	Z	U	V
0.0	80.0	-1040.0	187.1	159.0
10.0	220.0	-1040.0	190.1	287.2
10.0	400.0	-1040.0	183.8	463.8
100.0	140.0	-1040.0	278.1	218.2
110.0	300.0	-1040.0	284.1	368.1
220.0	230.0	-1040.0	387.1	307.1
250.0	80.0	-1040.0	413.1	177.1
290.0	400.0	-1040.0	449.9	468.1
0.0	80.0	-1410.0	184.8	253.2
10.0	220.0	-1410.0	187.2	345.8
10.0	400.0	-1410.0	183.1	468.2
100.0	140.0	-1410.0	249.1	295.3
110.0	300.0	-1410.0	253.1	401.9
220.0	230.0	-1410.0	326.1	358.0
250.0	80.0	-1410.0	346.9	263.0
290.0	400.0	-1410.0	370.8	471.9

Table 2.3 Typical calibration data

The transformation matrix produced from the calibration points above:

EQ. 2.66
$$\mathbf{C} = \begin{bmatrix} -5.64 & 0.119 & -0.739 & -161.773 \\ -0.199 & -5.689 & 0.659 & 2508.81 \\ -0.0012 & 0.0011 & 0.0069 & 1 \end{bmatrix}$$

This is the fundamental matrix, which contains all (linear) information about the camera. By use of analogy to the 2D-1D case and the symbolic operation feature in MathCad®, I will now on the basis of **C** find the analytical expressions for all parameters in the extended pinhole model.

2.2.2 The Extended Pinhole Model

The extended pinhole model can be described by successive use of appropriate transformation matrices (also here the short notation for sinus and cosines is applied):

EQ. 2.67
$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & -dx \\ 0 & 1 & 0 & -dy \\ 0 & 0 & 1 & -dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
Translation of focal centre

EQ. 2.68
$$\mathbf{Rx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & ca & sa & 0 \\ 0 & -sa & ca & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
1st rotation of camera

EQ. 2.69
$$\mathbf{Ry} = \begin{bmatrix} cb & 0 & sb & 0 \\ 0 & 1 & 0 & 0 \\ -sb & 0 & cb & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
2nd rotation of camera

EQ. 2.70
$$\mathbf{Rz} = \begin{bmatrix} cc & sc & 0 & 0 \\ -sc & cc & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
3rd rotation of camera

EQ. 2.71
$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -f & 0 \end{bmatrix}$$
Perspective transformation (f = 1/focal length)

EQ. 2.72

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & xh \\ 0 & 1 & yh \\ 0 & 0 & 1 \end{bmatrix}$$

Origin in image plane

EQ. 2.73

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scale of V relative to U

EQ. 2.74

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ sin(\beta) & cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Skewness of V relative to U

Again, we see that the total number of unknowns in the matrices matches the number of variables in the transformation matrix **C**. Multiplying the matrices:

EQ. 2.75

$$\mathbf{K = WSUPRzRyRxD}$$

The full matrix **K** is a quite complex matrix when written in analytical form. However, the form of each of the terms is important for the derivation of the subsequent equations, so the analytical form of matrix **K** is presented below.

EQ. 2.76

$$\mathbf{K} = \begin{bmatrix} cc\,cb - xh\,f\,sb & sc\,ca + sa\,cc\,sb + sa\,xh\,f\,cb & sc\,sa - ca\,cc\,sb - ca\,xh\,f\,cb & K_{03} \\ K_{10} & K_{11} & K_{12} & K_{13} \\ -\,f\,sb & f\,cb\,sa & -\,f\,cb\,ca & f\,sb\,dx - f\,cb\,sa\,dy + f\,cb\,ca\,dz \end{bmatrix}$$

where

K_{03}: $-dx\cdot cc\cdot cb + dx\cdot xh\cdot f\cdot sb - dy\cdot sc\cdot ca - dy\cdot sa\cdot cc\cdot sb - dy\cdot sa\cdot xh\cdot f\cdot cb - dz\cdot sc\cdot sa + dz\cdot ca\cdot cc\cdot sb + dz\cdot ca\cdot xh\cdot f\cdot cb$

K_{10}: $cb\cdot p\cdot sin(\beta)\cdot cc - cb\cdot p\cdot cos(\beta)\cdot sc - yh\cdot f\cdot sb$

K_{11}: $ca\cdot p\cdot sin(\beta)\cdot sc + ca\cdot p\cdot cos(\beta)\cdot cc + sa\cdot sb\cdot p\cdot sin(\beta)\cdot cc - sa\cdot sb\cdot p\cdot cos(\beta)\cdot sc + sa\cdot yh\cdot f\cdot cb$

K_{12}: $sa\cdot p\cdot sin(\beta)\cdot sc + sa\cdot p\cdot cos(\beta)\cdot cc - ca\cdot sb\cdot p\cdot sin(\beta)\cdot cc + ca\cdot sb\cdot p\cdot cos(\beta)\cdot sc - ca\cdot yh\cdot f\cdot cb$

K_{13}: $-dx\cdot cb\cdot p\cdot sin(\beta)\cdot cc + dx\cdot cb\cdot p\cdot cos(\beta)\cdot sc + dx\cdot yh\cdot f\cdot sb - dy\cdot ca\cdot p\cdot sin(\beta)\cdot sc - dy\cdot ca\cdot p\cdot cos(\beta)\cdot cc$
$-dy\cdot sa\cdot sb\cdot p\cdot sin(\beta)\cdot cc + dy\cdot sa\cdot sb\cdot p\cdot cos(\beta)\cdot sc - dy\cdot sa\cdot yh\cdot f\cdot cb - dz\cdot sa\cdot p\cdot sin(\beta)\cdot sc$
$-dz\cdot sa\cdot p\cdot cos(\beta)\cdot cc + dz\cdot ca\cdot sb\cdot p\cdot sin(\beta)\cdot cc - dz\cdot ca\cdot sb\cdot p\cdot cos(\beta)\cdot sc + dz\cdot ca\cdot yh\cdot f\cdot cb$

2.2.2.1 Relations between C and K

First, we notice that **K** is the unnormalised version of **C**:

EQ. 2.77

$$\mathbf{C} = \mathbf{K} / K_{23}$$

From EQ. 2.76 and EQ. 2.77 we can derive the following relations:

EQ. 2.78

$$K_{20}^2 + K_{21}^2 + K_{22}^2 = f^2 \Rightarrow$$

EQ. 2.79

$$\frac{K_{20}^2}{K_{23}^2} + \frac{K_{21}^2}{K_{23}^2} + \frac{K_{22}^2}{K_{23}^2} = \frac{f^2}{K_{23}^2} \Rightarrow$$

EQ. 2.80

$$C_{20}^2 + C_{21}^2 + C_{22}^2 = \left(\frac{f}{K_{23}}\right)^2 \Rightarrow$$

EQ. 2.81

$$K_{23} = \frac{f}{\pm\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}}$$

The sign of K_{23} is determined from the sign of the exterior parameters (see later). We now have the general relationship:

EQ. 2.82
$$C_{ij} = \frac{K_{ij}}{K_{23}} = \frac{\pm\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}}{f} K_{ij}$$

Where C_{ij} are normalised coefficients and f=1/focal length

2.2.2.2 3D Focal Centre

Like in EQ. 2.24 we can rearrange the transformation matrix **C** in following way:

EQ. 2.83
$$\mathbf{C}\mathbf{x}_{hom} = \mathbf{u} \quad \textit{is equivalent to} \quad \mathbf{M}\mathbf{x} + \mathbf{T} = \mathbf{u}$$

or

EQ. 2.84
$$\begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} C_{03} \\ C_{13} \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ v \\ t \end{pmatrix}$$

where

EQ. 2.85
$$\mathbf{M} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix}$$

and

EQ. 2.86
$$\mathbf{T} = \begin{pmatrix} C_{03} \\ C_{13} \\ 1 \end{pmatrix}$$

Now **M** is quadratic and we obtain:

EQ. 2.87
$$\mathbf{x} = \mathbf{M}^{-1}\mathbf{u} - \mathbf{M}^{-1}\mathbf{T}$$
Inverse perspective projection

putting in (u,v)=(Ut,Vt)

EQ. 2.88
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathbf{M}^{-1}\begin{pmatrix} U \\ V \\ 1 \end{pmatrix} t - \mathbf{M}^{-1}\begin{pmatrix} C_{03} \\ C_{13} \\ 1 \end{pmatrix}$$

From EQ. 2.88 we get

EQ. 2.89
$$\mathbf{V} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \mathbf{M}^{-1}\begin{pmatrix} U \\ V \\ 1 \end{pmatrix}$$
Viewing direction

EQ. 2.90
$$\mathbf{F} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = -\mathbf{M}^{-1}\begin{pmatrix} C_{03} \\ C_{13} \\ 1 \end{pmatrix}$$
Focal centre

If we try to perform the same operation on **K** as on **C** when we derived the world position of focal centre **F**, we first note that the normalisation procedure is factored out. The question now is whether the extended pinhole model, like in the 2D-1D case, leads to the same result. Let

EQ. 2.91
$$\mathbf{M'}=\begin{bmatrix} cccb - xh\,f\,sb & scca + saccsb + saxh\,f\,cb & scsa - caccsb - caxh\,f\,cb \\ K_{10} & K_{11} & K_{12} \\ -\,f\,sb & f\,cbsa & -\,f\,cbca \end{bmatrix} \quad ; \quad \left|\mathbf{M'}\right| = -f\,p\cos(\beta)$$

EQ. 2.92
$$\mathbf{T'}=\begin{pmatrix} K_{03} \\ K_{13} \\ f\,sbdx - f\,cbsady + f\,cbcadz \end{pmatrix}$$

Again the mark indicates that the expressions are derived from the unnormalised matrix **K** instead of the normalised **C**. The expressions for K_{10}, K_{11}, K_{12} is given below EQ. 2.76.

EQ. 2.93
$$\mathbf{F} = -\mathbf{M'}^{-1}\mathbf{T'} = -\mathbf{M}^{-1}\mathbf{T} = \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}$$
Focal centre

So, also in the 3D-2D case, the offset described in EQ. 2.67 can be calculated from **C** by using EQ. 2.93. Again we note that an equivalent way of finding **F** is to apply Cramers rule to the complete 4x3 transformation matrix (see EQ. 2.36).

2.2.2.3 Origin in image plane

By analogy to the 2D-1D case we see that the first and second columns in $\mathbf{M}^{'-1}$ are expanding the image plane. The cross product of first and second column is therefore a vector perpendicular to the image plane (and third column ($\mathbf{M}^{'-1<2>}$)+ focal centre (**F**) is a point on the image plane). (U_0,V_0) is defined as the point where a line (focal line) through the focal centre is intersecting the image plane with an angle of 90 degrees. Knowing the normal to the image plane (in world coordinates) and the transformation **[MT]** which maps the world points to the image plane, (U_0,V_0) can be calculated as follows:

- $\mathbf{M}^{-1^{<0>}} \times \mathbf{M}^{-1^{<1>}}$ (and $\mathbf{M}^{'-1^{<0>}} \times \mathbf{M}^{'-1^{<1>}}$) is perpendicular to the image plane
- $\mathbf{F}+(\mathbf{M}^{-1^{<0>}} \times \mathbf{M}^{-1^{<1>}}$) is a point lying on the focal line

EQ. 2.94
$$(U_0,V_0) = \mathbf{M}(\mathbf{F} + (\mathbf{M}^{-1^{<0>}} \times \mathbf{M}^{-1^{<1>}})) + \mathbf{T} \Rightarrow$$

EQ. 2.95
$$(U_0,V_0) = \mathbf{M}(\mathbf{M}^{-1^{<0>}} \times \mathbf{M}^{-1^{<1>}})$$

or by analogy to EQ. 2.41 (with the argument of Strat[ix])[1]

EQ. 2.96
$$U_0 = \frac{\mathbf{M}^{T^{<0>^T}} \mathbf{M}^{T^{<2>}}}{\left|\mathbf{M}^{T^{<2>}}\right|^2} \quad ; \quad V_0 = \frac{\mathbf{M}^{T^{<1>^T}} \mathbf{M}^{T^{<2>}}}{\left|\mathbf{M}^{T^{<2>}}\right|^2}$$
Origin of image plane

Again noting that the normalisation factor is disappearing, we will try to apply EQ. 2.96 to the extended pinhole model:

EQ. 2.97
$$\mathbf{M'}^{T} = \begin{bmatrix} cccb - xh\,f\,sb & K_{10} & -\,f\,sb \\ scca + saccsb + saxh\,f\,cb & K_{11} & f\,cbsa \\ scsa - caccsb - caxh\,f\,cb & K_{12} & -\,f\,cbca \end{bmatrix} \quad ; \quad \left|\mathbf{M'}^{T}\right| = -f\,p\cos(\beta)$$

[1] Actually, it can be shown that: $\mathbf{M}^{-1<0>} \times \mathbf{M}^{-1<1>} \equiv \mathbf{M}^{T<2>}$

EQ. 2.98
$$\mathbf{M}'^{T<0>} = \begin{pmatrix} cc\,cb - xh\,f\,sb \\ sc\,ca + sa\,cc\,sb + sa\,xh\,f\,cb \\ sc\,sa - ca\,cc\,sb - ca\,xh\,f\,cb \end{pmatrix}$$

EQ. 2.99
$$\mathbf{M}'^{T<1>} = \begin{pmatrix} K_{10} \\ K_{11} \\ K_{11} \end{pmatrix}$$

EQ. 2.100
$$\mathbf{M}'^{T<2>} = \begin{pmatrix} -\,f\,sb \\ f\,cb\,sa \\ -\,f\,cb\,ca \end{pmatrix}$$

EQ. 2.101
$$U_0 = \frac{\mathbf{M}'^{T<0>^T}\,\mathbf{M}'^{T<2>}}{\left|\mathbf{M}'^{T<2>}\right|^2} = xh \quad ; \quad V_0 = \frac{\mathbf{M}'^{T<1>^T}\,\mathbf{M}'^{T<2>}}{\left|\mathbf{M}'^{T<2>}\right|^2} = yh$$

So, EQ. 2.96 is the exact formula for calculating the origin on the image plane of the camera, when the extended pinhole model is assumed.

2.2.2.4 Focal length

From EQ. 2.76 we can derive following relation:

EQ. 2.102
$$K_{00}^2 + K_{01}^2 + K_{02}^2 = xh^2 f^2 + 1 \Rightarrow$$

EQ. 2.103
$$\left(\frac{f}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}}\right)^2 \left(C_{00}^2 + C_{01}^2 + C_{02}^2\right) = xh^2 f^2 + 1 \Rightarrow$$

remembering that $xh = U_0$

EQ. 2.104
$$\frac{1}{f} = \sqrt{\frac{C_{00}^2 + C_{01}^2 + C_{02}^2}{C_{20}^2 + C_{21}^2 + C_{22}^2} - U_0{}^2} \qquad \textit{Focal length}$$

This is the exact formula for calculating the focal length of the camera when the extended pinhole model is assumed.

2.2.2.5 Aspect ratio

From EQ. 2.76 we also get:

EQ. 2.105
$$K_{10}^2 + K_{11}^2 + K_{12}^2 = yh^2 f^2 + p^2 \Rightarrow$$

and by use of EQ. 2.82

EQ. 2.106
$$p^2 = f^2 \frac{C_{10}^2 + C_{11}^2 + C_{12}^2}{C_{20}^2 + C_{21}^2 + C_{22}^2} - yh^2 f^2$$

which gives the formula for calculating aspect ratio (with $yh = V_0$)

EQ. 2.107

$$p = f\sqrt{\frac{C_{10}^2 + C_{11}^2 + C_{12}^2}{C_{20}^2 + C_{21}^2 + C_{22}^2} - V_0{}^2} \qquad \textit{Aspect ratio}$$

Here we use the knowledge that aspect ratio is a positive number.

2.2.2.6 External angles

From EQ. 2.76 we get:

EQ. 2.108

$$\frac{K_{21}}{K_{22}} = -tan(a)$$

using that the normalisation factor disappears we get

EQ. 2.109

$$a = \begin{cases} a_0 = atan(\frac{-C_{21}}{C_{22}}) \\ a_1 = atan(\frac{-C_{21}}{C_{22}}) + \pi \end{cases}$$

from EQ. 2.76 and EQ. 2.82 we get:

EQ. 2.110

$$K_{20} = -f\ sin(b) \Rightarrow$$

EQ. 2.111

$$b = asin(-\frac{K_{20}}{f}) \Rightarrow$$

EQ. 2.112

$$b = asin(\frac{-C_{20}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}}) \Rightarrow$$

EQ. 2.113

$$b = \begin{cases} asin(\frac{-C_{20}}{\sqrt{C_{20}^2+C_{21}^2+C_{22}^2}}) & positive\, K_{23} \\ \pi - asin(\frac{-C_{20}}{\sqrt{C_{20}^2+C_{21}^2+C_{22}^2}}) & \\ \hline asin(\frac{C_{20}}{\sqrt{C_{20}^2+C_{21}^2+C_{22}^2}}) & negative\, K_{23} \\ asin(\frac{-C_{20}}{\sqrt{C_{20}^2+C_{21}^2+C_{22}^2}}) - \pi & \end{cases}$$

Where K_{23} is specified in EQ. 2.81. With the usual assumption that angle b is restricted to [-π/2; π/2], EQ. 2.113 reduces to

EQ. 2.114

$$b = \begin{cases} b_0 = asin(\frac{-C_{20}}{\sqrt{C_{20}^2+C_{21}^2+C_{22}^2}}) & positive\ K_{23} \\ b_1 = asin(\frac{C_{20}}{\sqrt{C_{20}^2+C_{21}^2+C_{22}^2}}) & negative\ K_{23} \end{cases}$$

The deduction of angle c is more complex. From EQ. 2.76 we get:

EQ. 2.115
$$cos(a)K_{01} + sin(a)K_{02} = sin(c) \Rightarrow$$

EQ. 2.116
$$sin(c) = cos(a)(K_{01} + tan(a)K_{02}) \Rightarrow$$

using EQ. 2.109 and some trigonometry

EQ. 2.117
$$sin(c) = \frac{-C_{22}}{\pm\sqrt{C_{21}^2 + C_{22}^2}}(K_{01} - \frac{C_{21}}{C_{22}}K_{02}) \Rightarrow$$

and by use of EQ. 2.82

EQ. 2.118
$$sin(c) = \pm f \frac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{21}^2 + C_{22}^2}\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}}$$

Finally from EQ. 2.76 and EQ. 2.82 we get:

EQ. 2.119
$$\frac{K_{00}}{K_{20}} = \frac{-cos(c)}{f\,tan(b)} + xh \Rightarrow$$

EQ. 2.120
$$cos(c) = -f\left(\frac{C_{00}}{C_{20}} - xh\right)tan(b) \Rightarrow$$

EQ. 2.121
$$cos(c) = -f\left(\frac{C_{00}}{C_{20}} - xh\right)\frac{-C_{20}}{\sqrt{C_{21}^2 + C_{22}^2}} \Rightarrow$$

EQ. 2.122
$$cos(c) = \left(\frac{C_{00} - xhC_{20}}{\sqrt{C_{21}^2 + C_{22}^2}}\right)f$$

which divided into EQ. 2.118 gives (with *(xh,yh) = (U₀,V₀)*)

EQ. 2.123
$$tan(c) = f\frac{C_{02}C_{21} - C_{01}C_{22}}{\pm\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)} \quad \Rightarrow$$

EQ. 2.124
$$c = \begin{cases} c_0 = atan\left\{f\frac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)}\right\} & positive\,K_{23} \\ c_1 = atan\left\{f\frac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)}\right\} + \pi & \\ c_2 = \text{-}atan\left\{f\frac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)}\right\} & negative\,K_{23} \\ c_3 = \text{-}atan\left\{f\frac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)}\right\} + \pi & \end{cases}$$

Where K_{23} is specified in EQ. 2.81. We are now ready to determine the sign of the angles. We have 4 combinations of angle a and b as possible solutions.

(a_0,b_0) or (a_1,b_0)	Positive K_{23}
(a_0,b_1) or (a_1,b_1)	Negative K_{23}

And for each sign of K_{23} there exist two possibilities for angle c. All together, 8 possible combinations (and only one correct).

From EQ. 2.76 it is possible uniquely to determine the angles a,b and thereby also the sign of K_{23}, reducing the number of possibilities to only two:

EQ. 2.125
$$\begin{pmatrix} K_{20} \\ K_{21} \\ K_{22} \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = f \begin{pmatrix} -sin(b) \\ cos(b)sin(a) \\ -cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = -K_{23} = \frac{-f}{\pm\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}}$$

So we can put up following condition:

EQ. 2.126
$$\left| \begin{pmatrix} -sin(b) \\ cos(b)sin(a) \\ -cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \right| = \frac{1}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}}$$

From EQ. 2.91 we saw that

EQ. 2.127
$$|M'| = |M| \left(\frac{f}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}} \right)^3 = -f p \cos(\beta) \Rightarrow$$

EQ. 2.128
$$cos(\beta) = -\frac{|M| \left(\frac{f}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}} \right)^3}{f p}$$

which can be expressed in angles (a,b) and **F**

EQ. 2.129
$$cos(\beta) = -\frac{|M| \left\{ f \begin{pmatrix} sin(b) \\ -cos(b)sin(a) \\ cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \right\}^3}{f p}$$

Knowing the skewness β is always in the range [-π/2; π/2] (actually for any realistic camera, β is very close to 0), EQ. 2.129 leads to following condition:

EQ. 2.130
$$\begin{pmatrix} -sin(b) \\ cos(b)sin(a) \\ -cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} > 0$$

Which together with EQ. 2.126 leads to the final condition for determining the angles a and b and the sign of K_{23}.

EQ. 2.131

$$\begin{pmatrix} -sin(b) \\ cos(b)sin(a) \\ -cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \frac{1}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}}$$

EQ. 2.131 can only be fulfilled for one set of the 4 possible combinations of angle a,b. Since the sign of K_{23} is known from the unique determination of angle b, only 2 possible solutions remain: ($a,b,c_0/c_1$) or ($a,b,c_2/c_3$). From EQ. 2.76 we have

EQ. 2.132

$$\begin{pmatrix} K_{00} \\ K_{01} \\ K_{02} \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} cos(c)cos(b) - xh\,f\,sin(b) \\ sin(c)cos(a) + sin(a)cos(c)sin(b) + xh\,f\,sin(a)cos(b) \\ sin(c)sin(a) - cos(a)cos(c)sin(b) - xh\,f\,cos(a)cos(b) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = -K_{03} = \frac{-f}{\pm\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}} C_{03}$$

Giving the final condition:

EQ. 2.133

$$\left| \begin{pmatrix} cos(c)cos(b) - xh\,f\,sin(b) \\ sin(c)cos(a) + sin(a)cos(c)sin(b) + xh\,f\,sin(a)cos(b) \\ sin(c)sin(a) - cos(a)cos(c)sin(b) - xh\,f\,cos(a)cos(b) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \right| = \left| \frac{f}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}} C_{03} \right|$$

EQ. 2.133 uniquely distinguish c_0 from c_1 (if K_{23} positive) or c_2 from c_3 (if K_{23} negative).

2.2.2.7 Skewness

With all external angles determined it is now possible to find the correct sign of the skewness. The magnitude of the skewness β is found in EQ. 2.128 (and EQ. 2.129)

(EQ. 2.128)

$$cos(\beta) = -\frac{|M|\left(\frac{f}{\sqrt{C_{20}^2+C_{21}^2+C_{22}^2}}\right)^3}{f\,p}$$

In order to find the correct sign of β, we have to look at the 2nd row of matrix **C,** where another expression for β can be found. From EQ. 2.76 and EQ. 2.82 we get:

EQ. 2.134

$$cos(a)K_{11} + sin(a)K_{12} = p\,cos(c)\{tan(c)sin(\beta) + cos(\beta)\}$$

let

EQ. 2.135

$$n = tan(c)$$

EQ. 2.136

$$m = \frac{cos(a)K_{11} + sin(a)K_{12}}{p\,cos(c)} = f\,\frac{cos(a)C_{11} + sin(a)C_{12}}{p\,cos(c)\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}}$$

$$= f\,\frac{cos(a)C_{11} + sin(a)C_{12}}{p\,cos(c)} \begin{pmatrix} sin(b) \\ -cos(b)sin(a) \\ cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}$$

this reformulates EQ. 2.134 to

EQ. 2.137

$$n \cdot sin(\beta) + cos(\beta) = m$$

which has the solutions

EQ. 2.138

$$\beta = 2atan\left[\frac{n \pm \sqrt{1+n^2-m^2}}{1+m}\right]$$

Because of the factor 2 in front of arctangent, EQ. 2.138 only gives two possible solutions for β, and only the physically correct one also appears in EQ. 2.128.

All parameters in the extended pinhole model are hereby uniquely determined in terms of the 4x3 transformation matrix **C**.

2.2.3 Equation summary

For quick access the formulas derived above are concatenated in this subchapter.

(EQ. 2.58)

$$\mathbf{C}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ v \\ t \end{pmatrix} \quad , \quad \mathbf{C} = \begin{bmatrix} C_{00} & C_{01} & C_{02} & C_{03} \\ C_{10} & C_{11} & C_{12} & C_{13} \\ C_{20} & C_{21} & C_{22} & 1 \end{bmatrix}$$

(EQ. 2.85)

$$\mathbf{M} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix} \quad ; \quad \mathbf{T} = \begin{pmatrix} C_{03} \\ C_{13} \\ 1 \end{pmatrix}$$

(EQ. 2.93)

$$\mathbf{F} = \quad -\mathbf{M}^{-1}\mathbf{T} = \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}$$

Focal Centre

(EQ. 2.96)

$$U_0 = \frac{{\mathbf{M}^{T^{<0>}}}^{T} \mathbf{M}^{T^{<2>}}}{\left|\mathbf{M}^{T^{<2>}}\right|^2} \quad ; \quad V_0 = \frac{{\mathbf{M}^{T^{<1>}}}^{T} \mathbf{M}^{T^{<2>}}}{\left|\mathbf{M}^{T^{<2>}}\right|^2}$$

Origin of image plane

(EQ. 2.104)

$$\frac{1}{f} = \sqrt{\frac{C_{00}^2 + C_{01}^2 + C_{02}^2}{C_{20}^2 + C_{21}^2 + C_{22}^2} - {U_0}^2}$$

Focal length

(EQ. 2.107)

$$p = f\sqrt{\frac{C_{10}^2 + C_{11}^2 + C_{12}^2}{C_{20}^2 + C_{21}^2 + C_{22}^2} - {V_0}^2}$$

Aspect ratio

(EQ. 2.109)

$$a = \begin{cases} a_0 = atan(\frac{-C_{21}}{C_{22}}) \\ a_1 = atan(\frac{-C_{21}}{C_{22}}) + \pi \end{cases}$$

(EQ. 2.114)

$$b = \begin{cases} b_0 = asin(\frac{-C_{20}}{\sqrt{C_{20}^2+C_{21}^2+C_{22}^2}}) & \text{positive } K_{23} \\ b_1 = asin(\frac{C_{20}}{\sqrt{C_{20}^2+C_{21}^2+C_{22}^2}}) & \text{negative } K_{23} \end{cases}$$

(EQ. 2.124)

$$c = \begin{cases} atan\left\{ f \dfrac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)} \right\} & positive\,K_{23} \\ atan\left\{ f \dfrac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)} \right\} + \pi & \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \cdots\cdots\cdots \\ -\,atan\left\{ f \dfrac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)} \right\} & negative\,K_{23} \\ -\,atan\left\{ f \dfrac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)} \right\} + \pi & \end{cases}$$

(EQ. 2.131)

$$\begin{pmatrix} -sin(b) \\ cos(b)sin(a) \\ -cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \frac{1}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}} \qquad \textit{Condition 1}$$

(EQ. 2.133)

$$\left| \begin{pmatrix} cos(c)\,cos(b) - xh\,f\,sin(b) \\ sin(c)\,cos(a) + sin(a)\,cos(c)\,sin(b) + xh\,f\,sin(a)\,cos(b) \\ sin(c)\,sin(a) - cos(a)\,cos(c)\,sin(b) - xh\,f\,cos(a)\,cos(b) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \right| = \left| \frac{f}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}} C_{03} \right| \qquad \textit{Condition 2}$$

(EQ. 2.138 & EQ. 2.129)

$$\beta = \begin{cases} acos\left(-\dfrac{|\mathbf{M}|\left\{ f \begin{pmatrix} sin(b) \\ -cos(b)sin(a) \\ cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \right\}^3}{fp}\right) \\ -\,acos\left(-\dfrac{|\mathbf{M}|\left\{ f \begin{pmatrix} sin(b) \\ -cos(b)sin(a) \\ cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \right\}^3}{fp}\right) \\ 2atan\left(\dfrac{n + \sqrt{1 + n^2 - m^2}}{1 + m}\right) \\ 2atan\left(\dfrac{n - \sqrt{1 + n^2 - m^2}}{1 + m}\right) \end{cases}$$

where $n = tan(c)$ and $m = f\dfrac{cos(a)C_{11} + sin(a)C_{12}}{p\,cos(c)} \begin{pmatrix} sin(b) \\ -cos(b)sin(a) \\ cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}$

The physically correct angle β appears twice in the expression above.

2.2.4 Applying results

In EQ. 2.66 we found a typical transformation matrix based on physical data measured with a standard video camera and frame grabber:

(EQ. 2.66)

$$\mathbf{C} = \begin{bmatrix} -5.640 & 0.119 & -0.739 & -161.773 \\ -0.199 & -5.689 & 0.659 & 2508.810 \\ -0.0012 & 0.0011 & 0.0069 & 1 \end{bmatrix}$$

Let us now extract all the interesting features about the camera that produced this matrix and the data behind it.

EQ. 2.139

$$\mathbf{M} = \begin{bmatrix} -5.64 & 0.119 & -0.739 \\ -0.199 & -5.689 & 0.659 \\ -0.0012 & 0.0011 & 0.0069 \end{bmatrix}$$

EQ. 2.140

$$\mathbf{T} = \begin{bmatrix} -161.773 \\ 2508.81 \\ 1 \end{bmatrix}$$

EQ. 2.141

$$\mathbf{M}^{-1} = \begin{bmatrix} -0.173 & -0.007 & -17.953 \\ 0.003 & -0.172 & 16.822 \\ -0.030 & 0.026 & 139.805 \end{bmatrix}$$

EQ. 2.142

$$\mathbf{F} = \mathbf{M}^{-1}\mathbf{T} = \begin{bmatrix} -0.173 & -0.007 & -17.953 \\ 0.003 & -0.172 & 16.822 \\ -0.03 & 0.026 & 139.805 \end{bmatrix} \begin{pmatrix} -161.773 \\ 2508.81 \\ 1 \end{pmatrix} = \begin{pmatrix} 7.68 \\ 416.34 \\ -210.44 \end{pmatrix} [\mathrm{mm}]$$

This result is in very fine accordance with what I was able to measure manually (it is not possible exactly to locate the focal point in a lens since it is a virtual point based on the pinhole model).

EQ. 2.143

$$\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2} = \pm\sqrt{-0.001^2 + 0.001^2 + 0.007^2} = \pm 0.007055$$

EQ. 2.144

$$\mathbf{M}^T = \begin{bmatrix} -5.64 & -0.199 & -0.0011 \\ 0.119 & -5.689 & 0.0012 \\ -0.739 & 0.659 & 0.0069 \end{bmatrix}$$

EQ. 2.145

$$U_0 = \frac{\mathbf{M}^{T<0>^T}\mathbf{M}^{T<2>}}{\left(\mathbf{M}^{T<2>}\right)^2} = 31.7 \quad ; \quad V_0 = \frac{\mathbf{M}^{T<1>^T}\mathbf{M}^{T<2>}}{\left(\mathbf{M}^{T<2>}\right)^2} = -29.3$$

$(U_0,V_0)=(xh,yh)$ is measured relative to the theoretical centre of the CCD chip which is (288,384).

EQ. 2.146

$$focal = \frac{1}{f} = \sqrt{\frac{C_{00}^2 + C_{01}^2 + C_{02}^2}{C_{20}^2 + C_{21}^2 + C_{22}^2} - U_0{}^2} = 805.9\ [pixels]$$

and knowing that the size of a ½” CCD chip is approximately 5.94mm x 3.96mm or 768x576 pixels, the camera constant can be estimated in millimetres by division with 129.3[pixels/mm]. The focal length in millimetres is then approximately: 6.2 mm. That result must be compared with the specified size of the lens: 6.5 mm.

EQ. 2.147

$$p = \sqrt{\frac{C_{10}^2 + C_{11}^2 + C_{12}^2}{C_{20}^2 + C_{21}^2 + C_{22}^2} - V_0{}^2} = 1.007 \qquad \textit{Aspect ratio}$$

EQ. 2.148
$$atan\left(\frac{-C_{21}}{C_{22}}\right) = atan\left(-\frac{0.001093}{0.006873}\right) = -9.04^\circ \Rightarrow a = \begin{cases} a_0 = -9.04^\circ \\ a_1 = 170.96^\circ \end{cases}$$

EQ. 2.149
$$asin\left(\frac{-C_{20}}{\sqrt{C_{20}+C_{21}+C_{22}}}\right) = asin\left(\frac{0.001157}{0.007055}\right) = 9.44^\circ \Rightarrow$$

EQ. 2.150
$$b = \begin{cases} b_0 = \ \ 9.44^\circ \\ b_1 = -9.44^\circ \end{cases}$$

EQ. 2.151
$$\left| \begin{pmatrix} -sin(b_1) \\ cos(b_1)sin(a_0) \\ -cos(b_1)cos(a_0) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \right| = \frac{1}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}} \quad \Rightarrow$$

EQ. 2.152
$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -9.04^\circ \\ -9.44^\circ \end{pmatrix} and\ K_{23} < 0$$

EQ. 2.153
$$atan\left\{ f\frac{C_{02}C_{21} - C_{01}C_{22}}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}\left(C_{00} - U_0C_{20}\right)} \right\} = 2.36^\circ \quad \Rightarrow$$

EQ. 2.154
$$c = \begin{cases} c_0 = \quad -2.36^\circ \\ c_1 = -182.36^\circ \end{cases}$$

EQ. 2.155
$$\left| \begin{pmatrix} cos(c_0)cos(b) - U_0\ f\ sin(b) \\ sin(c_0)cos(a) + sin(a)cos(c_0)sin(b) + U_0\ f\ sin(a)cos(b) \\ sin(c_0)sin(a) - cos(a)cos(c_0)sin(b) - U_0\ f\ cos(a)cos(b) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \right| = \left| \frac{f}{\sqrt{C_{20}^2 + C_{21}^2 + C_{22}^2}} C_{03} \right| \Rightarrow$$

EQ. 2.156
$$(a,b,c) = (-9.04^\circ, -9.44^\circ, -2.36^\circ)$$

The only remaining question is β. For that we calculate:

EQ. 2.157
$$n = tan(c) = -0.041$$

EQ. 2.158
$$m = f\frac{cos(a)C_{11} + sin(a)C_{12}}{p\,cos(c)} \begin{pmatrix} sin(b) \\ -cos(b)sin(a) \\ cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = 0.999993$$

EQ. 2.159

$$\beta = \begin{cases} acos\left(-\dfrac{|\mathbf{M}|\left\{ f\begin{pmatrix} sin(b) \\ -cos(b)sin(a) \\ cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}\right\}^3}{fp}\right) \\ -\, acos\left(-\dfrac{|\mathbf{M}|\left\{ f\begin{pmatrix} sin(b) \\ -cos(b)sin(a) \\ cos(b)cos(a) \end{pmatrix}^T \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}\right\}^3}{fp}\right) \\ 2atan\left(\dfrac{n+\sqrt{1+n^2-m^2}}{1+m}\right) \\ 2atan\left(\dfrac{n-\sqrt{1+n^2-m^2}}{1+m}\right) \end{cases}$$

EQ. 2.160

$$\beta = \begin{cases} 0.01^\circ \\ -0.01^\circ \\ -4.72^\circ \\ 0.01^\circ \end{cases} \Rightarrow \beta = 0.01^\circ$$

So the correct physical value of β is 0.01° and now we are able to reconstruct the transformation matrix and compare it with the original:

EQ. 2.161

$$\mathbf{K} \quad = \begin{bmatrix} 0.992 & -0.021 & 0.130 \\ 0.035 & 1.001 & -0.116 \\ 0.000 & -0.000 & -0.001 \end{bmatrix}$$

Unnormalised matrix

EQ. 2.162

$$\frac{\mathbf{K}}{K_{23}} = \begin{bmatrix} -5.64 & 0.119 & -0.739 & -161.773 \\ -0.199 & -5.689 & 0.659 & 2508.81 \\ -0.0012 & 0.0011 & 0.0069 & 1 \end{bmatrix}$$

Normalised matrix

Which is exactly the same as the one presented in EQ. 2.66.

2.3 3D Measuring

With fully calibrated cameras it is now possible to do 3D measuring of points in space. I will focus on 2 different types of measuring based on, respectively, a) more than one fixed cameras and b) only one movable camera.

2.3.1 3D measuring with fixed cameras

When the camera is fixed in some external configuration, it is necessary to have at least two different cameras looking at the same object in order to determine the 3D position of that object.

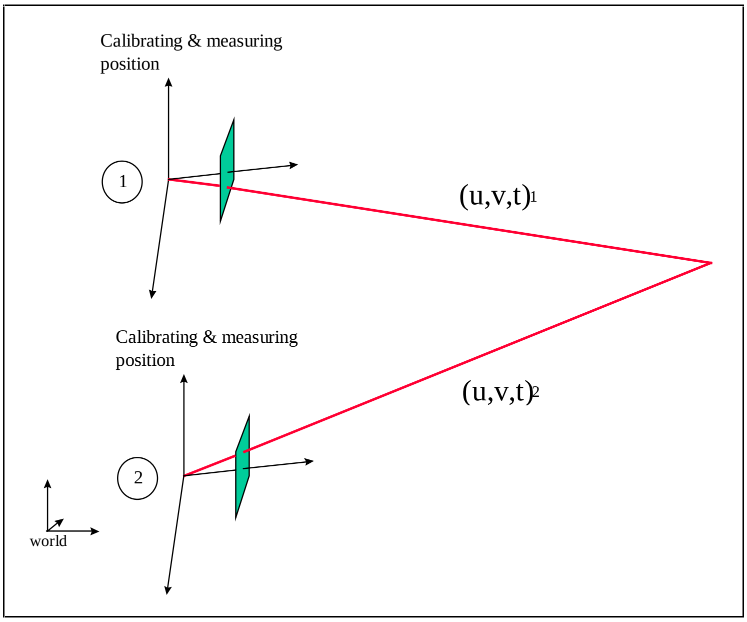

Calibrating & measuring position
1
(u,v,t)1
Calibrating & measuring position
(u,v,t)2
2
world

Fig 2-4 Fixed cameras measurement

The advantage is that calibration and measuring is done at exactly the same location and no additional uncertainty is introduced. The parametric form of a line through the focal centre and a point in space is taken from EQ. 2.88 - EQ. 2.90.

EQ. 2.163 $$\mathbf{x} = \mathbf{F} + \mathbf{V}t$$ *Parametric line description*

If only two cameras are used, the 3D point is found where the two lines of interpretation are intersecting. In practice, they are not intersecting and the point is estimated to be in the middle of the shortest distance from line 1 to line 2. This point can be calculated according to following formula[xxiv]:

EQ. 2.164 *Crossing values*

$$t_1 = \frac{\mathbf{V}_1((\mathbf{V}_1 \times \mathbf{V}_2) \times (\mathbf{F}_1 - \mathbf{F}_2))}{(|\mathbf{V}_1 \times \mathbf{V}_2|)^2}$$

$$t_2 = \frac{((\mathbf{F}_2 - \mathbf{F}_1) \times \mathbf{V}_1)(\mathbf{V}_2 \times \mathbf{V}_1))}{(|\mathbf{V}_1 \times \mathbf{V}_2|)^2}$$

EQ. 2.165 $$\mathbf{P} = \frac{(\mathbf{V}_1 \cdot t_1 + \mathbf{F}_1) + (\mathbf{V}_2 \cdot t_2 + \mathbf{F}_2)}{2}$$ *Crossing point*

When more than 2 cameras are available a more robust method based on Least Squares or similar can be applied.

This method was tried on following data

World			Pos 1		Pos 2	
X	Y	Z	U	V	U	V
0.0	80.0	-1020.0	201.5	201.0	185.8	191.0
10.0	220.0	-1020.0	209.8	341.7	182.2	303.2
10.0	400.0	-1020.0	208.9	522.6	167.2	461.2
100.0	140.0	-1020.0	299.3	263.6	265.1	240.9
110.0	300.0	-1020.0	307.8	420.2	264.2	376.7
220.0	230.0	-1020.0	412.1	353.0	366.1	320.3
250.0	80.0	-1020.0	444.8	211.0	396.3	199.0
290.0	400.0	-1020.0	475.3	513.2	424.1	476.0
0.0	80.0	-1470.0	201.8	258.9	233.0	306.9
10.0	220.0	-1470.0	206.9	349.1	232.1	386.2
10.0	400.0	-1470.0	206.0	465.3	223.8	493.8
100.0	140.0	-1470.0	265.0	298.3	288.0	343.2
110.0	300.0	-1470.0	270.2	401.1	288.1	437.0
220.0	230.0	-1470.0	339.9	357.2	356.1	399.0
250.0	80.0	-1470.0	359.2	263.1	377.9	313.8
290.0	400.0	-1470.0	381.2	464.0	394.1	503.9

Table 2.4 Calibration data from 2 different camera positions

And 2 points in space were measured

World			**Position 1**		**Position 2**	
X	Y	Z	U	V	U	V
220	230	-1256	367.2	356.0	360.0	369.8
220	230	-1200	376.8	355.0	361.1	359.2

Table 2.5 Measuring data from 2 different camera positions

The data in Table 2.4 produces following calibration matrices

$$\mathbf{C}_1 = \begin{bmatrix} -4.036 & 0.024 & 0.949 & 197.52 \\ -0.31 & -3.87 & 1.716 & 1294.87 \\ -0.001 & -7e-5 & 0.005 & 1 \end{bmatrix} \qquad \mathbf{C}_2 = \begin{bmatrix} -17.55 & 2.607 & 7.449 & 3462.78 \\ -0.496 & -15.654 & 12.35 & 9817.94 \\ 0.001 & 0.005 & 0.022 & 1 \end{bmatrix}$$

which by use of EQ. 2.88 - EQ. 2.90 leads to

$$\mathbf{F}_1 = \begin{pmatrix} 199.27 \\ 489.41 \\ -166.65 \end{pmatrix} \qquad \mathbf{V}_1 = \begin{pmatrix} -0.835 \\ 10.168 \\ 42.8 \end{pmatrix}$$

EQ. 2.166

$$\mathbf{F}_2 = \begin{pmatrix} 1.46 \\ 242.36 \\ -208.14 \end{pmatrix} \qquad \mathbf{V}_2 = \begin{pmatrix} -42.78 \\ 2.341 \\ 205.05 \end{pmatrix}$$

and from EQ. 2.164

EQ. 2.167

$$t_1 = 25.487 \qquad t_2 = 5.118 \qquad \text{Point A}$$

$$t_1 = 24.165 \qquad t_2 = 4.836 \qquad \text{Point B}$$

we finally get to

World			Calculated		
X	Y	Z	X	Y	Z
220	230	-1256	220.47	230.31	-1257.50
220	230	-1200	220.26	229.73	-1197.78

Table 2.6 Measured data compared to calculated

We see an extremely high accuracy in the calculation of points in 3D space when the cameras are untouched from calibration to measurement.

