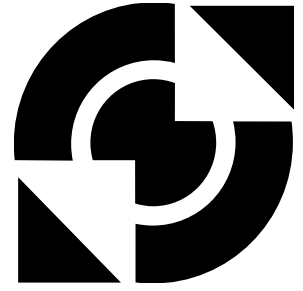


University of Twente

Faculty of
Engineering Technology



Laboratory of Mechanical Automation

A flexible seam detection technique for robotic laser welding

J.O. Entzinger
Enschede, September 2005



1013

University of Twente
Faculty of Engineering Technology
Laboratory of Mechanical Automation

A flexible seam detection technique for robotic laser welding

Master Thesis on

Image undistortion and world coordinate calibration
for seam tracking with structured light
on the Stäubli RX130 robot

by

J.O. Entzinger

Exam-committee:

Chairman:	Prof.dr.ir. J. Meijer
Mentor:	Ir. D. Iakovou
Member:	Dr.ir. R.G.K.M. Aarts
Member:	Dr.ir. D.F. de Lange
External-member:	Dr.ir. F. van der Heijden

WA-1013
23 September 2005

Copyright ©2005, University of Twente. All rights reserved. No part of this report may be used or reproduced in any form or by any means, or stored in a database or retrieval system without prior written permission of the University of Twente, except in the case of brief quotations embodied in critical articles and reviews.

Abstract

This thesis deals with the problem of seam detection for robotic laser welding applications. Within the Mechanical Automation laboratory of the Engineering Technology faculty, a compact, lightweight and multi-purpose welding head is developed. The integrated welding head is attached to a robot and should be able to detect seams, learn trajectories, do the laser welding and inspect the welds. A video camera is attached to the laser focussing optics to obtain a real-time stream of images of the work piece when the robot moves the head along the seam.

This report focusses on compensating the camera images for radial distortions introduced by the welding optics, detection of the seam in the camera images and the translation of image coordinates (in pixels) to real world coordinates (in millimetres).

For estimating the distortion of the images a calibration procedure based on Zhang's algorithm [1] has been implemented in MATLAB. Also an algorithm for automated extraction of reference points from calibration images and an undistortion function have been implemented. Extensive testing has shown that the effect of distortions can be well compensated for after calibration of the optical system.

The welding optics are not optimised for imaging applications. Together with the wish for a large field of view, this results in images with large distortions and aberrations. Image processing algorithms have been developed to cope with these problems in a robust way.

The seam detection algorithm makes use of a structured light projection onto the work piece. Using the triangulation principle the seam position can be estimated from the camera images. Tests have been done with a single straight line projection and with a projection of two crossing lines over an overlap joint. In these tests both methods proved to be capable of tracking several test seams.

For the world coordinate calibration an automated procedure has been developed which detects a marker in the camera images. The marker is positioned at the tool centre point, the focussing spot of the welding laser (or at a known offset). Position changes of the marker in the images are then related to prescribed robot movements, in order to determine the position and orientation of the welding head with respect to the marker.

Keywords: Camera calibration, Seam detection, Structured light, Radial distortion, Image processing, Robot tool calibration

Samenvatting in het Nederlands (Summary in Dutch)

In dit verslag wordt gekeken naar naaddetectie methoden voor robotisch laserlassen. Binnen de vakgroep Werktuigbouwkundige Automatisering van de faculteit Construerende Technische Wetenschappen wordt een compacte, lichtgewicht en veelzijdige laskop ontwikkeld. Deze geïntegreerde laskop wordt bevestigd aan een robot en moet in staat zijn lasnaden te detecteren, in te leren, te lassen en de gelegde las achteraf te inspecteren. Een video camera kijkt via de optica voor het focuseren van de laser bundel mee en verschaft zo real-time beelden van het werkstuk terwijl de robot de laskop langs de naad beweegt.

Dit rapport gaat in op het compenseren van radiale verstoringen in de camerabeelden (deze worden geïntroduceerd door de laser optiek), de detectie van de lasnaad in de camerabeelden en de bepaling van de wereldcoördinaten (onder andere de verhouding tussen pixels in het camera beeld en millimeters in het werkstuk).

Om de verstoringen in de camerabeelden te bepalen, is een MATLAB programma geschreven dat op basis van het algoritme van Zhang [1] een set camera en lens parameters bepaalt. Hier is een algoritme aan toe gevoegd voor automatische extractie van referentiepunten uit beelden van een calibratiepatroon. Bovendien is een programma geschreven dat op basis van de geschatte parameters beelden van hun verstoringen kan ontdoen. Uitgebreide tests hebben bewezen dat met deze methode het effect van de lensverstoringen vrijwel ongedaan gemaakt kan worden.

De optieken die nodig zijn voor het laser lasproces zijn niet geoptimaliseerd voor het nauwkeurig weergeven van een beeld van het werkstuk. Samen met de wens om een groot zichtveld te hebben voor de camera, veroorzaakt dit veel ruis en onscherpte in de camerabeelden. Verschillende beeldbewerkingstechnieken zijn daarom ontwikkeld om een programma op te leveren dat robuust is voor kwalitatief slechte beelden.