2.3.2 3D measuring with a movable camera

Working with fixed cameras has many limitations, especially when dealing with robots. It is therefore necessary to develop algorithms that can make it possible to move a camera away from its calibration position and still achieve accurate measurements. The basic ideas are described in the following (see figure below).

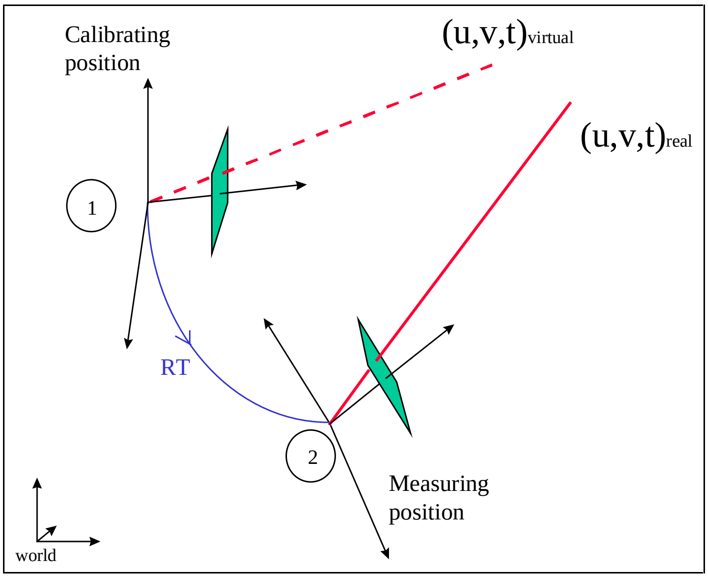


Fig 2-5 Movable camera

If the measurement has been done in position 1 the 3D point in space would have been calculated according to the usual formula:

EQ. 2.168 $$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{world}^{virtual} = \mathbf{V}t + \mathbf{F}$$ *Virtual*

Since the measurement actually is done in position 2, the real position is calculated by multiplying with the rotation+translation **RT**:

EQ. 2.169 $$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{world} = \mathbf{RV}t + (\mathbf{F} + \mathbf{T})$$

or

EQ. 2.170
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{world} = \left\{ \mathbf{RM}^{-1} \begin{pmatrix} U \\ V \\ 1 \end{pmatrix} \right\} t + \left\{ \mathbf{T} + \mathbf{F} \right\}$$

The equation above is also just a parametric line description and with 2 different positions of the movable camera, the crossing point can be calculated by using EQ. 2.164 and EQ. 2.165. With a movable camera however there will usually be more than 2 measurements available and the Least Squares Method will probably be the most applicable.

2.3.3 Testing the movable camera solution

The problem with moving cameras is of course the additional contribution to uncertainty coming from the movement **RT**. It is also difficult to do reliable tests without a very accurate device for moving the camera. A possible testing method could be to calibrate the camera in two different positions, calculate these positions by splitting the matrices into pinhole parameters and from them calculate the matrix **RT**. The problem with this method is, as we shall see in chapter 2.4, that taken one by one, some of the parameters in the extended pinhole model are very poorly determined, only the complete calibration matrix being always reliable. That means the calculation of **RT** will be defective, unless the pinhole estimation is improved. Fortunately, there are methods for improving the accuracy of the extended pinhole parameters and especially if we require fixed internal parameters for all calibration positions, the calculation of **RT** is quite precise as shown in following example.

One camera is moved to position 1 and calibrated, then a 3D measurement of a point in space is made. The camera is then moved to position 2, the point in space is measured again and finally the camera is calibrated again in this new position. The obtained data is shown in Table 2.7.

World			**Pos 1**		**Pos 2**	
X	Y	Z	U	V	U	V
0.0	80.0	-1040.0	187.1	159.0	176.0	86.0
10.0	220.0	-1040.0	190.1	287.2	185.9	227.8
10.0	400.0	-1040.0	183.8	463.8	188.0	403.2
100.0	140.0	-1040.0	278.1	218.2	275.2	148.1
110.0	300.0	-1040.0	284.1	368.1	283.9	305.8
220.0	230.0	-1040.0	387.1	307.1	388.2	238.9
250.0	80.0	-1040.0	413.1	177.1	419.8	95.2
290.0	400.0	-1040.0	449.9	468.1	445.1	395.8
0.0	80.0	-1410.0	184.8	253.2	178.0	125.8
10.0	220.0	-1410.0	187.2	345.8	184.9	223.1
10.0	400.0	-1410.0	183.1	468.2	185.9	345.8
100.0	140.0	-1410.0	249.1	295.3	246.8	168.1
110.0	300.0	-1410.0	253.1	401.9	252.9	277.8
220.0	230.0	-1410.0	326.1	358.0	326.2	230.8
250.0	80.0	-1410.0	346.9	263.0	348.0	130.0
290.0	400.0	-1410.0	370.8	471.9	369.1	342.8

Table 2.7 Calibration data from 2 different camera positions

The point in space

World			Position 1		Position 2	
X	Y	Z	U	V	U	V
110	300	-1250	263.9	390.2	263.9	287.2

Table 2.8 Point measured from 2 different camera positions

With independent calibration routines (different internal parameters) the data in Table 2.7 produces following calibration matrices

$$\mathbf{C}_1 = \begin{bmatrix} -5.970 & 0.434 & 1.241 & 126.23 \\ -0.644 & -5.269 & 3.298 & 2892.81 \\ -0.001 & 0.001 & 0.007 & 1 \end{bmatrix} \qquad \mathbf{C}_2 = \begin{bmatrix} -3.670 & -0.103 & -0.77 & 204.18 \\ -0.191 & -3.640 & 0.900 & 936.06 \\ -0.001 & -5e-4 & 0.004 & 1 \end{bmatrix}$$

These two matrices are split into the 11 pinhole parameters

	X	Y	Z	a	b	c	asp	focal	skew	U_0	V_0	Residual
$\mathbf{C}_1$	7.7	416.3	-210.4	-9.04	-9.44	-2.36	1.007	805.9	0.01	319.7	354.7	0.60
$\mathbf{C}_2$	5.5	204.8	-210.7	6.61	-9.56	0.22	1.005	804.2	-0.28	317.9	306.5	0.71

Table 2.9 Pinhole parameters from 2 different camera positions

Here we see that the internal parameters are differing despite the fact that the same camera is used in both positions. By use of same procedure as described under “Fixed cameras” the position of the point in space can be estimated.

World			Calculated		
X	Y	Z	X	Y	Z
110	300	-1250	110.48	299.83	-1255.08

Table 2.10 Measured data compared to calculated

Again, we see a fine accuracy in the calculation of points in 3D space when the measurement is done in same place as the calibration. The result is not quite as good as the result presented in Table 2.6, but some minor noise is always introduced when you have to touch the camera between measurements and this example just shows how sensitive things are.

With a joint calibration routine (same internal parameters, see Table 2.23 and Table 2.24), the data in Table 2.7 produces following calibration matrices

$$\mathbf{C}_1 = \begin{bmatrix} -6.76 & 0.60 & 1.38 & 91.91 \\ -0.71 & -5.85 & 3.72 & 3227.49 \\ -0.001 & 0.002 & 0.008 & 1 \end{bmatrix} \qquad \mathbf{C}_2 = \begin{bmatrix} -3.69 & -0.13 & 0.78 & 206.97 \\ -0.20 & -3.68 & 0.90 & 939.35 \\ -0.001 & -0.001 & 0.004 & 1 \end{bmatrix}$$

These two matrices split into two other sets of pinhole parameters

	X	Y	Z	a	b	c	asp	focal	skew	U_0	V_0	Residual
$\mathbf{C}_1$	7.8	416.4	-211.7	-10.90	-9.29	-2.68	1.007	805.3	-0.15	318.7	328.4	0.78
$\mathbf{C}_2$	5.0	205.2	-205.0	8.08	-9.59	0.50	1.007	805.3	-0.15	318.7	328.4	0.91

Table 2.11 Pinhole parameters from 2 different camera positions

Here we note that the internal parameters are the same for both positions at the cost of a minor rise in the residual. With fixed internal parameters it is possible to find a transformation **RT** which exactly brings $\mathbf{C}_1$ into $\mathbf{C}_2$, and a 3D measurement performed with $\mathbf{C}_1$ and $\mathbf{C}_2$ gives exactly the same result as using EQ. 2.170.

EQ. 2.171

$$\mathbf{T} = \begin{pmatrix} 7.8 \\ 416.4 \\ -211.7 \end{pmatrix} - \begin{pmatrix} 5.0 \\ 205.0 \\ -205.0 \end{pmatrix} = \begin{pmatrix} 2.8 \\ 211.4 \\ -6.7 \end{pmatrix}$$

EQ. 2.172

$$\mathbf{R} = \mathbf{M}_2^{-1}\mathbf{M}_1 = \begin{bmatrix} 0.998 & -0.055 & 0.006 \\ 0.054 & 0.947 & -0.317 \\ 0.012 & 0.317 & 0.948 \end{bmatrix}$$

The moving device (robot) normally gives the values found in above equations and only the calibration in position 1 is used:

EQ. 2.173

$$\mathbf{C}_1 = \begin{bmatrix} -6.76 & 0.60 & 1.38 & 91.91 \\ -0.71 & -5.85 & 3.72 & 3227.49 \\ -0.001 & 0.002 & 0.008 & 1 \end{bmatrix} \Rightarrow$$

EQ. 2.174

$$\mathbf{M}_1^{-1} = \begin{bmatrix} -6.76 & 0.60 & 1.38 \\ -0.71 & -5.85 & 3.72 \\ -0.001 & 0.002 & 0.008 \end{bmatrix}^{-1} = \begin{bmatrix} -0.153 & -0.007 & 30.97 \\ 0.002 & -0.151 & 72.33 \\ -0.026 & 0.029 & 119.80 \end{bmatrix}$$

EQ. 2.175

$$\mathbf{F}_1 = \mathbf{M}_1^{-1}\begin{pmatrix} 91.91 \\ 3227.49 \\ 1 \end{pmatrix} = \begin{pmatrix} 7.8 \\ 416.4 \\ -211.7 \end{pmatrix}$$

The two lines of interpretation are now given by

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_1 = \left\{\mathbf{M}^{-1}\begin{pmatrix} 263.9 \\ 390.2 \\ 1 \end{pmatrix}\right\}t + \mathbf{F}_1$$

EQ. 2.176

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_2 = \left\{\mathbf{R}\mathbf{M}_1^{-1}\begin{pmatrix} 263.9 \\ 287.2 \\ 1 \end{pmatrix}\right\}t + (\mathbf{T} + \mathbf{F}_1)$$

using EQ. 2.164 and EQ. 2.165 produces the final result

World			Calculated		
X	Y	Z	X	Y	Z
110	300	-1250	111.00	299.90	-1258.87

Table 2.12 Measured data compared to calculated

So even for a moving camera it is possible to achieve good results if the camera is well calibrated and the movement of the camera is well known. Note that the accuracy of any measurement in space always can be improved by taking more images from different positions. The example above is using the minimal set of images and still the error in the estimation of Z is less than 1 percent.

2.4 Variability of parameters

It is a well-known problem that the stability of some of the parameters in the extended pinhole model is poor[xiv] when determined by traditional calibration. With the derived formulas from the previous section, I will make a thorough analysis of which parameters are stable and which are not. Also any kind of correlation between the parameters will be checked. I will use 2 different methods for achieving the results.

2.4.1 Methods for analysis

Method A[xxv]: From a measured set of data (world points + corresponding image points) we find the best transformation matrix **C**. We assume the measurements of world points as well as image points to be disturbed by noise of Gaussian nature. We then can construct new “data” from the measured set of world points, assuming the variance of the noise on the measured image points is approximately equal to the variance of the first measurement (in Numerical Recipes[xxvi] it is described how to construct a generator which produces Gaussian distributed numbers). This seems fair since the fitting of the transformation matrix **C** is based on an assumption of Gaussian noise. For each new set of data it is now possible to find the best transformation matrix **C** and subsequently the 11 parameters contained in the pinhole model. With enough pseudo data, it is possible to estimate mean values, variance and correlation and to make graphical representations from following formulas:

EQ. 2.177

$$\bar{k} = \frac{1}{max}\sum_{j=1}^{max} k_j^{exact}$$

EQ. 2.178

$$VAR(k) = \overline{k^2} - \bar{k}^2$$

EQ. 2.179

$$C^{\alpha\beta} = \overline{k^\alpha k^\beta} - \overline{k^\alpha}\,\overline{k^\beta}$$

Method B[xxvii]: Another method based on Bayes[xxviii] theorem expresses the probability of a parameter value as a function of measured data. A mathematical expectation value can be written:

EQ. 2.180

$$\begin{aligned} E(k^\alpha) &= \frac{1}{\nu}\int_M dk\, k^\alpha \sigma_M(k) \\ M &= \text{Model space} \\ \nu &= \int_M dk\, \sigma_M(k) \\ \nu\sigma_M(k) &= \text{propability density of parameter } K = \rho^M L(k) \\ \rho^M &= \text{a priori knowledge (eg. } > 0) \\ L(k)dk &= \text{the probability that the parameter has value } k \text{ in } (k, k+dk) \end{aligned}$$

With the assumption of Gaussian noise on measured data, *L(k)* can be written:

EQ. 2.181

$$\begin{aligned} L(k) &= \frac{1}{c} e^{-\sum_i (g(k)-d_i)^2 / 2\sigma^2} \\ g &= \text{the model} \\ d &= \text{measured data} \\ \sigma &= \text{uncertainty of data} \\ c &= \text{normalisation factor} \end{aligned}$$

When calculating the mean value of k^α in practice, we need to integrate over model space *M*. In other words calculate the sum:

EQ. 2.182
$$\overline{k^{\alpha}} = \sum k^{\alpha} P(\underline{k})$$
$$P(\underline{k}) = \frac{1}{\nu} e^{-\sum_{j} \left(y_j^{teo} - y_j^{measured} \right)^2 \big/ 2\sigma^2}$$
$$\nu = \sum_{\overline{k}} e^{-\sum_{j} \left(y_j^{teo} - y_j^{measured} \right)^2 \big/ 2\sigma^2}$$

The camera model involves 11 parameters, so going systematically through the complete parameter space would be a huge task. It is therefore of great advantage to use a Monte-Carlo process with a Markov chain of guesses on {k$_i$}. A Markov chain is a transition from one state to the next which fulfil the following two requirements. The transition must be based on stochastic events and the present state influences on future states but the steps which lead to current state does not influence future steps. In other words, a Markov process is a stochastic process with shortest possible "memory" (1 step).

In Method B the same values are calculated as follows:

EQ. 2.183
$$\overline{k^{\alpha}} = \nu^{-1} \sum_{n=1}^{N} k^{\alpha} L(\underline{k}_n)$$

EQ. 2.184
$$C^{\alpha\beta} = \nu^{-1} \sum_{n=1}^{N} k^{\alpha} k^{\beta} L(\underline{k}_n) - \overline{k^{\alpha}}\,\overline{k^{\beta}}$$

where

EQ. 2.185
$$\nu = \sum_{n=1}^{N} L(\underline{k}_n)$$

and

EQ. 2.186
$$L(\underline{k}_n) = e^{-\sum_{i} \frac{\left| y_i^{teo}(n) - y_i^{measured} \right|^2}{2\sigma_i^2}}$$

Where $y^{measured}$ are measured pixel values, y^{teo} corresponding pixel values calculated from the extended pinhole model with the current set of parameters and σ$_i$ the standard deviation, which I assume to be constant over the whole image.

The best value of all parameters is known (the question is how stable they are); in this case it is possible to speed up the procedure of finding the mean value of k^{α}. The technique was introduced by Metropolis[xxix] and describes how to exchange an ordinary random number generator with one that produces sets of parameters with the distribution $L(\underline{k}_n)$. This avoids all infinitesimal contributions since all sampling is done in the interesting domain. The equations are now reduced to:

EQ. 2.187
$$\overline{k^{\alpha}} = N^{-1} \sum_{n=1}^{N} k^{\alpha}$$
$$C^{\alpha\beta} = N^{-1} \sum_{n=1}^{N} k^{\alpha} k^{\beta} - \overline{k^{\alpha}}\,\overline{k^{\beta}}$$
$$L(\underline{k}_n) \text{ - } distributed \left\{ k_i \right\}$$

The generator which produces set of k with distribution $L(\underline{k}_n)$ is based on following algorithm:

```
1) Given a vector $k_n$
2) Calculate $L(\underline{k}_n) = e^{-\sum_i \frac{\left|y_i^{teo}(n) - y_i^{measured}\right|^2}{2\sigma_i^2}}$
3) Find a new $\underline{k} = \underline{k}'$ from $\underline{k}$ with an ordinary random number generator
4) Calculate $L(\underline{k}')$
5) If $L(\underline{k}') \geq L(\underline{k}_n)$:                          $\underline{k}_{n+1} = \underline{k}'$   and   $L(\underline{k}_{n+1}) = L(\underline{k}')$
6) If $L(\underline{k}') < L(\underline{k}_n)$ but $L(\underline{k}') \geq p\, L(\underline{k}_n)$:   $\underline{k}_{n+1} = \underline{k}'$   and   $L(\underline{k}_{n+1}) = L(\underline{k}')$
7) If $L(\underline{k}') < L(\underline{k}_n)$:                          $\underline{k}_{n+1} = \underline{k}_n$   and   $L(\underline{k}_{n+1}) = L(\underline{k}_n)$
8) Go to 3)

In 6) p is a random number [0;1].
```

Table 2.13 $L(\underline{k}_n)$ *distributed generator*

Method B is despite its excellence only used for checking the results of method A. As we will see, method A gives some possibilities for varying the parameters in a way which is not straightforward in method B. So after checking mean, variance, correlation etc. for first data set, I will only use method A in the rest of the calculations.

2.4.2 Results

After a traditional calibration based on 16 world points with 16 corresponding measured image points, new synthetic image points are generated by multiplying the 16 world points with the achieved 3D-2D calibration matrix and subsequently adding Gaussian noise. The variance of the noise is set to the residual of the initial. For each new set of synthetic data, the transformation matrix and subsequently the corresponding pinhole parameters are found. The total number of data set is 2000 for Method A and the total number of Markov steps in Method B is 1.000.000.

Position 1 **(residual: 0.6 pixel/data point)**							
k	k^{exact}	$\bar{k}$	$\sqrt{\overline{k^2}-\bar{k}^2}$	k^{min}	k^{max}	$\bar{k}^{\,*}$	$\sqrt{\overline{k^2}-\bar{k}^2}^{\,*}$
F_x	7.68	8.31	1.09	4.45	11.74	8.06	1.25
F_y	416.34	415.24	1.33	410.68	420.25	416.01	1.33
F_z	-210.44	-216.14	6.09	-236.90	-194.35	-213.24	6.01
a	-9.04	-8.91	0.48	-10.78	-7.13	-9.46	0.57
b	-9.44	-9.50	0.52	-11.19	-7.73	-9.64	0.51
c	-2.36	-2.33	0.10	-2.65	-2.03	-2.43	0.12
aspect ratio	1.00734	1.00727	0.00209	1.00062	1.01447	1.00773	0.00233
focal length	805.88	801.05	5.48	782.21	820.72	803.01	5.26
skew	0.01005	0.00651	0.11861	-0.42060	0.39070	-0.00963	0.13088
U_0	319.74	320.28	7.29	295.51	343.52	322.66	7.08
V_0	354.66	356.34	6.67	329.55	380.28	348.93	7.96

Table 2.14 Variation analysis of pinhole parameters for calibration position 1. The values in the 2 last columns (marked with an asterisk) are found with method B.

The distribution of the parameters is shown below. On all graphs the horizontal resolution is (maximum-minimum)/32. The vertical values are divided by the horizontal resolution. This keeps the shape of the distribution unaltered while a direct comparison between related parameters is made possible.

External parameters:

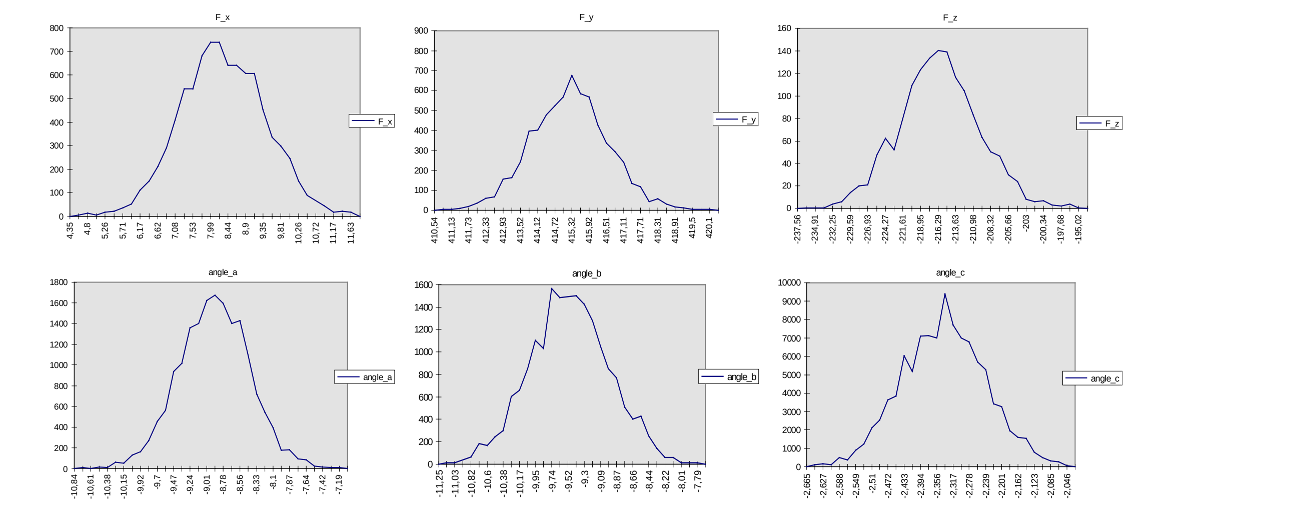


Fig 2-6 Graphical representation of stability of external parameters

Internal parameters:

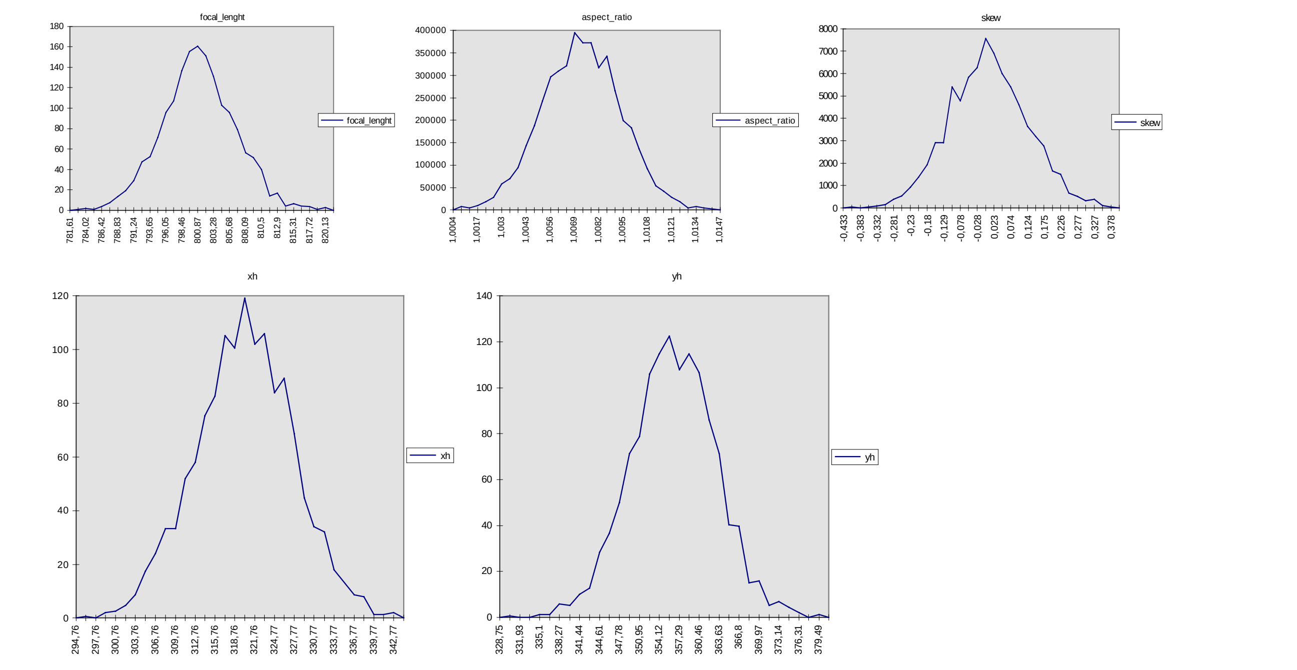


Fig 2-7 Graphical representation of stability of internal parameters

The distributions can also be compared relatively. All parameters have an approximate Gaussian distribution and the mean value is close to the optimum value. In order to show the accuracy of the parameters relative to each other, the distributions are presented on the same graph in Fig 2-8 and Fig 2-9. There is no point in putting all distributions in one graph, because the significance of the variance is different for different types of parameters. The variance also depends on the choice of units (for instance a variance of 1 degree in angles is generally worse than a variance of 1 mm in position).

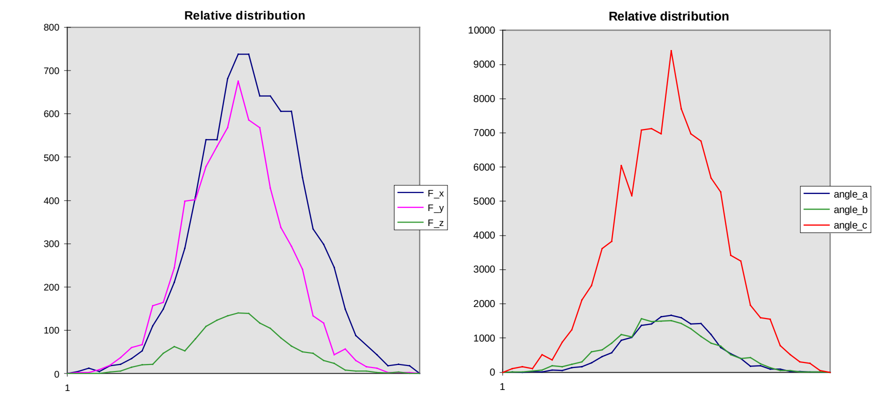


Fig 2-8 Graphical representation of relative stability of external parameters

Here we see that the z-position is 5 times more inaccurate than the x- and y-positions, and angle c 5 times more accurate than angle a and b. This is as intuitively expected since a translation towards or away from an object is much more difficult to detect than a translation keeping the distance fixed. In addition, a rotation around the z-axis is much easier to detect than around the other axes since you see the full effect of such a rotation. Rotation around the other axes will, for small angles, be difficult to distinguish from small translations and is much more difficult to recognise.

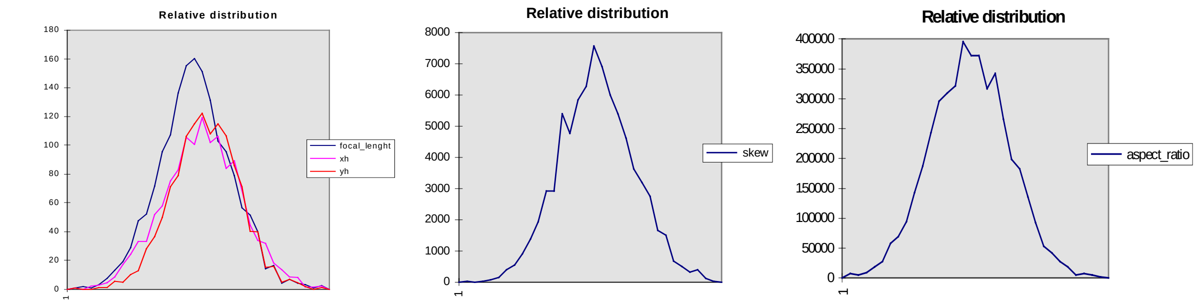


Fig 2-9 Graphical representation of relative stability of internal parameters

Aspect ratio and skewness are extremely well defined and surprisingly (U_0,V_0) is comparable with focal length (also measured in pixels) in accuracy. So why is (U_0,V_0) varying so much from one calibration position to another while for instance focal length is very stable? In order to answer that question we will start by looking at the correlations.

Correlations (Method A)

	F_x	F_y	F_z	a	b	c	asp.	f	skew	U_0	V_0
F_x	1.00	-0.36	-0.51	-0.14	0.05	-0.21	-0.05	-0.48	-0.09	0.01	-0.14
F_y	-0.36	1.00	0.72	0.03	0.18	-0.01	-0.12	0.72	0.05	-0.15	0.10
F_z	-0.51	0.72	1.00	0.26	0.23	0.12	-0.09	0.97	-0.05	-0.19	0.26
a	-0.14	0.03	0.26	1.00	0.16	0.62	-0.21	0.25	-0.11	-0.18	0.99
b	0.05	0.18	0.23	0.16	1.00	0.02	-0.10	0.30	0.00	-0.99	0.12
c	-0.21	-0.01	0.12	0.62	0.02	1.00	-0.08	0.10	0.45	-0.05	0.62
aspect ratio	-0.05	-0.12	-0.09	-0.21	-0.10	-0.08	1.00	-0.25	0.00	0.09	-0.22
focal length	-0.48	0.72	0.97	0.25	0.30	0.10	-0.25	1.00	-0.05	-0.26	0.24
skew	-0.09	0.05	-0.05	-0.11	0.00	0.45	0.00	-0.05	1.00	-0.01	-0.09
U_0	0.01	-0.15	-0.19	-0.18	-0.99	-0.05	0.09	-0.26	-0.01	1.00	-0.15
V_0	-0.14	0.10	0.26	0.99	0.12	0.62	-0.22	0.24	-0.09	-0.15	1.00

Table 2.15 Correlation between parameters in optimum for calibration position 1

Correlations (Method B)

	F_x	F_y	F_z	a	b	c	asp.	f	skew	U_0	V_0
F_x	1.00	-0.21	-0.41	-0.24	0.16	-0.28	-0.10	-0.37	-0.12	-0.06	-0.23
F_y	-0.21	1.00	0.63	-0.05	0.10	-0.03	-0.08	0.65	0.11	-0.06	0.02
F_z	-0.41	0.63	1.00	0.14	0.04	0.08	0.04	0.96	0.01	0.01	0.12
a	-0.24	-0.05	0.14	1.00	-0.05	0.66	-0.11	0.10	0.02	-0.01	0.99
b	0.16	0.10	0.04	-0.05	1.00	-0.17	-0.06	0.12	-0.11	-0.99	-0.08
c	-0.28	-0.03	0.08	0.66	-0.17	1.00	-0.03	0.04	0.51	0.11	0.66
aspect ratio	-0.10	-0.08	0.04	-0.11	-0.06	-0.03	1.00	-0.16	-0.03	0.05	-0.12
focal length	-0.37	0.65	0.96	0.10	0.12	0.04	-0.16	1.00	0.00	-0.07	0.09
skew	-0.12	0.11	0.01	0.02	-0.11	0.51	-0.03	0.00	1.00	0.10	0.04
U_0	-0.06	-0.06	0.01	-0.01	-0.99	0.11	0.05	-0.07	0.10	1.00	0.02
V_0	-0.23	0.02	0.12	0.99	-0.08	0.66	-0.12	0.09	0.04	0.02	1.00

Table 2.16 Correlation between parameters in optimum for calibration position 1

Both methods shows that in optimum, U_0 is completely correlated with angle b and V0 with angle a (and z with focal length). These results are very similar to the ones presented by Li and Lavest[xxx]. Does that mean that the extended pinhole model contains too many parameters and some of them could simply be expressed in terms of some others? No, not at all!

Let us check the variability of the internal parameters if the externals are allowed to move freely. New synthetic data are generated by keeping the internal parameters fixed, varying the external parameters freely, generating the 3D-2D transformation matrix, multiplying it with the 16 world points and finally add Gaussian noise to the synthetic image points. This is exactly the same method as used in first analysis except that in the first case no parameters were allowed to move at all, only Gaussian noise was added.

Alternating external parameters (residual: 0.6 pixel/data point)					
k	k^{exact}	$\bar{k}$	$\sqrt{\overline{k^2} - \bar{k}^2}$	k^{min}	k^{max}
aspect ratio	1.00734	1.00727	0.00241	0.95897	1.02201
focal length	805.88	805.95	2.17	793.50	825.89
skew	0.01005	0.00980	0.12501	-0.55540	1.29559
U_0	319.74	319.70	2.18	300.53	343.78
V_0	354.66	354.61	2.14	332.43	371.19

Table 2.17 Variation analysis with alternating external parameters

As in first the experiment the correlations can be calculated as well.

Correlation with alternating external parameters											
	F_x	F_y	F_z	a	b	c	asp.	f	skew	U_0	V_0
F_x	1.00	0.01	0.01	0.00	0.03	0.03	0.03	0.04	-0.05	-0.05	0.01
F_y	0.01	1.00	0.01	0.01	0.01	0.01	0.02	-0.04	0.04	0.02	0.01
F_z	0.01	0.01	1.00	0.01	0.01	0.01	-0.01	0.06	-0.04	-0.04	-0.01
a	0.00	0.01	0.01	1.00	-0.01	0.06	0.01	0.02	0.00	0.00	0.02
b	0.03	0.01	0.01	-0.01	1.00	-0.01	-0.01	0.05	-0.02	-0.04	0.00
c	0.03	0.01	0.01	0.06	-0.01	1.00	-0.05	0.01	0.02	-0.01	0.03
aspect ratio	0.03	0.02	-0.01	0.01	-0.01	-0.05	1.00	-0.31	-0.12	-0.11	-0.01
focal length	0.04	-0.04	0.06	0.02	0.05	0.01	-0.31	1.00	-0.04	-0.11	0.01
skew	-0.05	0.04	-0.04	0.00	-0.02	0.02	-0.12	-0.04	1.00	0.06	-0.01
U_0	-0.05	0.02	-0.04	0.00	-0.04	-0.01	-0.11	-0.11	0.06	1.00	0.11
V_0	0.01	0.01	-0.01	0.02	0.00	0.03	-0.01	0.01	-0.01	0.11	1.00

Table 2.18 Correlation with alternating external parameters

This removes all correlation and the accuracy of the 3 internal parameters previously with a correlated external partner increases significantly. This shows that although some parameters are correlated in the optimum, they certainly are not in general and all parameters in the extended pinhole model are needed. The central problem is that the optimum is very flat as illustrated in the following experimental test. Without changing the internal parameters, I now move the camera to another position and recalibrate with new world and image points.

Position 2 (Residual: 0.7 pixel/data point)					
k	k^{exact}	$\bar{k}$	$\sqrt{\overline{k^2}-\bar{k}^2}$	k^{min}	k^{max}
F_x	5.49	6.07	1.22	2.07	10.02
F_y	204.78	204.85	1.03	201.54	208.21
F_z	-210.67	-215.64	6.64	-236.16	-190.96
a	6.61	6.69	0.55	4.62	8.70
b	-9.56	-9.61	0.60	-11.42	-7.48
c	0.215	0.230	0.116	-0.138	0.589
aspect ratio	1.00549	1.00534	0.00253	0.99831	1.01311
focal length	804.21	799.93	6.00	780.68	821.63
skew	-0.27940	-0.28261	0.14413	-0.76120	0.17132
U_0	317.86	318.34	8.31	289.14	342.69
V_0	306.52	307.32	7.59	279.53	334.28

Table 2.19 Variation analysis for calibration position 2

A graphical representation of the variability of the parameters gives similar results as shown earlier and is omitted. We note that especially angle a and V0 is changed. That relationship is in accordance with what could be expected from the correlation table. The correlation table for this new position is also similar to the previous one and is shown in Table 2.20.

Correlation 2											
	F_x	F_y	F_z	a	b	c	asp.	f	skew	U_0	V_0
F_x	1.00	0.04	-0.52	0.04	0.06	0.01	-0.08	-0.49	0.01	0.00	0.00
F_y	0.04	1.00	-0.10	-0.27	-0.07	-0.17	0.06	-0.09	0.09	0.06	-0.18
F_z	-0.52	-0.10	1.00	0.00	0.19	-0.05	-0.02	0.97	-0.01	-0.15	0.08
a	0.04	-0.27	0.00	1.00	0.16	0.62	-0.33	-0.07	-0.07	-0.16	0.99
b	0.06	-0.07	0.19	0.16	1.00	0.04	-0.12	0.24	0.25	-0.99	0.19
c	0.01	-0.17	-0.05	0.62	0.04	1.00	-0.14	-0.12	0.44	-0.03	0.61
aspect ratio	-0.08	0.06	-0.02	-0.33	-0.12	-0.14	1.00	-0.16	-0.02	0.11	-0.32
focal length	-0.49	-0.09	0.97	-0.07	0.24	-0.12	-0.16	1.00	0.01	-0.20	0.00
skew	0.01	0.09	-0.01	-0.07	0.25	0.44	-0.02	0.01	1.00	-0.25	-0.05
U_0	0.00	0.06	-0.15	-0.16	-0.99	-0.03	0.11	-0.20	-0.25	1.00	-0.18
V_0	0.00	-0.18	0.08	0.99	0.19	0.61	-0.32	0.00	-0.05	-0.18	1.00

Table 2.20 Correlation between parameters in optimum for calibration position 2

So now, the situation is that for two different calibration positions we get different internal values. Especially U_0,V_0 are changing with changing external parameters, and that is reflected in high correlation in optimum for these parameters.

If we (incorrectly) assume that the internal parameters found in the first calibration were the correct ones and feed them into the pinhole model, we see from the table below that the residual is growing to near 50 pixels per data point (this is, not very surprising, more or less the same as the deviation in V0). But then allowing the external values to reorganise, and accepting these dictated internal values brings the residual down to only 1.6 pixels per data point (and a residual of 1.1 in total), which again tells us that the optimum in one single calibration is very flat or degenerated with respect to the parameters with high correlations.

Internal parameters fixed 1			
k	Pos 1	Pos 2	Pos 2 (opt)
F_x	7.68	5.49	5.46
F_y	416.34	204.78	205.47
F_z	-210.44	-210.67	-197.16
a	-9.04	6.61	9.842
b	-9.44	-9.56	-9.576
c	-2.36	0.215	0.563
aspect ratio	1.00734		
focal length	805.88		
skew	0.01005		
U_0	319.74		
V_0	354.66		
Residual	0.6	47.6	1.6

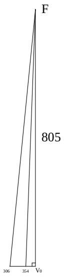
Table 2.21 Internal parameters found in position 1 used in both positions

Moreover, if we instead keep the internal values from calibration position 2 fixed we get another configuration in the flat minimum. This time the total residual is as low as 1.0.

Internal parameters fixed 2			
k	Pos 2	Pos 1	Pos 1 (opt)
F_x	5.49	7.68	6.99
F_y	204.78	416.34	415.78
F_z	-210.67	-210.44	-212.96
a	6.61	-9.04	-12.41
b	-9.56	-9.44	-9.22
c	0.215	-2.36	-2.99
aspect ratio	1.00549		
focal length	804.21		
skew	-0.27940		
U_0	317.86		
V_0	306.52		
Residual	0.7	47.9	1.2

Table 2.22 Internal parameters found in position 2 used in both positions

It now appears more clearly why the parameters are acting as they are. As is well known for small changes, rotation is nearly indistinguishable from translation and we see that when forcing V_0 to change from 354 to 306 (position 1) or vice versa (position 2), the external angle is compensat-



ing with a shift of approximately 3.3 degrees. Knowing the focal length approximately (805 pixels) this shift could have been estimated from $\alpha = atan\left(\frac{354 - 306}{805}\right) = 3.4^\circ$ (see drawing). Here I also assume that we are near the correct value of V_0 and therefore are having a near-rectangular triangle.

So small changes in an external angle cause big changes in U_0 or V_0; the estimation of the internal offset is ill-conditioned when finding it from only one calibration position. Obviously, the camera cannot have different internal parameters for different external positions, so the external values as well as the internal values in each calibration position must be wrong. This leaves us with 2 questions:

- Do we care at all?
- If yes, how can the correct values of the internal parameters (and thereby the external) be calculated?

In many cases the first question put up should be answered with a no: we don’t care whether or not we know the exact values of the external and internal parameters. As long as the total calibration matrix is sufficiently precise and producing the results we are looking for with sufficiently high accuracy, as shown in previous subchapter. When we are working with fixed camera systems it is not important (although interesting) to know the exact values. For instance in the application in B4 (see chapter 6), nearly all parameters are mixed, but the measuring method is nevertheless very accurate.

On the other hand, when working with moving cameras, for instance cameras mounted on a robot, it is essential to know all internal as well as external parameters exact. Otherwise, any movement (rotation) described by a matrix multiplication would give troubles. The problem of unknown exact parameters is similar when looking at a CAD model, with a virtual camera described in OpenGL (see next chapter).

2.4.3 A New Calibration Method

So if we really need the exact internal and external values, we have to find a better method for calibration and splitting. One obvious way is to calibrate the camera in different external positions and put on the constraint that the internal parameters have to be the same for all calibrations. It can be quite tricky to find the joint optimum but a robust method is described below.

1. Calibrate the camera in position 1 and 2. This produces $\mathbf{C}_1$ and $\mathbf{C}_2$
2. Split $\mathbf{C}_2$ into pinhole parameters
3. Use the internal parameters from $\mathbf{C}_2$ as fixed values for position 1 and find the (sub-) optimal set of external parameters. Put in the 11 parameters to produce $\mathbf{C}_1$'.
4. Generate $\mathbf{C}(\alpha) = (1-\alpha)\ \mathbf{C}_1 + \alpha\ \mathbf{C}_1'$ α: -1.0,-0.9...0.0,...1.0...,2.0
5. Split $\mathbf{C}(\alpha)$ into pinhole parameters.
6. Minimise the residual for position 1 by optimising external parameters
7. Minimise the residual for position 2 by optimising external parameters
8. Calculate total residual as an average between the residuals in both positions
9. Go to 4.

Table 2.23 Routine for finding optimal set of internal pinhole parameters

The value of α giving the lowest total residual is found and the corresponding internal and external parameters are stored. Note that α=0 gives Table 2.21 and α=1 gives Table 2.22. A result based on the method above and the previously used calibration positions is shown in the table below.

Joint calibration		
k	Pos 1	Pos 2
F_x	7.79	4.99
F_y	416.39	205.20
F_z	-211.67	-204.96
a	-10.895	8.080
b	-9.293	-9.585
c	-2.681	0.495
aspect ratio	1.007	
focal length	805.3	
skew	-0.15	
U_0	318.7	
V_0	328.4	
Residuals	0.78	0.91
Total	0.85	

Table 2.24 Joint calibration

We see a residual only slightly above the value found for the separate calibrations, again indicating a very flat optimum. But this time it is more reasonable to trust the accuracy of the single parameters since the internal parameters cannot adapt freely to just one position of the camera. This also forces the external parameters to assume more physically correct values although the variation between the 2 calibration positions is too small to be sufficient for a correct estimation of all parameters. In Fig 2-10 the value of U_0, V_0 and the residual as a function of α are shown.

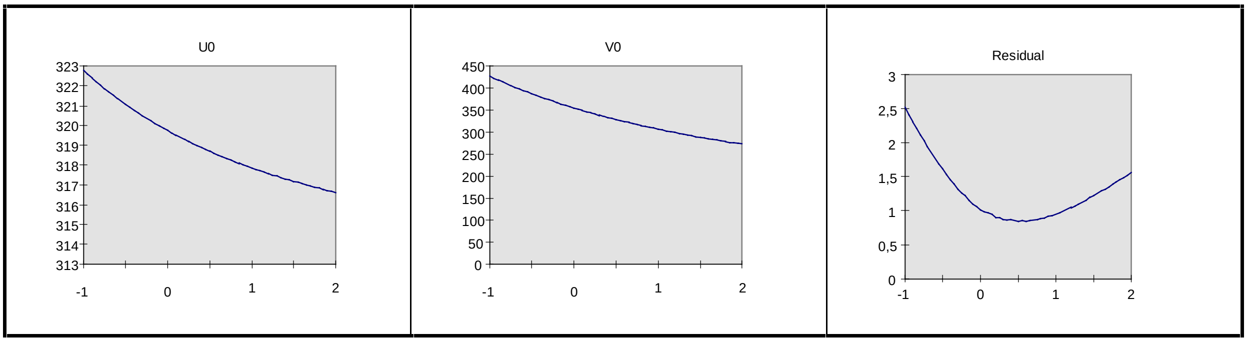


Fig 2-10 U_0, V_0 and the residual as a function of α

We see that for the joint calibration there exists a minimum in the residual different from the separate calibrations. On the other hand, it also seems as if we could expect some other values of U_0, V_0 if the two calibration positions differed more. Unfortunately, time and the capabilities of the calibration device do not allow for further tests. But it can be concluded that if we insist on working with exact parameters, these values can be found with a more comprehensive calibration procedure at the expense of a little increment in the residual and thereby a little less accuracy in the measurements [1].

2.5 Conclusion

Careful and thorough calibration is crucial for the success of any application, whether it is meant for quality measurements or for positioning. This chapter has taken us through the principles of calibrating with DLT. The Extended Pinhole Model is the linear model describing image mapping and is generally not sufficient if one wants to do high accuracy measurements. In that case, the non-linear contributions from the lens distortion have to be incorporated into the model. This however, does not make the pinhole model any less important when trying to understand the basic principles.

A new method for finding the pinhole parameters on the basis of a 3x4 transformation matrix has been presented. The method is simple and any parameters can be calculated by use of a simple desk calculator. An important spin-off from the derivations of the formulas is a deeper insight into how image mapping can be described by appropriate matrix multiplications.

The methods used in section 2.4 have revealed most interesting characteristics of the extended pinhole model and illustrated some of the problems that always crop up when calibrating a vision system for future high precision measurements. It has been shown how the optimum for the extended pinhole model is very flat, not owing to ill conditioned parameters, but owing to a high degree of correlation between internal and external parameters in that area.

A new method for finding better internal parameters has been described. The method is, owing to various constraints, not fully developed, but it has already shown a significant increase in the accuracy of the calibration matrices.

[1] I realise that these statements need to be confirmed by further research. However, the experiments done so far are all in line with the conclusions.

2.6 References

[i] S. K. Ghosh. *Analytical Photogrammetry*. Pergamon press, 1979 Bog, chapter 5.

[ii] Lauridsen O. Brande. *Fotogrammetri*. AUC,1988, pp. 8-11

[iii] Jacobi Ole, *Fotogrammetri. Matematisk grundlag.*1984

[iv] Roberts, L. G. *Machine perception of three-dimensional solids*.1963 J. Tippett et al. (ed.). Optical and Electro-Optical Information Processing, MIT Press, Cambridge, MA, pp. 159-197

[v] Duda, R. O. and Hart, P. E. *Pattern Classification and Scene Analysis*.1973 Chapter 10, Wiley & sons

[vi] Tsai, R. Y., *An efficient and accurate camera calibration technique for 3D machine vision*, 1986, CH2290-5/86 IEEE

[vii] Dueholm, Keld, Institute of Fotogrammetry, *Lecture notes* 1994

[viii] Ganapathy, S. *Decomposition of transformation matrices for robot vision*, 1984, 0167-8655/84, Elsevier

[ix] Strat, Thomas M. , *Recovering the camera parameters from a transformation matrix,* 1984. Proc. DARPA Image Understanding Workshop, New Orleans, LA 264-271

[x] Faugeras, O. D. and Toscani, G. *The calibration problem stereo.* 1986. Proc. IEEE Conf. on Computer Vision and Pattern recognition, Miami Beach, FL, USA. pp. 15-20

[xi] Shih, T. Y. and Faig, W. *Physical interpretation of the extended DLT model.* 1987. Proceedings of ASPRS Fall Convention, American Society for Photogrammetry and Remote Sensing, Reno, Nevada, pp. 385-394

[xii] Faugeras, O. D. and Toscani, G. *Camera calibration for 3D computer vision*. 1987. Proc. Int. Workshop on Industrial Application of Machine Vision and Machine Intelligence, Silken, Japan, pp. 281-305

[xiii] Melen Trond. *Geometrical modelling and calibration of video cameras for underwater navigation.* 1994. NTH Norway. ITK-rapport 1994:103-W

[xiv] Puget P. & Skordas T. *An optimal solution for mobile camera calibration.* Faugeras O.(Ed.) , *Lecture Notes in Computer Science*, 1990, First European Conference on Computer Vision

[xv] M. Li and J-M *Lavest Some aspects of zoom-lens camera calibration* February 1995, CVAP172

[xvi] Wilson Reg. G. and Schafer Steven A *What is the centre of the image? April 1993 CMU-CS-93-122*

[xvii] Wang L.-L. and Tsai W.-H. *Computing camera parameters...* , 1990 Machine Vision and Applications, 3(3):129-141

[xviii] Seetharaman G. et al. *Calibration of camera parameters using vanishing points.* 1994. Journal of the Franklin Institute, Elsevier Vol. 331B, No 5 pp. 555-585

[xix] Vernon, D. , 1991 *Machine Vision*. ISBN 0-13-543398

[xx] Ballard , D. H. , Brown C. M. *Computer Vision*. 1982. ISBN 0-13-165316-4, pp 477-487

[xxi] Immerkaer J. , *Algorithms to Determine Position ...* 1991. Master thesis Odense University Printing

[xxii] Hansen, Per B. , June *Householder reduction of linear equations* 1992. ACM Computing Surveys, Vol 24, No 2

[xxiii] Dueholm K. , *Image aquisition by camera II: Geometry*, 1996. Course 8817 note IMM DTU

[xxiv] Sarabi, Hamid , *Efficient methods for finding ...* 1993. Master thesis Odense University Printing

[xxv] Press W.H. , Flannery B.P. , Teukolsky S.A. , Vetterling W.T. 1986 *Numerical recipes* chapter 14.5

[xxvi] Press W.H. , Flannery B.P. , Teukolsky S.A. , Vetterling W.T. 1986 *Numerical recipes* chapter 7.2

[xxvii] Tarantola A. *Inverse Problem Theory* chapter 3. 1987

[xxviii] http://ic-www.arc.nasa.gov/ic/projects/bayes-group/group/html/bayes-theorem-long.html

[xxix] http://www.sst.ph.ic.ac.uk/angus/Lectures/compphys/node84.html

[xxx] M. Li and J-M *Lavest Some aspects of zoom-lens camera calibration* February 1995, CVAP172

3. Camera and CAD

3.1 Introduction

One of the biggest challenges in industrial vision is the great demand for flexibility in the applications. Either you put strict requirements on the position of the object (typically, electrical components at an electronic factory or perhaps bottles for inspection at a brewery etc.) or you are doing some kind of marking of the object (with possibly some special light like stroboscope). None of these aids can be applied at OSS.

At OSS all parts of the ship are described in CAD before production. That means that any part one wishes to search for by camera has an ideal description in the CAD system. Moreover it is generally the case that one knows roughly the position of the camera relative to the searched object.

The idea now is to produce templates from CAD and feed the vision system with them in order to extract specific measures. In the case of 3D localisation, it should be possible to develop methods that avoid the difficult classical correspondence problem. One method could be an iterative one, where the virtual camera is moved around until the virtual image matches the real image optimally. Another more efficient and direct method can be applied if it is possible to identify corresponding features despite some differences due to translation and rotation and if the features found in the CAD image can be identified by their 3D CAD coordinates. If the same features are uniquely identified in the real image, we are left with a classical pose estimation problem where the "world" coordinates X,Y,Z are the CAD values and the image coordinates U,V are the positions of the features in the real image.

The matching between real images and virtual images can be made in two ways. Either the edges in the real images are extracted and compared with the wireframe of the CAD model, or the real image is compared to a surface description of the CAD model on pixel basis. In both cases, an ideal tool for the generation of virtual images is OpenGL[i]:

- OpenGL makes it possible to user specify the 9 most important parameters in the (linear) transformation matrix (which is delivered from a previous calibration routine). Only skewness and aspect ratio are not straightforward to implement[1]. OpenGL produces a very realistic image of the CAD file instead of just an ideal projection often seen in proprietary CAD simulation systems.
- OpenGL is fast in geometrical operations like clipping and panning. That is very useful when moving a virtual camera around in a neighbourhood, grabbing many images.
- In OpenGL you can control light, colours, textures etc. which is essential when real images are matched with computer generated images.
- OpenGL is invented by Silicon Graphics and now supported by HP and IBM as well. Most important however is that Microsoft NT is supporting it and like it or not the evolution of industrial application goes in the direction of Windows NT.

[1] As shown in the previous chapter, it is possible to post-process skewness and aspect ratio to the image. The gain however is very limited since skewness is always very close to 0 degrees and aspect ratio close to 1. This post-processing is therefore omitted.

3.2 Implementation

Chapter 2 gives us the necessary instruments for taking synthetic images with a virtual camera that has correct internal parameters (lens size and image centre) and correct external parameters (position and orientation). In the following, the most important functions for describing a virtual camera are presented.

3.2.1 Defining the internal parameters

In OpenGL the internal parameters of a camera are described by defining the viewing volume of the camera (see Fig 3-1).

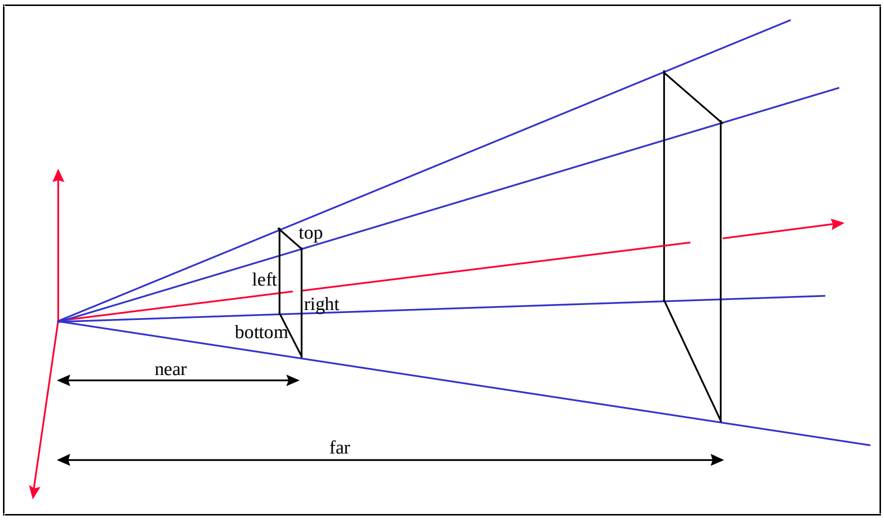


Fig 3-1 The perspective viewing volume specified by glFrustum

The viewing volume of the camera is defined by use of the function **glFrustum**:

```
glFrustum(3G)                  OpenGL Reference                  glFrustum(3G)

NAME
     glFrustum - multiply the current matrix by a perspective matrix

C SPECIFICATION
     void glFrustum( GLdouble left,
                     GLdouble right,
                     GLdouble bottom,
                     GLdouble top,
                     GLdouble near,
                     GLdouble far )

PARAMETERS
     left, right Specify the coordinates for the left and right vertical
                 clipping planes.

     bottom, top Specify the coordinates for the bottom and top horizontal
                 clipping planes.

     near, far   Specify the distances to the near and far depth clipping
                 planes.  Both distances must be positive.

DESCRIPTION
     glFrustum describes a perspective matrix that produces a perspective
     projection.  (left, bottom, -near) and (right, top,  -near) specify the
     points on the near clipping plane that are mapped to the lower left and
     upper right corners of the window, respectively, assuming that the eye is
     located at (0, 0, 0).  -far specifies the location of the far clipping
     plane.  Both near and far must be positive.  The corresponding matrix is
```

$$\begin{bmatrix} \frac{2\cdot near}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2\cdot near}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & \frac{-\ far+near}{far-near} & \frac{-2\cdot\ far-near}{far-near} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

```
The current matrix is multiplied by this matrix with the result replacing
     the current matrix.  That is, if M is the current matrix and F is the
     frustum perspective matrix, then M is replaced with M o F.

     Use glPushMatrix and glPopMatrix to save and restore the current matrix
     stack.

NOTES
     Depth buffer precision is affected by the values specified for near and
     far.  The greater the ratio of far to near is, the less effective the
     depth buffer will be at distinguishing between surfaces that are near
     each other.  If
```

$$r=\frac{far}{near}$$

```
     roughly log r bits of depth buffer precision are lost.  Because r
     approaches infinity as near approaches zero, near must never be set to
     zero.

ERRORS
     GL_INVALID_VALUE is generated if near or far is not positive.

     GL_INVALID_OPERATION is generated if glFrustum is executed between the
     execution of glBegin and the corresponding execution of glEnd.

ASSOCIATED GETS
     glGet with argument GL_MATRIX_MODE
     glGet with argument GL_MODELVIEW_MATRIX
     glGet with argument GL_PROJECTION_MATRIX
     glGet with argument GL_TEXTURE_MATRIX
```

Table 3.1 glFrustum manual

The parameters in the function glFrustum can be calculated from the extended pinhole model:

left	= -V_0
right	= 768-V_0
bottom	= -U_0
top	= 576-U_0
near	= focal_ length
far	= “huge”

Table 3.2 Pinhole values in glFrustum

Far = “huge” just means that the camera has to look sufficiently far (and not much farther owing to the depth buffer precision).

3.2.2 Defining the external parameters

The external parameters are described in OpenGL with the Utility Library routine **gluLookAt**:

```
gluLookAt(3G)                OpenGL Reference                 gluLookAt(3G)

NAME
     gluLookAt - define a viewing transformation

C SPECIFICATION
     void gluLookAt( GLdouble eyex,
                     GLdouble eyey,
                     GLdouble eyez,
                     GLdouble centerx,
                     GLdouble centery,
                     GLdouble centerz,
                     GLdouble upx,
                     GLdouble upy,
                     GLdouble upz )

PARAMETERS
     eyex, eyey, eyez                Specifies the position of the eye point.
     centerx, centery, centerz       Specifies the position of the reference point.
     upx, upy, upz                   Specifies the direction of the up vector.

DESCRIPTION
     gluLookAt creates a viewing matrix derived from an eye point, a reference
     point indicating the center of the scene, and an up vector.  The matrix
     maps the reference point to the negative z axis and the eye point to the
     origin, so that, when a typical projection matrix is used, the center of
     the scene maps to the center of the viewport.  Similarly, the direction
     described by the up vector projected onto the viewing plane is mapped to
     the positive y axis so that it points upward in the viewport.  The up
     vector must not be parallel to the line of sight from the eye to the
     reference point.

     The matrix generated by gluLookAt postmultiplies the current matrix.

SEE ALSO
     glFrustum, gluPerspective
```

Table 3.3 gluLookAt manual

The parameters in the function gluLookAt can also be calculated from the pinhole model:

eyex	= focal_center.x
eyey	= focal_center.y
eyez	= focal_center.z
centerx	= focal_center.x + scale*$M_{2,0}$
centery	= focal_center.y + scale*$M_{2,1}$
centerz	= focal_center.z + scale*$M_{2,2}$
upx	= $M'_{0,0}$
upy	= $M'_{1,0}$
upz	= $M'_{2,0}$

Table 3.4 Pinhole parameters in gluLookAt

The first three parameters in gluLookAt are equal to the Focal Centre found with EQ. 2.93. The next three parameters are describing a point on Focal Line. Such a point can be found by use of the same arguments as used on page 20, or by use of the argument described by Strat[ix]. He points out that the first three elements in the last row of the 3x4 transformation matrix is a vector parallel with the focal line. The scale factor is introduced for numerical reasons since the last row in the calibration matrix normally contains very small numbers. Scaling does of course not affect the orientation of the vector. The last 3 parameters in gluLookAt describe the up-vector of the camera. As also described on page 20, this vector is equal to the first row in the inverse of matrix **M** (EQ. 2.85).

3.2.3 Defining the dimensions of the image

The last needed GL routine is the one that describes the size of the image. The size is of course chosen to be the same as the size of the real image (or the half of it). The routine for defining the image on the screen is **glViewPort**:

```
glViewport(3G)                 OpenGL Reference                 glViewport(3G)

NAME
     glViewport - set the viewport

C SPECIFICATION
     void glViewport( GLint x,
                      GLint y,
                      GLsizei width,
                      GLsizei height )
PARAMETERS
     x, y Specify the lower left corner of the viewport rectangle, in pixels.
          The default is (0,0).

     width, height
          Specify the width and height, respectively, of the viewport.  When a
          GL context is first attached to a window, width and height are set
          to the dimensions of that window.
DESCRIPTION
     glViewport specifies the affine transformation of x and y from normalized
     device coordinates to window coordinates.  Let (x  , y  ) be normalized
```

device coordinates. Then the window coordinates $\left(\frac{x}{w},\frac{y}{w}\right)$ are computed as

follows:

$$\frac{x}{w}=\left(\frac{x+1}{w}\right)\left(\frac{width}{2}\right)+x$$

$$\frac{y}{w}=\left(\frac{y+1}{w}\right)\left(\frac{height}{2}\right)+y$$

```
     Viewport width and height are silently clamped to a range that depends on
     the implementation.  This range is queried by calling glGet with argument
     GL_MAX_VIEWPORT_DIMS.

ERRORS
     GL_INVALID_VALUE is generated if either width or height is negative.

     GL_INVALID_OPERATION is generated if glViewport is executed between the
     execution of glBegin and the corresponding execution of glEnd.

ASSOCIATED GETS
     glGet with argument GL_VIEWPORT
     glGet with argument GL_MAX_VIEWPORT_DIMS
```

Table 3.5 glViewPort manual

The parameters in the function glViewPort are simply given by:

x	= 0
y	= 0
width	= w (768)
height	= h (576)

Table 3.6 Pinhole values for glViewPort

Width and height are actually controlled via the window manager, which then is asked to produce a window(w,h) with (w,h) equal to (768,576).

3.3 Results

We are now ready to make our first comparison between a real image and a similar synthetic image. In order to avoid as many error sources as possible, I start with images of the test calibration bench situated at the office (see Fig 6-6). Here I have fully controllable conditions and the result has to be close to exact match. Two images used for one 3D calibration are shown in Fig 3-2. In the 3rd image, the two first images are added together and the resulting image has been histogram equalised in order to enhance the templates.

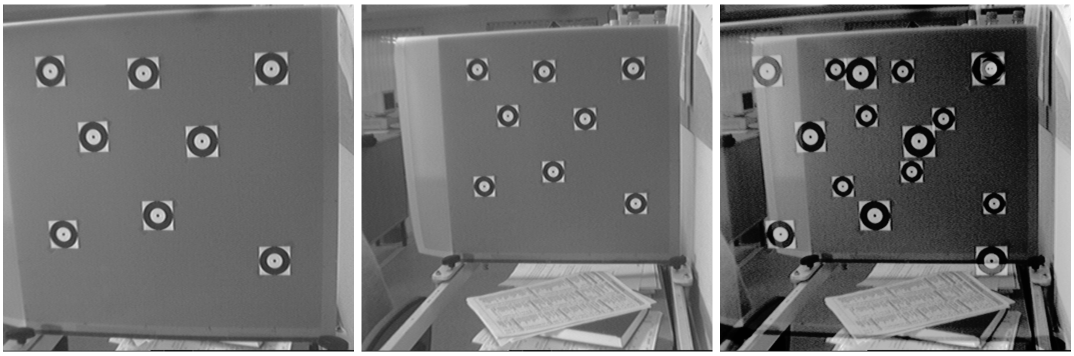


Fig 3-2 Images taken from the calibration test bench

The plane containing the circular templates is translated an exact known distance between the two images thereby giving the needed 3D information. The calibration based on Fig 3-2 produced the data shown in Table 2.7 (Pos 1) and the pinhole parameters shown in Table 2.9 (C_1).

For these data, the parameters in the function glFrustum can be calculated to (see Table 3.2 & Table 3.7):

left	= -354.7
right	= 413.3
bottom	= -319.7
top	= 256.3
near	= 805.9
far	= “huge”

Table 3.7 Actual pinhole values in glFrustum

The parameters in the function gluLookAt can be calculated to (see Table 3.4 & Table 3.8):

eyex	=	7.7
eyey	=	416.3
eyez	=	-210.4
centerx	=	7.7 - 0.162
centery	=	416.3 + 0.186
centerz	=	-210.4 + 0.969
upx	=	-0.986
upy	=	0.013
upz	=	-0.167

Table 3.8 Actual pinhole parameters in gluLookAt

Here, the $M_{2,0-2}$ and the $M'_{0-2,0}$ from Table 3.4 have been scaled with their respective normalisation factor. Putting in all these parameters into the OpenGL program and constructing[1] templates with same 3D coordinates as specified in Table 2.7 produces the image shown in Fig 3-3.

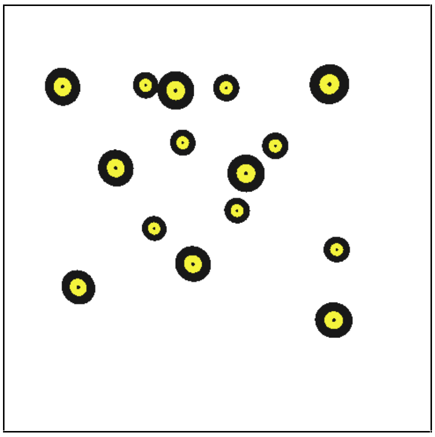


Fig 3-3 Synthetic image of calibration grid.

In the program, all templates in both distances were created simultaneously, something, which is not possible in real life with the kind of calibration equipment I used. However, it is now possible to do a direct match between Fig 3-3 and the last image in Fig 3-2. The result is shown in Fig 3-4 where the two images are added together.

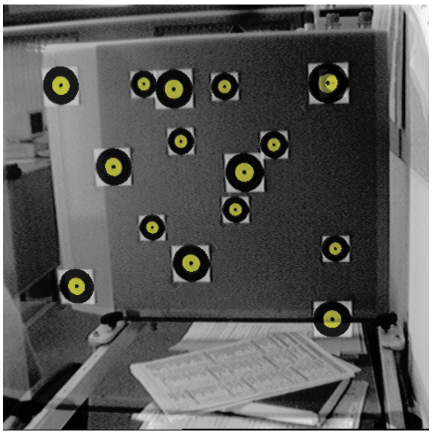


Fig 3-4 Real image added to synthetic image.

Fig 3-4 shows that the method gives a very fine accordance between real images and computer generated images, at least for scenarios where all parameters are well known. Another way of

[1] Constructing templates is done the same way as any CAD model is constructed. The format is called promos#1 and is proprietary to OSS. It includes possibilities for describing any kind of geometrical feature (point, line, arc, polygon, spline, NURBS ... For the templates, only features used were Plate and Face.

checking the result is to use the same routine, which originally found the position of the templates in the real image(s), to find the templates in the synthetic image with subpixel accuracy. The result is shown in Table 3.9

<table>
<tr><th colspan="2">Measured</th><th colspan="2">Virtual</th><th colspan="4">Difference</th></tr>
<tr><td>187,1</td><td>159,0</td><td>185,6</td><td>158,5</td><td colspan="2">1,5</td><td colspan="2">0,5</td></tr>
<tr><td>190,1</td><td>287,2</td><td>189,7</td><td>287,0</td><td colspan="2">0,4</td><td colspan="2">0,2</td></tr>
<tr><td>183,8</td><td>463,8</td><td>182,6</td><td>462,4</td><td colspan="2">1,2</td><td colspan="2">1,4</td></tr>
<tr><td>278,1</td><td>218,2</td><td>277,8</td><td>219,4</td><td colspan="2">0,3</td><td colspan="2">-1,2</td></tr>
<tr><td>284,1</td><td>368,1</td><td>283,8</td><td>367,5</td><td colspan="2">0,3</td><td colspan="2">0,6</td></tr>
<tr><td>387,1</td><td>307,1</td><td>386,4</td><td>307,1</td><td colspan="2">0,7</td><td colspan="2">0,0</td></tr>
<tr><td>413,1</td><td>177,1</td><td>412,0</td><td>176,0</td><td colspan="2">1,1</td><td colspan="2">1,1</td></tr>
<tr><td>449,9</td><td>468,1</td><td>450,2</td><td>467,4</td><td colspan="2">-0,3</td><td colspan="2">0,7</td></tr>
<tr><td>184,8</td><td>253,2</td><td>184,0</td><td>253,5</td><td colspan="2">0,8</td><td colspan="2">-0,3</td></tr>
<tr><td>187,2</td><td>345,8</td><td>186,9</td><td>345,0</td><td colspan="2">0,3</td><td colspan="2">0,8</td></tr>
<tr><td>249,1</td><td>295,3</td><td>249,0</td><td>295,6</td><td colspan="2">0,1</td><td colspan="2">-0,3</td></tr>
<tr><td>253,1</td><td>401,9</td><td>252,7</td><td>400,7</td><td colspan="2">0,4</td><td colspan="2">1,2</td></tr>
<tr><td>326,1</td><td>358,0</td><td>325,9</td><td>357,2</td><td colspan="2">0,2</td><td colspan="2">0,8</td></tr>
<tr><td>346,9</td><td>263,0</td><td>346,3</td><td>263,1</td><td colspan="2">0,6</td><td colspan="2">-0,1</td></tr>
<tr><td>370,8</td><td>471,9</td><td>370,1</td><td>470,6</td><td colspan="2">0,7</td><td colspan="2">1,3</td></tr>
<tr><td>mean</td><td>mean</td><td>mean</td><td>mean</td><td>mean</td><td>abs</td><td>mean</td><td>abs</td></tr>
<tr><td>286,1</td><td>322,5</td><td>285,6</td><td>322,1</td><td>0,55</td><td>0.59</td><td>0,44</td><td>0.70</td></tr>
</table>

Table 3.9 Comparison between real and synthetic templates

We see that the mean values (0.55 and 0.44) are significantly different from 0. This is not a result of the neglected parameters (skewness and aspect ratio) in the synthetic image, because it gives approximately the same result if the virtual image points are compared to image points generated from a 3x4 transformation matrix with the same constraints on skewness and aspect ratio (and with no compensation for non-linear distortion). But, they are all very small deviations with a total average difference less than 1 pixel, and generally we see a very fine correspondence between the virtual and the real image, so for the present I will not worry about these minor perturbations.

The next step is to move into production-like environments. In the welding lab of OSS, an 11 axes robot and a 3x4x2 metre mock-up was build for testing purposes during the Esprit project Cleopatra[1] and, as always, a complete CAD model of the block was constructed as well. The robot manipulator and the mock-up are shown in Fig 3-5 and Fig 3-6 respectively.

[1] Cleopatra was partly a robot controller project and partly a vision project where OSS worked closed together with Thomson (F) with regard to vision implementations.

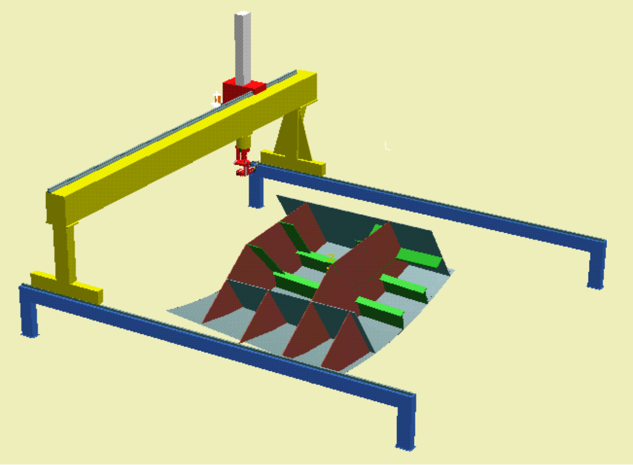


Fig 3-5 The 11 axes robot manipulator

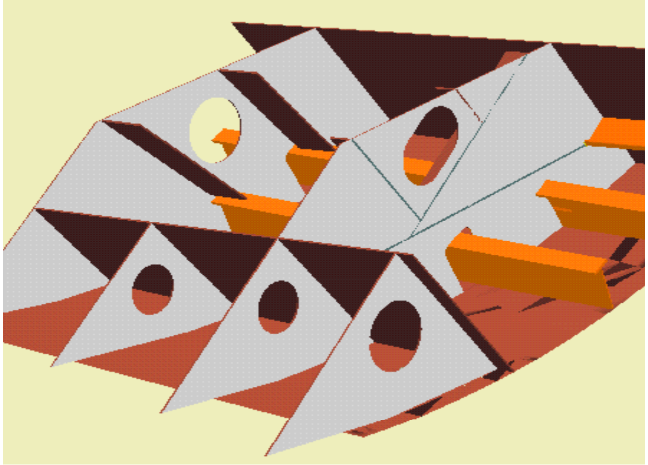


Fig 3-6 CAD model of mock-up

A camera was mounted on the 6th axis of the manipulator and a series of images were taken from different positions (Fig 3-7).

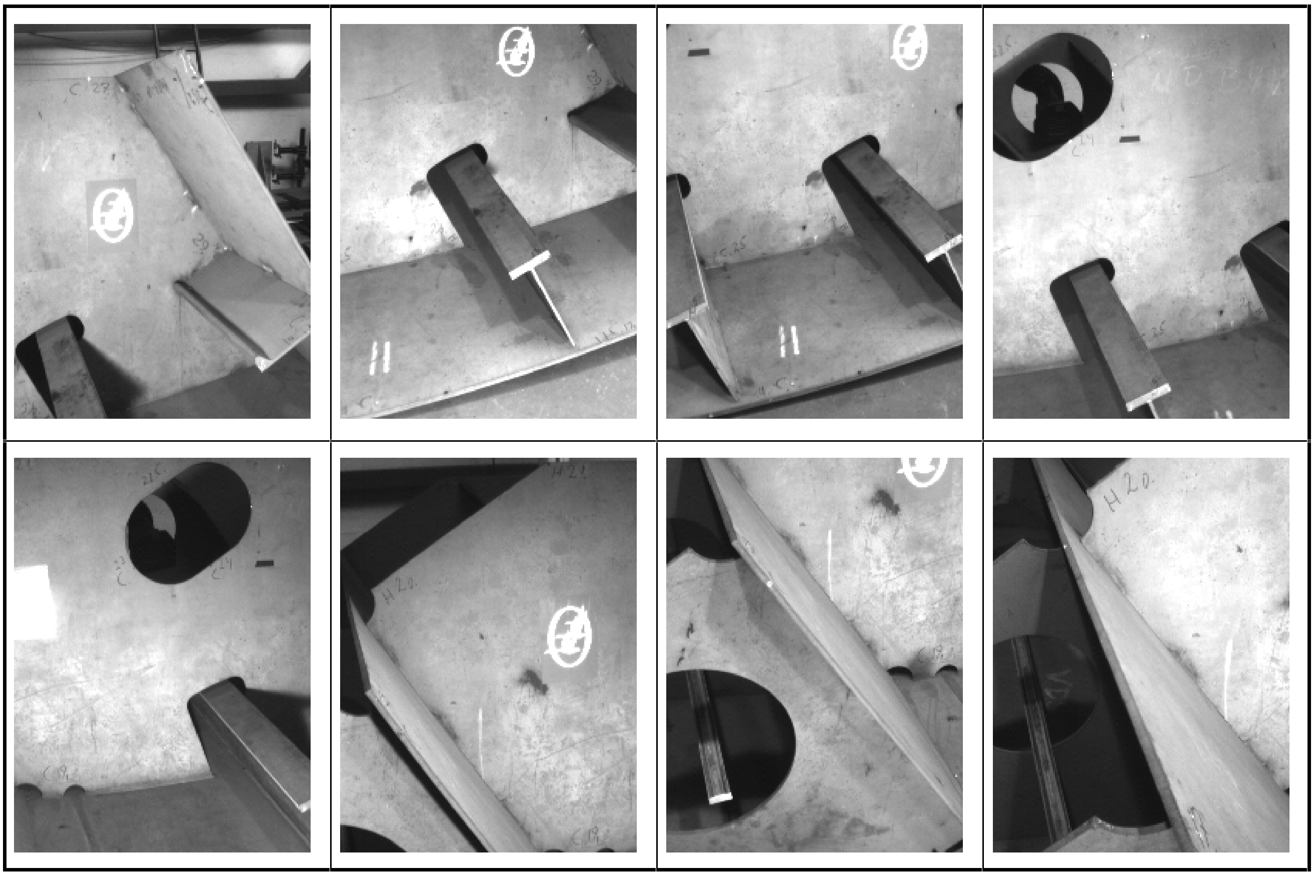
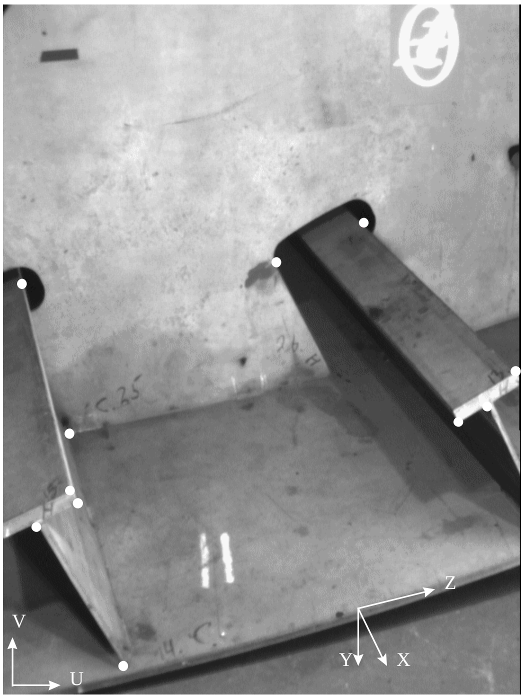


Fig 3-7 Images of the mock-up.

One image was picked out and 11 points in the image could be identified by their pixel coordinates (white spots in Fig 3-8) and respective 3D CAD coordinates (see Table 3.10).



X	Y	Z	U	V
41376	20322	13349	134	34
41376	19932	13448	37	188
41376	19947	13516	82	212
41376	19923	13521	75	225
40490	20221	13370	75	290
41376	20313	14158	506	303
41376	20320	14228	538	320
41376	20303	14310	571	359
40490	20192	14085	305	472
40490	20223	14324	407	514
40490	19822	13532	19	458

Table 3.10 The mock-up calibration data.

Fig 3-8 Selected image with identified calibration points

The 3x4 transformation matrix was constructed with a residual of 4.4 pixels. The pinhole parameters have been extracted and are shown in Table 3.11.

Pinhole parameters for mock-up											
X	Y	Z	a	b	c	asp	focal	skew	U_0	V_0	Residual
42679	19263	13916	91.25	49.30	56.92	0.994	913.4	5.01	420.5	207.1	4.40

Table 3.11 Pinhole parameters constructed on basis of mock-up data

Especially the value for skewness indicates that the parameters should be taken with a grain of salt. But, as seen in chapter 2, wrong internal parameters can be compensated a good deal by a (wrong) change in the external parameters, resulting in a nice-looking final image. When putting in the extracted parameters for the CAD program, one must remember that the program works with no skewness and equal scale on the axes. That is clearly not the case for these pinhole parameters. In Fig 3-9, the result is presented together with the real image for comparison.

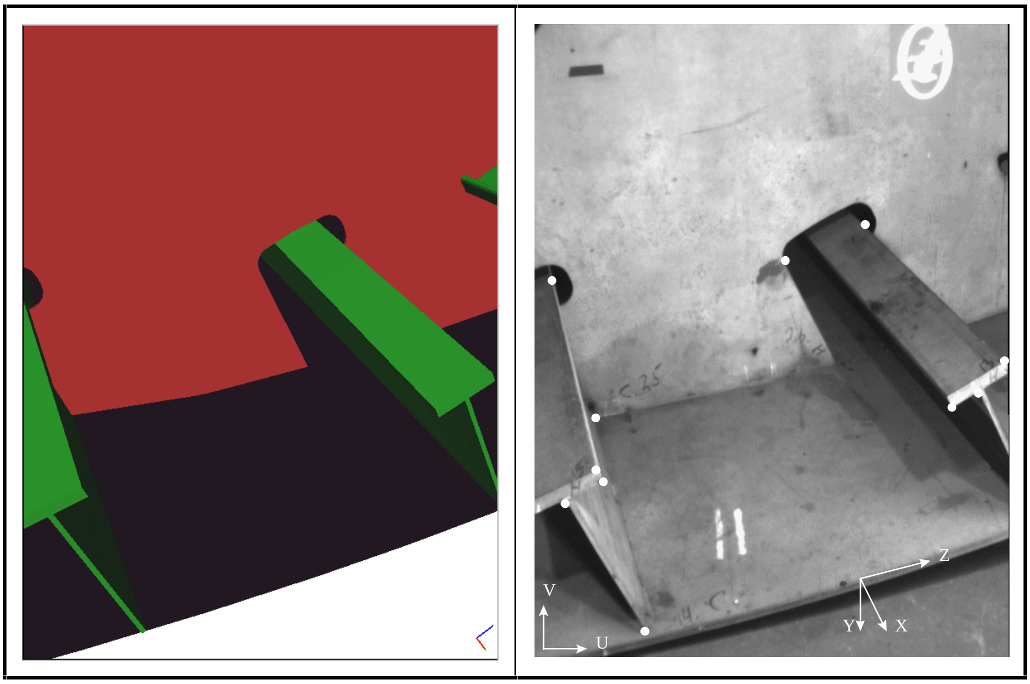


Fig 3-9 Virtual and real mock-up image based on pure mock-up calibration

The module for reading in the CAD model has been provided by the Promos group[1]. At a first glance, the result is quite impressive when remembering the sparse number of calibration points, but adding the wire-frame model to the real image some of the major errors in the virtual image are revealed (Fig 3-10).

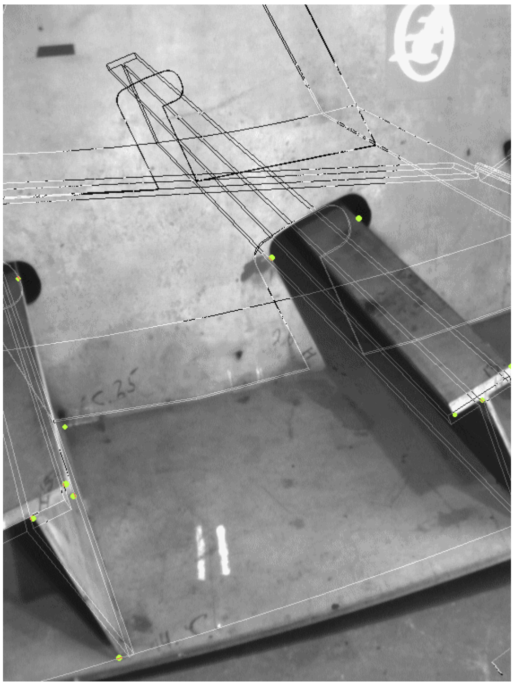


Fig 3-10 Wire-frame and real image

Due to the missing skewness “correction” in the synthetic image, the residual between the virtual model and the real image calculated at the 11 calibration points is bigger than the calculated residual of 4.4. There is currently no routine implemented for hidden line removal, but the routine is prepared and the integration will be done in near future. The non-parallelism between the long straight lines in the image indicates that the estimation of the focal length (and thereby the correlated distance from object to camera) is wrong. Nevertheless for a first trial, the result is promising and the method seems to be quite robust.

In order to get a better match, I then de-mounted the camera from the robot manipulator without touching the lens and brought it back to the office for a more accurate calibration with the test bench. The internal parameters, found by use of the test bench, were used to find the new corresponding external parameters giving minimal residual under the constraint of keeping the internal parameters fixed. The

[1] The Promos group at OSS is a team working on the creation of a complete product model of the ship including process data and 3D geometry (which I used for this specific purpose)

new external values were found by use of simulated annealing although probably more direct methods also would be applicable and certainly a lot faster. The new pinhole parameters are shown in Table 3.12.

Corrected pinhole parameters for mock-up											
X	Y	Z	a	b	c	asp	focal	skew	U_0	V_0	Residual
43122	19106	14021	88.01	65.58	55.54	1.026	1185	0.08	299.0	416.5	9.51

Table 3.12 Corrected pinhole parameters

We note that all parameters have changed significantly and with these new values, a new CAD image can be generated. The result is shown in Fig 3-11 with the non-corrected result shown in Fig 3-12 for comparison.