Het naaddetectie algoritme maakt gebruik van een projectie van gestructureerd licht op het werkstuk. Door middel van triangulatie kan vervolgens de positie van de lasnaad bepaald worden uit de camerabeelden. Er zijn tests gedaan waarbij één rechte lijn of twee kruisende lijnen werden geprojecteerd op elkaar overlappende staalplaten. Beide methoden bleken in deze tests in staat een lasnaad te kunnen volgen in verschillende testobjecten.

Een geautomatiseerde procedure is ontwikkeld om de wereldcoördinaten te bepalen. In deze procedure wordt de positie van een marker in het camerabeeld

gerelateerd aan bewegingen van de robot. Door een serie voorgedefinieerde bewegingen van de robot, kan daarmee de positie van de marker ten opzichte van de robotflens bepaald worden. Daarmee ligt ook de focus positie van de las-laser vast.

Trefwoorden: Camera kalibratie, Naaddetectie, Gestructureerd licht, Radiale verstoringen, Beeldbewerking, Robotgereedschapskalibratie

Acknowledgements

In the first place I would like to thank my supervisor Dimitrios for his enthusiasm and help during these 10 months of research. I have appreciated the discussions on a conceptual level and the assistance with the VC++ implementation of the code.

To my exam committee I would like to express my gratitude for their flexibility regarding the postponement of my graduation and the hand-in date for the final report. Although time is always running out and always more things could be investigated, I think the experiments I did in the last weeks have completed my research.

Last but not least I must say that I have very much appreciated the help and support of all my colleagues, family, friends and roommates. You helped me through the sometimes difficult and stressful times and made, apart from an educational time, a pleasant time as well.

Jorg Entzinger

Contents

1	Introduction	3
1.1	The ‘Integrated Laser Welding Head’ Project	3
1.2	Research Overview	4
2	Image Undistortion	7
2.1	Introduction to Distortions and Calibration	8
2.1.1	Problems Due to Distortions	8
2.1.2	Projective Geometry	9
2.1.3	Calibration	12
2.2	Calibration Pattern & Image Acquisition	14
2.3	Keypoint Extraction	14
2.3.1	Keypoint Identification	14
2.3.2	Keypoint Ordering	15
2.4	Camera and Lens Distortion Estimation	19
2.4.1	Finding the Homography	19
2.4.2	Camera Intrinsic & Extrinsic Parameter Estimation	20
2.4.3	Radial Distortion Parameter Estimation	22
2.4.4	Solution Refinement	23
2.5	Images Undistortion	23
3	Camera Calibration Experiments	25
3.1	Introduction	25
3.1.1	Overview of Experiments	25
3.1.2	Performance Standards	26
3.2	Keypoint Extraction	27
3.3	Simulations	28
3.3.1	Simulation Data	28
3.3.2	Simulation Images	30
3.4	Zhang’s Test Images	34
3.4.1	Zhang’s Test Data	35
3.4.2	Zhang’s Test Images	37
3.5	Photo Camera Calibration	37
3.6	Welding Head Video Camera Calibration	40
3.6.1	A Final Calibration Result	45
3.7	Guidelines for Taking Good Calibration Images	47

4	Seam Detection	49
4.1	Structured Light & Triangulation	49
4.1.1	Line Deformations	50
4.1.2	Calculating the Seam Position	51
4.1.3	Calculating the Workpiece Orientation (Single Line)	52
4.1.4	Calculating the Workpiece Orientation (Crossing Lines) . .	53
4.2	Image Processing	54
4.2.1	Seam Detection with a Single Line	54
4.2.2	Seam Detection with Crossing Lines	54
4.3	Robot Coordinate Systems	55
4.4	Real-World Coordinate Calibration	57
4.4.1	Scaling Factor and Rotation	57
4.4.2	X-Y Position of the TCP	58
4.4.3	Z Position of the TCP	59
4.4.4	Laser Diode Calibration	60
5	Seam Tracking Experiments	61
5.1	Test Objects	61
5.2	Single Line Seam Detection Experiments	62
5.3	Crossing Lines Seam Detection Experiments	63
5.4	Seam Tracking Experiments	65
5.5	Real-World Coordinate Calibration Experiments	65
5.5.1	Scaling Factor and Rotation	65
5.5.2	X-Y Position of the TCP	66
6	Conclusions & Recommendations	69
6.1	Camera and Lens Calibration	69
6.2	Seam Detection and Real-World Coordinate Calibration	70
6.3	Recommendations	71
	Appendix	73
A	Morphological Image Processing Functions	75
A.1	Images	75
A.2	Thresholding	76
A.3	Labelling	77
A.4	Erosion & Dilation	77
B	Theory	81
B.1	Choice of Calibration Objects and Keypoints	81
B.2	Distortion Formulas	82
B.2.1	Radial Distortions	83
B.2.2	Extension with Tangential Distortions	84
B.3	Rodrigues' Rotation Formula	84
B.4	The Root Mean Square (RMS)	85
B.5	Distance Between a Point and a Line	86