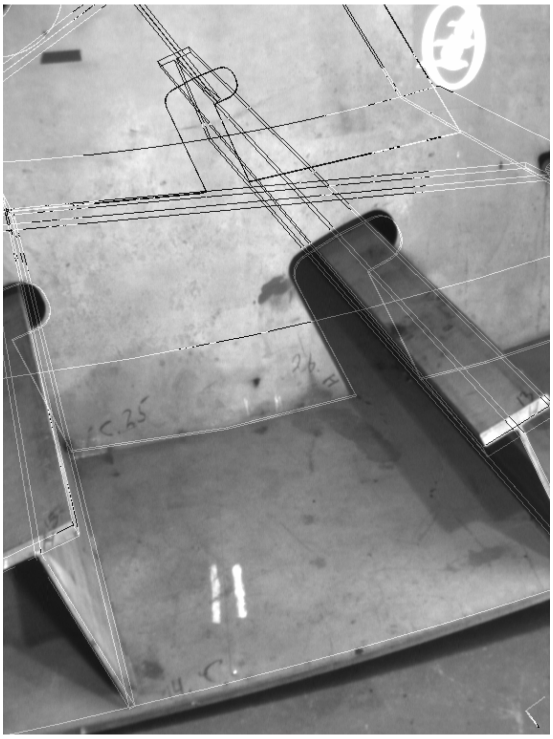


Fig 3-11 Corrected internal parameters

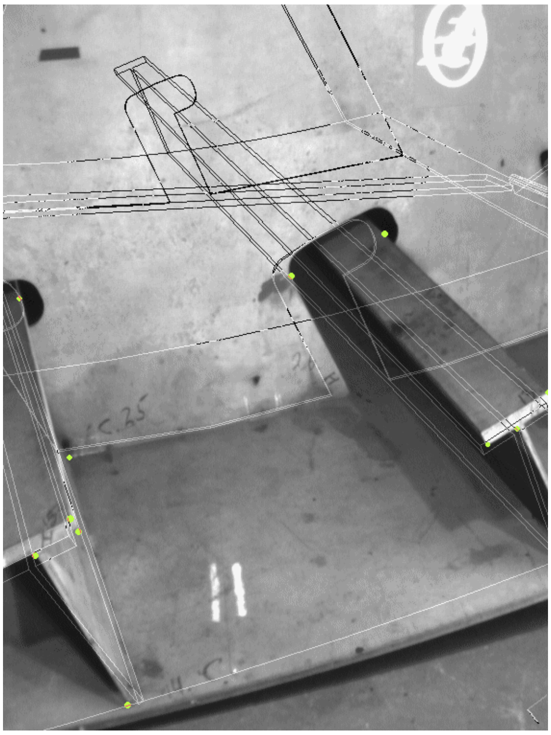


Fig 3-12 Mock-up based internal parameters

The result is significantly improved. All lines in the CAD model are lying very nicely along the lines in the image. If the comparison is to be done between wire-frame models, the lines in the image can be extracted with many different operators. The success rate is quite high since it is not so important if the operator finds too many lines as long as it finds all the physically correct ones as well. Another example also illustrating the robustness of the method is given below.

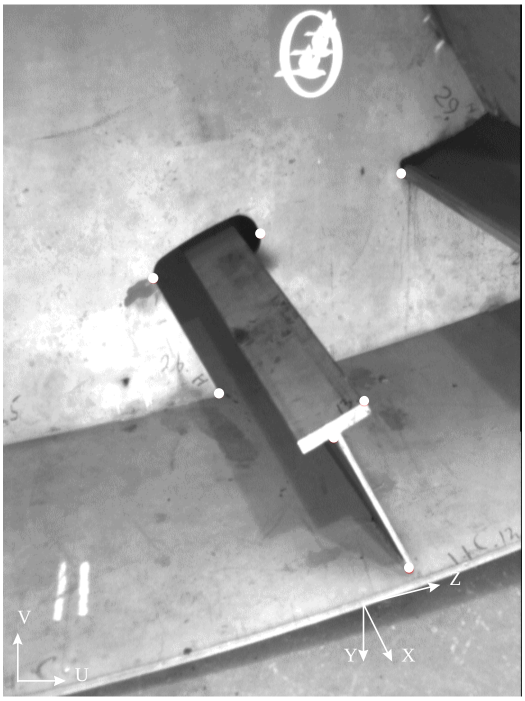


Fig 3-13 Another image with only 7 calibration points

X	Y	Z	U	V
41376	20719	14188	452	139
41376	20319	14227	367	287
41376	20302	14310	401	327
40490	20570	14035	241	334
40490	20191	14085	168	458
40480	20277	14365	285	512
40490	20389	14770	443	579

Table 3.13 The calibration points

A direct calibration using only the 7 points indicated on Table 3.13 is shown in Table 3.14, together with a calibration where the test-bench produced internal parameters are used.

Pinhole parameters for mock-up												
	X	Y	Z	a	b	c	asp	focal	skew	U_0	V_0	Residual
Direct	42838	19307	14260	104.03	54.26	42.49	1.053	1020	4.16	502.6	396.3	0.82
Optimal	43109	19197	14113	97.92	63.47	46.50	1.026	1185	0.08	299.0	416.5	11.60

Table 3.14 Pinhole parameters for Table 3.13.

A residual of only 0.82 indicates that the number of calibration points is too small to give a good result (we remember from EQ. 2.60 that the minimum number of points for doing 3D calibration was 6). In addition, the huge skewness shows that the result is doubtful, even when using the optimal internal parameters. However, the result is shown in Fig 3-14 and Fig 3-15.

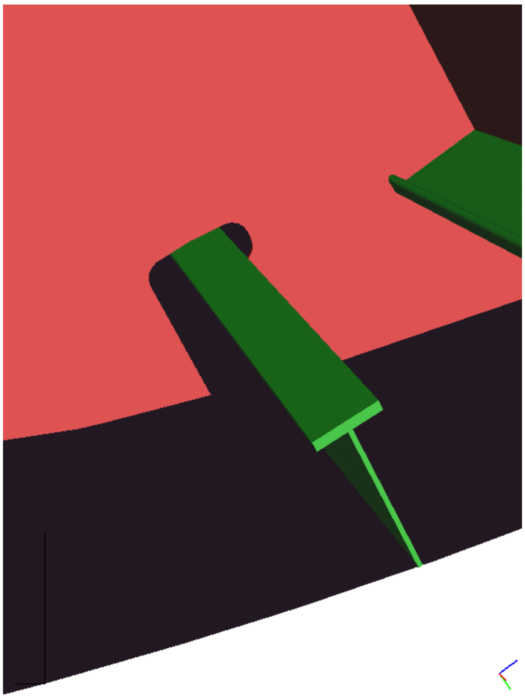


Fig 3-14 CAD image with optimal parameters

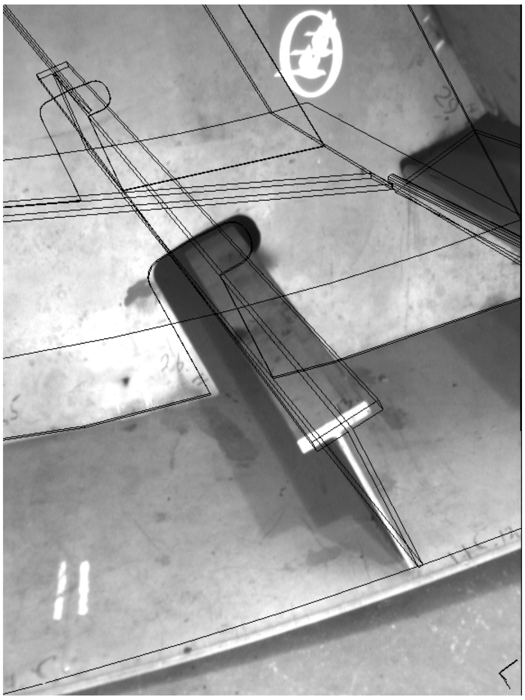


Fig 3-15 Wireframe and real image

The number of calibration points is close to minimum, but the result is still reasonably good when using optimal calibration parameters.

In general the pose of the camera is known and the final goal of our work is to find the exact model pose relative to the camera coordinate system. Next steps are

1. Development of a routine for finding the correspondence between common features in the real and the synthetic image. Special attention will be paid to the so-called Aspect Graph representation.
2. Development of a routine for finding the best match as a function of the external parameters, generally referred to as the Pose Estimation problem.
3. Generation of a synthetic image only on basis of a transformation matrix and a previous calibration. This requires exact knowledge about the transformation between the coordinate system of the moving device and the calibration coordinate system. This involves the solving of the Hand-Eye problem.

3.4 Correspondence between synthetic and real features

The matching between features found in the CAD image and features found in the real image, is one of two difficult problems to solve. I quote from Peurach[ii]: "*Model based vision problems are simply stated as finding the 'best' match between an object extracted from an image and the model database. This is an instance of the general problem of finding a solution in a solution space. These types of problems are notoriously difficult. Domains such as game play and problem solving demonstrates the difficulty in searching solution spaces for the best answer. In machine recognition systems the problem is significantly more difficult. The reason is simple: How do you know what is the 'best' match? The question 'Does this fit?' can not longer be used, but rather 'How good a match is this?' must be asked. This implies an increased search space since a complete search must be conducted, not one which stops when the 'right' answer is located.*"

The definition and extraction of features is the second difficult problem to attack. There exist two obvious possibilities based on *line* extracted features and *area* extracted features respectively.

3.4.1 Line-based feature extraction & matching

Edge detection with subsequent line estimation provides suitable configurations of lines that can be used for a matching with the CAD wireframe model. In Fig 3-16 the image has been processed with the **canny** edge detector giving the white lines, and in Fig 3-17 the wire frame has been added (black lines).

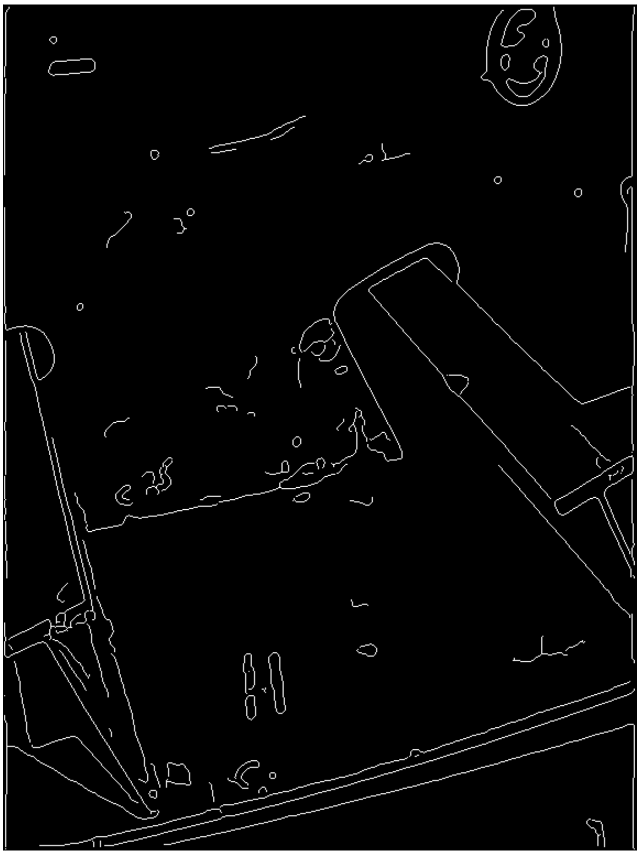


Fig 3-16 Canny-extracted image

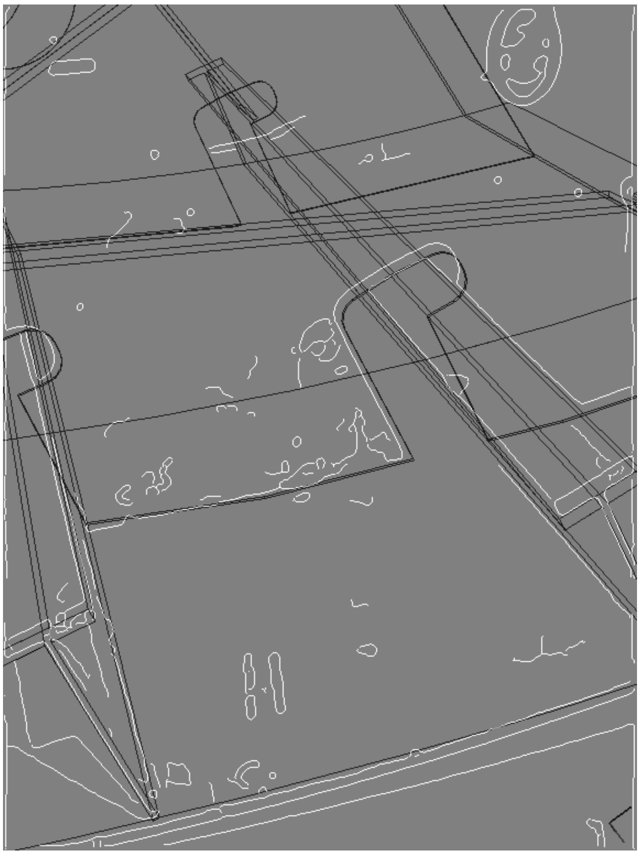


Fig 3-17 Canny-extracted plus wire-frame

The result is very promising. It is obvious that the lines in both images are lying very close, but how do we measure it? One measure could start from global transformations of the Hough transformation type. Another possibility is to work directly in the image, knowing that any corresponding set of lines (or arcs) is lying very near. This is work that will be done in near future, but due to time restrictions, not in this thesis.

3.4.1.1 Aspect Graphs

It is a well-known problem that general pattern recognition based on CAD models is very complicated. The solutions mostly spoken for are transformation of various CAD features into topological invariants. The program could then be built up like a tree of interpretation. An often advocated matching method is based on the construction of model graphs and so-called aspect graphs. The identification will then be done using a sub-graph isomorphy by means of a sub-graph search algorithm. In the area of *object recognition*, the generation of an aspect graph might be the right tool, although the traditional definition of the topology should be enriched in some way. An aspect graph of an object is a graph structure in which

- Each node represents a *general view* of the object as seen from some maximal, connected cell of viewpoint space. That means, the topological relations are the same in the complete cell (the topological relations are typical defined in terms of number of vertices, edges and faces).
- Each arc represents an accidental view that occurs on the boundary between two cells of general viewpoint. So each boundary signals a change in the topology of the silhouette (and maybe other features) of the object.
- There is a node for each possible general view of the object.
- There is an arc for each possible boundary.

Object recognition is not the most important application at OSS. But the very much related model-POSE estimation is a crucial point in the vision strategy. Likewise, a subsequent exhaustive identification of features inside the boundary of the object is most valuable for quality control.

The aspect graph representation has some severe problems, which has to be considered very carefully before a possible implementation in a daily production environment[iii]. One major problem is that relevant changes in topology are dependent on the choice of scale and there is no mathematical method to distinguish between relevant and not-relevant changes. In other words, there is no practical means to reduce the complexity of the computation of an aspect graph in general. Another weakness in the usual definition of an aspect graph is that a node is defined in terms of feature topology (vertex, edge and face). Whenever the topological structure changes as with respect to viewpoint a new aspect (node) is generated. From the CAD file these different aspects can be exactly computed but from the VISION side it is impossible to guarantee that all relevant topological relations are extracted from an image (a problem which is amplified in case of occlusion). An aspect graph consisting of only “recoverable aspects” (Joe Mundy[iii]) is difficult to produce but would be much more applicable.

The reasons for considering the aspect graph representation despite its general problems is that all vision at OSS is performed in a *structured environment*. That means that the pose of the camera and the model pose of the object are roughly known. Otherwise it would be most hazardous even to try to bring a camera into a position where it is able to take pictures; and in case of fixed cameras it can always be assumed that the ship block to look for is somewhere in the work area. Another big advantage is that there is always only one stable pose; a shipblock will always lie with one predefined specific side down and furthermore the last rotational degree of freedom is mostly reduced to some tens of degrees. In general, the difficult task at OSS is not core object recognition but the accuracy of the measurements which have to be around 1 millimetre in a distance of up to 15 metres.

To conclude, for fixed cameras there remain only two translational and one rotational degrees of freedom to be calculated. For movable cameras (mounted on robots) the problem is in principle similar; the model pose again has 3 unknowns but the pose of the camera is not as well defined as for fixed, in-place-calibrated cameras. The accuracy of the camera pose is related to the accuracy of the mechanical equipment and of course the accuracy of the off-line calibration of the camera (with respect to the inner parameters). With movable cameras however, the accuracy of the measurements can be arbitrarily improved by taking sufficiently many images from different camera poses.

Originally, aspect graphs were defined purely as the topological structure of an image of the geometrical contours of the object, and furthermore the source of all information was based on line extraction. At OSS we can still rely on the line extraction mainly because all shipblocks consist of a lot of straight lines and mainly because the size of the blocks allows us to consider even significantly curved structures as straight lines in the region of interest (from the vision point of view). The aspect graph representation however, has to be enriched with at least line extraction from the complete interior of the contour (especially since the contour of a shipblock normally is much too big to be contained in one single aspect). But with the rich CAD representation of the object also other kind of features should be considered: grey level, colours, textures and even shadows if we in the real environment are able to control the light sources sufficiently. Otherwise, shadows can be a most disturbing error source in the generation of the aspect graph. To conclude, it is very complicated to construct the complete aspect graph of an object and other methods, using the 3D information in a more direct way, will often be more efficient. In the OSS case, there exists a lot of additional information and more direct methods will surely be possible to develop.

3.4.2 Area based feature extraction & matching

Calculation of the-cross correlation or similar between smaller parts (with high dynamic) in the real image and the CAD surface model is another possible way of estimating the correspondence. In Fig 3-18, our original image is shown together with a CAD image, taken with a virtual camera with wrong pose (the camera is rotated some degrees in various directions). The CAD image is for simplicity converted to greyscale.

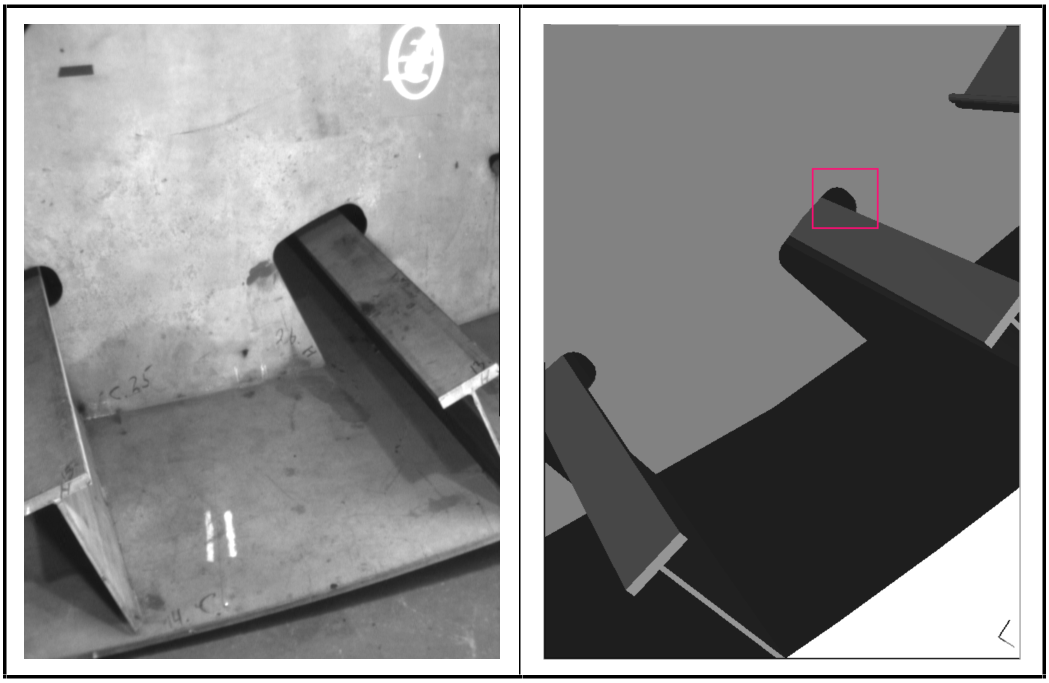


Fig 3-18 Real mock-up image and rotated virtual image (or vice versa!)

As a very first experiment I have tried to use a fairly simple method to find common features in the two images. A sub-window containing a salient feature that also appears in the real image is identified in the CAD image (small rectangle)[1]. The problem is that the feature has moved and been transformed due to scale and rotation. If however, we do perform a cross-correlation between the image and the sub-window we get the result shown in Fig 3-19.

[1] The problem of automizing the extraction of good features with high reliability is a huge task itself which will be attacked in an Esprit project recently accepted by the Commission.

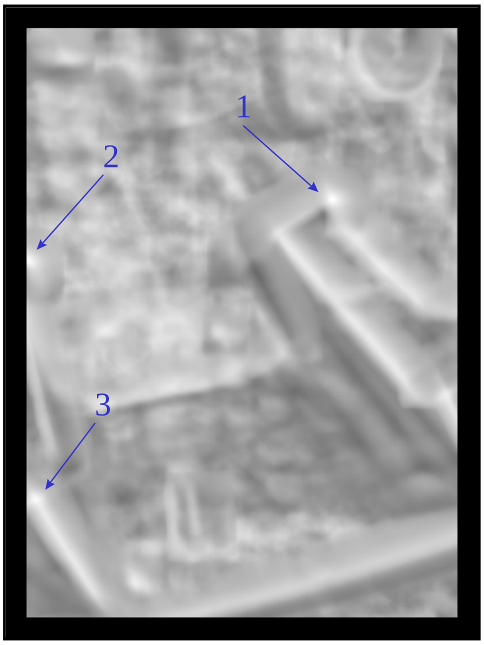
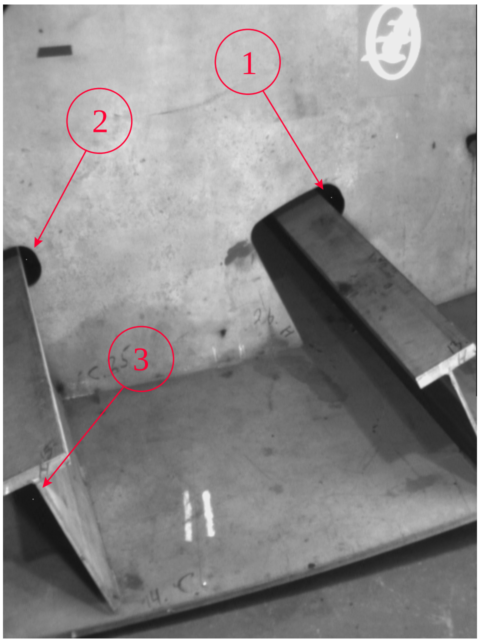
Identification of highest correlation values in image cross correlated with feature: ◢ (see Fig 3-18)	Identification of template on basis of cross correlation	Highest pixel values
		1 Peak(233,398) [0.9040] 2 Peak(308, 28) [0.8435] 3 Peak(598, 36) [0.8416]

Fig 3-19 Greyscale version of CAD image and extraction of sub-window.

We see that crosscorrelation is quite robust towards rotation. The correct position of the feature is found in the real image with fairly good accuracy and with high certainty since the second highest correlation point is significantly lower in value.

It is nevertheless possible to improve the result by transforming the feature extracted, before doing the crosscorrelation in the real image (Beveridge[iv] calls this kind of transformation "weak-perspective-2D"). In Table 3.15 three different rotations of the feature are cross-correlated with the image giving 3 different maximum peak values. In all cases the position is found but it is also notable that the correlation is significantly enhanced with optimal rotation and so is the gap to the first (wrong) position of the feature.

Feature	◢ (no rotation)	◢ (optimal rotation)	◢ (bad rotation)
Highest pixel values	1 Peak(233,398) [0.9040] 2 Peak(308, 28) [0.8435] 3 Peak(598, 36) [0.8416]	1 Peak(237,400) [0.9595] 2 Peak(312, 31) [0.8862] 3 Peak(601, 39) [0.7580]	1 Peak(232,397) [0.8914] 2 Peak(306, 26) [0.8435] 3 Peak(598, 35) [0.8416]

Table 3.15 Maximum peak values

Another example will illustrate the point even more. Consider another feature extracted from the CAD image and cross-correlated with the real image (Fig 3-20).

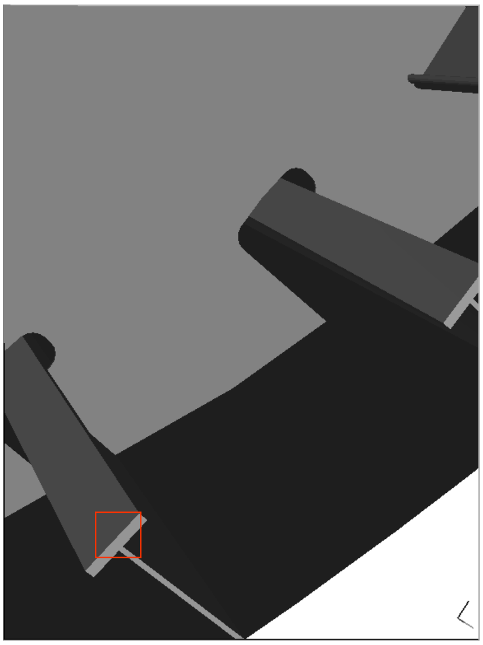
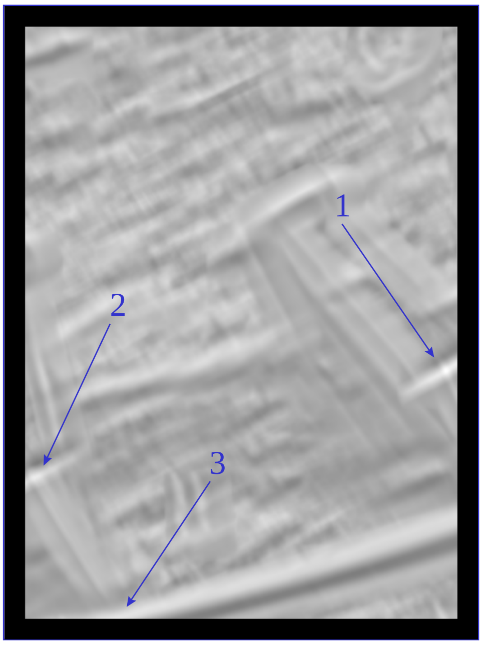
Rotated image	Cross correlation image
	

Fig 3-20 Rotated virtual image & cross correlation image (obtained with optimal feature rotation).

In Fig 3-21 and Table 3.16 we see that the method again points out the correct feature, although there are 2 similar features and the feature with the wrong position gives highest correlation value.

Feature	Highest pixel values
no rotation	1 Peak(441,530) [0.5158] 2 Peak(280,282) [0.4993] 3 Peak(569, 25) [0.4979]
optimal rotation	1 Peak(441,536) [0.8366] 2 Peak(571, 35) [0.6200] 3 Peak(741,135) [0.5463]
bad rotation	1 Peak(279,279) [0.4801] 2 Peak(68,494) [0.4483] 3 Peak(377, 81) [0.4373]

Table 3.16 Maximum peak values.

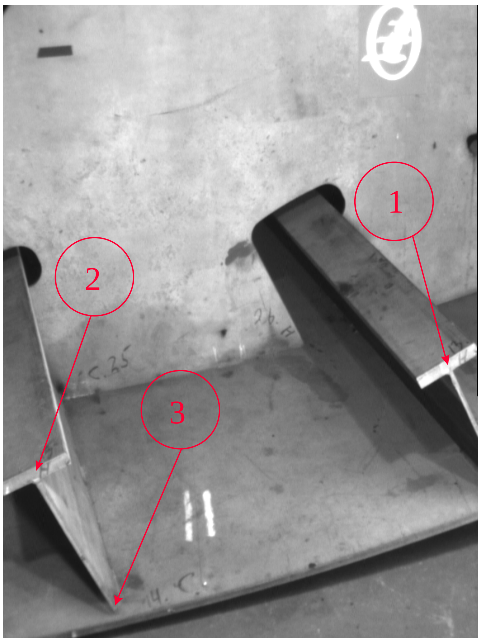


Fig 3-21 Another feature identified.

If more than two features are considered (which means that some kind of rigid transformation can be assumed), it should be possible to make a robust method. Especially to be able to find the correct feature in cases where more than one of the same kind appears in the real image. Horaud et al.[v] have developed an occlusion-tolerant method for finding a local feature in an image and searching for more features on basis of the position of the previous feature. The method uses a pre-created, ordered list of features with each feature having two sets of deduction, one to be made if the next feature is found and one if not. The creation of this difficult list is assumed done and not described in the paper. All objects are assumed lying on a tabletop and the image plane is assumed parallel to this tabletop: restrictions that hardly apply to the OSS case. The authors have developed a standard geometric model and a vision-oriented model. In this way, they become able to handle the search tree of feature matching. Despite the limitations of the approach, there might be some possibilities for porting this method to the OSS domain. The examples given above are only meant as some kind of brain-storming on the crucial issue of finding the object pose relative to the camera pose. Time doesn't leave possibility for further work on the matter in this thesis but the problem is the by far most important to be solved inside the frames of a typical OSS application and will be approached in the immediate future.

3.5 The pose estimation problem

With a correctly calibrated camera, finding the best match between a real image and a synthetic CAD image is equal to finding the same position of the CAD model (relative to the virtual camera) as the block has to the real camera. Generally, we assume working with correctly calibrated cameras and high accuracy manipulators, where the transformation matrices give very good descriptions of the real movements. At the same time we know roughly the position of the ship block, since it would be hazardous to let the robot move freely in unknown space. So, finding the absolute best match is a question of quite small corrections, i.e. moving the virtual camera a number of decimetres and rotating some degrees. The solution to this problem could be to keep moving the virtual camera until best fit between real and CAD image is obtained. From a (safe) position, an image of for instance a corner is taken from a camera mounted on a robot arm. From roughly the same relative position a virtual camera generates a virtual image from the CAD file and that image is compared with the real image. Then the virtual camera is moved around in the neighbourhood until best fit is achieved. This gives a (good) estimation of the position of the camera (and the corner). The real camera is moved to another (collision-safe) position, a new picture of the corner is taken, and the procedure is repeated. From these 2 (or more) images, it is now possible to make a very accurate 3D estimation of the position of the corner, firstly by optimising the correlation with the respective best CAD images and secondly (and most accurate) by solving the correspondence problem, now knowing which points are the same in the two real images. The procedure for finding best match is someway similar to the Hand-Eye calibration problem (see subchapter 3.6). In this case we know that we are close to the minimum and can therefore perform a sequential approach to find the solution. The sequential approach means: *First rotation, then translation.*

A more direct and efficient method can be developed but it will only work if following two conditions are met:

- It must be possible to find corresponding features in CAD and real images despite various deviations in the poses.
- It must be possible to identify the CAD-extracted features by their 3D CAD coordinates.

With the above conditions fulfilled, the problem reduces to a traditional pose estimation problem. Such pose problems are thoroughly described in the literature. Phong et al.[vi] presents a robust and efficient method to calculate the camera pose relative to a model. The method uses point or (preferably) line correspondences and applies a trust-region optimisation method that is much more efficient and stable than the traditional Newton method and at least as good as the generally preferred Levenberg-Marquardt algorithm. Only the trust-region method can be modified to handle so-called large-residual problems. Beveridge[iv] is presenting various methods for matching CAD features with image extracted features. He states that most work has replaced the pinhole model with less general imaging models. It is, for instance, often seen that a two-step process is applied where the CAD feature is projected into the image from a particular pose and subsequently all further changes are made by subjecting the 2D projection to 2D affine transformations (rotation, translation and scale). Other methods are based on scaled-orthographic projections (“weak-perspective-3D”) where it is possible to change the pose. Beveridge is presenting his own method, which uses full-perspective imaging with 4 different kind of pose finding algorithms. Christy et al[vii] are also using a weak-perspective camera model for initial pose estimation. The authors are analysing the differences between the weak-perspective approach and the so-called para-perspective approach. The conclusion is that although the para-perspective approach might be faster, the weak-perspective method should be preferred due to computational simplicity and ease of implementation. Another important result of their work is implementation of a line based pose estimation, which generally works just as well as traditionally point-based methods. The method might be the only possibility in cases where exact positions of points are hard to get. Peurach[ii] is describing how to calculate a fine pose given a coarse pose estimate. He is generating a wireframe model for each new and slightly different pose. The choice of next pose is based on a two-step algorithm instead of brute force search through the complete parameter space. For each of the real images the object is Hough-transformed into so-called feature space. The corresponding Hough transformations of the wireframes are constructed mathematically owing to the very time-consuming process. The approach described sounds quite relevant for the OSS applications and the conditions seem to be quite similar to many typical OSS installations. Some restrictions apply to the work of Milutinovic[viii], who in principle is doing roughly the same in his experiments as we might do at OSS. He is extracting scalar features like perimeter and distance of contour points from the contour of the object and similar features from the area image (area, number of holes, etc.). The estimation of the pose includes 1) identification of stable state 2) calculation of center of gravity 3) angle of rotation (by consulting a database). Majumdar[ix] is in a much smaller scale working in the same area but her work is also using application-specific techniques, which hardly can be implemented into a typical OSS installation. Modayur& Schapiro[x] are doing some matching based on line extraction and hidden line removal combined with finding features like straight lines. The overall goal of their work is to investigate the propagation of uncertainties in the measurement tasks. An important aspect, but a premature one for the tasks at OSS.

3.6 Hand-Eye calibration

In order to be able to generate virtual images only on the basis of a (thorough) calibration and some information about the translation/rotation of the camera away from this calibration position, it is necessary to calculate the position of the camera relative to the robot coordinate system. This is found on the basis of pair-wise sets of information about the position of the camera (in the calibration coordinate system), and the position of the frame of the robot arm on which the camera is mounted (in the robot coordinate system). The problem is referred to as the Hand-Eye calibration problem and is, despite its simple description, surprisingly difficult to solve. A presentation based on the work of Horaud[xi,xii] is given below.

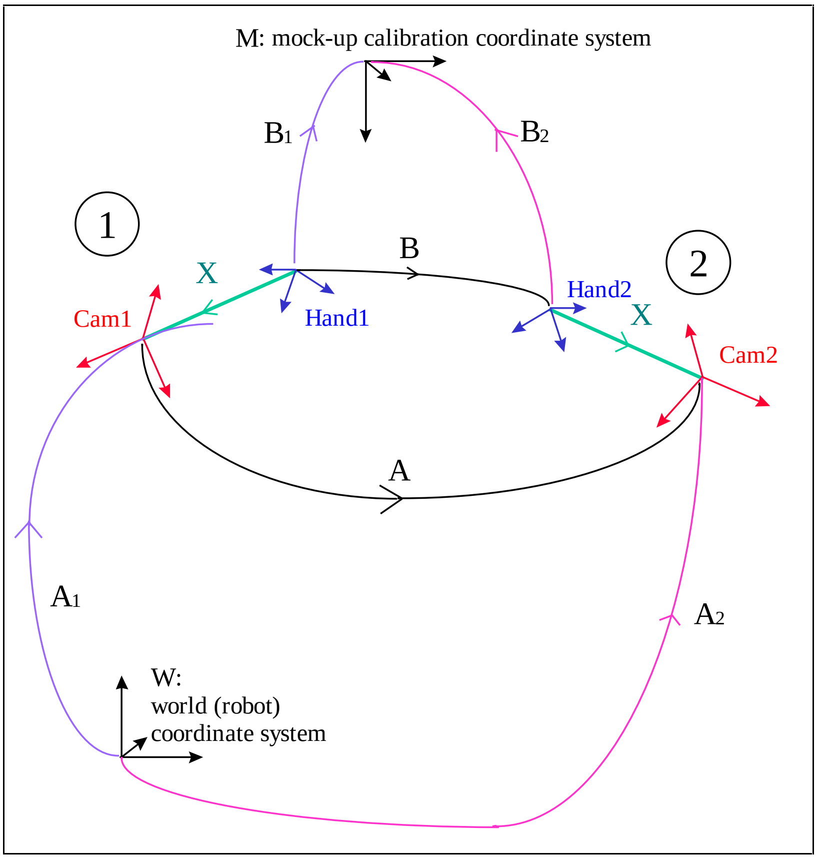


Fig 3-22 The Hand-Eye calibration problem

From Fig 3-22 following relations between frames can be derived:

EQ. 3.188
$$ {}^{M}T_{W} = {}^{M}T_{\text{Hand1}}\,{}^{\text{Hand1}}T_{\text{Cam1}}\,{}^{\text{Cam1}}T_{W} \quad \Rightarrow $$

EQ. 3.189
$$ \left.\begin{array}{l} {}^{M}T_{W} = B_1 X^{-1} A_1 \\ {}^{M}T_{W} = B_2 X^{-1} A_2 \end{array}\right\} \Rightarrow $$

EQ. 3.190
$$ B_2 X^{-1} A_2 = B_1 X^{-1} A_1 \quad \Rightarrow $$

EQ. 3.191
$$ A_2 A_1^{-1} X = X B_2^{-1} B_1 \quad \Rightarrow $$

EQ. 3.192
$$ \mathbf{AX} = \mathbf{XB} $$

where

EQ. 3.193
$$A = A_2 A_1^{-1}$$
$$B = B_2^{-1} B_1$$

EQ. 3.193 could also have been derived from following relation

EQ. 3.194 $${}^{W}T_{Cam1} = {}^{W}T_{Cam2}\,{}^{Cam2}T_{Cam1}$$

EQ. 3.192 (with EQ. 3.193) is the fundamental equation for solving the Hand-Eye calibration problem. It is commonly agreed that at least three different positions are needed to uniquely determine X. The solution is not at all simple, but the general approach is to decompose the equation into two equations:

- A matrix equation depending on rotation and
- A vector equation depending both on rotation and translation

The decomposition of EQ. 3.192 can then be written:

EQ. 3.195 $$R_A R_X = R_X R_B$$

and

EQ. 3.196 $$(\mathbf{R_A} - \mathbf{I})\mathbf{t_X} = \mathbf{R_X}\mathbf{t_B} - \mathbf{t_A}$$

From EQ. 3.195 we get

EQ. 3.197 $$R_A = R_X R_B R_X^T$$

which is a similarity transformation, since $\mathbf{R_X}$ is an orthogonal matrix. Therefore, $\mathbf{R_A}$ and $\mathbf{R_B}$ have the same eigenvalues, and for 3D rotation matrices one of the eigenvalues is equal to +1 (corresponding to the vector around which the rotation is performed. Let $\mathbf{n_B}$ be the eigenvector of $\mathbf{R_B}$ associated with this eigenvalue. Now EQ. 3.195 can be written

EQ. 3.198
$$R_A R_X n_B = R_X R_B n_B \qquad \Rightarrow$$
$$R_A R_X n_B = R_X n_B \qquad (eigenvalue = +1)$$

From EQ. 3.198 it follows that

EQ. 3.199 $$\mathbf{n_A} = \mathbf{R_X}\mathbf{n_B}$$

So, solving EQ. 3.192 is equivalent to solving for EQ. 3.196 and EQ. 3.199 which can be done following either of following approaches.

1) Rotation then translation
2) Rotation and translation simultaneously

The first approach is the simplest and can be solved in closed form. Unfortunately, this method is quite unstable and often unable to find the correct solution at all. The second approach is a non-linear optimisation problem that can be solved by various methods. Due to time restrictions I will not, in this thesis, go any deeper into the possible best solution for the OSS scenario. I will only conclude that it is essential to construct a method that precisely determines the transformation between the vision calibrating coordinate system and the robot coordinate system.

3.7 References

[i] Neider J. et al., *OpenGL Programming Guide*, 1993, Addison Wesley

[ii] Peurach T., et al. *3D Model Matching for Fine Pose Determination.* SPIE Vol. 2588, pp 485-493. 1995

[iii] Faugeras et al. *Why aspect graphs are not (yet) practical for computer vision*. Workshop panel report. CVGIP: Image Understanding. Vol. 55, no 2, March 1992

[iv] Beveridge, J. R. *Optimal geometric Model Matching Under Full 3D Perspective*. 0-8186-5310-8/94 IEEE.

[v] Horaud Radu & Skordas Thomas. *Model-based strategy planning for recognizing partially occluded parts.* Computer v 20 n8,1987, pp 58-65.

[vi] Phong, T. Q., Horaud R., Yassine A. *Object Pose from 2D to 3D Point and Line Correspondences.* IJCV, 15, 225-243, 1995

[vii] Christy S., Horaud R., *Iterative pose computation from line correspondences*. GRAVIR-IMAG & INRIA. France

[viii] Milutinovic, Dragan S. et al. *Model-based vision system for industrial parts using a small computer*. Robotics and Computer-Integrated Manufacturing v 3 n 4,1987 pp 439-450

[ix] Majumdar J, Levi P. & Rembold U. *3-D model-based Robot Vision by matching scene description with the object model from a CAD modeller*. Robotics v 5 n 1,1989, pp 69-83

[x] Modayur B. R. & Shapiro L. G. *Automated inspection of machine parts.* Proceedings. 11. IAPR International Conference on Pattern Recognition. Vol 1, Conf. A: Computer Vision and Applications, 1992, pp 57-60.

[xi] Horaud, Radu , Dornaika Fadi, 1995 *Hand-Eye Calibration* International Journal of Robotics Research, Vol 14, pp. 195-210

[xii] Dornaika Fadi, Horaud, Radu. 1996. *Simultaneous Robot -World and Hand-Eye Calibration*

4. Method for subpixel estimation

When using cameras for the exact measurement of an object, it is essential that the uncertainty of the equipment is reduced to a negligible level compared with the desired accuracy. This can be achieved by systematically calibrating the complete vision system. Calibration means in this case developing a model for describing how points in the object are mapped on to the camera image. Grid points are typically chosen as reference and all other points are interpolated from the transformation of these points.

It is however also a task in itself to determine the exact location of a point in an image if the accuracy has to be higher than the density of the pixels. This means that in order to construct a reliable model for straightening out the image, it is necessary to determine the grid points with subpixel accuracy. A method for determining the position of a template in an image with subpixel accuracy is described below.

4.1 The calibration grid

As search area is chosen an image of a 3-dimensional calibration grid with exact (<1 mm) placed measurement points (Fig 4-1).

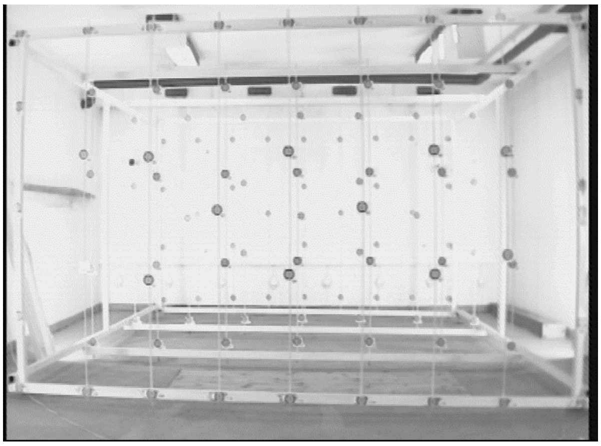


Fig 4-1 The calibration installation

This very accurate installation is placed in the cellar of the Institute of Photogrammetry and has for many years served as calibration tool for photo cameras. The template to look for in the search area is a measurement point which is build up of concentric black and white circles (Fig 4-2). The internal dimensions of the template are well known, but the size of the template (in the image) is dependent upon the size of the lens and the distance to the camera. All measurement points in a row have approximately the same size (same distance to the camera). The total number of measurement points is about 100, divided into four groups with different distances from the camera.

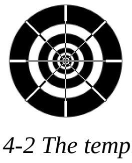


Fig 4-2 The template

In order to find the location of the measurement points with ordinary 1-pixel accuracy, the method of calculating correlation between the template and a moving window in the search area is used. The result is shown in Fig 4-3.

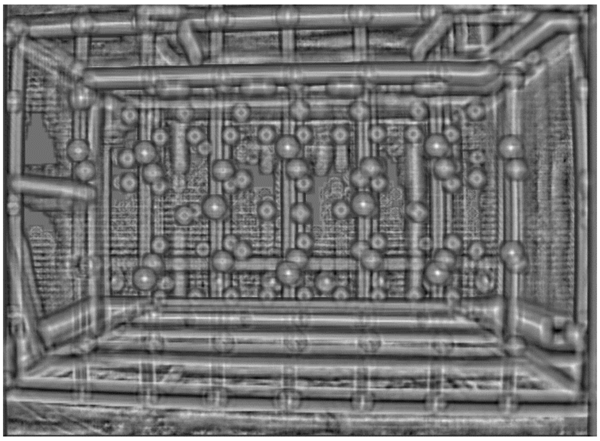
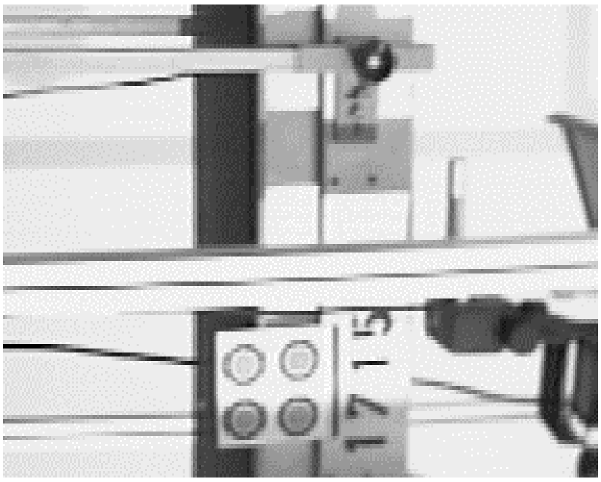


Fig 4-3 The cross-correlated image

The location of the templates is with one-pixel accuracy found at the lightest points in the correlated image. In order to achieve higher (sub-pixel) accuracy, it is necessary to look at the shape of the correlation top and determine the skewness. Sub-pixel estimations demand a completely reliable template without inaccuracies. The centre of the circles must be exactly placed "in the middle of a pixel". Therefore the template is generated synthetically (see Fig 4-2) in order to ensure that all subpixel displacements refer to the location of the measurement points.

4.2 The method

The following calculation is done on a smaller image containing only four measurement points (Fig 4-5 & Fig 4-4), but the principle will be exactly the same for the complete image of the calibration grid.



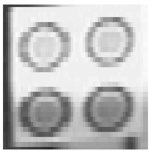


Fig 4-4 Features in test image

Fig 4-5 The test image

The first step is to find the most suitable template (right size). This is done by optimising the geometric parameters with respect to the average of the four correlation peaks.

The result of the cross-correlation between the best template and the complete search area is seen on Fig 4-6 and Fig 4-7 showing the four pixels around which the precise location of the grid points has to be found.

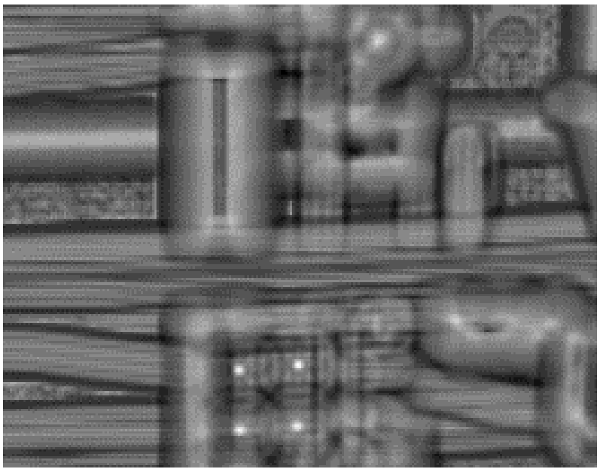


Fig 4-6 The cross-correlated testimage

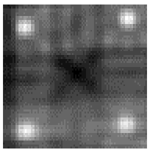


Fig 4-7 Region of interest

The idea is first to look at the shape of the autocorrelation of the template in order to find a parametric function that describes that curve. When the analytical function is known (with optimal parameters), the cross correlation peaks are fitted to this function by shifting the centre of the function incrementally away from the centre of the pixel. When the RMS between the correlation peak and the analytical function is minimal, the exact position of the peak is found. The following pages will clarify the principle.

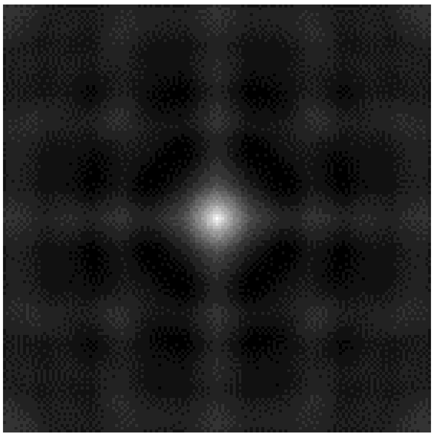


Fig 4-8 The ideal autocorrelation image

Fig 4-8 shows the auto-correlation image of the template from Fig 4-2. This image is ideal in the sense that the template used consists of many more pixels than the actual template for calibration contains. The ideal template consists of 49 times 49 pixels while the template giving the highest cross correlation contains 23 times 23 pixels. However for finding the best suitable analytical function (and for getting the nicest pictures), it is better to use a high resolution template which gives a nice auto-correlation curve without any strange distortion to coarse representation. A 3D perspective view of the auto-correlation is shown on Fig 4-9. Due to the symmetry of the calibration mark, the auto-correlation is also symmetric with respect to rotation. It is now the aim to find a suitable analytical function, which matches the auto-correlation peak perfectly. The symmetry of the function implies that it is possible to look for an analytical function in 2D and then afterwards expand the function into 3D. A slice through the autocorrelation peak is shown in Fig 4-10.

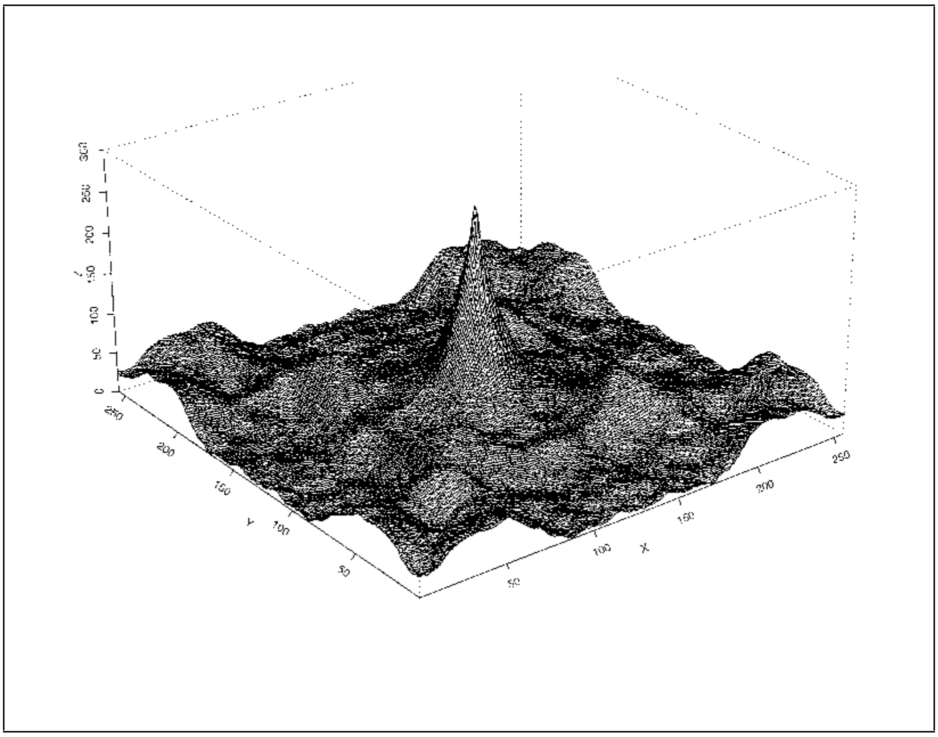


Fig 4-9 Autocorrelated image (full resolution)

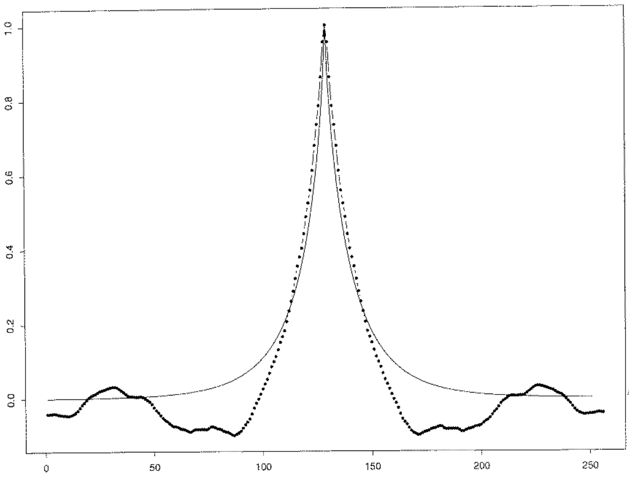


Fig 4-10 Analytical function and slice through autocorrelated image..

The shape of the peak is clearly "exponential-like" and it proved to be a good idea to look for an analytical function like:

EQ. 4.1
$$f(x) = e^{-a\left|\frac{x}{l}\right|^{p}}$$

Where a and p are the parameters to be fitted and 2l+1 is the size of the peak.

For the ideal autocorrelation peak, the optimal parameters were calculated to: a : 4.5 p : 1.2

The analytical function is also shown in Fig 4-10.

A simple expansion into 3D gives the analytical function:

EQ. 4.2
$$f(x,y) = e^{-a\left[\left(\frac{x}{l}\right)^{2}+\left(\frac{y}{l}\right)^{2}\right]^{\frac{p}{2}}}$$

2l+1: length (width) of window containing top.

This analytical function with RMS-optimised parameters is shown in Fig 4-11. Fig 4-9 is reproduced in Fig 4-12 with equal resolution. In order to check the correspondence between Fig 4-11 and Fig 4-12 the difference in each point are calculated and shown in Fig 4-13. As we can see the peak is completely removed concluding that the analytical function describes the autocorrelation peak very well (note the different scale on the z-axis).

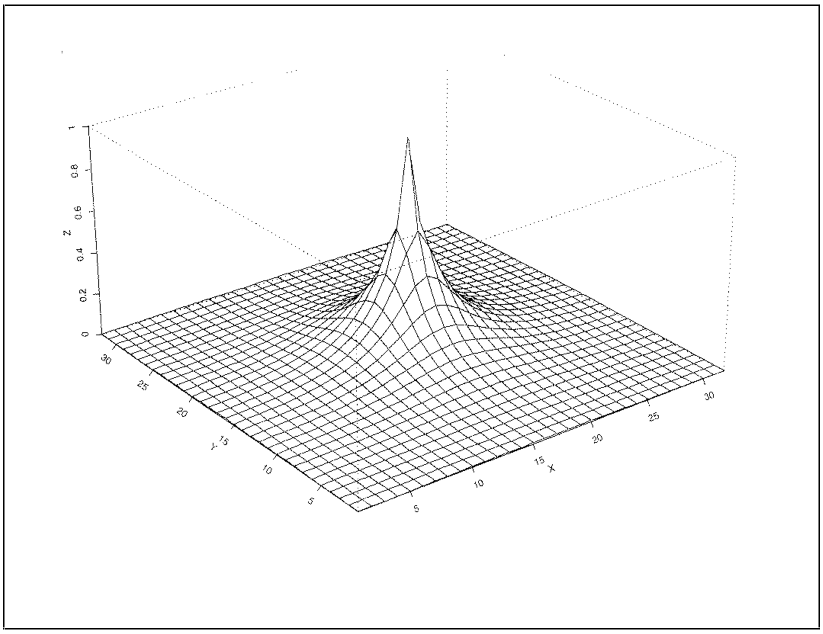


Fig 4-11 3D analytical function.

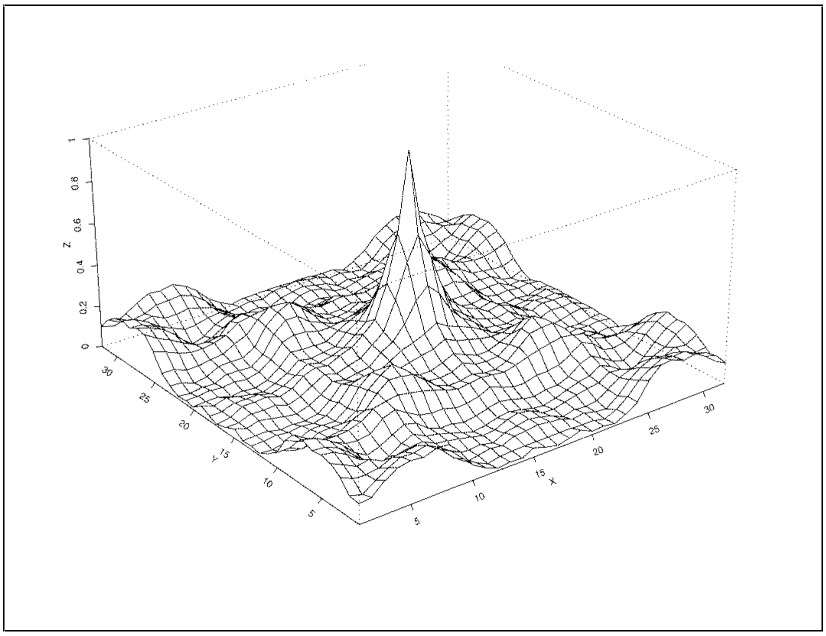


Fig 4-12 Autocorrelated image (scaled).

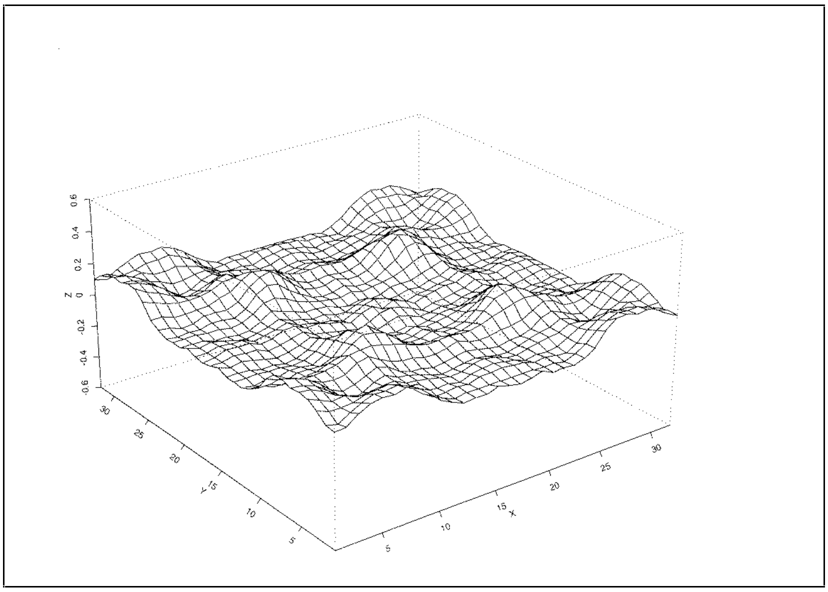


Fig 4-13 3D difference function.

For the test image (Fig 4-5), the template had the size of 23 x 23 pixels giving the coarse autocorrelation peak shown on Fig 4-14. A slice through the peak is shown in Fig 4-15. It is clearly affected by the digitalisation problems but remains exponentially shaped. In addition, from Fig 4-15 we can deduce that the extent of the peak is 9 x 9 (l = 4) pixels before ringing occurs.

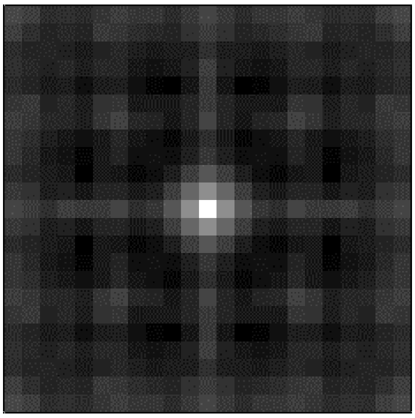


Fig 4-14 23x23 autocorrelation top

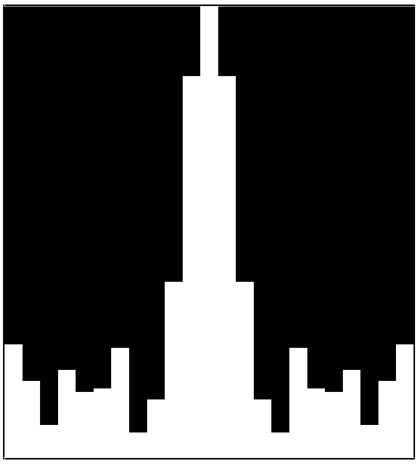


Fig 4-15 Slice of 23x23

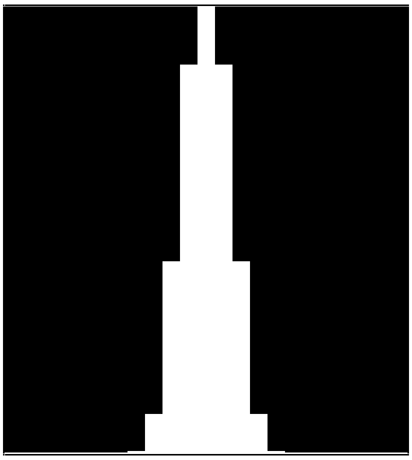


Fig 4-16 Analytical function

The analytical function best describing this autocorrelation peak was calculated to

EQ. 4.3

$$f(x,y)=e^{-8\left[\left(\frac{x}{4}\right)^2+\left(\frac{y}{4}\right)^2\right]^{\frac{1.72}{2}}}$$

A slice through this function is shown in Fig 4-16 with same discretization as the slice on Fig 4-15.

We now have a good analytical description of the autocorrelation peak and can return to the original problem: The exact location of the templates in Fig 4-4. The crosscorrelation of one of the 4 peaks in Fig 4-4 and Fig 4-5 is shown in Fig 4-17 and a slice of the peak in Fig 4-18.

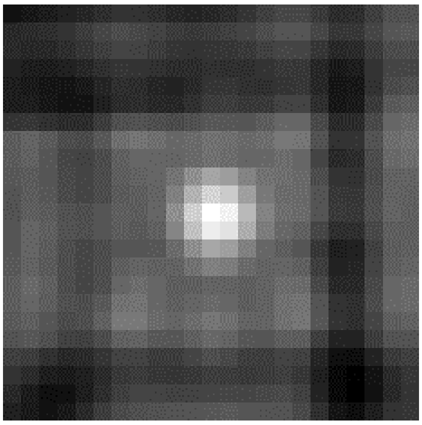


Fig 4-17 A cross-correlation peak

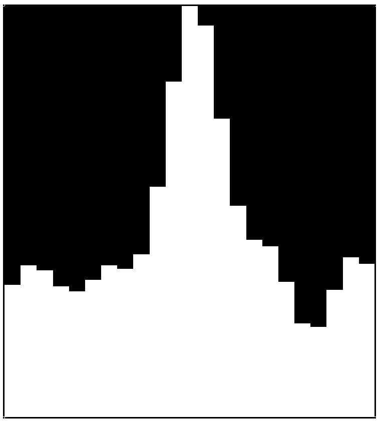


Fig 4-18 Slice of cross-correlation

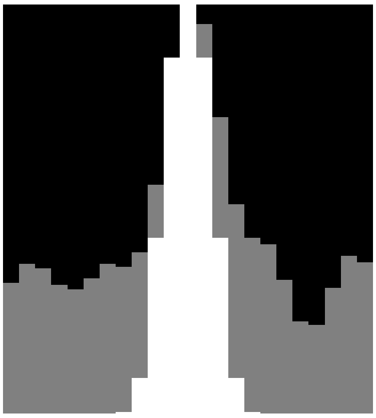


Fig 4-19 Auto- and cross-correlation

From Fig 4-19 which is Fig 4-18 overlaid by Fig 4-16 we can see that the cross-correlation peak is biased. The real maximum of the crosscorrelation is lying some tenths of a pixel away from the pixel with highest crosscorrelation value.

The idea now is to incrementally move the centre of the analytical function describing the auto-correlation peak:

EQ. 4.4
$$f(x,y) = e^{-a\left[\left(\frac{x+\partial x}{l}\right)^2+\left(\frac{y+\partial y}{l}\right)^2\right]^{\frac{p}{2}}}$$

and minimise the RMS with respect to (∂x,∂y):

EQ. 4.5
$$RMS = \frac{1}{N}\sum_{x,y}(f(x,y) - I(x,y))^2$$

N: Number of pixels $(2l+1)^2$

I(x,y): Points in cross-correlation peak.

4.3 Results

In Table 4.1 is illustrated how much the RMS is reduced when optimising the centre of the auto-correlation function. The results are shown for all 4 peaks.

Best values of (∂x,∂y) for a=8 and p=1.72.					
	(∂x,∂y)	RMS(∂x,∂y)	RMS(0,0)	l	Max corr.
Peak 1	0.14 0.36	0.0091	0.102	5	0.701
Peak 2	-0.30 0.84	0.0118	0.0138	9	0.659
Peak 3	0.42 0.08	0.0185	0.0204	4	0.654
Peak 4	0.26 -0.24	0.0076	0.0080	7	0.634

Table 4.1 Best value of a and p for all peaks

The results were obtained with a=8 and p=1.72, but since the optimum of the analytical function is rather wide, it is necessary to check whether the subpixel estimation is sensitive with respect to small changes of the parameters a,p. In Table 4.2 and Table 4.3, it is illustrated how much the sub-pixel position changes for different values of a and p.

Variability of p			
a: 8.0	p	(∂x,∂y)	RMS(∂x,∂y)
Peak 1	1.64	-0.16 -0.36	0.0100
	1.72	**-0.14 -0.36**	**0.0091**
	1.80	-0.14 -0.34	0.0084
Peak 2	1.64	0.30 -0.82	0.0107
	1.72	**0.30 -0.84**	**0.0118**
	1.80	0.32 -0.88	0.0131
Peak 3	1.64	-0.44 -0.10	0.0198
	1.72	**-0.42 -0.08**	**0.0185**
	1.80	-0.42 -0.08	0.0174
Peak 4	1.64	-0.26 0.24	0.0076
	1.72	**-0.26 0.24**	**0.0076**
	1.80	-0.24 0.24	0.0078

Table 4.2 Variability of p for fixed a

Variability of a			
p: 1.72	a	(∂x,∂y)	RMS(∂x,∂y)
Peak 1	7	-0.14 -0.36	0.0074
	8	**-0.14 -0.36**	**0.0091**
	9	-0.16 -0.36	0.0108
Peak 2	7	0.34 -0.94	0.0137
	8	**0.30 -0.84**	**0.0118**
	9	0.28 -0.78	0.0107
Peak 3	7	-0.42 -0.10	0.0159
	8	**-0.42 -0.08**	**0.0185**
	9	-0.44 -0.08	0.0209
Peak 4	7	-0.26 0.24	0.0072
	8	**-0.26 0.24**	**0.0076**
	9	-0.26 0.24	0.0082

Table 4.3 Variability of a for fixed p

The results in Table 4.2 and Table 4.3 clearly indicate that the calculation of the subpixel position is stable and reproduces consistent results despite minor perturbations. In other words, the method is applicable to a broader range of problems where the task is to find specific well-known features in the image. The only constraint is the necessity for a completely reliable and error-free template.

No exact calculations are made however about the uncertainty of the method. But from Fig 4-20 some estimations can be made. From a strictly vision-based point of view, it seems reasonable to expect about 0.1 pixel accuracy for the method.

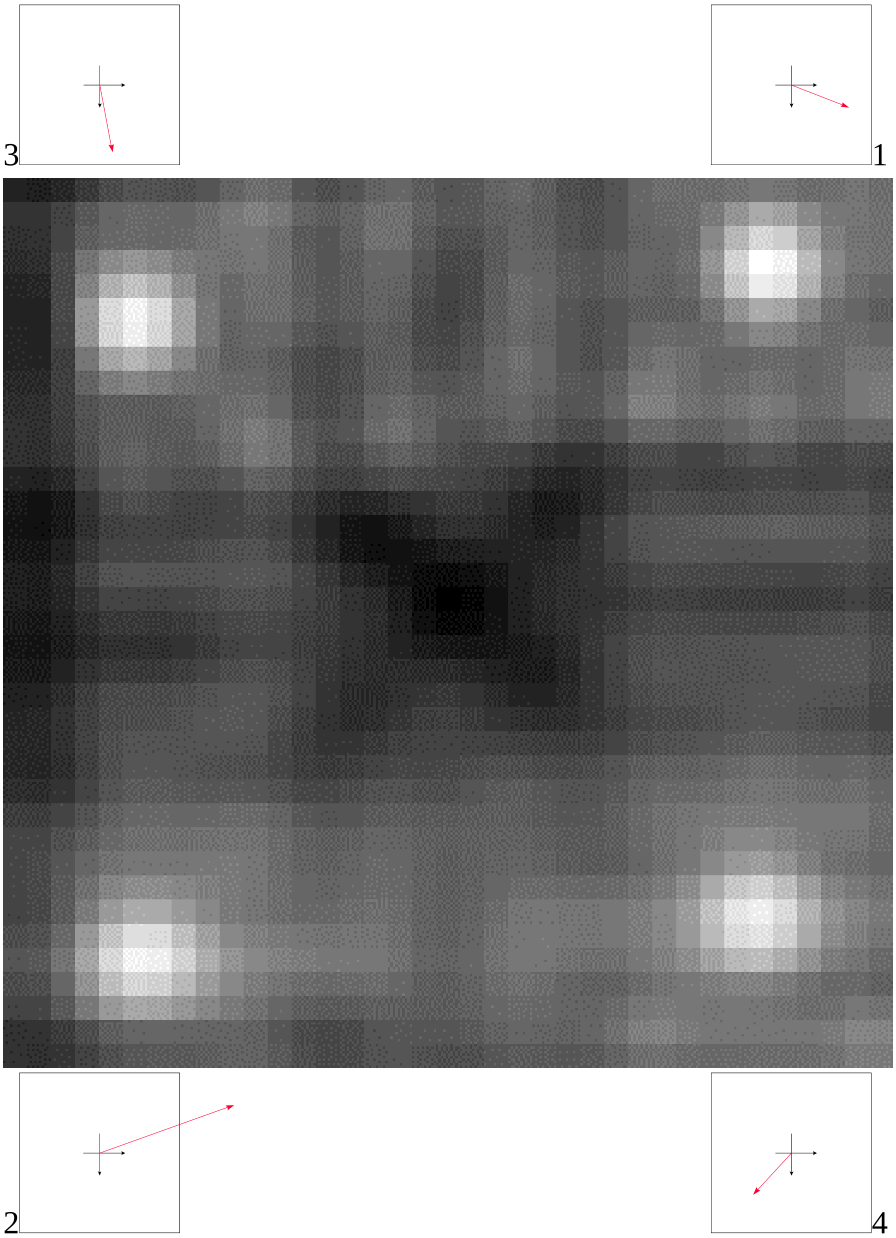


Fig 4-20 Illustration of how the centre of the templates is shifted with subpixel accuracy. The boxes surrounding the image illustrate one pixel and the red arrow shows how much the centre should be shifted (see also Table 4.1)

4.4 Conclusion

A new method for subpixel estimation has been presented. Many other methods for finding features in an image can be found in the literature[i,ii,iii]. Generally, some methods use cross correlation combined with some kind of fitting function[iv], while other methods are based on some kind of an edge detection, especially straight lines[v]. The method presented here is different in its use of the auto correlation curve as the basis for the fitting surface. The algorithm can, besides being used alone, also serve as an independent method for checking the result of many line-detection based applications.

The development of the algorithms was done early in the project, and at present this method has shown high reliability and applicability. For instance all results obtained in chapter 2 are based on data found by this subpixel method and the reduction in the residual as calculated in Table 2.9 proved to be up to 20 percent.

4.5 References

[i] Brand, P. and Mohr R. *Accuracy in Image Measure*. SPIE Vol. 2350 Videometrics III, p. 218-228, 1994

[ii] Valkenburg, McIvor and Power. *An evaluation of subpixel feature localisation methods for precision measurement.* SPIE Vol. 2350 Videometrics III, p. 229-238

[iii] Shortis, M. R., Clarke, T. A., Short, T. A comparison of some techniques for the subpixel location of discrete target images. SPIE Vol. 2350 Videometrics III, p. 239-250, 1994

[iv] Gleason, Hunt and Jatko. *Subpixel measurement of image features based on paraboloid surface fit*. Oak Ridge National Lab.

[v] Gramkow C. *Quality control of plasma cutting*. Quarterly report 2 ATV (EF 660). June 1997

5. Vision components

In a shipyard a vision system has to operate under extreme light conditions changing from a bright summer day to a dark winter afternoon, or even in the middle of the night where only electric light is illuminating the shipblocks. It is therefore very important that the system constantly is producing an image with the highest possible accuracy, contrast and sharpness.

The basic components in any vision application are camera + lens + frame grabber + computer. When selecting the components one has to focus on accuracy and applicability from the total system with respect to a given application. For instance, some frame grabbers dedicated for colour images take only poor images from grey tone cameras and likewise (cheap) colour cameras produce images with less qualified resolution than b/w cameras. In some applications more than one camera is needed which makes it desirable to have a framegrabber able to take more than one input.

5.1 Cameras

Cameras are roughly divided into 2 main groups: Surveillance cameras and machine vision cameras[i]. A surveillance camera normally enhances the image in order to give the best possible image for the human eye. The most typical image enhancements performed are:

Gamma correction modifies the output image according to following formula[ii]:

EQ. 5.1

$$I_{out} = I_{max}\left(\frac{I_{in}}{I_{max}}\right)^{\gamma}$$

This means that for $\gamma \neq 1$ there exists a non-linear relationship between the light distribution in the image and the real world. Surveillance cameras typically have a gamma value of 0.6, which makes dark areas lighter relative to the light areas. This feature alone weakens the edges in an image and is nearly always implemented together with the feature “Edge enhancement” (see next). When making accurate measurements with a surveillance camera it is therefore important either to find out how to switch off the gamma correction or to *gamma correct* the image by the transform

EQ. 5.2

$$I_{out} = I_{max}\left(\frac{I_{in}}{I_{max}}\right)^{\frac{1}{\gamma}}$$

Edge enhancement (over- and undershoot). In order to make the edge stand out clearly, dark areas are made even darker near the edge and light areas are made lighter (see Fig 5-1). A similar process is actually always produced by the human eye, the camera just adds further to this.

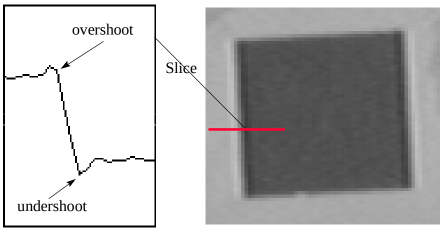


Fig 5-1 Over- and undershooting

Furthermore, surveillance cameras are often equipped with various devices in order to compensate for varying light conditions:

Auto shutter adapts the exposure time to achieve the correct amount of light per image. The problem with auto shutter can be that in situations with very intensive light, the exposure time gets so small that the CCD chip is not allowed time to completely transfer the previous image to the register. This causes a local spreading of the image called *smear*. This is an extreme situation as generally the auto shutter is a good way to automatically control the light input when doing accurate machine vision.

AGC (Auto Gain Control) measures the voltage of the video signal and adjusts the signal to constant 0.7 Volt, which is the typical value for a camera output. This means that when the light level is decreasing, the AGC is amplifying the complete signal including the noise coming from the camera itself. This will some times result in unacceptable noisy images. It is also generally preferable to adjust the signal instead of the light input. In case of intense illumination, some charge in the sensor cells will overflow into neighbouring CCD cells causing so-called *blooming*: a phenomenon which AGC will not register at all.

Cameras normally have a resolution of 720 x 575 pixels and they contain a CCD chip of size 1/3, 2/3 or ½ inch (see Fig 5-2)[1].

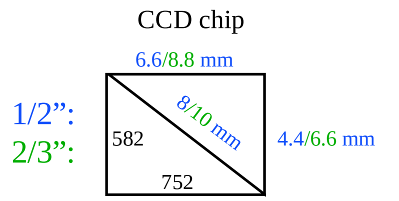


Fig 5-2. Dimensions of the CCD chip

Generally, the effective view area is assumed to be 90 % of the total. This leads to the dimensions:

$$\frac{1}{2}\ \text{inch: } 6.6\times 4.4 \xrightarrow{\div 10\%} 5.94\times 3.96$$

$$\frac{2}{3}\ \text{inch: } 8.8\times 6.6 \xrightarrow{\div 10\%} 7.92\times 5.94$$

which means that a $\frac{2}{3}$ inch CCD has the width w = 7.92 mm.

and a $\frac{1}{2}$ inch CCD has the width w = 5.94 mm.

Using above dimensions for the CCD chip together with EQ. 2.4 the relation between focal length, viewing area and viewing distance can be expressed in a diagram like the one shown in Fig 5-3[iii]. From Fig 5-3 it can for instance be seen that a 16 mm lens in 6 metres distance gives a viewing area of approx. 4 square metres, which again means a resolution of $^{2100}\!/_{720} \approx 3$ mm per pixel.

[1] Apparently, 2/3 “ cameras are being phased out.

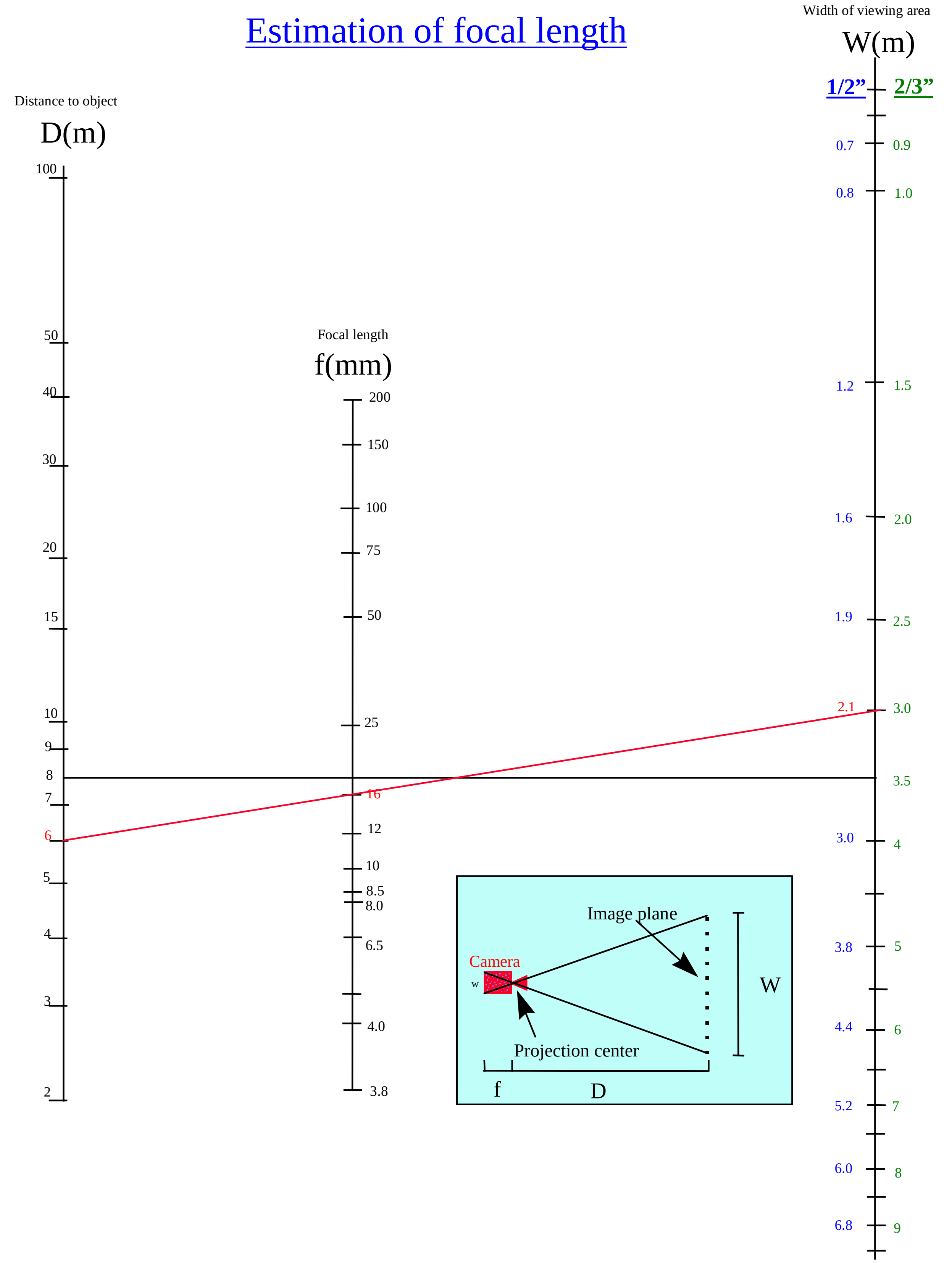


Fig 5-3. Diagram for quick estimation of suitable lenses (redrawn from working papers of H. Rasmussen[iii])

The cameras used so far at OSS are standard JAI b/w ½" CCD cameras with some extra features. Although AGC is a problem, intelligent gain control (IGC) is a most attractive feature. IGC means that it is possible from the computer to amplify the video signal on basis of measurements done only in regions of interest in the image. The cameras used at OSS is equipped with a facility that makes it possible to do IGC via an analogue potential (6-0 V). With a simple D/A converter, it is possible with 3 digital I/O's (provided by the framegrabber) to amplify the video signal from 0 to 100 percent in steps of 15-20 percent. In Fig 5-4 the relation between potential and gain is shown (6 Volt is equal to zero gain which means that the camera is "active low").

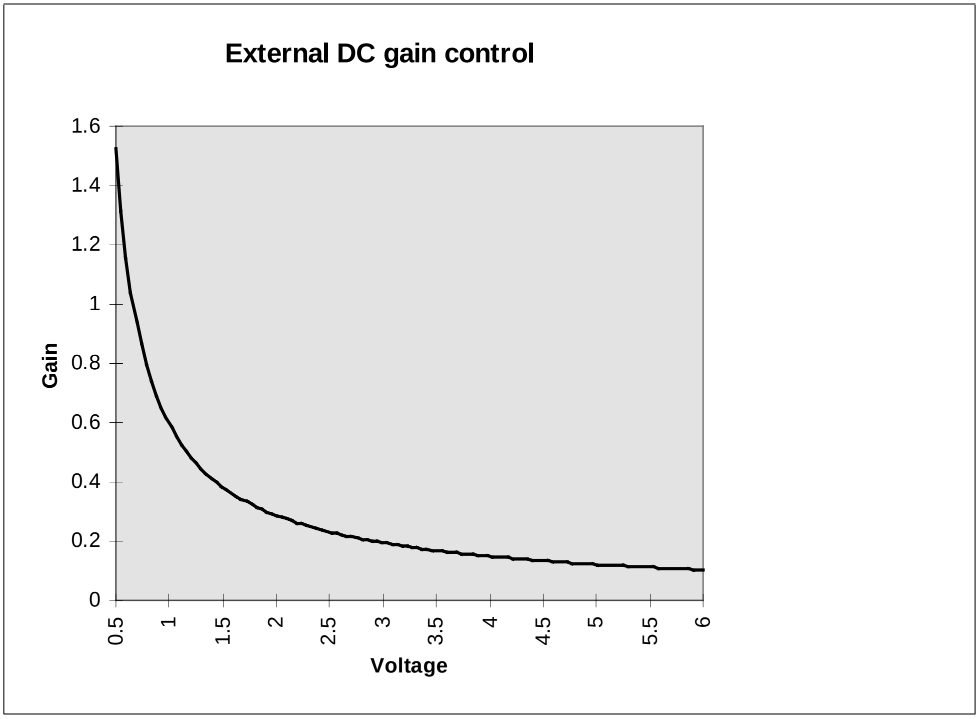


Fig 5-4 Amplification of video signal as a function of an external DC voltage.

Another feature that the cameras have is the possibility for setting the shutter time via digital I/O's (also provided by the framegrabber). With two I/O's it is possible to select one of the following four shutter settings (in milliseconds): 1/60, 1/125, 1/250, 1/500.

For specific applications, the vision processing can be made easier if some aid is provided by structured light in the shape of laser beams. Lasers are emitting light with wavelengths from around 500 nm up to 1200 nm. It is therefore necessary to know the spectral sensitivity of the camera. In Fig 5-5 the spectral sensitivity as a function of the wavelength is shown for a SONY CCD chip. This chip is used in all JAI cameras.