B.6	Determination of a Circle's Centre From Point Data	86
B.6.1	A Geometrical Approach	87
B.6.2	A Least Squares Fitting Approach	89
C	Thresholding Algorithms	91
C.1	Static Thresholding	91
C.1.1	Black/White Balancing Thresholding	91
C.1.2	MATLAB Thresholding	91
C.2	Dynamic Thresholding	92
C.2.1	Dynamic Thresholding I	92
C.2.2	Dynamic Thresholding II	93
C.3	Advanced Keypoint Determination.	94
D	Hardware Specifications	99
D.1	Welding Head Parameters	99
D.1.1	Laser Diodes	99
D.1.2	Lenses	99
D.1.3	Camera & Frame Grabber	99
D.2	Calibration Patterns	100
D.2.1	Paper Patterns	100
D.2.2	Laser-engraved Patterns	100
D.2.3	Simulated Images	102
D.3	Photo Camera	103
	References	105

Notation

The following convention is used throughout this thesis. Vectors are in bold, lower case letters, e.g. \mathbf{x} and \mathbf{m} , except for vectors corresponding to real world 3D points or model points, which are printed in upper case, typewriter letters, e.g. \mathbf{M} . Matrices are in bold, uppercase letters, e.g. \mathbf{R} . Tilde \sim is used to denote an augmented vector such that $\tilde{\mathbf{x}} = [\mathbf{x}^T 1]^T$.

The most used symbols and variables are:

\cdot	: dot/inner/scalar product, $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y}$
\times	: cross product of two vectors, e.g. $\mathbf{x} \times \mathbf{y}$
$^{-1}$: superscript, inverse of a matrix
T	: superscript, transpose of a vector or matrix
$^{-T}$: superscript, transpose of the inverse of a matrix
$\ \mathbf{x}\ $: 2-norm of a vector
$\ \mathbf{x}\ _{fro}$: Frobenius norm of a matrix, $\sqrt{\sum diag(\mathbf{X}^T \mathbf{X})}$
$\mathbf{0}$: vector with all elements equal to zero
\mathbf{A}	: 3×3 matrix with the camera intrinsic parameters
$\tilde{\mathbf{x}}$: augmented vector, $[\mathbf{x}^T 1]^T$
Δ	: a difference, e.g. Δx , a change in parameter x
α	: normalised pixel scaling factor of a camera in width direction
β	: normalised pixel scaling factor of a camera in height direction
γ	: skew factor of the camera image (non-perpendicularity of axes)
u_0	: position of the optical centre in width direction
v_0	: position of the optical centre in height direction
κ	: radial distortion parameter
τ	: tangential distortion parameter
f_c	: focal length
f	: factor containing the focal length and pixel sizes
\mathbf{F}	: 3×3 diagonal matrix with diagonal $[f, f, 1]$
\mathbf{H}	: 3×3 homography matrix
\mathbf{h}_i	: the i^{th} column of \mathbf{H}
$\bar{\mathbf{h}}_i$: the i^{th} row of \mathbf{H}
\mathbf{I}	: identity matrix
\mathbf{k}	: vector with radial distortion parameters
\mathbf{m}	: image point $[u, v]^T$
\mathbf{M}	: real world point $[\mathbf{x}, \mathbf{y}, \mathbf{z}]^T$ or point derived from a model

- R** : 3×3 rotation matrix
r_i : the i^{th} column of **R**
t : translation vector $[\Delta x, \Delta y, \Delta z]^T$
 $[]_c$: subscript, in camera coordinates
 $[]_n$: subscript, normalised values
RMS : Root mean square, a measure for fitting accuracy, see §B.4

Chapter 1

Introduction

Since the early 1980s, laser welding has developed from a tool only used for exotic applications into a full-fledged part of the metalworking industry. Nowadays innumerable laser welds are produced for a variety of common products such as cigarette lighters, razor blades, pacemakers and car parts. A bundle of laser light is focussed onto a weld joint, which makes the material heat up and melt locally, resulting in a weld. The well controllable heat input, the small heat affected zone and the high welding speeds result in a fast production of high quality welds, which is the main benefit of laser welding over conventional welding techniques.

The small heat affected zone and the high welding speeds however, also result in very tight tolerances on the accuracy of positioning the weld spot on the seam. Especially when a robot is needed to carry the welding optics along complex geometries, it becomes difficult to assure accurate positioning. Therefore it is important to have a system that detects the position of the seam with respect to the robot. Using the data provided by such a system the robot can be trained to track the seam accurately, before the actual welding starts.

1.1 The ‘Integrated Laser Welding Head’ Project

The work covered by this report is part of the ‘Integrated Laser Welding Head’ Project[2, 3]. This project’s aim is to develop a fully integrated head for robotic laser welding applications. The multifunctional head should be both compact and light. The capabilities of the end product (hard- and software) should include:

- Seam detection
- Seam trajectory learning
- Laser welding
- Welding process control (from weld pool observation)
- Weld inspection

The compactness of the welding head is important to be able to access products that should be welded from the inside. A light welding head is beneficial because it will allow robots to move it around more accurate. The integration of all functionality is a matter of cost saving and needed to have full control over the process. Due to these practical goals, industrial partners are highly interested in the current research.

This report deals with the seam detection part and a number of accompanying problems.

1.2 Research Overview

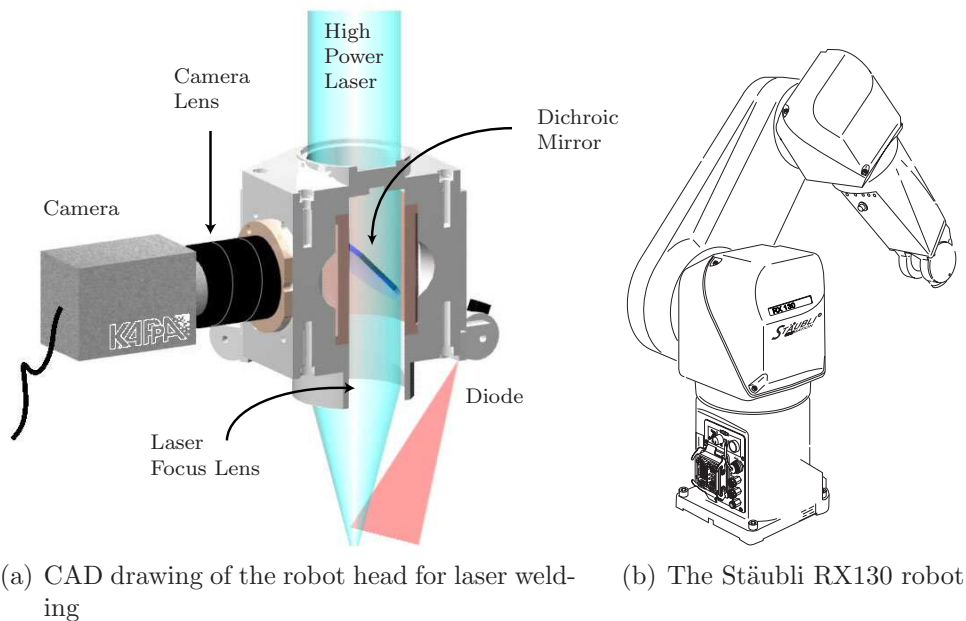


Figure 1.1: The robot and the laser welding head

The designed head (Figure 1.1(a)) contains a dichroic mirror, which reflects an image of the workpiece to a video camera while letting high power laser light pass through. The benefit of this setup is that the working point is always in the camera view. A disadvantage is that the image passes the laser focus lens, which has not been optimised for imaging purposes, but for a well focussed laser bundle. Because a large field of view is desired, not only the relatively distortion-free centre but also the outer parts of the laser focus lens will need to be used. This introduces large errors in the image, such as aberrations, radial distortions and coma (for explanations of these terms, please refer to [4, 5]). In Chapter 2 a procedure will be presented for determination and compensation of several camera and lens distortions. Chapter 3 deals with the results of experiments using this procedure.

The images grabbed from the video camera will be corrected for distortions and used to determine the seam position. To calculate the seam position from the images, the distance between the workpiece and the welding head must be

known. To determine this distance, laser diodes attached to the welding head will project lines onto the surface of the workpiece (see Figure 1.2(a)). The position of the projected lines in the images provides information about the position and orientation of the welding head relative to the workpiece. The fixed relation between the camera and the light planes created by the laser diodes makes it possible to calculate this position and orientation. This is called triangulation. For an overlap joint the seam itself usually appears as a break in the line projection, as can be seen from Figure 1.2(b). Chapter 4 elaborates on triangulation and the determination of the seam position and experimental results are presented in Chapter 5.

The head is attached to a Stäubli RX130 robot (Figure 1.1(b)) of which the dynamical behaviour is thoroughly examined by other research groups within the mechanical automation laboratory. A calibration of the tool – which includes the determination of the laser focus point relative to the robot tip and the determination of a pixels-to-millimetres scaling – is described in §4.4.