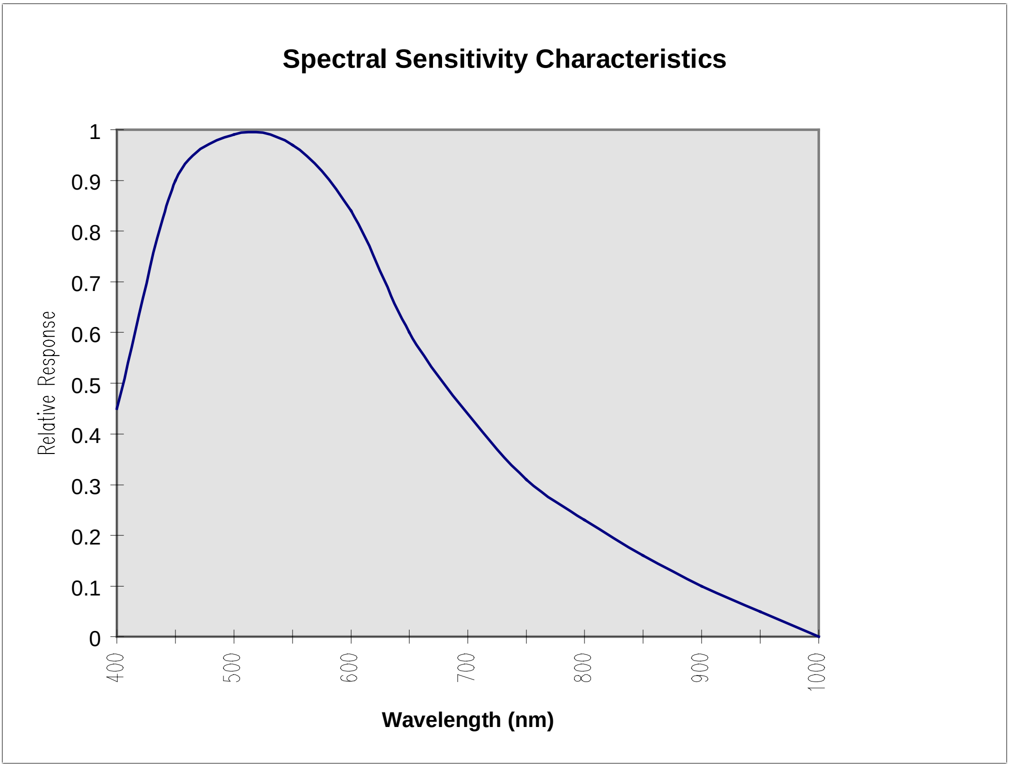


Fig 5-5 Spectral Sensitivity Characteristics for Sony CCD chip.

5.2 Lenses

Lenses are divided into 3 categories depending on the viewing angle:

< 15° :	wide angle
15° < α < 60° :	normal
60° < α :	tele

Knowing the distance to the camera and the viewing area, it can be derived which lens is the most suitable for a specific application as seen in previous subchapter. The camera constant c is changing when changing the focus. After focusing, the lens must be kept fixed and the complete system has to be calibrated. The viewing angle α and the camera constant have the following relationship:

$$tan(\frac{a}{2}) = \frac{l}{2c} \quad \Rightarrow \quad \alpha = 2atan(\frac{l}{2c}) \quad \Rightarrow \quad c = \frac{l}{2\,tan(\frac{\alpha}{2})}$$

The aperture ("F"-value, not to be mistaken with the focal length, abbreviated f) written on most lenses is a measure of how much light the lens is letting through. Strictly speaking, F is expressing the relation between optical length and the optical diameter of the lens. This means that low F values correspond to much light through the lens, which again gives images with higher quality[iii].

Some examples: $F = \frac{focal\ length}{lens\ diameter} = \frac{16mm}{20mm} = 0.8 \qquad F = \frac{focal\ length}{lens\ diameter} = \frac{16mm}{10mm} = 1.6$

This also means that wide-angle lenses let more light through than tele lenses. Normally the value of F is between 1.2 and 22. A higher sensitivity than 1.2 is special and expensive while optics with F higher than 22 require a special design. At F=22 the optics start behaving like the pinhole model.

It is also possible to use lenses with controllable apertures; either automatic or remote-controlled from a computer. The advantage is that the correct amount of light always reaches the CCD without risking smearing, blooming or noise. There might however be a problem when using varying apertures in extremely accurate measurements (0.1 pixel or better). If one with a full open lens does a complete calibration of the vision system down to say 0.1 pixel, this calibration has taken into account the non-linear contributions from the whole lens and especially those near the edge. If then, in a measuring situation, the amount of light has increased, the aperture decreases and only the inner part of the lens is producing the image on the CCD. That might require another calibration or maybe an interpolation between previous calibrations for different apertures. Another problem with automatic aperture is that areas in the image which are of no interest can influence the resolution of the image, e.g. a strong illuminated spot will make the aperture small and reduce the illumination of the rest of the image unnecessarily.

Focus is achieved by adjusting the length from the lens to the CCD chip (focal length). This is mainly done by adjusting the lens but additionally it is possible to adjust “back plane focus”. The procedure for adjusting back plane focus is first to find an object in infinity (> 20 m), adjust the lens to infinity, make the aperture as big as possible without destroying the image and then adjust back plane focus to a sharp image.

Sharpness of an image is in principle only achievable for exactly one focal length for one distance from lens to object. However, in practice a high F value will result in a less sensitive adjustment of the focal length[iv]. In other words, if you have problems with too small a field of depth you just increase the amount of light and decrease the aperture. The normal procedure for ideal image quality is first to make the aperture as big as possible, focus sharply and then reduce the aperture to the desired level.

5.3 Framegrabbers and computers

The framegrabbers used so far are partly UNIX based and partly NT based. When the applications are closely linked to the cell-controller environment we normally choose a SUN workstation with a simple and reliable framegrabber able to grab colour as well as b/w images. The framegrabber is not suitable for real time display nor able to handle many inputs but many planned applications do not need these utilities. When a cheap, stand-alone solution is preferable we mostly go for a PC solution with NT as the operative system. For instance, we will in the near future see applications where we have to put together 4-6 cameras in order to measure plates of the size 4 times 16 metres. This requires a dedicated framegrabber able to handle many inputs fast.

The most important single error source when grabbing images from a composite analogue video stream is incorrect synchronisation between camera and framegrabber: the so-called line-jittering problem. Line-jittering is seen when there exists a time shift between when the camera reports that a new image is starting and when the framegrabber thinks so. This time shift is stochastic and can, for a good framegrabber, vary from -5 nano-seconds to +5 nano-seconds. The influence on the sub-pixel accuracy can roughly be estimated from following calculation:

1 field in a frame takes 1/50 s	= 20 milli-seconds (ms)
1 line in a field takes 20/288 ms.	≈ 70 micro-seconds (μs)
1 pixel in a line takes 70/768 μs.	≈ 90 nano-seconds

So, in the worst case, line-jittering causes an error in the subpixel estimation of size 6/90 ≈ 0.07 pixel. This is not much, but on the other hand, it is in the area where we often operate. However, if the application is based on line detection or similar, one should expect the influence from line-jittering to be minimal since the mean error is zero and many lines in the CCD chip contribute to the line detection. If the error is still unacceptable (and price is not important), accuracy can be greatly improved by using the best cabling method available[v]. Using the simplest connection (the composite video from a camera) may not always make the most sense. Three typical types of cabling configurations are described below, in order of ascending accuracy.

Composite Sync Mode. This is the most common method of connecting a camera to a frame grabber. The composite video signal from the camera is simply fed into the composite input of the frame grabber. The frame grabber locks onto the sync pulses from the camera, which are fed along with the video information. For boards with a digital synchronisation circuit, pixel jitter will be approximately 5 nano-seconds, worst case. Boards with an analogue phase-locked loop (PLL) will result in approximately 8 to 10 nano-seconds jitter. This is the easiest of the cabling configurations, but also results in the highest jitter.

Sync Master Mode. This is when the horizontal and vertical drives from the frame grabber are used to drive the camera. The camera locks these signals to its outgoing video signal that is fed back into the frame grabber, reducing jitter to approximately 1 to 2 nano-seconds. This is also sometimes referred to as "Gen Lock" mode. This is a very common configuration that doesn't require an expensive camera and gives excellent results.

External Clock Mode. This uses the camera's pixel clock and horizontal and vertical sync signals to drive the frame grabber. This is the most accurate of the configurations. However, only the more expensive cameras have horizontal sync, vertical sync, and pixel clock outputs. In addition, the frame grabber must have a variable scan front-end, so that it can synchronise onto the Horizontal and Vertical sync pulses. Jitter in this configuration is typically in the sub-nanosecond range, or less than 1 nano-second.

5.4 References

[i] Læssøe, Jørgen,. *Working papers*

[ii] Immerkær, John, *Lecture Notes*

[iii] Rasmussen, Henning, SEMCO. *Working papers*

[iv] Jähne, B., *Practical handbook on image processing...*, 1997, ISBN 0-8493-8906-2, page 133-136

[v] www.datx.com/faqtips

6. First implementation: B4

As part of the project, it was decided by the steering group together with me, to make a half-year break in the studies and do an implementation of some of the results that had been achieved so far. The justification for this break in the studies was twofold: Firstly, it was important for me to do some "real" work and get hands-on experience of the problems which inevitably occur, when an implementation is tested in a daily production environment. Secondly it was important for the whole project to become visible at the shipyard with an installation earning money like any other (advanced) equipment installed at the premises of OSS.

It was decided, on request from OSS, that the implementation should be performed on a robot installation, which were constructed at the same time in one of the big assembly halls, B4 (see Fig 6-1). This installation is by far the largest, the most complex and the most ambitious robot installation ever made at OSS, and actually ever made in the shipbuilding industry. This of course had several impacts on work done from my side. The advantages were that "Vision at OSS" could get a flying start, the use of cameras became highly visible immediately and the cameras didn't have to be added on existing (and working) equipment, but could be integrated from the beginning (more or less!). The disadvantage was that all problems were bigger and very visible and nearly all problems in any area of the equipment had a severe impact on the performance of the vision module.

Unfortunately the implementation work took considerably longer than expected, mainly owing to the constraints, which follows when you have to integrate modules into equipment in a production line. And because of a very tight time schedule, the physical installation of the cameras was not possible before the equipment was taken into daily use. But at last, the workers have now started using the cameras in actual production and the cameras can start demonstrating an increase in accuracy and thereby productivity.

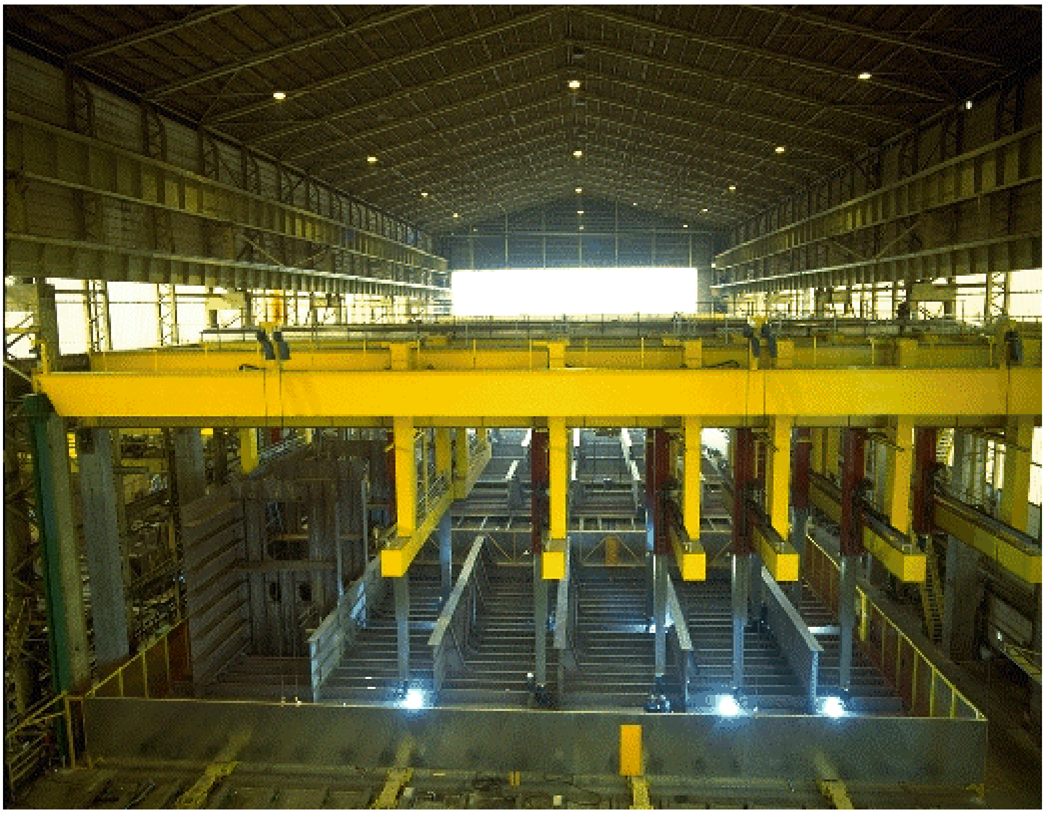


Fig 6-1 The B4 robot installation

6.1 Introduction

I have taken this approach first to give a description of the basic problem of vision-based object localisation which I had to solve. This includes construction of a 2D-2D transformation matrix (calibration) and analysis of the parameters contained in the matrix. Secondly a description of the equipment which carries the cameras and which will use the resulting co-ordinates from the vision module is given. The vision module is very sensitive to uncertainties in the mechanical devices. Therefore some considerations have been made concerning the accuracy of subparts of the equipment, as well as the aggregated accuracy of the device. In the Appendix a description of the functionality of the various programs involved in the application is given. During construction and implementation of the programs many interesting considerations had to be taken into account and therefore under each description several subchapters will occur, describing the nature of the problem and the approach for solving it. The concluding chapter discusses the status of the objectives and the remaining work to be done in B4 in order to fully optimise the vision module.

6.2 Why vision?

The specific problem in B4 was the requirement for exact alignment of the ship blocks. Due to the special construction of the mechanical device combined with the off-line generated robot programs, it was necessary to require that the block be placed parallel to the production line. The deviation was not allowed to exceed some tenths of a degree. Generally, the placing of the block with sufficient accuracy would not be that difficult if only the workers could be provided with a tool for measuring the rotation. The main problem was to create a module, able to measure with sufficiently high accuracy. At the same time, it was important that the module should be user-friendly, reliable and open for future automation. The first module provided was merely a pointing device composed of a laser mounted on a known position on the equipment. The workers then manually had to move the equipment until the laser spot was right over a predefined point. From the position of the equipment it was then possible to estimate the position of the ship block. The drawbacks in this method were numerous. The accuracy depended on the accuracy of the manual positioning of the laser, it took long time to position the laser and, most important, the process was impossible to automate further. These drawbacks could all be eliminated by introduction of vision and no obvious new disadvantages were introduced if the implementation were done carefully.

6.3 The Problem

The basic task for the robots is of course to weld as much as possible of the ship sections. And to do it as fast as possible and with the highest achievable quality. The role of the vision module in this task will be described below.

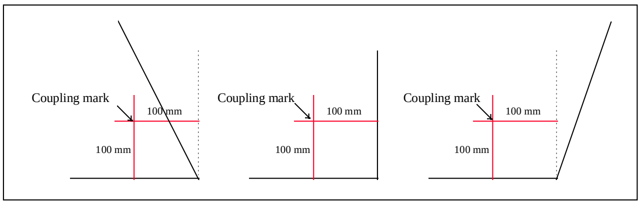
Each ship section is in the programming environment divided into many (10 - 75) separate cells. A cell typically contains about 2 hours of welding time for one robot. To each of these cells a program is generated with an off-line programming tool named ROBIN, and any of the 12 robots can in principle execute this program. The programs assume a theoretical position of the welding lines (relative to the equipment). Before execution of the program, the correct position of the welding lines has to be measured and the program modified accordingly.

By measuring the position of the complete ship block, the positions of all weld lines are implicitly given. In order to find the position of the ship block, it is sufficient to know the co-ordinates of just 2 (actually 1½) measuring points in both co-ordinate systems, because it is assumed that the measuring points always have same z-value (only moving in x,y-plane). These points are called "coupling marks" and are placed in all regular corners of the baseplate of the shipblock[1]. The co-ordinates of the coupling marks in the CAD file (which forms the basis for the programs) are pre-written in the top of each program. When the position of these marks in the robot co-ordinate system is known, the transformation of the programs can be calculated.

A coupling mark is basically just a little hole in the baseplate. Any kind of (rotation symmetric) template with a spike in the centre can easily be placed exactly over the coupling mark. So the problem to be solved by the vision system can be expressed simply as:

Determine the position of a template in a fixed (6 metres) distance from a movable camera with a known position.

[1] The distance from the marks to the edges of the baseplate is ideally 100 mm (see figure below) but in practice the distance to the edges is not well defined while the distance to the welding lines is very accurate, and that is the important measure for this application. The reason for the imprecise distance to the edges can be found in bad production techniques. The cutting machinery especially is error prone.



Coupling marks for most common kind of corners.

6.3.1 Overview

For the specific problem in B4 the equations of 3D-2D reconstruction could be simplified by making the approach that the plane in which the cameras are moving is assumed parallel to the plane where the templates can be placed[1] (Fig 6-2). A fixed viewing distance (z_0) reduces the backward transformation matrix to a 3x3 matrix, and requires only calibration points at the same fixed distance from the camera.

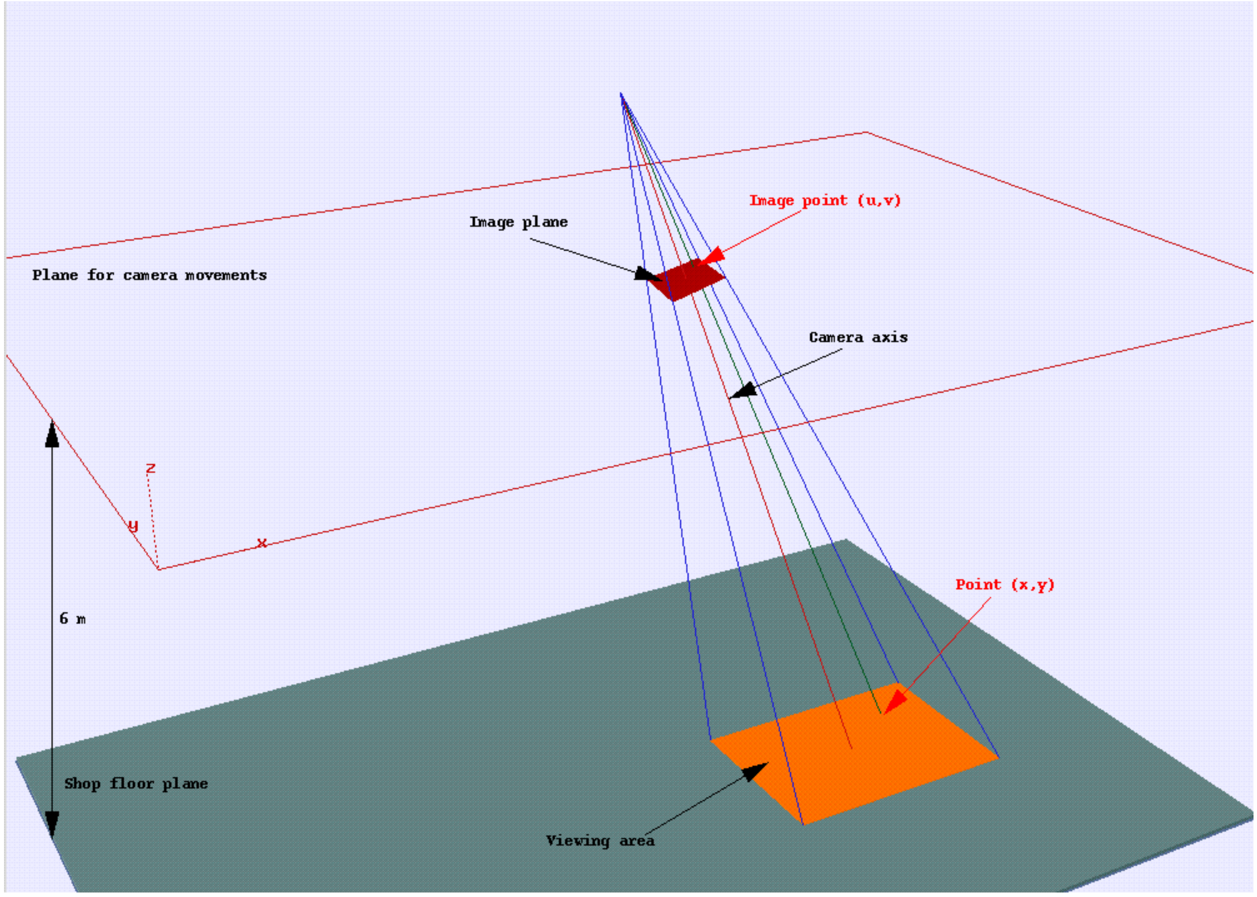


Fig 6-2. Fundamental arrangement of vision-based object identification

6.3.2 Construction of a 2D-2D transformation matrix

The equations (in homogenous co-ordinates) are as follows:

EQ. 6.1
$$\text{image po int}: \begin{pmatrix} u \\ v \\ t \end{pmatrix} \quad , \quad \text{world po int}: \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

The transformation matrix **C**, mapping two-dimensional world points to corresponding two-dimensional image points satisfies the equation:

EQ. 6.2
$$\mathbf{C}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ v \\ t \end{pmatrix} \quad , \quad \mathbf{C} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{21} & C_{22} & 1 \end{bmatrix}$$

[1] Note that it is not an requirement that the image plane should be parallel with the plane containing the templates. But the angles between these planes have to be constant.

expanding:

EQ. 6.3
$$\left.\begin{aligned} C_{00}x + C_{01}y + C_{02} &= u = Ut \\ C_{10}x + C_{11}y + C_{12} &= v = Vt \\ C_{20}x + C_{21}y + 1 &= t \end{aligned}\right\} \Rightarrow$$

EQ. 6.4
$$\begin{aligned} C_{00}x + C_{01}y + C_{02} - UC_{20}x - UC_{21}y - U &= 0 \\ C_{10}x + C_{11}y + C_{12} - VC_{20}x - VC_{21}y - V &= 0 \end{aligned}$$

The equations in EQ. 6.4 have 8 unknowns. The minimum number of calibration points is therefore 4 for this special case of image mapping.

Let

EQ. 6.5
$$\mathbf{A} = \begin{bmatrix} x^1 & y^1 & 1 & 0 & 0 & 0 & -U^1x^1 & -U^1y^1 \\ 0 & 0 & 0 & x^1 & y^1 & 1 & -V^1x^1 & -V^1y^1 \\ x^2 & y^2 & 1 & 0 & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x^N & y^N & 1 & 0 & 0 & 0 & -U^Nx^N & -U^Ny^N \\ 0 & 0 & 0 & x^N & y^N & 1 & -V^Nx^N & -V^Ny^N \end{bmatrix}, \mathbf{B} = \begin{pmatrix} U^1 \\ V^1 \\ U^2 \\ \vdots \\ U^N \\ V^N \end{pmatrix}, \mathbf{C} = \begin{pmatrix} C_{00} \\ C_{01} \\ C_{02} \\ C_{10} \\ C_{11} \\ C_{12} \\ C_{20} \\ C_{21} \end{pmatrix}$$

giving

EQ. 6.6
$$\mathbf{AC} = \mathbf{B}$$

where each superscript denotes an observation (calibration) point and **C** is the unknown vector we want to find. As previously, the problem is solved by a Least Squares method:

EQ. 6.7
$$\mathbf{C} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B}$$

Having determined the transformation matrix **C**, it is possible to find a point in the real world on basis of its imaged position:

EQ. 6.8
$$\left.\begin{aligned} C_{00}x + C_{01}y + C_{02} &= u = Ut \\ C_{10}x + C_{11}y + C_{12} &= v = Vt \\ C_{20}x + C_{21}y + 1 &= t \end{aligned}\right\} \Rightarrow$$

EQ. 6.9
$$x = \frac{b_0c_1 - b_1c_0}{a_0b_1 - a_1b_0}$$
$$y = \frac{a_1c_0 - a_0c_1}{a_0b_1 - a_1b_0}$$

where

EQ. 6.10
$$\begin{aligned} a_0 &= C_{00} - UC_{20} \\ b_0 &= C_{01} - UC_{21} \\ c_0 &= C_{02} - U \\ a_1 &= C_{10} - VC_{20} \\ b_1 &= C_{11} - VC_{21} \\ c_1 &= C_{12} - V \end{aligned}$$

The expressions for x and y in EQ. 2.5 are sometimes called the equations of radiation.

For the special case of parallel planes giving a quadratic transformation matrix, it would probably have been easier simply to find the transformation **K** from the image plane to the real world:

EQ. 6.11

$$\mathbf{K}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ t \end{pmatrix}$$

However, the gain in simplicity is not very high and by using the traditional method the programs are tuned for the full 3D case which soon will be the next implementation task.

6.3.3 The template

The template to look for was constructed to be rotation symmetric (see Fig 6-3) and the size (diameter: 120 mm) was chosen so that the 8 templates would cover the viewing area uniformly and densely (see also Fig 6-14 and App-fig 3 in the appendix).

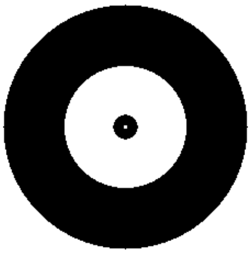


Fig 6-3. The template

A rotation symmetric template has several advantages. The method used for finding the template in the image is based on cross correlation. With a symmetric template, it is not important how the camera is rotated as long as the camera axis is approximately perpendicular to the template plane. The total aspect ratio is not 1:1, which can be seen from Fig 6-4 where a vertical and a horizontal slice through a template in an image is compared with best-fit template. It clearly shows that the image is stretched a little more vertically than horizontally and therefore the best choice of a (circular) template has to be a compromise between the horizontal and the vertical size of the imaged template. In fact, Fig 6-4 is very efficient in finding the best size of the template for cross correlation.

If the planes are not perpendicular, it will result in a stretching of the image. And this stretching will not necessarily follow the (u,v)-axes of the image plane. But knowing that the planes are roughly parallel (< 5°), simple calculations show that it will only alter the aspect ratio by up to 0.4%. This is negligible compared to the aspect ratios measured in the transformation matrices (up to 10%).

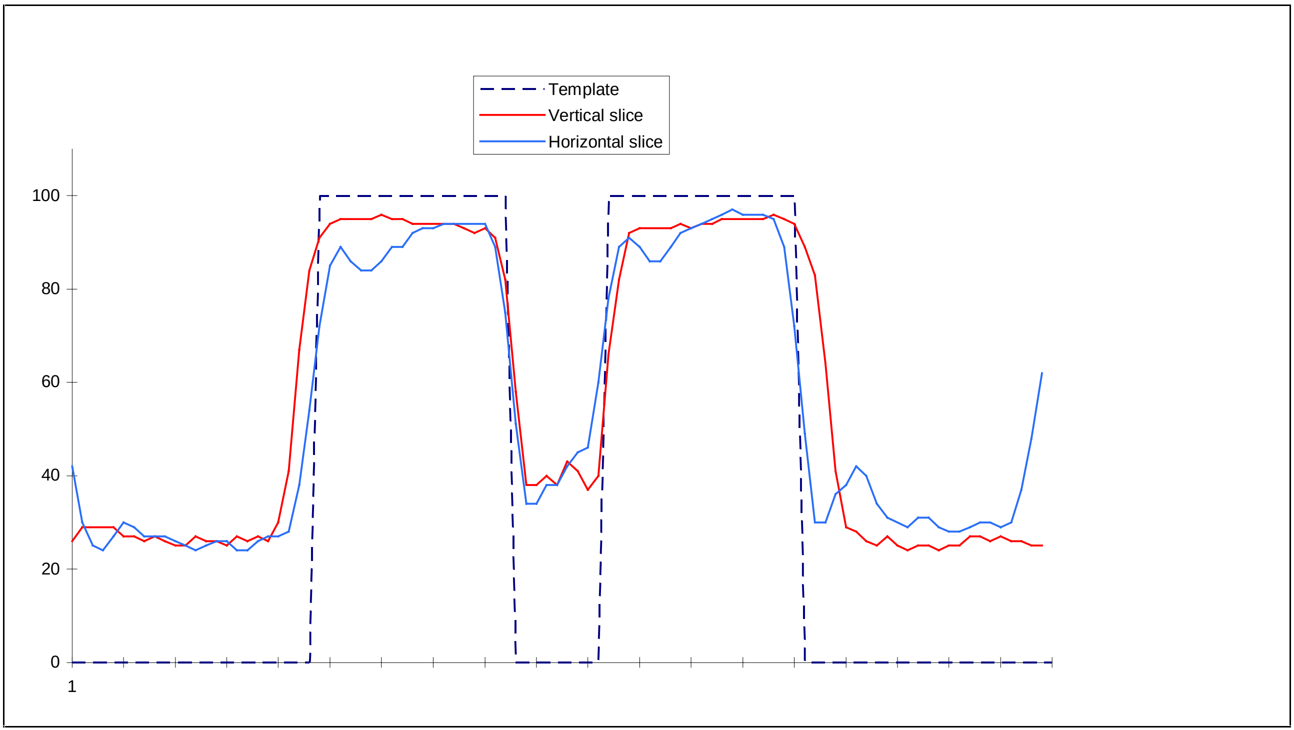


Fig 6-4. Vertical and horizontal slices of template compared with best size of template.

The stretching of the template results in a non-rotation symmetric correlation peak (see Fig 6-5). However, with fixed rotational axes of the camera between calibration and measuring, the cross correlation peak will have the same shape from calibration to measurement. A positioning method based on cross correlation (and sub pixel estimation) is therefore still consistent and applicable. By use of sub-pixel estimation, it is reasonable to estimate an accuracy of 1 millimetre per pixel. As will be shown later the basic resolution in a typical image is approximately 3 millimetres. And if the back projection matrix is based on an over-determined set of equations, the estimate on the accuracy of the algorithm is probably better than 1 millimetre. However in practice, it is not important whether the accuracy is 1 or 3 millimetres since much bigger error sources are disturbing the result.

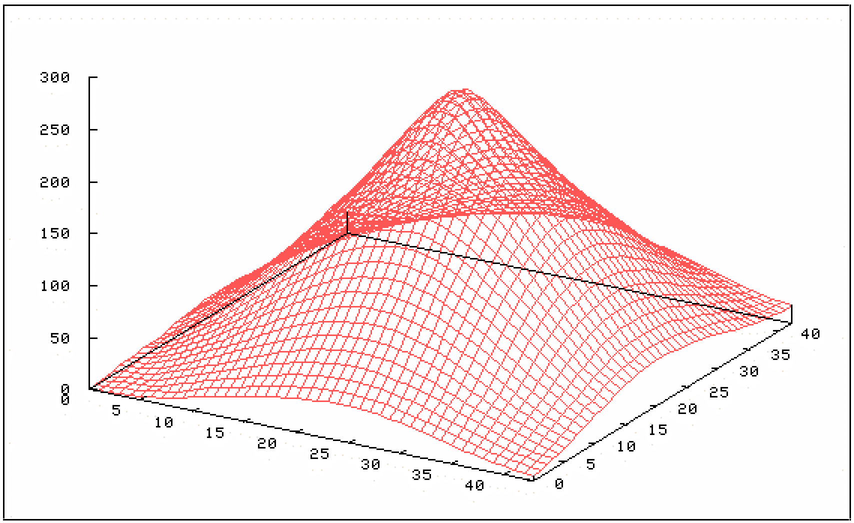


Fig 6-5. GNU plot of cross correlation between template and image

6.3.4 Distortions in the transformation matrix

In a typical 3x3 transformation matrix, the scaling of the x and y axes can be calculated to:

$$Scale(x) = \sqrt{C_{00}^2 + C_{10}^2} = 1.59$$

$$Scale(y) = \sqrt{C_{01}^2 + C_{11}^2} = 1.46$$

Which gives a total aspect ratio of $$Scale(x) \big/ Scale(y) = 1.087$$

and a skewness angle of $$Skew = \frac{\pi}{2} - cos^{-1}\left\{\frac{\bar{x} * \bar{y}}{|x| * |y|}\right\} = 0.23°$$

This skewness[1] is mainly a consequence of the fact that the camera is not transforming the originally rectangular pixels into exactly quadratic pixels, and partly a consequence of not having exact parallelism between the image plane and the plane containing the template. Fortunately, the linear transformation matrix can compensate for skewing as well as scaling, if all parameters are allowed to move freely.

The justification for using a 3x3 matrix instead of a 2x2 matrix for rotation plus a (x,y) displacement vector is that the 3x3 matrix can absorb all linear errors and also the non-linear perspective transformation which is represented in the lowest row of the homogeneous matrix. Perspective transformation does not appear in a 2x2 matrix. In other words, if it was assumed that a 2x2 matrix would do the job equally well we should expect a 3x3 matrix to look like Mat 6-1 (below) or at least like Mat 6-2, whereas a "real" transformation matrix typically looks like Mat 6-3.

$$\begin{bmatrix} cos(a) & -sin(a) & -dx \\ sin(a) & cos(a) & -dy \\ 0 & 0 & 1 \end{bmatrix}$$

Mat 6-1. An ideal transformation

$$\begin{bmatrix} C_{00} & C_{01} & -dx \\ C_{10} & C_{11} & -dy \\ 0 & 0 & 1 \end{bmatrix}$$

Mat 6-2. A less ideal transformation

$$\begin{bmatrix} 4.4 \cdot 10^{-3} & 6.8 \cdot 10^{-1} & -1.5 \cdot 10^{4} \\ -6.2 \cdot 10^{-1} & 2.1 \cdot 10^{-3} & 5.1 \cdot 10^{3} \\ -1.5 \cdot 10^{-6} & 8.0 \cdot 10^{-6} & 1 \end{bmatrix}$$

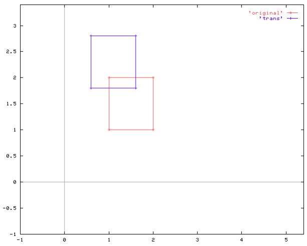
Mat 6-3 A real transformation matrix

The semantics of the 3x3 transformation matrix is taken from Ballard & Brown[xx] and described in Mat 6-4. The effect of the various transformations is illustrated by some examples in the following figures.

Scale X	Skew	Trans X
Skew	Scale Y	Trans Y
Perspective distortion		1

Mat 6-4. The semantics of the 3x3 transformation matrix

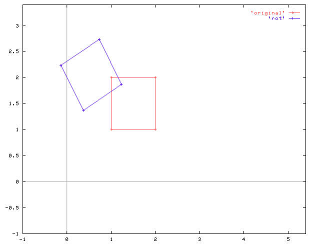
Translation: $$\begin{bmatrix} 1 & 0 & -0.4 \\ 0 & 1 & 0.8 \\ 0 & 0 & 1 \end{bmatrix}$$



[1] NOTE: When speaking about skewness in this context, I am talking about the overall skewness of the system. It cannot directly be compared to the skewness defined in chapter 2.

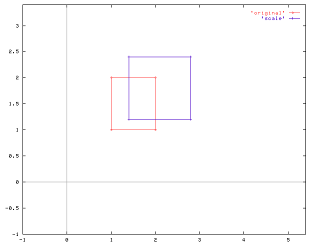
Rotation:

$$\begin{bmatrix} 0.87 & -0.5 & 0 \\ 0 & 0.87 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



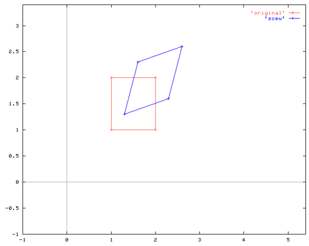
Scaling:

$$\begin{bmatrix} 1.4 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Skewing:

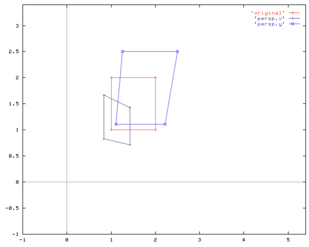
$$\begin{bmatrix} 1 & 0.3 & 0 \\ 0.3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Perspective distortion:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.2 & 0.2 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0.1 & -0.1 & 1 \end{bmatrix}$$

(reduce) (enlarge)



The examples given above shall only serve as guiding tools. As shown previously, the transformation matrix is not divided so nicely into the basic transformations, but the overall picture is valuable to have when making a first reading of a transformation matrix.

6.4 The Mechanics

6.4.1 A test installation

As working conditions were very difficult at the production site, a temporary test bench was built up in the office (Fig 6-6). The main advantage was that there, in contrast to the production site, the camera and lenses could be adjusted whenever needed, without having to wait for a stop in the production or the availability of the right staff (electricians etc.) or necessary equipment.

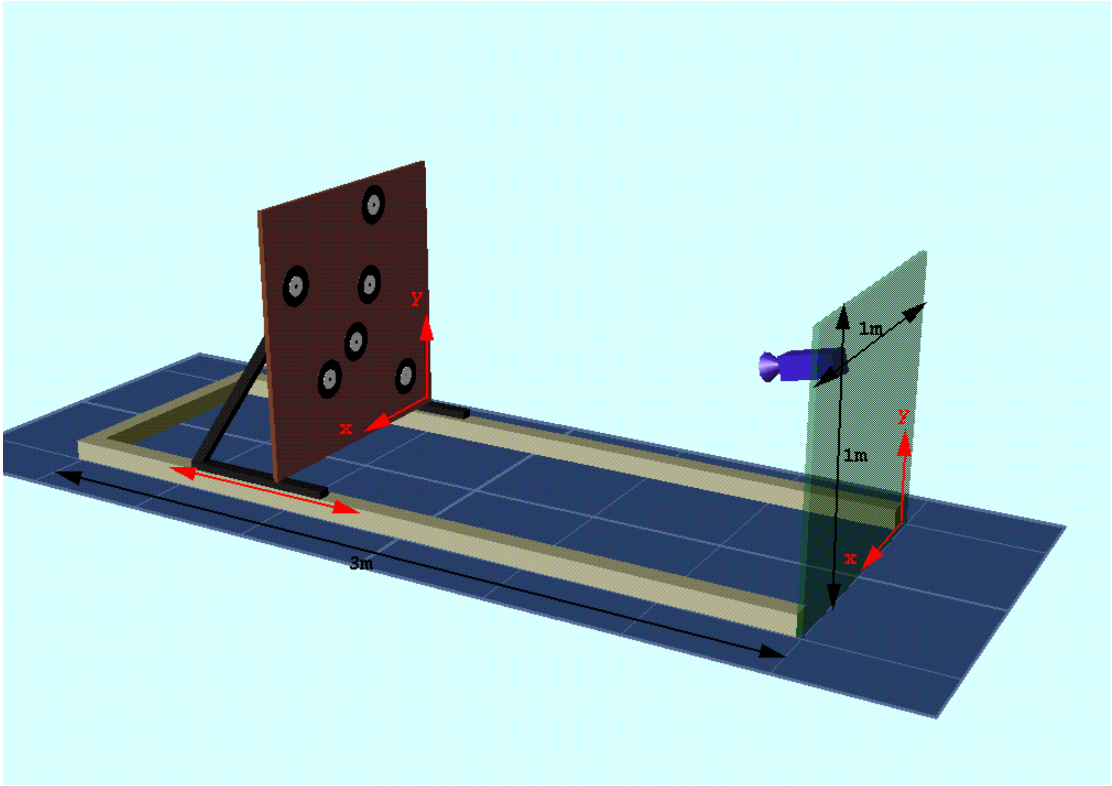


Fig 6-6. The test bench

The test bench was constructed with 5 degrees of freedom (DOF) in order to test all possible situations that we could expect to meet at the production site. The only missing DOF was the unimportant rotation of the camera along the camera axis. The camera was mounted on an old drawing board, giving the x and y moveability. The mounting was constructed with two links making it possible to twist the camera in 2 planes perpendicular to the drawing board and perpendicular to each other. The distance to the plate with the templates could be changed to any value from 0 to 3 metres. The idea was to apply this facility for calculating some of the internal parameters of the camera; but for the actual application it was not used. Nevertheless, the ability to change viewing distance keeping all other parameters fixed was extremely valuable for other analyses.

The viewing distance in the test bench is roughly 1/3 of the distance at the production site so in order to imitate reality as well as possible, the templates used here were also 1/3 of the real templates (keeping same lenses).

Instead of using the controller for getting the position of the camera, the drawing board was covered with graph paper making it possible to read the camera position with accuracy better than 1 mm. Also the plane containing the templates was covered with graph paper. The templates were made with 4 small marks at the circumference of the outer black circle. Between each mark was exactly 90° (see Fig 6-7).

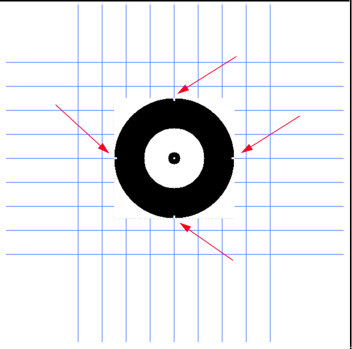


Fig 6-7. Placing of template

This little trick made it possible to place the template extremely accurately on the drawing paper. This again implied that it was possible to check the complete module without worrying about the accuracy of the control data. It should be noted that there were no requirements on the relative positions of the origos of the co-ordinate systems in the two graph papers. Also the angle between the (x,y)-axes of the systems was free. The only requirements were that the planes should be parallel (constant z), and the orientation of the camera had to be fixed from calibration to measurement (the viewing direction is not necessarily completely perpendicular to the planes).

The test bench served its purpose very well with one exception: The moving of the camera to another (x,y) position was mechanically too unstable, giving rise to several millimetres inaccuracy in the final result. But it did not prevent us from performing the test, and the accuracy of the method could be estimated by using a fixed camera position and moving the templates instead. For future tests, however, it would be worthwhile to invest in a better device for moving the camera.

The results derived from the test bench were very promising. The accuracy of the method was extremely high. All templates were detected with an accuracy of 0.1 mm, even with calibration matrices based on only 4 templates.

6.4.2 The B4 installation

The goal of this installation was to build a mechanical system able to weld ship sections with dimensions as big as 32 metres in length, 22 metres width and 6 metre high, and also allow sections with a height up to 12 metre to pass under the gantries. This task was solved by building a flexible system with 12 minor (11 metres long) gantries hanging down from 3 big (37 metres long) beams, mounted across the complete production hall at a height of 17 metres (see Fig 6-1, Fig 6-8 and Fig 6-9).