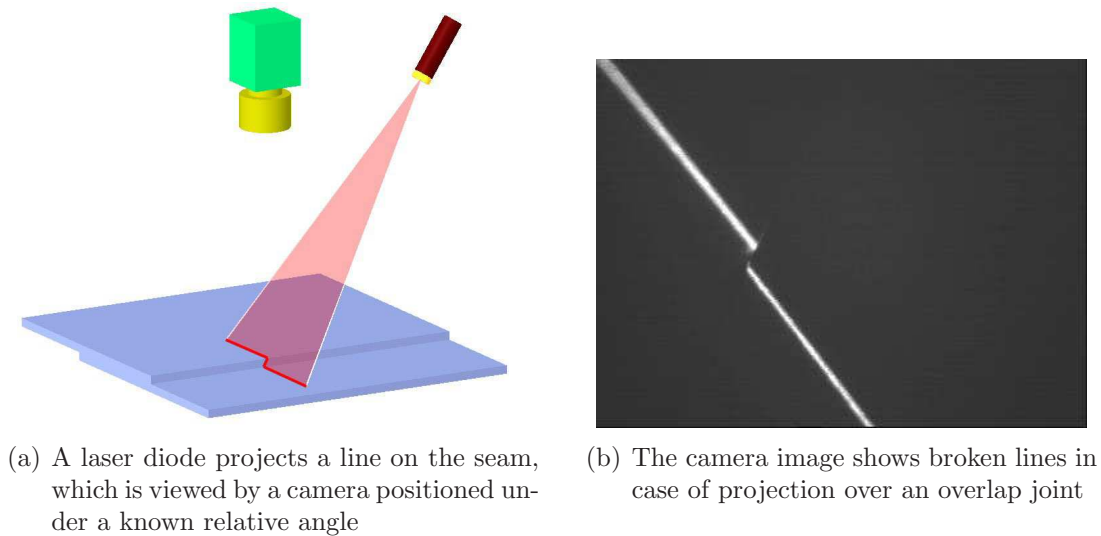


Figure 1.2: The principle of triangulation

The final goal of this part of the project is to provide algorithms for accurate, real-time, image based seam detection using one or two laser line projections. The conclusions (Chapter 6) show whether this goal was reached and what improvements could be made.

Chapter 2

Image Undistortion

Almost all images we see are distorted somehow. Especially when lenses are used distortions become clearly visible. This can be seen when you put on somebody else's glasses, when you look through a peep hole or fish eye in the door, or when you are close to a webcam. Professional camera lenses are optimised to minimise these effects, but small, cheap lenses or lenses designed for purposes other than imaging (such as the laser focus lens in the integrated welding head) often suffer from severe distortions.

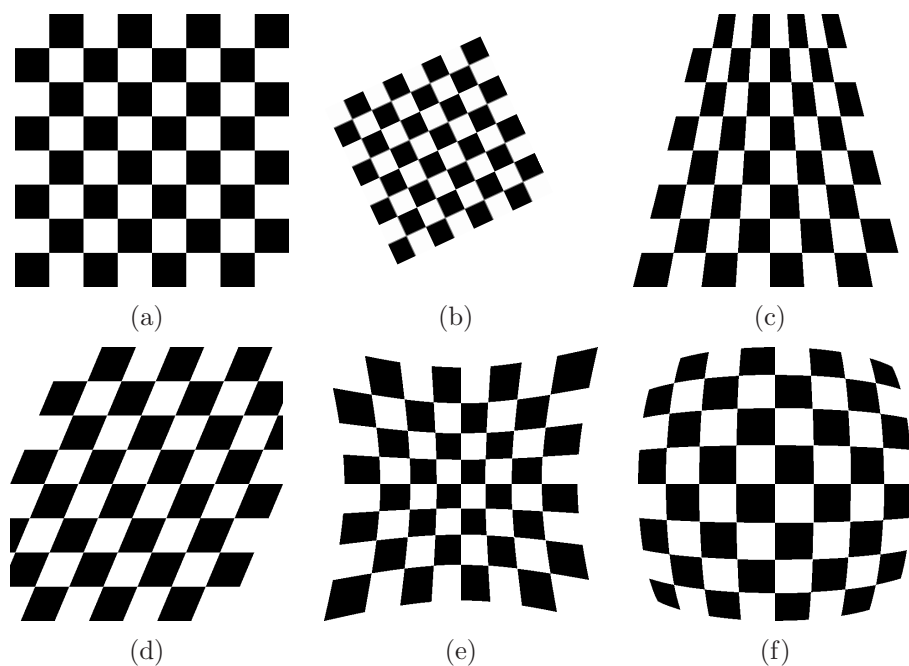


Figure 2.1: Several transformations:

- (a) Original
- (b) Linear Conformal
- (c) Projective / Perspective
- (d) Affine / Skew
- (e) Radial: Pincushion
- (f) Radial: Barrel (negative pincushion distortion)

2.1 Introduction to Distortions and Calibration

Several transformations occur when a 2D image is taken of a 3D scene. Rotation and perspective (Figures 2.1(b)–(c)) for instance depend on the viewpoint relative to the scene. Skew, pincushion and barrel transformations (Figures 2.2(d)–(f)) on the other hand, depend on the optics and camera properties. Scaling depends on both.