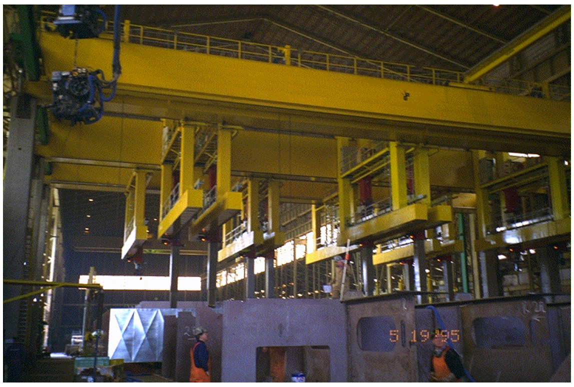


Fig 6-8. The height of the installation is 17 metres

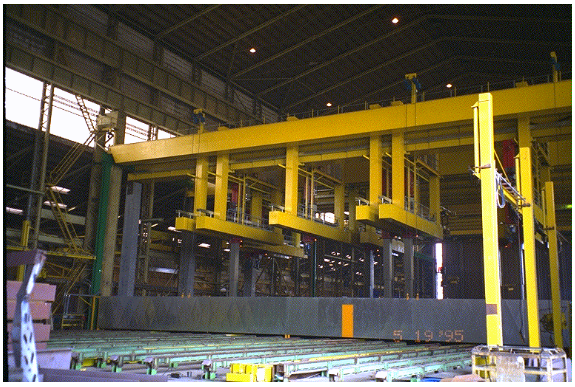


Fig 6-9. The width of the installation is 32 metres.

Fig 6-10 shows a diagram of the installation. Each gantry carries a 6 metres long vertical beam from which a robot is suspended. Due to the fact that the vertical beam is quite heavy and always hanging down, it is possible to control it only by use of wires. All gantries are autonomous production units, which work completely independent of each other, and all of them get fed by programs from a main computer, which takes care of distributing the right programs to the correct production unit at the right time.

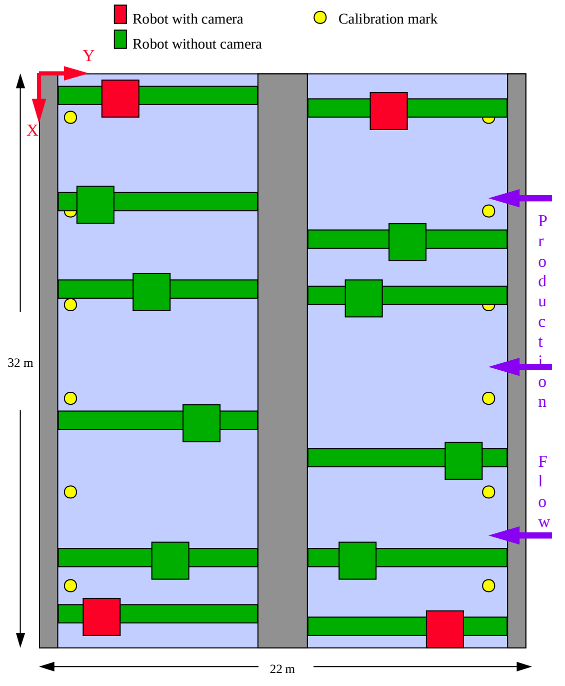


Fig 6-10. Diagram of robot installation

The 4 robots placed in the corners are each equipped with a camera and a laser. These cameras are used for determining the position of the coupling marks. The camera (and the laser) is mounted on the cart moving along the y-axis (6 metres above the floor). It is thereby possible to move each camera in x and y and cover nearly all of the work area. The camera and the laser are mounted very tightly on the cart, and the cart is moving with high accuracy, keeping the orientation of the camera and the laser fixed. The laser beam is nearly parallel to the camera axis. This means that the laser spot always appears at the same position in the image 6 metres away.

The lasers are installed as a guiding tool for the users on the shop floor. When the laser spot is near the coupling mark, both will be in the viewing area. It is important to notice that there is no requirement on the accuracy of the positioning of the laser, except that it should not be placed directly over the template, because its high luminance will disturb the image. The lasers can as a fallback solution also be used for measuring the position of the coupling marks although the accuracy, the performance and the automation level are significantly lower.

The ship sections going through the production line are blocks with straight welding lines and easy accessibility. The size of the blocks can vary a lot, from blocks filling up the entire workspace down to blocks of 10 x 3½ metres, in which case up to 7 blocks will be placed under the robots at the same time. On Fig 6-11 a possible (large) B4 ship section is shown.

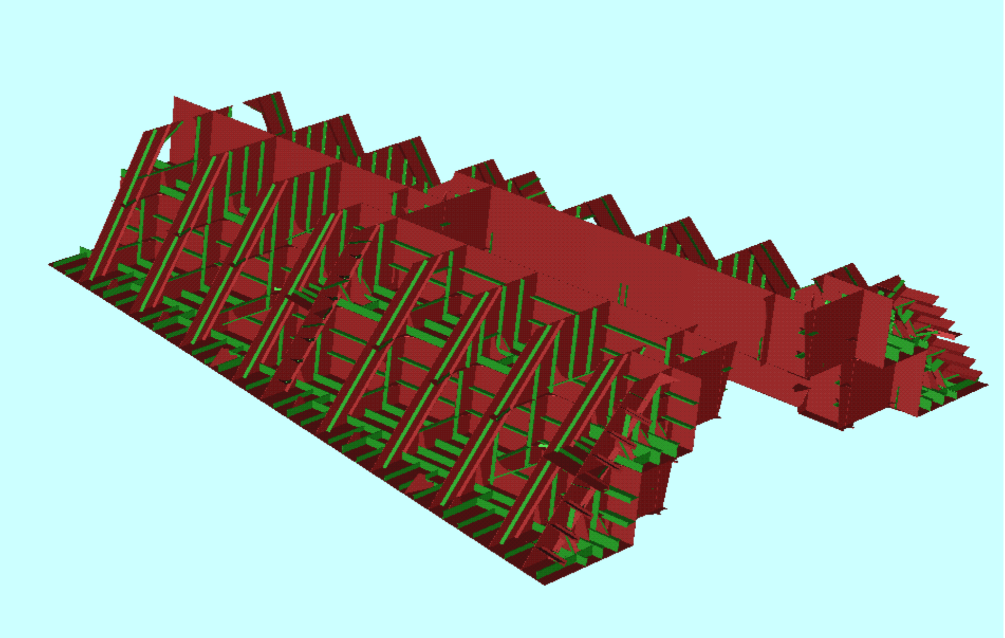


Fig 6-11. Typical block for welding in B4.

6.4.3 Calibration of mechanics

6.4.3.1 Monmos

For such a big installation (96 axes) where a lot of subsystems have to work together, it is extremely important to be able to calibrate the system frequently and with an accuracy as high as possible. For this purpose a theodolite-like system **Monmos** was used to identify strategic calibration marks where the robots could go and check their internal co-ordinates (Fig 6-10). The position of these marks was measured by Monmos with an accuracy estimated to about 3 mm (see Table 6.1).

1st Measurement				2nd Measurement							
x	y	z		x	y	z		Dx	Dy	Dz	
31018	22277	-3986		31016	22279	-3986		2	-1,3	-0,1	
27106	22277	-3993		27105	22281	-3994		0,7	-3,7	0,5	
22559	22279	-3999		22555	22280	-3998		4,6	-1,8	-1,4	
17963	22280	-3997		17957	22278	-3997		5,6	1,7	0,5	
13104	22279	-3990		13099	22276	-3990		4,8	2,3	0	
8202,9	22284	-3993		8197	22283	-3993		5,9	1,6	0,1	
31096	1789,6	-3996		31098	1787,8	-3997		-2,4	1,8	0,4	
27206	1788,7	-3996		27206	1785,6	-3998		-0,3	3,1	1,1	
22511	1789	-3997		22507	1786,9	-3998		3,2	2,1	1,5	
18014	1787,7	-3993		18010	1783,4	-3994		3,6	4,3	1	
13107	1790	-3998		13105	1786,2	-3998		2	3,8	0,4	
7950,1	1790,5	-3994		7947,9	1786,6	-3994		2,2	3,9	-0,1	
							Mean	2,6583	1,4833	0,325	
							Std. dev.	2,4854	2,4928	0,7387	

Table 6.1 Monmos measurements

The measurements in Table 6.1 only give a rough estimation of the accuracy of Monmos, but as we will see later , the contribution from Monmos to the total error is limited and therefore a more exhaustive analysis has been omitted. Besides, the error introduced by the Monmos measurements is almost always caused by human interference. If the same position is measured several times without moving Monmos, the variation of the values is of the magnitude 0.2 mm. The numbers in Table 6.1 also indicate that the changes from first to second measurement are caused by human error rather than some kind of Gaussian noise.

6.4.3.2 Deflection and other error sources

In robot systems 2 kind of accuracies are considered:

- **Absolute:** The accuracy with which the robot is able to reach an arbitrary position in space. In other words: How precise is the robot when moving away from the calibration position?
- **Repeatability:** The accuracy with which the robot is able to reach the same position repeatedly. In other words: How precise is the robot when approaching the calibration area again?

For the installation in B4 the accuracy of the system is important at 3 positions:

1. At Tool Centre Point (TCP) where the welding is performed.
2. At the laser spot on the ground; this is nearly the same as the TCP but the accuracy is better.
3. Six metres above floor where the camera and the laser are mounted. Only x,y-error.

In Table 6.2 the accuracy of these three positions is listed. The measurements are not done for me or by me. They are done as part of the complete calibration of the system and the number of measurements is not high enough to give a complete overview of the errors in terms of mean value and standard deviation. The values indicated are worst case in a series of maximum 4 observations. The reason for this rather low number of values is again the time constraints, considerations for production and a balance between “need to know” and “nice to know”.

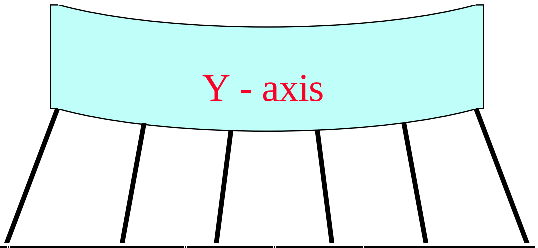
	ABS		REL		ABS	REL
	x	y	x	y	total	total
TCP	±20	±20	±15	±15	±30	±20
Spot	±15	±15	±10	±10	±20	±15
X,Y	±5	±2	±1	±½	±5	±1

Table 6.2 Worst case accuracy of installation at important positions.

The physical reasons for the main errors comes from several sources:

Some backlash exists in the gear wheels of the x- and y-axes. Backlash generally just acts as an introduction of random noise in the measurements and can be very difficult to handle. The best thing to do is actually to use good materials, which do not corrode, get worn, get skewed or malfunction in any other way.

Despite the enormous dimensions of the installation, deflection is still a big error source in the x-direction as well as the y-direction. Deflection can never be avoided fully, but the good thing about deflection is that it is predictable and smooth, so normally it is not a problem that can not be handled, as long as the bending isn’t too big or too elastic. For instance in Fig 6-12 and Fig 6-13, the difference between the y-value at the rack (6 metres above floor) and at the floor is measured for the laserspot. The reason for only having positive differences lies in the choice of reference and difficulties in defining when the laser beam is pointing directly down. The argumentation remains valid: deflection is predictable, smooth and reproducable.



Movement at floor	0	2269	4408	6993	8778	9676
Movement at rack	0	2268	4404	6984	8766	9660
Deviation	0	1	4	9	12	16

Fig 6-12. Deflection along the y-axis.

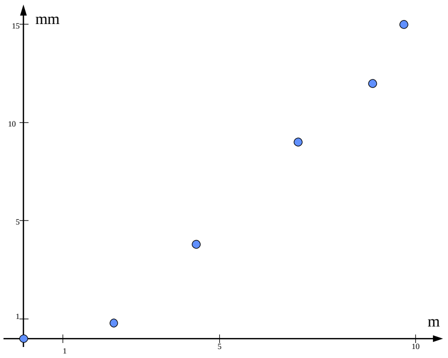


Fig 6-13. Deviation as function of y.

Vibration is in same family as deflection. It can not be avoided completely, but as long as the devices are not moving too fast, and (when needed) the vibration is allowed time to fade out, it is not a severe problem.

On the y-axis, another error source is important to be aware of. When the cart is moving along the rack, it is twisting a fraction of a degree. This causes on the floor a shift of several millimetres along the x-axis[1]. Intensive work has been performed in order to map this deviation as a function of the position (x,y), but to be brief, the twisting demonstrated the same behaviour as the deflection. This basically means that the same twisting is reproduced when the cart is approaching the same position again. This is all I need to know!

6.4.3.3 Comparison with laser point method

Calibration of the system has by far been the most time-consuming single task in the whole installation. Several sources contribute to the uncertainty of the calibration. In order to illustrate the advantage of using cameras, I will describe a manual way of measuring a position on a ship section and compare that with the vision-based method.

As mentioned before, near each camera a laser source is mounted. The laser beam is nearly parallel to the optical axis of the camera. When the laser spot points at a position on the ship block, this position can be calculated when knowing the (x,y)-distance Δ_{laser} between gantry TCP ($(x,y)_{\text{controller}}$) and the laser spot. Δ_{Laser} is measured by moving the laser-spot to a Monmos measured calibration mark on the floor and calculating the difference between $(x,y)_{\text{Monmos}}$ and $(x,y)_{\text{controller}}$:

$$\Delta_{\text{Laser}} = (x,y)_{\text{Monmos}} - (x,y)_{\text{controller}}$$

Using the results from page 104 and 105, the method has the following uncertainty contributions (the uncertainty in the z direction is not important):

[1] A twisting of 0.1 degree will move a point at a distance of six metres away 10 mm! Generally, the biggest change in x across the y-axis was of magnitude 10-15 mm.

Laser point based measurement	
Uncertainty source	**Estimated uncertainty (mm)**
Monmos measurement of calibration mark	>3
Manual positioning of the laser spot at calibration mark	>3
Manual positioning of the laser spot at measured point	>3
Absolute x,y positioning of laserspot at measured point	20
Total:	>20 mm

Table 6.3. Error contributions when using laser-based measurements

From Table 6.3 it can be seen that the overwhelming source of uncertainty originates from the deflection of the beams and the twisting of the cart on the y-axis. The twisting makes the telescope axis, the laser beam and the optical axis of the camera move a little. This twisting is a non-linear function of the positioning in x,y of the cart. However, tests have shown that the **relative** accuracy of the laserspot is significantly higher, in other words the twisting is the same for the same position; and the changes are smooth when moving away. This is also what would be expected, and we are now able to estimate the accuracy of the vision-based measuring method (Table 6.4).

Vision-based measurement	
Uncertainty source	**Estimated uncertainty (mm)**
Monmos measurement of 4 calibration marks	1
Repeated x,y positioning of view centre at measured point	10 (max.)
Pixel identification	1
Total:	ca. 10 mm

Table 6.4. Error contributions when using vision-based measurements

The point is that the complete work area is covered with a grid of calibration positions (gridsize ≅ 1 metre), thereby moving the focus from the absolute accuracy to the repeated accuracy of the system. Furthermore, the backward calculation of the image is calculated with the 4 nearest calibration positions and a weighted mean is made. The absolute twisting of the cart is of no interest, as long as it is repeated every time the cart approaches the same position. The twisting is simply absorbed in the local transformation matrix. Another big advantage is that all manual influence has been removed, thereby guarding the system against fatal errors, which inevitably will occur, when human beings are involved. The estimation of accuracy due to pixel identification is based on the calculations made on page 105. The accuracy of the Monmos measurements is better in Table 6.4 than in Table 6.3 because we here know the exact distances between the calibration marks and can use that for smoothing (see later).

6.4.3.4 Calibration of the complete work area

In order to find the transformation matrix between image points and points in the real world, measured in the robot co-ordinate system[1], it is necessary to know the position of at least 4 points in the image as well as in the real world. More points are an advantage since each extra point contributes to the total accuracy of the transformation and a clear mistake will be easy to detect.

Therefore, a plate with 8 templates was constructed. The templates were placed very accurately (estimated precision of 0.2 mm with the same method as described in subchapter 6.4.1) on a piece

[1] Robot co-ordinate system is short for the global co-ordinate system which the complete gantry system with 12 robots is referring all its movements to. It is NOT the local robot co-ordinate system which each single robot uses for internal movements.

of high-precision graph paper. Ordinary graph paper was considered not to be accurate enough. Due to absorption of humidity from the air, it showed to be incorrect by several millimetres measured over a distance of 1 metre. Now the position of the template was known exactly in a local co-ordinate system (on the plate). In order to find their positions in robot co-ordinates, 4 measuring marks ("reflectors") used by Monmos were also placed on the plate and with the same high accuracy (see Fig 6-14).

In the calibration situation, the plate was positioned in 15 different positions in the work area, and for each position, the 4 marks were measured by Monmos. Then the two-dimensional shift and rotation between the local positions and the global positions of the Monmos marks were found by applying a simple deepest descent algorithm and minimising the RMS between theoretical global points and measured global points:

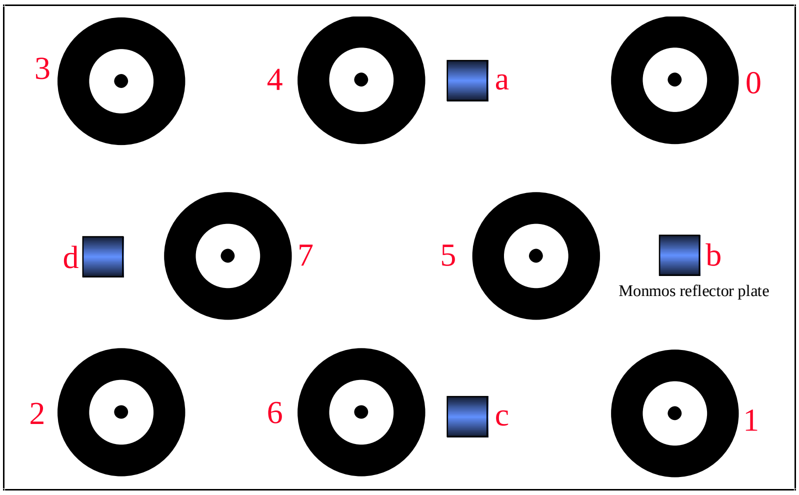


Fig 6-14. The calibration plate.

EQ. 6.12

$$RMS = \sqrt{\frac{1}{4}\sum_{i=1}^{4}(\mathbf{A}\mathbf{x}_i^{local} - \mathbf{x}_i^{measured})^2}$$

where **A** is the matrix representing the transformation and rotation

and $\mathbf{x}_i$ is the coordinate (x,y) of Monmos mark number i.

The best fitting transformation between local measured Monmos points and global measured Monmos points is then used to calculate the global positions of the 8 calibration marks. Since the internal distances should be unaltered, it is easy to see how good Monmos is measuring a point position, at least relative to the other points. Some of the distances between the Monmos points on the calibration plate are shown in Table 6.5 together with the exact (measured) values.

	a-b	**b-c**	**c-d**	**d-a**
Graph paper	212.6	212.6	331.1	331.1
Monmos	211.9	212.8	331.4	330.8

Table 6.5. Distances between Monmos reflectors

The algorithm uses one of the Monmos measured world points as an initial guess on (Δx,Δy), and the result of the optimisation is shown in Table 6.6.

	(Δx, Δy)	α	RMS
Start	(4647.6 , 999.5)	0.00	155.44
End	(4487.7 , 1004.8)	-1.90	0.55

Table 6.6. Result of transforming local data to global.

In Table 6.7 smoothed positions of the Monmos marks are shown. Notice how the deviation from the measured values is distributed around zero.

Monmos	Corrected	Dx,Dy
(4647.7, 999.5)	(4647.6, 999.5)	(0.1, 0.0)
(4492.4, 1144.7)	(4493.2, 1144.7)	(-0.8, 0.0)
(4656.9, 1279.3)	(4656.5, 1279.6)	(0.4, -0.3)
(4952.1, 1129.5)	(4951.9, 1129.3)	(0.2, 0.2)

Table 6.7. Corrected positions of the Monmos marks

Applying the optimised matrix on the 8 templates yields the global positions that are shown in Table 6.8.

Local	Global
(x,y)	(x,y)
(0,0)	(4487.7, 1004.8)
(0,280)	(4497.0, 1284.6)
(460,280)	(4956.8, 1269.4)
(460,0)	(4947.5, 989.5)
(260,0)	(4747.6, 996.2)
(130,140)	(4622.3, 1140.4)
(260,280)	(4756.9, 1276.0)
(360,140)	(4852.2, 1132.8)

Table 6.8. Transformation of templates to global coordinates

6.4.3.5 Interpolation of calibration positions

A measuring procedure is never done exactly on the same spot as a previous calibration procedure. There will always be a shifting vector Δ. So in order to find a point (x, y) on the shop floor in robot co-ordinates on the basis of an identified print (u,v) in the image, it is necessary first to perform the backward transformation on the image, and then add to the resulting world point the vector Δ. When more than one calibration position is available, each of these calibrations can give an estimate of the position of the point (x,y). In theory they will all give the same value but in practice they differ, and the bigger the distance from calibration to measuring, the less reliable the result. The program **bilinear** uses a set of rules to make the best estimate of the position:

1. If the measuring position is close (< 1 metre) to a calibration position, then use only that position.
2. If that is not the case: use linear or bilinear interpolation of the estimates from the involved calibration positions.
3. If that is not possible: use only the nearest calibration position.

Some examples might be appropriate:

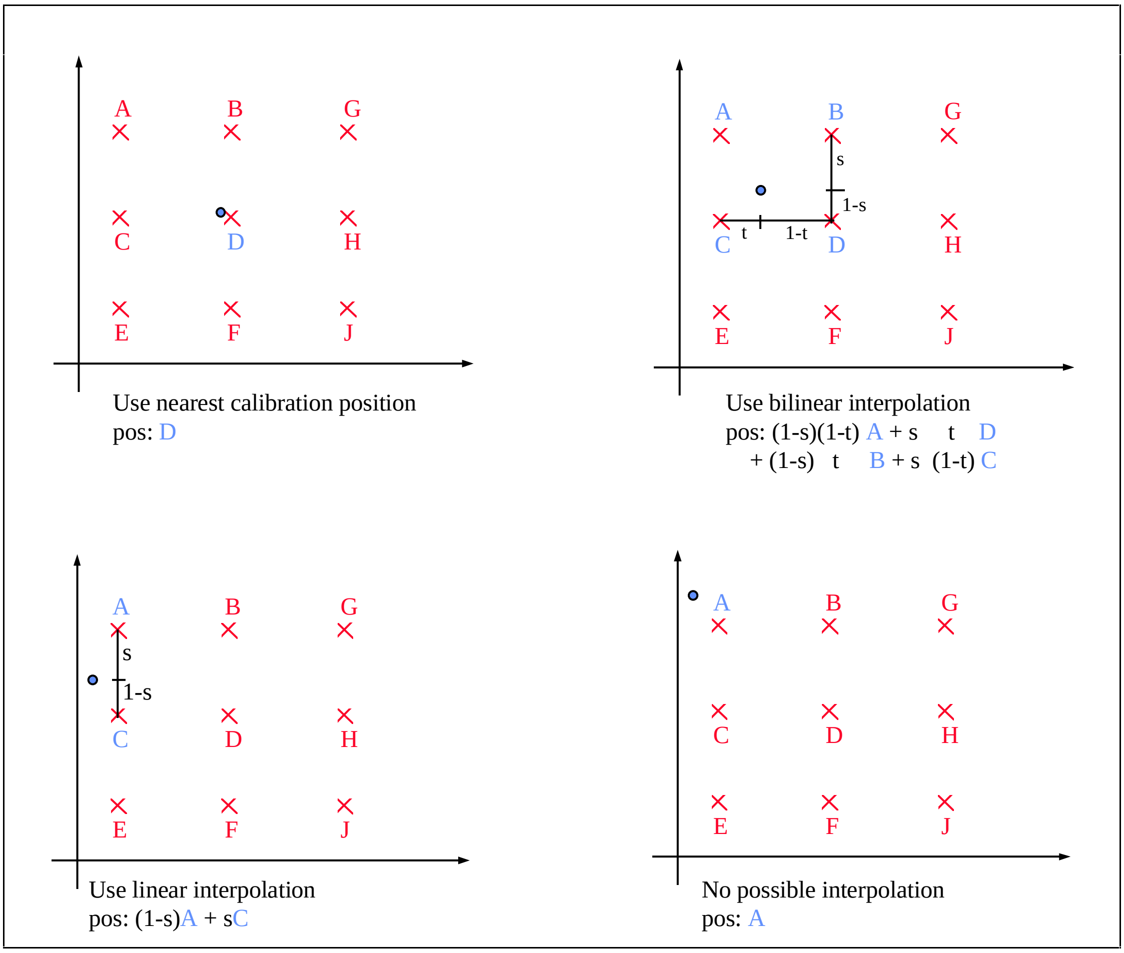


Fig 6-15. Examples of how to use the calibration grid

The calibration points do not (as shown in Fig 6-15) lie on a perfect grid, so in a measuring situation the ideal (rectangular) bilinear interpolation has to be replaced with the general formula for bilinear interpolation:

EQ. 6.13
$$f(x,y) = ax + by + cxy + d$$

Here the only requirement is that there are 4 points with known values *f(x,y)* to interpolate with. We then have 4 equations with 4 unknowns *(a,b,c,d)*, which can be solved, after which any point in the region can be estimated.

6.4.3.6 Fast re-calibration and measurement

Calibrating the complete work area is a most time consuming and tedious task. The 4 x 15 placings of the calibration plate, followed by the Monmos measurement, requires 3 hours of work for 2 men, plus the occupation of the expensive equipment for the same period. So clearly, such a calibration can only be done once, otherwise the concept is useless. Fortunately it is not necessary to do such a global calibration more than once. The reasons for that are the same as given in 6.4.3.2 “Deflection and other error sources”; the repeatability accuracy is high giving a static deflection pattern. The high precision of Monmos, when not moved between measurements, makes it possible to do a very efficient calibration of the complete robot work area, and at the same time to reduce all future calibrations to a simple procedure, which furthermore can be completely automated. And as a spin-off, it was possible to produce a complete mapping of the deviations in x and y of the robot as a function of its position (x,y). With one placing of Monmos it was possible to measure all positions one robot can reach, and the accuracy with which the 15 positions of the calibration plate was measured, was around 0.2 mm. With the method of interpolation between calibration points, it is possible to compare any point in the work area with a chosen reference position and from that to split the error contributions into mechanics and vision+Monmos. The procedure is a bit difficult to describe but an example is given below.

The work area of robot 12 is approximately 1.6-11.6 metres in y direction and 4.5-13.0 metres in x direction. Some of the 15 calibration positions are shown in Fig 6-16. In each calibration position, the exact positions of the templates are known (from Monmos). If one calibration position (e.g. position d) estimates the position of one of the templates in e.g. calibration position b (position of camera: (5640,6321) and position of indicated template: (5077,4763)), there will be a deviation (here (+2,+13)) from the correct (Monmos) value. If the opposite estimation is calculated as well, we get nearly the same result with negative sign (here: (-1,-14)). From that result, it can be concluded that the mechanical error (twisting and deflection etc.) between these two positions is 1-2 millimetres in x and 14-15 millimetres in y, and the accuracy of Monmos + Vision is around 1 mm.

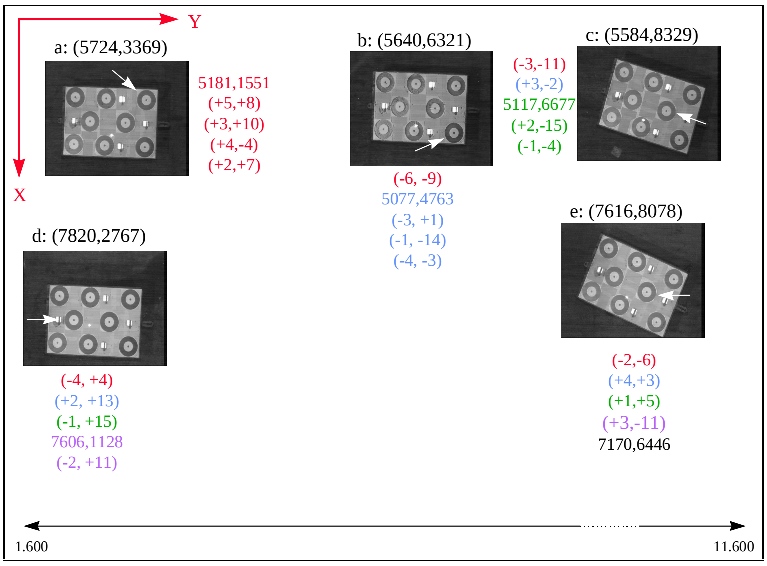


Fig 6-16 Calibration positions for robot 12

Such comparisons can of course be done between any two pairs of calibration positions thereby giving a complete map of the mechanical error contributions and some statistics of the Monmos measurements plus the vision-based template identification. In Table 6.9 and Table 6.10, some other measurements (from robot 6) are shown, and we see an average camera error of ca. half a pixel and a worst-case mechanical error of 18 millimetres.

	12.8	15.0	17.7	19.8	22.3
4.5 m	X		X	X	X
9.0 m		X			
13.0 m	X			X	X

Table 6.9 Selected positions in working area

Pos	Deviation 1	Deviation 2	Total error	Camera error
X	(0.7,-6.4)	(-1.3,6.9)	(1.0,6.7)	(0.6,0.5)
X	(-13.0,-8.2)	(12.8,8.0)	(13.0,8.1)	(0.2,0.2)
X	(0.9,17.5)	(-0.9,-18.4)	(0.9,18.0)	(0.0,0.9)
X	(3.2,4.0)	(-2.7,-3.6)	(3.0,3.8)	(0.5,0.4)

Table 6.10 Selected positions in working area

Such calculations have been done for all 4 robots mounted with a camera and a complete result (for robot 6) is shown in Table 6.11.

X		4.5					9.0					13.0				
	Y	**12.8**	**15.0**	**17.7**	**19.8**	**22.3**	**12.8**	**15.0**	**17.7**	**19.8**	**22.3**	**12.8**	**15.0**	**17.7**	**19.8**	**22.3**
4.5	**12.8**	0.0,0.0	-7.1,5.7	-12.4,7.5	-6.4,14.1	0.7,14.9	-0.4,4.9	-1.3,6.9	-8.0,9.0	0.6,10.9	0.1,20.6	-0.2,-2.7	-6.3,1.0	-11.2,4.5	-9.3,10.7	0.2,15.9
	15.0	7.1,-5.7	0.0,0.0	-5.2,1.8	0.8,8.4	7.8,9.2	6.8,-0.9	5.8,1.2	-0.8,3.3	7.7,5.2	7.2,14.9	6.9,-8.4	0.8,-4.7	-4.1,-1.2	-2.2,5.0	7.3,10.1
	17.7	12.6,-7.1	5.3,-2.0	0.0,0.0	6.1,6.9	13.3,7.5	12.4,-3.1	11.3,-1.0	4.7,1.1	13.4,2.9	12.6,12.8	12.3,-10.0	5.9,-6.5	1.3,-2.9	3.3,3.2	12.8,8.0
	19.8	6.5,-14.0	-0.8,-8.8	-6.1,-6.8	0.0,0.0	7.2,0.7	6.3,-9.8	5.2,-7.8	-1.3,-5.7	7.4,-3.9	6.5,6.0	6.3,-16.9	-0.1,-13.3	-4.8,-9.8	-2.7,-3.6	6.8,1.2
	22.3	-0.5,-15.4	-8.2,-9.7	-13.5,-8.1	-6.9,-1.2	0.0,0.0	-3.1,-11.0	-4.1,-8.7	-10.9,-7.4	-2.5,-5.7	-3.5,4.5	-0.9,-18.4	-7.6,-14.0	-12.6,-11.4	-10.6,-5.5	-1.8,0.5
9.0	**12.8**	-0.3,-4.6	-7.4,0.7	-12.7,2.7	-6.7,9.4	0.6,10.1	0.0,0.0	-0.9,2.0	-7.5,4.3	1.2,6.1	0.4,15.8	-0.2,-7.3	-6.5,-3.9	-11.2,-0.2	-9.3,6.1	0.4,10.9
	15.0	0.7,-6.4	-6.5,-1.2	-11.8,0.8	-5.8,7.5	1.6,8.3	0.9,-2.0	0.0,0.0	-6.6,2.3	2.1,4.1	1.3,13.9	0.7,-9.2	-5.6,-5.7	-10.4,-2.1	-8.4,4.2	1.3,9.0
	17.7	7.4,-8.7	0.1,-3.4	-5.1,-1.4	0.9,5.3	8.2,6.0	7.5,-4.2	6.6,-2.2	0.0,0.0	8.7,1.9	7.9,11.6	7.4,-11.4	1.0,-8.0	-3.7,-4.3	-1.7,2.0	7.9,6.8
	19.8	1.9,-8.2	-5.4,-3.1	-11.2,-1.4	-5.0,5.7	1.5,6.4	-0.5,-5.4	-2.4,-3.2	-8.8,-1.7	0.0,0.0	-1.4,10.3	0.4,-11.3	-6.4,-7.3	-10.9,-4.7	-8.8,1.1	0.0,6.2
	22.3	-0.1,-19.3	-8.0,-14.5	-13.2,-12.4	-6.9,-5.5	0.7,-5.2	0.0,-16.0	-1.1,-14.2	-7.6,-11.9	1.3,-10.2	0.0,0.0	-0.3,-22.6	-7.2,-19.1	-11.5,-15.5	-9.3,-9.4	0.4,-5.0
13	**12.8**	0.1,2.8	-7.1,8.1	-12.4,10.0	-6.4,16.8	0.9,17.5	0.2,7.2	-0.8,9.2	-7.4,11.4	1.3,13.2	0.5,23.0	0.0,0.0	-6.3,3.5	-11.0,7.1	-9.0,13.4	0.6,18.2
	15.0	6.3,-0.7	-0.9,4.6	-6.2,6.5	-0.1,13.3	7.1,14.0	6.4,3.7	5.4,5.8	-1.1,8.0	7.5,9.8	6.8,19.6	6.3,-3.5	0.0,0.0	-4.8,3.6	-2.8,9.9	6.8,14.7
	17.7	11.1,-4.4	3.8,0.9	-1.4,2.9	4.6,9.6	11.9,10.3	11.1,0.1	10.2,2.1	3.6,4.3	12.3,6.2	11.5,15.9	11.0,-7.1	4.7,-3.6	0.0,0.0	1.9,6.3	11.6,11.1
	19.8	9.9,-9.9	2.4,-4.9	-3.0,-2.9	3.2,4.0	10.1,4.7	8.8,-6.3	7.3,-4.2	0.8,-2.4	9.5,-0.7	8.6,9.5	9.0,-13.0	2.5,-9.2	-2.1,-6.0	0.0,0.0	9.2,4.9
	22.3	-0.6,-15.5	-7.8,-10.2	-13.0,-8.2	-7.0,-1.5	0.2,-0.8	-0.5,-11.0	-1.4,-9.0	-8.0,-6.8	0.7,-4.9	-0.1,4.8	-0.6,-18.2	-6.9,-14.8	-11.7,-11.1	-9.7,-4.8	0.0,0.0

Table 6.11 Deviations between measurements and Monmos values for robot 6.

For better readability, some of the measurements in pairs have been marked. The deviations in Table 6.11 can be illustrated graphically and the result is shown on Fig 6-17 and Fig 6-18.

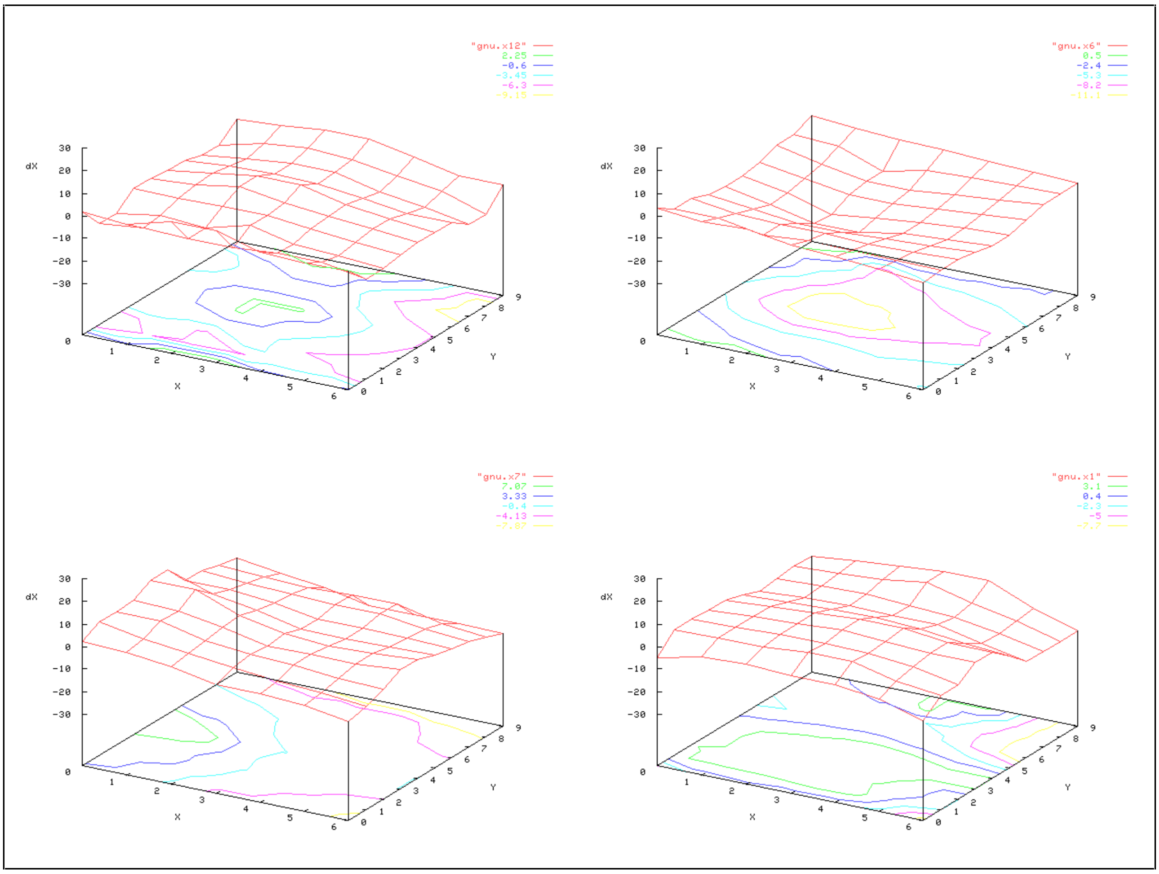


Fig 6-17 Deviations in x for robot 1,6,7 and 12

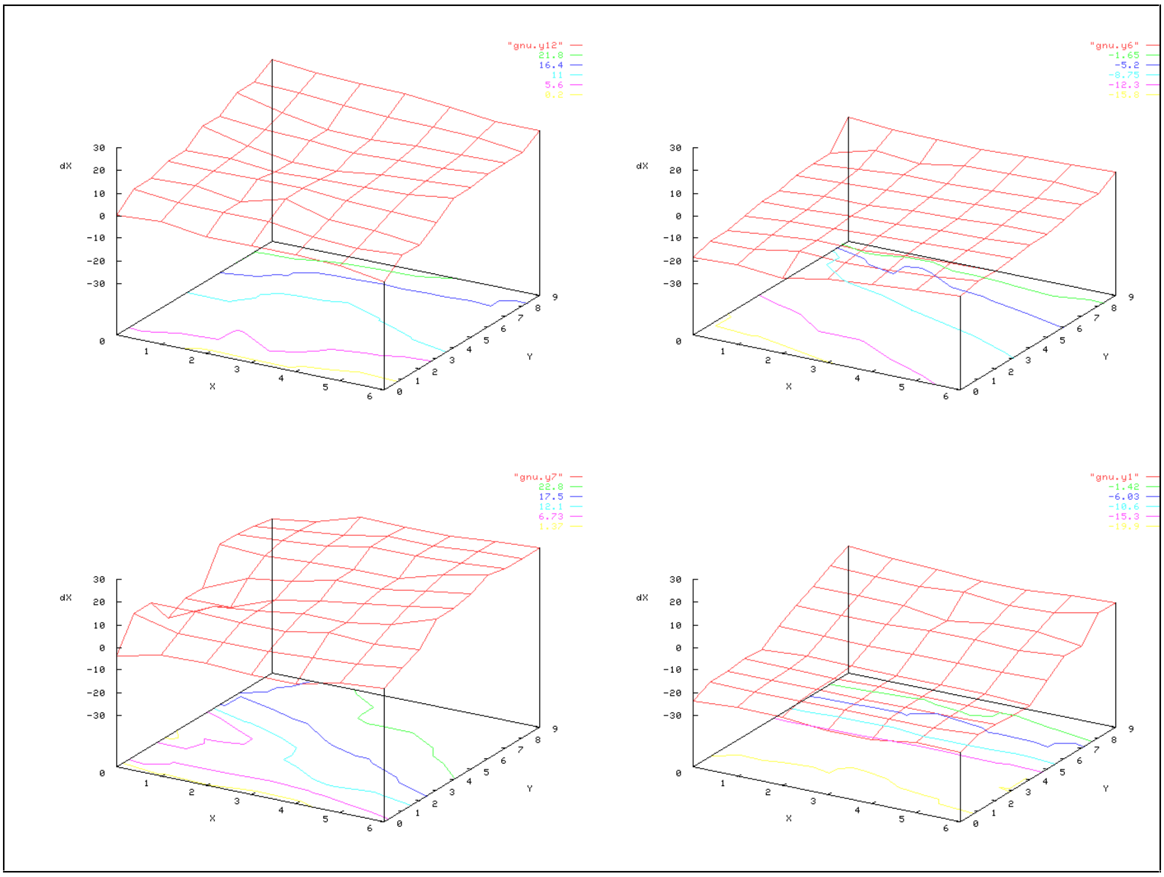


Fig 6-18 Deviations in y for robot 1,6,7 and 12

The big advantage is that this description of how the mechanics deviates as a function of the position is constant in time. This means that a measurement anywhere in the work space is done with just one calibration matrix, and the offset in x and y is added according to the values shown in on Fig 6-17 and Fig 6-18. In addition, if a re-calibration should be necessary, the only thing to do is to measure the position of one calibration point in the work area and compare that with the Monmos measured value. The offset between Monmos and the vision-based measurement will be a constant to add all over the work area (together with the offset from on Fig 6-17 and Fig 6-18).

6.4.4 Data transfer

For the transfer of programs from the offices to the cell controller on the shop floor, standard Ethernet (based on fibre optics) is used. The programs used in the vision module are physically on the same UNIX workstation as the cell controller. For practical reasons all communication with the robot controllers has to be transferred wireless. This also includes the images transferred from the cameras to the frame grabber in the workstation. Several attempts were made before finding a reliable and robust method for data transfer.