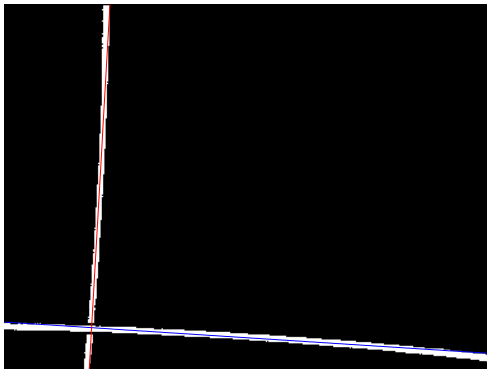
Here we will adopt the definition of ‘distortion’ given by Merriam-Webster’s dictionary[6]: *a lack of proportionality in an image resulting from defects in the optical system*. This rules out the viewpoint-dependent transformations, because they do not originate in the optical system. It also rules out spherical aberrations and astigmatism, because they do not introduce a lack of proportionality in the image, but only a blurring effect.

2.1.1 Problems Due to Distortions

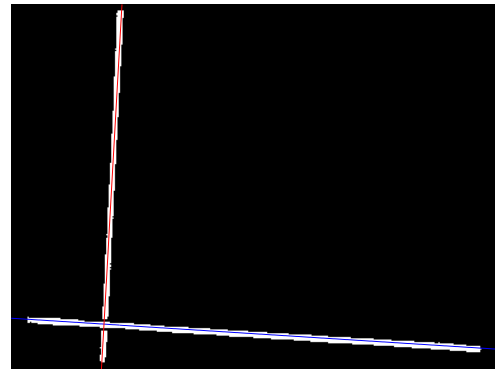
The main distortions in an image are *radial distortions*, where pixels are moved from or towards the optical centre with an amount related to their distance from this centre. Radial distortions cause straight line(s) in the real world to be displayed as a curve in the camera image, as Figure 2.2(a) shows. Fitting a straight line through a curve will result in inaccurate results, as the fitted line will always deviate from the line in the image at several places.

Another problem occurring due to radial distortions, is that two objects in the centre of the image may be 10 pixels apart, whereas two other objects, equally distant in the real world, may appear 15 pixels apart when positioned at the edge of the image. Figure 2.1(e) illustrates this: the once equally sized squares of figure Figure 2.1(a) appear larger at the image edges after distortion. This means a line at the edge is not only curved, it will also be shifted.

It is therefore important correct any radial distortions in the images, before any feature extraction – such as seam detection – is done.



(a) The distorted image shows curved lines, which are difficult to fit accurately.



(b) After the images is undistorted, the lines are straight again which makes accurate fitting possible.

Figure 2.2: The problem of extracting features from radially distorted images.

2.1.2 Projective Geometry

To determine the influence of distortions on the image, some basic knowledge about image projections is needed. When relating features in images to the real world, a transformation between the 2D image and the 3D world must be defined. The camera model used here is that of a pinhole camera[7]. This widely used model is simple yet powerful and applicable to most off-the-shelf cameras. The pinhole camera is an extension of the perspective projection which gives the ideal transformation from a 3D point $\mathbf{M} = [\mathbf{x}, \mathbf{y}, \mathbf{z}]^T$ to a 2D image point $\mathbf{m} = [u, v]^T$, assuming that all light rays pass through a pinhole C (also called optical centre) at distance f_c from the image plane.

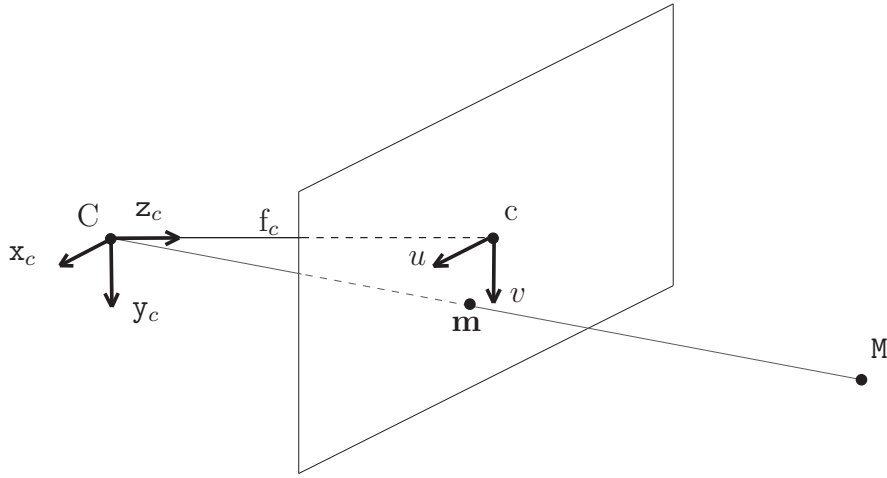


Figure 2.3: Virtual image plane.