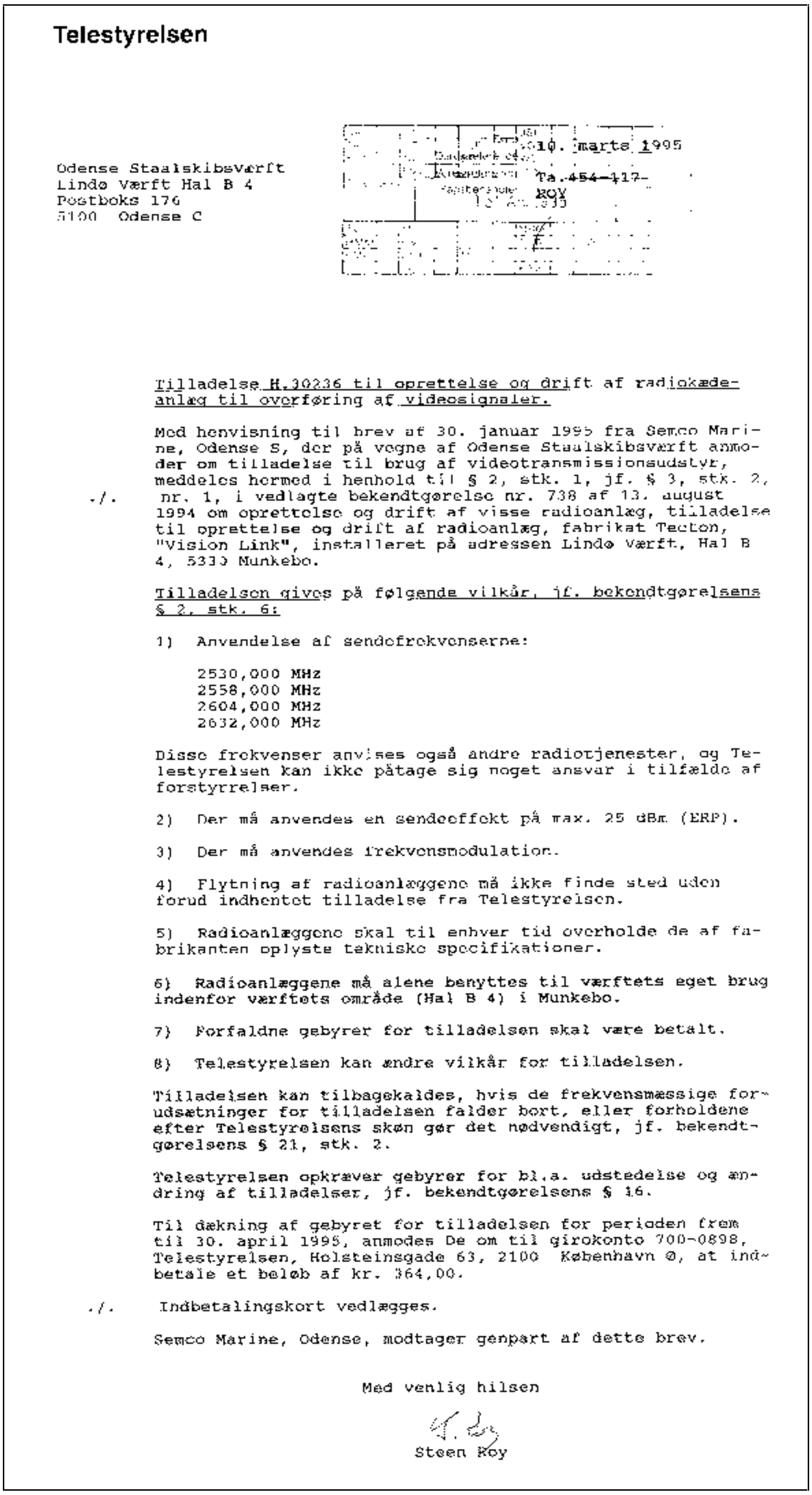
The first attempt was based on laser technology, but it showed to be too sensitive if the laser beam did not exactly hit focus on the receiver.

Secondly, infra-red transmission of the video signal was tried; a method which is commonly used in many other areas, especially for surveillance. However, the quality of the image was simply not good enough for doing accurate image processing.

Finally, it was decided to use so-called micro-waves (~2500 MHz). Unfortunately, this was also by far the most expensive solution, and it required a special licence from the Danish P&T (see Fig 6-19).

Nevertheless, the Yard was generous and the goal was achieved; the transmission of video signals is reliable and of a very high quality.

The position-transformed programs are sent to the separate robot controllers via another wireless system also based on micro-waves. The reason for not using the same device is that the two applications have very different needs. The bandwidth of wireless vision is big but one-way, while the cell controller needs to have a two-way dialogue on a small bandwidth with each robot controller.



Telestyrelsen

Odense Staalskibsværft
Lindø Værft Hal B 4
Postboks 176
5100 Odense C

10. marts 1995

Ta.454-117-
ROY

Tilladelse H.30236 til oprettelse og drift af radiokædeanlæg til overføring af videosignaler.

Med henvisning til brev af 30. januar 1995 fra Semco Marine, Odense S, der på vegne af Odense Staalskibsværft anmoder om tilladelse til brug af videotransmissionsudstyr, meddeles hermed i henhold til § 2, stk. 1, jf. § 3, stk. 2, nr. 1, i vedlagte bekendtgørelse nr. 738 af 13. august 1994 om oprettelse og drift af visse radioanlæg, tilladelse til oprettelse og drift af radioanlæg, fabrikat Tecton, "Vision Link", installeret på adressen Lindø Værft, Hal B 4, 5330 Munkebo. ./.

Tilladelsen gives på følgende vilkår, jf. bekendtgørelsens § 2, stk. 6:

1) Anvendelse af sendefrekvenserne:

2530,000 MHz
2558,000 MHz
2604,000 MHz
2632,000 MHz

Disse frekvenser anvises også andre radiotjenester, og Telestyrelsen kan ikke påtage sig noget ansvar i tilfælde af forstyrrelser.

2) Der må anvendes en sendeeffekt på max. 25 dBm (ERP).

3) Der må anvendes frekvensmodulation.

4) Flytning af radioanlæggene må ikke finde sted uden forud indhentet tilladelse fra Telestyrelsen.

5) Radioanlæggene skal til enhver tid overholde de af fabrikanten oplyste tekniske specifikationer.

6) Radioanlæggene må alene benyttes til værftets eget brug indenfor værftets område (Hal B 4) i Munkebo.

7) Forfaldne gebyrer for tilladelsen skal være betalt.

8) Telestyrelsen kan ændre vilkår for tilladelsen.

Tilladelsen kan tilbagekaldes, hvis de frekvensmæssige forudsætninger for tilladelsen falder bort, eller forholdene efter Telestyrelsens skøn gør det nødvendigt, jf. bekendtgørelsens § 21, stk. 2.

Telestyrelsen opkræver gebyrer for bl.a. udstedelse og ændring af tilladelser, jf. bekendtgørelsens § 16.

Til dækning af gebyret for tilladelsen for perioden frem til 30. april 1995, anmodes De om til girokonto 700-0898, Telestyrelsen, Holsteinsgade 63, 2100 København Ø, at indbetale et beløb af kr. 364,00.

./. Indbetalingskort vedlægges.

Semco Marine, Odense, modtager genpart af dette brev.

Med venlig hilsen

Steen Roy

Fig 6-19. Official licence from Danish P & T

6.5 Results

In order to check the stability of the system and to get an idea of what kind of accuracies we are actually dealing with, I have prepared an additional feature which first calculates the distance between 2 vision-measured coupling marks, then finds the corresponding CAD file and calculates the theoretical distance between the marks. The difference between these 2 measures arises partly from inaccuracies in the measuring method (due to inaccurate mechanics) and partly from a badly produced ship baseplate (or badly placed coupling marks). In addition, the angle between the vectors between the coupling marks is calculated because it is a production demand that these vectors have to be close to parallel. In Table 6.12 some results from daily production are shown. The angle is measured in degrees, the rest in millimetres.

Distances between coupling marks						
Cam-1	Cam-2	Corners	Cam-dist	CAD-dist	Diff.	Angle
7	6	2-3	30484.8	30499.4	14.6	0.10
12	7	1-3	18469.0	18466.6	-2.5	0.06
6	1	2-4	18802.7	18838.0	35.2	0.03
6	6	1-2	2467.9	2460.9	-7.0	0.01
12	7	1-3	28614.2	28619.7	5.5	0.04
7	12	1-3	28479.8	28494.0	14.2	0.04
1	1	3-4	7761.3	7759.0	-2.3	0.02
12	7	1-3	31357.1	31354.0	-3.1	0.08
6	12	1-2	15322.1	15332.0	9.9	0.03
7	12	1-3	20358.1	20350.3	-7.8	0.04
12	7	1-3	31060.1	31054.0	-6.1	0.03
12	6	1-2	15319.9	15332.0	12.1	0.00
6	1	2-4	28947.2	28941.0	-6.2	0.03
12	7	1-3	31361.3	31354.0	-7.3	0.02
6	12	1-2	15329.8	15345.8	16.0	0.01
12	6	1-2	15640.9	15646.2	5.4	0.01
12	12	1-2	1896.1	1894.3	-1.8	0.10
12	7	1-3	25591.3	25598.0	6.7	0.02
6	6	1-2	5362.0	5363.0	1.0	0.11
6	1	2-4	27739.5	27747.2	7.7	0.17
12	7	1-3	27736.4	27747.2	10.8	0.21
6	1	2-4	25583.2	25598.0	14.8	0.02
6	12	1-2	19173.6	19179.1	5.5	0.07
6	7	2-3	32825.8	32838.2	12.4	0.05
1	12	1-4	31984.4	31985.9	1.5	0.03
1	7	3-4	20573.0	20569.2	-3.8	0.03
12	7	1-3	25634.4	25635.7	1.4	0.04
6	1	2-4	28601.9	28595.9	-6.0	0.01
12	7	1-4	29673.6	29664.3	-9.3	0.05
12	7	1-3	31051.0	31054.0	3.0	0.01
12	7	1-3	19110.4	19132.0	21.6	0.01
6	1	2-4	19113.4	19132.0	18.6	0.06
12	7	1-3	26372.3	26374.0	1.7	0.00
12	7	2-4	28580.6	28588.5	7.8	0.01
6	1	1-3	28584.7	28588.8	4.1	0.02

Table 6.12 Some results from the camera application in B4.

So what we can deduce from Table 6.12 is that generally the system works fine with some few outliers (which most probably are caused by uncertainties in the mechanics; the baseplate is not wrong in that scale). The average of all measurements is ca. 5 mm. That is somewhat mysterious since it should be expected that the measurements would fall equally around zero. But for some reason the physical measurement is on average 5 mm below the theoretical value coming from the CAD file. I checked if that could be because of the fact that the welding of the baseplate of the block is done after the making of the mandrel holes but before measurement. However, the welding experts have told me that this welding does not introduce any measurable shrinking of the complete baseplate, so for the moment the phenomenon is not explained. The mean deviation is ca. 10 mm, which is very much in line with the estimated accuracy of the mechanical equipment. The accuracy of the measurement is generally inside the required tolerances, and the extreme low angles show that the workers are quite good at placing these giant blocks very accurately with respect to rotation.

6.6 Outlook

The objectives of the implementation work have been achieved. It is possible to use cameras for object localisation in production sites at Odense Steel Shipyard. Some minor but important improvements of the production flow are very much recommended. For instance, the cameras should very soon also be able to automatically calibrate the complete mechanical system, just by frequently taking a number of images of a well-known set of calibrations marks fixed permanently on for instance the shop floor.

It should also be possible to move the cameras via the monitor and a remote control without having the user leaving the cell controller.

The cameras are looking for templates in form of a white circle. These white circles are manually put on the block with a piece of chalk. Each circle is placed exactly over a little spot made by a mandrel. This is ridiculous, since the only reason for putting the small spot is to indicate where the white circle shall be placed. So when the right tool for putting marks on the plate is found, the mandrel holes will be replaced with direct painting of the circles.

The cameras have recently been connected directly to their respective robots via the cell controller interface. That means that it is now impossible to combine a robot position with a false camera; an error which frequently occurred in the past.

With the mentioned improvements in automation and the introduction of online calibration, it is possible to do a simple measurement of torsions induced by the welding. The measuring positions of the cameras before welding will be stored and reused after welding. New measurements are performed automatically and the deviations can be estimated.

7. Conclusion

The complete project has been a success. When the project started in 1993 there was absolutely no knowledge, no know-how and no strategy for using vision in production. Instead, there was a big and growing need for new solutions in the areas of quality control, positioning, process monitoring, safety surveillance and other more specific applications. Today, shipbuilding is ready for extensive use of vision. The exceptionally low prices for high quality equipment and the fast development of the cameras, giving better and better resolution, are perfectly timed with the new level of technology that Odense Steel Shipyard has achieved via foresighted and massive investments. The perspective of the vision concept at Odense Steel Shipyard is very promising and the most exciting thing about the vision method is the numerous possibilities and very high degree of automation it offers.

This thesis has made the first step towards a wide introduction of vision applications at Odense Steel Shipyard. At the hour of writing many other vision applications are being specified with the intention of being implemented before the end of 1997. In order to illustrate the diversity and extent of the vision-based application, I will briefly describe some of the most important installations which are being/going to be implemented in 97/98.

B13 (1997): A robot welding station equipped with 3 robots shall weld incoming parts as fast as possible. Due to the very low amount of work per item, the total positioning time has to be less than 3 minutes. The problem will be solved with one camera under the roof to make a rough estimation of the position of the various items (up to 4 at the time). The rough estimation is used to guide the robots (each carrying a camera) to the predefined positions where an accurate measurement can be performed. The new aspects in this application are the interaction between cameras, the visual guidance of the robots and the tough performance requirement.

Outdoor stock (1997). Quality control of incoming raw steel plates. The dimensions of the plates are measured outside the factory in order to identify and reject bad steel plates. This vision installation has its challenge primarily in the uncontrollable light conditions.

Cutting factory (1997). Quality control of cut plates. The accuracy of the plasma cutting machines is not always good enough, costing The Yard a lot of money in the periods when bad plates are produced and sent on. The requirement for the vision module is extreme accuracy. In a distance of 11 metres, a plate shall be measured with +/-1 mm. In addition, the plate is wet and dirty.

LASOS (1997). A brand new laser installation has just been installed and a vision module shall do positioning, 2D quality control of the cutting and 3D quality control of the welding. The positioning is easy, but the quality control has to be online connected to the CAD information.

B9 (1998). The most difficult part of the shipyard to automize is situated in the production hall B9. The blocks are very complex and the accessibility is limited. The introduction of a new generation of many-axis robots is closely linked to the use of cameras for guiding and positioning, based on CAD information

B4 (1998-1999). 3D measuring of large ship blocks is the ultimate goal in quality control. The tolerances have to be 1-2 mm and the measurements have to be purely passive. This means a close co-operation with the CAD model and full control of all factors like non-linearity, light conditions, and synchronisation. A very difficult task, which is the final goal of another ATV project, recently started at OSS.

Sale (1998-). The Yard is selling technology, mainly as turnkey solutions. An integrated part of the sales program is vision modules for 2D and 3D positioning.

The project has achieved nearly all the objectives set up. In chapter 2, a thorough and well-founded presentation of the theory for camera calibration is given. Furthermore straightforward formulas for calculating the model parameters have been developed, and the stability of these parameters has been examined and explained. The results in chapter 2 are crucial and form the basis of the work presented in chapter 3 and chapter 6.

In chapter 3 the breakthrough from world to CAD is presented. Here we see how it is possible to create synthetic images nearly indistinguishable from real images. As demonstrated from the examples described above, the applicability of vision is heavily increased when vision can be accompanied with CAD. This project has built the first bridge from the real world back to the CAD world. Many various applications are waiting to be developed, but already now, the results so far are used in practice. For instance when the physical installation is unreachable or maybe even not built, it is now possible to do a completely realistic simulation that makes the developer able to control and tune his program dramatically earlier than otherwise. This is a most valuable (and originally not recognised) feature. In a typical industrial installation, the vision module starts getting real images on almost the very same day the production equipment is taken into use. Enormous pressure is put on the vision-developers in order to finish their work almost on the same day as they get the first images. With the programs developed in chapter 3, it is now possible to experiment with the layout (distance, orientation, lens etc.) and to create 100 percent realistic images several months before production starts. This provides the developers with enough time to refine their work.

Chapter 4 presents a new, robust method for finding templates with subpixel accuracy. The program has been used for several years now and has never failed. Besides its own ability to find features with high accuracy, one of the big advantages of the method is that it is not based on edge detection. Therefore, the program offers an independent method for checking applications (calibration routines) which are based on edge detection.

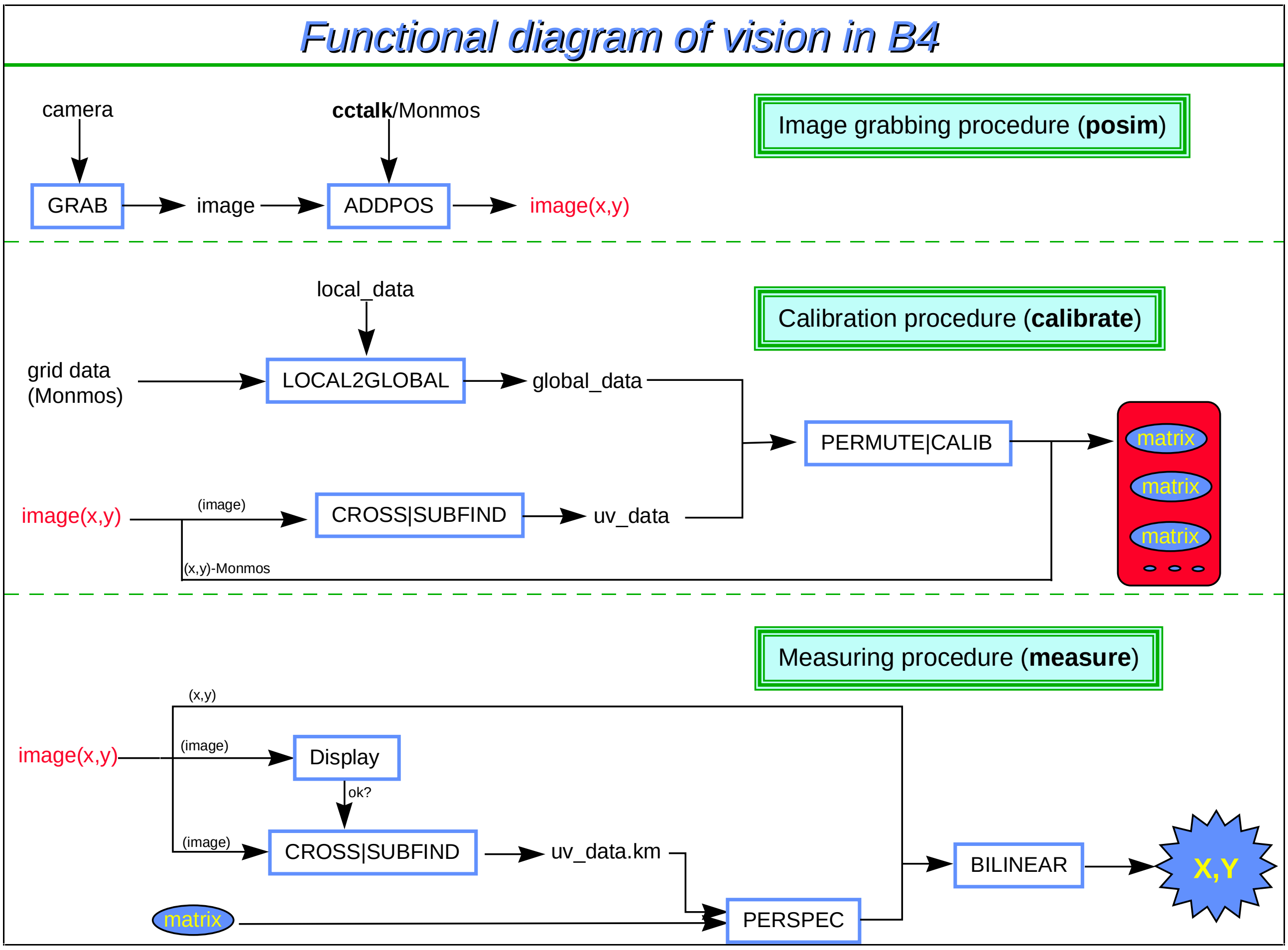
Chapter 5 presents an overview of the experience and know-how collected during the project. The collection of experience was another important objective of the project that has been fully met.

Chapter 6 presents the first result of the preceding scientific work. The implementation in B4 was extremely valuable for the project. A lot of solutions (and mistakes) were tested, and the constant demand from the production staff has put an extra dimension on the implementation experience achieved.

The last but not least important contribution from this dissertation is the new awareness of all developers at Odense Steel Shipyard that vision is a tool which should always be considered and which very often offers the best solution to a specific problem.

Appendix A: The B4 Programs

A functional diagram of the interacting programs used in B4 is shown below.



App-fig 1 Functional diagram of program interactions

The work consisted of 2 tasks, calibration and measuring, plus the basic task of constructing a method for grabbing an image and obtaining information about current the position of the camera. Clearly this method had to be constructed with high robustness and reliability, since it would be the key for getting information to the system during calibration, as well as measuring. The image grabbing procedure was split into two functions:

Image grabbing

For practical reasons, the calibration of the cameras is performed off-line. This means that all images and camera positions from the shop floor are collected on a workstation (HP) at the office. The image and position grabbing procedure is contained in a shell program (**posim**, Prog 1), which first via the cell controller gets the position of the indicated camera. The communication with the cell controller is contained in another shell program named **cctalk** (Prog 2). The returned (x,y)-position of the camera is then used as parameters in the image grabbing programs.

```
#!/bin/bash
# This program "posim" needs 2 parameters:
# $1: Robot number (1,6,7,12)
# $2: Image index

if [ $# = 1 ]
then
 ext=$1
else
 if [ $# = 2 ]
 then
  ext=$1.$2
 else
  echo "USAGE: posim <argument> <argument (optional)>"
  echo "(fx. "posim 12 name" where the number is the robot number)"
  exit 1
 fi
fi

if [ $1 != 1 -a $1 != 6 -a $1 != 7 -a $1 != 12 ]
then
 echo "Wrong robot number (robot $1 has no camera)"
 exit 1
fi
robot_no=$1
echo Camera: $robot_no
echo Image:  $ext

# Program "cctalk" gets info from MBX
string=‘cctalk $robot_no‘
echo String: $string
echo
STATUS=‘echo $string |cut -d" " -f 1‘
echo "STATUS for robot no. $1: $STATUS"

if [ "$STATUS" = "0" ]
then
 X_POS=‘echo $string |cut -d" " -f 3‘
 X1_POS=‘echo $string |cut -d" " -f 4‘
 Y_POS=‘echo $string |cut -d" " -f 6‘
 echo Robot position: $X_POS $Y_POS

# Grab image of plate with 1 mark
# Add position of camera in header of image
# Store image in image.$ext

 grab |add_pos $X_POS $Y_POS >image.$ext

else
 echo "ERROR: Error in reading from robot$robot_no (STATUS = $STATUS)"
 exit 1
fi
```

*Prog 1. Shell program (***posim***) grabbing position and image for off-line calibration*

```
#!/bin/bash

if [ $# = 1 ]
then
 if [ $1 != 1 -a $1 != 6 -a $1 != 7 -a $1 != 12 ]
 then
  echo "NOTE: robot $1 has no camera"
 fi
else
 echo "Wrong no. of arguments ($#)"
 exit 1
fi

robot_no=$1

#
# Flush mailbox for old messages
#
mbx_flush -n p13mbx vision

mbx_open -n p13mbx vision
mbx_put -n p13mbx -r vision robot "read robot$1 camera_pos" >/dev/null

while [ 1 ]
do
        mbx_wait -n p13mbx -d vision >pos_rob$1

        #
        # Check if message is from correct robot
        #
        grep -s "robot$1" pos_rob$1

        if [ $? = 0 ]
        then
                break;
        fi
done

#grep "robot$1" pos_rob$1

# Format of data: //c x read robot$1 camera_pos 123 456
STATUS=‘grep camera_pos pos_rob$1 |cut -d" " -f 2‘
X_POS=‘grep camera_pos pos_rob$1 |cut -d" " -f 6‘
X1_POS=‘grep camera_pos pos_rob$1 |cut -d" " -f 7‘
Y_POS=‘grep camera_pos pos_rob$1 |cut -d" " -f 8‘
echo $STATUS
echo "( $X_POS $X1_POS ) $Y_POS"
rm pos_rob$1
```

*Prog 2. Shell communication program (***cctalk***)*

Grab

A program using the VideoPix framegrabber installed in the SUN workstation. This frame grabber can run only under the old operative system of SUN (not SOLARIS!). The framegrabber is a cheap and well-proven grabber that works fairly well. The most serious problem is its rectangular representation of pixels. This representation is then transformed to quadrants in software, which is not particularly accurate. Fortunately, the way the framegrabber is used in the application diminishes the problem significantly as described on page 97. The program output is a greyscale (0-255) image in PGM raw format with a time stamp in the header. This time stamp (together with **Add_pos**) ensures a reliable book keeping of the images.

Add_pos

A program which takes the PGM raw image from the framegrabber program "GRAB" and adds the position of the camera to the header. Together with the time stamp, this identifies the image completely. A typical image header is shown in Prog 3

```
P5
#Camera pos: 8175.40 21616.90  # grab: Fri. Jul 21 09:28:08 1995?
720 575
255
```

Prog 3. Typical image header (PGM format)

Local2global

This program converts the 8 local templates on the plate to global coordinates as described on page 107.

Cross

Cross correlation is a mathematical expression for measuring the similarity between two images and is calculated as described in EQ. 7.1.

EQ. 7.1

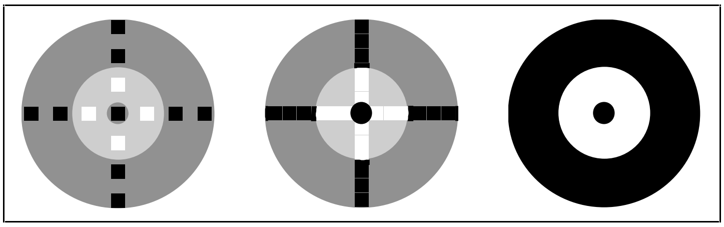
$$CC = \frac{\sum_{i,j} T(i,j) * Im(i,j)}{\sqrt{\sum_{i,j} T(i,j)^2 * \sum_{i,j} Im(i,j)^2}}$$

where T(i,j) denotes a pixel in the template and Im(i,j) a pixel in the image.

Normally cross correlation is calculated using two rectangular images, and very often it is an advantage to transform the images into the Fourier domain before doing the calculations. However, in cases with rotation symmetric templates, it is necessary to do cross correlation on circular images, which means that Fourier transformation is not straightforward. But knowing exactly what the template looks like in the image, a method based on cross correlation in the Cartesian domain can still be applicable. And if you can add that the size of the template is known and the orientation is known or unimportant (rotation symmetric), it is a very reliable and reasonably quick method.

Several techniques can be applied in order to speed up the calculations across the image. Some are mentioned in Brown & Ballard[i] and in the program **cross** the following speed-up tricks are used:

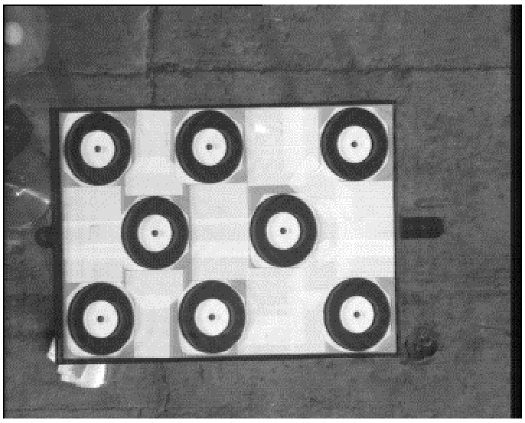
The first sweep through the image is only calculating the cross correlation for some pixels in two perpendicular slices of the template (see App-fig 2, first image). This identifies the regions of interest (ROI) in the image roughly. At the same time, the template is moved through the image in steps bigger than 1 pixel.



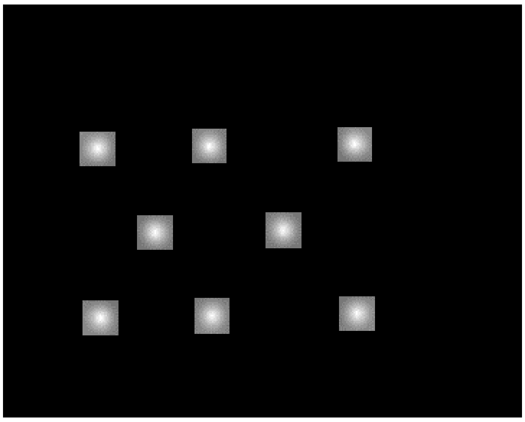
App-fig 2 Subimages of template

The second run-through is using the complete slices of the template (see App-fig 2, -second image) and moves through the ROIs 1 pixel per step. This identifies the exact ROIs.

The final iteration is calculating the cross-correlation between the full template and all positions in all ROIs in the image. A result of **cross** working on a typical image is shown in App-fig 3 and App-fig 4.



App-fig 3 Typical image from a calibration session



App-fig 4 Image in App-fig 3 cross correlated with template

Spot

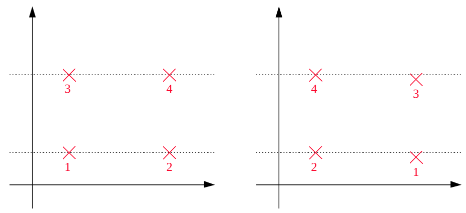
The program **cross** proved to be unnecessarily overprecise and too slow in daily use, so I constructed a much faster and less accurate algorithm which looks for a white spot on a non white background. The algorithm runs through the complete image and for each pixel the gradient in a distance equal to the radius of the circle we are looking for is calculated. This is done in 8 different directions for each pixel. Clearly, the gradient is giving the same result in all directions only around the centre of the spot (see also the image in App-fig 6).

Subfind

This program is based on the algorithm described in chapter 3. It uses the shape of the auto correlation peak for the template to find an analytical expression describing similar kinds of peaks. This analytical peak is then moved around in the cross correlation image until the Root Mean Square is minimal. The idea is that an analytical peak can be moved in arbitrarily small steps giving an estimated accuracy of 0.1 pixel.

Permute

The cross correlation program finds 8 spots in an image used for calibration. The orders in which these points appear are not fully predictable. If for instance we choose to enumerate them according to increasing u-values and (if two or more have the same u-value) according to increasing v-value, the order will very often be different from the order obtained with the method used on the templates in the robot co-ordinate system.



App-fig 5 The order of the points changes with rotation

It was therefore necessary to construct an algorithm that could determine the correct matching of global co-ordinates with image co-ordinates. The program is written by John Immerkær and is based on the fact that the two sets of points must have roughly the same geometrical appearance except for scaling, rotation and translation (here we do not consider small perturbations). The job now is to:

- Choose an order of the points in the image
- Translate the points so that point no. 1 is lying on the same position as template no. 1
- Rotate and scale the points around point 1 so that point 2 is lying on the same position as template no. 2
- Check the rest of the points with the rest of the templates.
- If not OK go to 1.

Unfortunately this kind of problem grows in complexity with O(n!). And for 8 points there are 8! = 40.320 combinations which it took the computer 8 minutes to go through. Clearly an unacceptably long time. But knowing that the relationship between the internal distances should be the same in both reference frames it was possible to pre-sort the orders of points, thereby reducing the execution time to a few seconds (incl. the pre-sorting procedure)

Calib

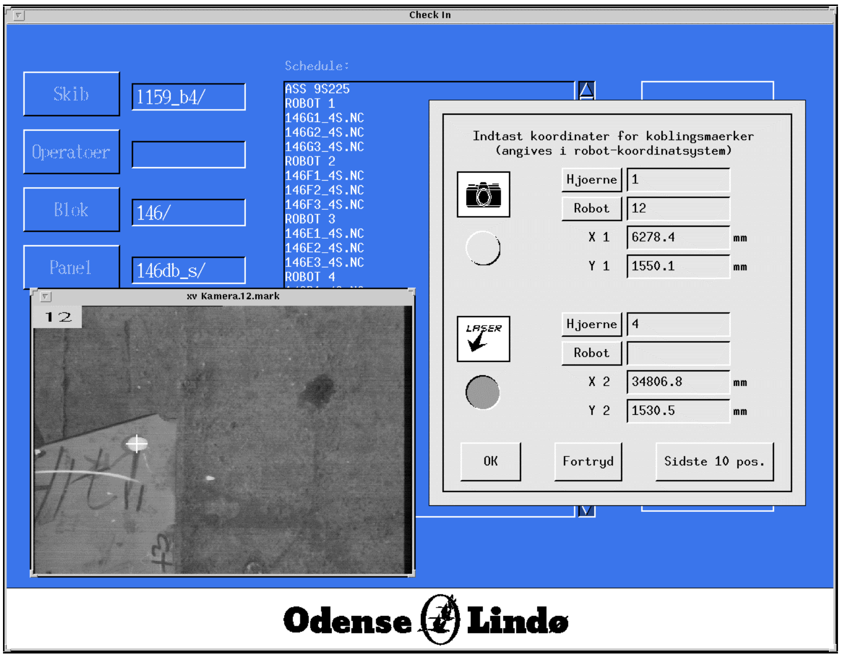
This program is the heart of the procedure for building up the backward transformation matrix. The code is written in awk and the syntax of the program is based on initial work by John Immerkær. The input to the program is global template co-ordinates (produced by **local2global**) and image points (found by **cross** & **subfind** and sorted by **permute**). A typical output from calib is shown in Prog 4.

```
#calibration position: 5680.60 21617.60
# mindiv = 0.0671664
# C:
  5.13464009e-03  8.43469761e-01 -1.85838002e+04
 -7.73028265e-01  6.05514690e-03  4.26073871e+03
 -8.18449183e-07  4.55327705e-08  1.00000000e+00

# Scale(x):     1.29359
# Scale(y):     1.18555
# Aspect ratio: 1.09113
# Skew:         90.0307

#    X      Y            Z  ;   U       V   ;    u+U0      v+V0
# 5051.8 22243.3   0.0 ;  204.3  491.9 ;  204.3278  491.7762
# 5048.4 22523.3   0.0 ;  441.4  496.0 ;  441.2145  496.1058
# 5308.4 22526.5   0.0 ;  445.2  294.4 ;  445.3562  294.5744
# 5508.4 22528.9   0.0 ;  448.4  139.5 ;  448.4912  139.4914
# 5511.8 22248.9    0.0 ;  211.6  134.8 ;  211.5151  135.1547
# 5311.8 22246.5    0.0 ;  208.3  290.2 ;  208.4190  290.2408
# 5180.1 22384.9    0.0 ;  324.7  393.1 ;  324.8301  393.1851
# 5410.1 22387.7    0.0 ;  328.7  215.5 ;  328.4462  214.8717
# SqErr = 0.757037
# sqrt(SqErr/n) = 0.307619
```

Prog 4. Typical transformation matrix generated by program **calib**

Measuring

Opposite to the calibration procedure, the measuring module has to operate on-line. This means that on request from the user the image is grabbed and checked. Then the point in the image is found and the nearest calibration matrices are identified and used to estimate the position of the template/coupling mark. The whole procedure is controlled and managed by the cell controller (see App-fig 6).



Check In

Skib: 1159_b4/
Operatoer:
Blok: 146/
Panel: 146db_s/

Schedule:

```
ASS 9S225
ROBOT 1
146G1_4S.NC
146G2_4S.NC
146G3_4S.NC
ROBOT 2
146F1_4S.NC
146F2_4S.NC
146F3_4S.NC
ROBOT 3
146E1_4S.NC
146E2_4S.NC
146E3_4S.NC
ROBOT 4
```

xv Kamera.12.mark

Indtast koordinater for koblingsmaerker
(angives i robot-koordinatsystem)

Hjoerne: 1
Robot: 12
X 1: 6278.4 mm
Y 1: 1550.1 mm

Hjoerne: 4
Robot:
X 2: 34806.8 mm
Y 2: 1530.5 mm

App-fig 6 The cell controller interface

The shell program lying behind the functionality of the vision part of the cell control interface is called “video_calc” and shown in Prog 5 (see also App-fig 1)

```
#!/bin/bash
#
# Author:   Ole Knudsen
#           Odense Steel Shipyard Ltd.
#           960423
#
# Modified:         960423 by Pia Holsting (Sending MBX reply back to mailbox)
#
# Syntax:   video_calc <reply_box> <robot_no>
#
# Action:           This is the first script of 2 which calculates the displacement
#                           coordinates using the video camera. This script shows the
#                           picture taken by the camera in the actual position.
#
```

```
# Replies:  //c 0 video_calc complete
#           //c 1 video_calc error
#

#
# Set name of mbx to reply to
#
mbx_name=p13mbx

robot_no=$2
CAMERA=Camera.$robot_no  # fx. Camera.7
#echo "Using $CAMERA in $HOME/video/b4/$CAMERA "

# Program "cctalk" gets info from MBX
string=‘$HOME/bin/cctalk $robot_no‘
STATUS=‘echo $string |cut -d" " -f 1‘

if [ "$STATUS" = "0" ]
then
 X_POS=‘echo $string |cut -d" " -f 3‘
 X1_POS=‘echo $string |cut -d" " -f 4‘
 Y_POS=‘echo $string |cut -d" " -f 6‘
# Grab image with 1 mark
# Add position of camera in header of image
# Store image in Kamera.$robot_no
 if (grab |add_pos $X_POS $Y_POS > $HOME/Images/Kamera.$robot_no)
then
#  if ( spot < $HOME/Images/Kamera.$robot_no >$HOME/Images/uv_data.km )
  if ( rsh iris1 run_b4 $robot_no >$HOME/Images/uv_data.km ) # REMOTE HANDLING !
  then
   number < $HOME/Images/Kamera.$robot_no |marker > $HOME/Images/Kamera.$robot_no.mark
   xv -geometry 535x427+10+127 -display ews-hip-169:0.0 $HOME/Images/Kamera.$robot_no.mark & pid=$!
   mbx_put -n $mbx_name $1 "Video: Process id of xv is $pid"
   echo "#current position $X_POS $Y_POS"             > $HOME/Images/xy_data
   echo ""                                            >> $HOME/Images/xy_data
    for MATRIX in $HOME/video/b4/$CAMERA/matrix.$robot_no*
   do
    grep ’calibration position’ $MATRIX       >>$HOME/Images/xy_data
    cat $MATRIX $HOME/Images/uv_data.km | perspec          >>$HOME/Images/xy_data
   done
DISPLACE=‘bilinear $HOME/Images/xy_data‘
 cat .bilinear_out1 >> $HOME/Images/LOGFILE
 cat .bilinear_out2 >> $HOME/Images/LOGFILE
echo  Camera: $CAMERA >> $HOME/Images/LOGFILE
echo  Block: $3 >> $HOME/Images/LOGFILE
program=‘ls -1 /home/pni/nc/l15*_b4/*/$3/*.nc | head -1‘
if [ "$program" != "" ]
then
 head -6 $program |grep X >> $HOME/Images/LOGFILE
fi
echo "------------------------------" >> $HOME/Images/LOGFILE
   mbx_put -n $mbx_name $1 "$DISPLACE"
   mbx_put -n $mbx_name $1 "//c 0 video_calc complete!!"
   exit 0
  fi # iris1
  mbx_put -n $mbx_name $1 "//c 1 video_calc error in finding mark in photo"
  exit 1
 fi # grab

 mbx_put -n $mbx_name $1 "//c 1 video_calc error in grabbing image"
 exit 1
fi #STATUS

mbx_put -n $mbx_name $1 "//c 1 video_calc error in reading from robot$robot_no (STATUS = $STATUS)"
exit 1
```

Prog 5. The shell program "video_calc"

Display

In order to make sure that the image grabbed really contains the template and that the quality is satisfactory, the image is displayed on the screen for approval by the user. In the case of a wrong image the program terminates, otherwise the position identification procedure starts.

Perspec

The theory of this program is contained in EQ. 6.9 and EQ. 6.10. The calculation of the world point is done for <u>one</u> calibration matrix in the working area of the robot, no matter how far away the measured points are.

Bilinear

The program finds the nearest neighbours (1,2 or 4) and from them calculates a mean value to which finally a global offset is added. Typical output from program **bilinear** is shown in Prog 6.

```
Inputfile: xy_data
xy_data opened
------------------
Current camera position: (5705.4,18762.6)

Four nearest calibration positions:
-----------------------------------
Calib. pos. (5650.0,16335.6) => (x,y) = (5069.2,19363.5)
Calib. pos. (5680.6,21617.6) => (x,y) = (5072.8,19357.1)
Calib. pos. (8353.6,16459.5) => (x,y) = (5073.4,19368.9)
Calib. pos. (8175.4,21616.9) => (x,y) = (5068.4,19361.2)
Interpolation quadrant:
(5650.0,16335.6) - (8353.6,21617.6)
s = 0.020491
t = 0.459485
Estimated (x,y) = (5070.8,19360.7)
```

Prog 6. Typical output from program **bilinear**

References

[i] Ballard , D. H. , Brown C. M. , 1982 *Computer Vision*. ISBN 0-13-165316-4, pp 107

Appendix B: Courses, presentations, publications

Courses attended

UNIX/C programming tools. 3 days. SuperUsers
C++ basic course. 2 days. SuperUsers
X basic course. 2 days. SuperUsers
MOTIF programming. 4 days. SuperUsers
Xlib programming. 3 days. SuperUsers
Digital image processing. 3 days. DIEU

Papers and presentations

Exhibition of IT results. EITC. Brussels
Presentation of industrial application. Vision day at IMM
Largest robot station in the world. Article in Intern Information
Vision developments at OSS. Article in Danish Industrial Robot Association
Technology at OSS. Interview in Tech World

Conferences

European IT Conference & Exhibition. Brussels. 5 days
MARitime Information Society (MARIS) Conference. Malta. 3 days
Quality Control with vision systems. Ebeltoft. 2 days
Vision day at IMM. Copenhagen. 1 day

Others

As local leader with responsibility for the Esprit project 8329 Cleopatra, a lot of experience has been gained. With a total budget above 100 mill DKK (hereof more than 7 mill. for OSS), not only scientific but also administrative and controlling skills have been developed. Furthermore a lot of inspiration for the PhD project has been gained during visits at the vision centres of AEG, Thomson, Deutsche Aerospace, BMW and many others. The Esprit project covered the construction of vision modules for a driverless car, an autonomous helicopter, an automatic reader able to read addresses and checks, and our own application: the development of a prototype for vision-based weld line identification and safety surveillance. The co-operation in this project has added at lot of value to the scientific content of this thesis.

Furthermore, experience in creating a consortium and a proposal has been achieved. Many contacts with people in the Commission in Brussels have been established and a lot of expertise in negotiations with EU representatives has been gained: skills which are most valuable in the future creation of scientific projects.