The perspective projection states that (see also Figure 2.3):

$$\frac{u}{x_c} = \frac{v}{y_c} = \frac{f_c}{z_c},$$

which can be rewritten to

$$\frac{z_c u}{f_c x_c} = \frac{z_c v}{f_c y_c},$$

or, in matrix-vector form with s set equal to z_c by the lower row in the equation:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_c & 0 & 0 \\ 0 & f_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_c,$$

or for short, using the augmented vector $\tilde{\mathbf{m}} = [\mathbf{m}^T, 1]^T$

$$s\tilde{\mathbf{m}} = \mathbf{F}_c \mathbf{M}_c \tag{2.1}$$

The perspective projection is extended with a camera transformation (see also Figure 2.4). This is needed because generally the camera image will not have its origin in the principal point c , but at some distance $[u_0, v_0]$ from it. Also the size of pixels in the image is in the extension, as the position of a point in the image is measured in pixels, rather than in millimetres. As the pixels need not to be square, different sizes for the u direction (k_u) and v direction (k_v) exist. The last parameter is the angle θ , which specifies the perpendicularity of the image axes. Because these parameters specify the internal behaviour of the camera, they are called the camera *intrinsic parameters*. The full pinhole camera model can now be written as:

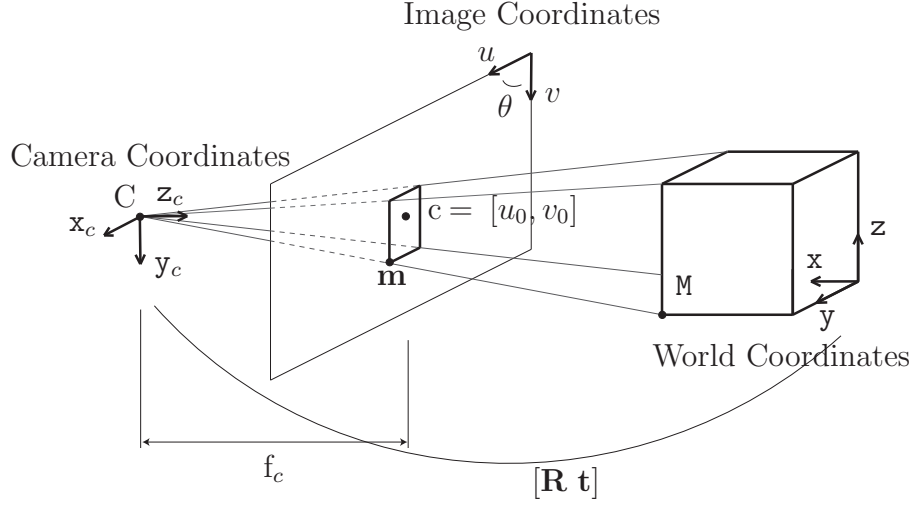


Figure 2.4: Coordinate Systems in Projective Geometry.

$$\begin{aligned}
 s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} k_u & k_u \cot \theta & u_0 \\ 0 & k_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_c & 0 & 0 \\ 0 & f_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_c \\
 &= \begin{bmatrix} f_c k_u & f_c k_u \cot \theta & u_0 \\ 0 & f_c k_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_c
 \end{aligned} \tag{2.2}$$

In this equation a change in the focal length f_c – i.e., the distance between the optical centre C and the principal point c – cannot be distinguished from a change in the pixel dimensions k_u and k_v . Therefore they are split in a slightly different way, only for convenience: the zooming part of the camera intrinsic matrix, which is the upper-left 2×2 matrix, is normalised to be non-magnifying:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_c$$

or for short

$$s\tilde{\mathbf{m}} = \mathbf{AFM}_c \tag{2.3}$$

With:

$$\left\| \begin{array}{cc} \alpha & \gamma \\ 0 & \beta \end{array} \right\|_{fro} = \sqrt{2} \quad (2.4)$$

and thus:

$$f = \frac{\left\| \begin{array}{cc} f_c k_u & f_c k_u \cot \theta \\ 0 & f_c k_v / \sin \theta \end{array} \right\|_{fro}}{\sqrt{2}} \quad (2.5)$$

Using Equation 2.3 a point expressed in the camera coordinate system can be projected onto the image plane. In practice the position of a point is not known with respect to the camera, but only with respect to a world coordinate system. This system can be rotated and translated with respect to the camera, as shown in Figure 2.4. A 3×3 rotation matrix \mathbf{R} and a translation vector $\mathbf{t} = [\Delta x, \Delta y, \Delta z]^T$ describe the transformation:

$$\mathbf{M}_c = \mathbf{R}\mathbf{M} + \mathbf{t},$$

or, written out in full:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_c = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix},$$

which can be written using an augmented vector as

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & \Delta x \\ r_{21} & r_{22} & r_{23} & \Delta y \\ r_{31} & r_{32} & r_{33} & \Delta z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},$$

or

$$\mathbf{M}_c = [\mathbf{R} \ \mathbf{t}] \tilde{\mathbf{M}}. \quad (2.6)$$

Because the transformation $[\mathbf{R} \ \mathbf{t}]$ takes place outside the camera, \mathbf{R} and \mathbf{t} are called the camera *extrinsic parameters*.

The full transformation from a 3D model, expressed in world coordinates to a 2D images can be written by combining Equations 2.3 and 2.6 to:

$$s\tilde{\mathbf{m}} = \mathbf{A}\mathbf{F}[\mathbf{R} \ \mathbf{t}]\tilde{\mathbf{M}}. \quad (2.7)$$

Special Case for a 2D Model

In the procedure to find the distortion parameters, a flat pattern (model) will be used. This means it can be assumed that every point in the pattern is at $z = 0$

of the world coordinate system¹. This results in

$$\begin{aligned}
 s\tilde{\mathbf{m}} &= \mathbf{A}\mathbf{F} \begin{bmatrix} r_{11} & r_{12} & r_{13} & \Delta x \\ r_{21} & r_{22} & r_{23} & \Delta y \\ r_{31} & r_{32} & r_{33} & \Delta z \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \\
 &= \mathbf{A}\mathbf{F} \begin{bmatrix} r_{11} & r_{12} & \Delta x \\ r_{21} & r_{22} & \Delta y \\ r_{31} & r_{32} & \Delta z \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}
 \end{aligned} \tag{2.8}$$

By abuse of notation, for the rest of the report $\tilde{\mathbf{M}}$ is redefined to $[x \ y \ 1]^T$ instead of $[x \ y \ z \ 1]^T$. The transformation matrices are gathered into the homography \mathbf{H} which thus relates the real world point (or model point) \mathbf{M} to the image point \mathbf{m} :

$$s\tilde{\mathbf{m}} = \underbrace{\mathbf{A}\mathbf{F}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]}_{\text{Homography } \mathbf{H}} \tilde{\mathbf{M}}. \tag{2.9}$$

2.1.3 Calibration

To compensate the distortions, the parameters of the camera and optical system must be known. Calibration is the art of finding these parameters. For the camera the intrinsic parameters (§2.1.2, Equation 2.3) will need to be determined and for the radial distortions two extra parameters have to be found. When these parameters are known, they can be used to undo images taken with that camera from distortions.

Most camera calibration procedures make use of a pattern of regularly spaced objects. These objects can for instance be points, lines or squares (see Figure 2.5). Multiple pictures of the pattern are taken under different angles. The main goal is then to find a ‘common factor’ in all images, to distinguish between the camera (varying) extrinsic parameters (§2.1.2, Equation 2.6) on the one side and the (constant) intrinsic parameters and distortions on the other side.

A model describes the positions of the objects in the pattern based on features (‘keypoints’) that can later be detected in the images (such as a centroid, crossings or corners). The parameters can be found by fitting the keypoints extracted from the images to the mathematically transformed and distorted model positions. Most procedures (like those of Tsai[8, 9], Heikkila[10] and Savii[11]) require a model containing the absolute 3D position of calibration points, or well defined relative movements between two calibration images. Zhang’s method [1] however, is flexible and only needs relative 2D positions of a 2D pattern for the model points. This is why Zhang’s method was chosen for camera and lens calibration².

¹ It could be assumed that the model exists at $x = 0$ or $y = 0$ instead.

² After implementation it appeared that Zhang’s method was found to have the best (but not the fastest) convergence according to [12]

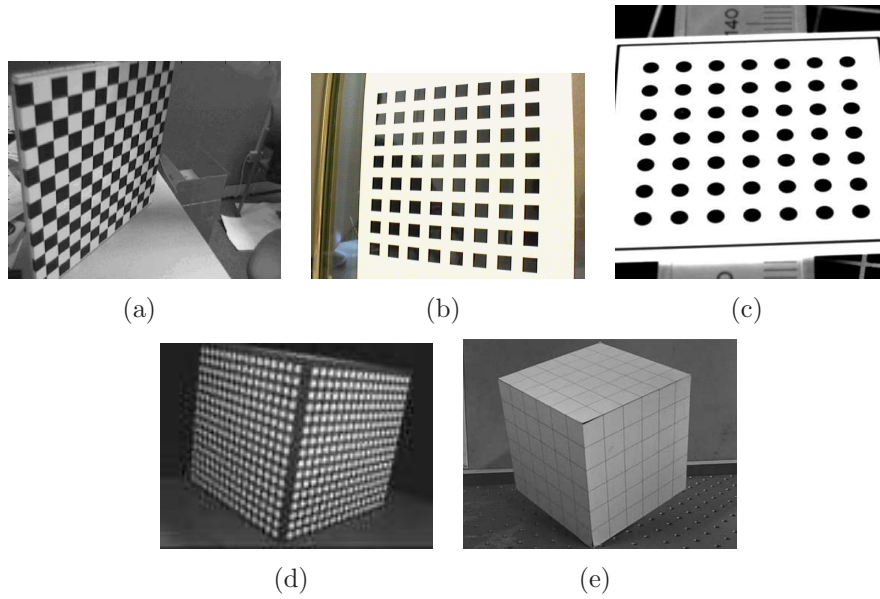


Figure 2.5: Common calibration patterns:

- (a) The checkerboard image used for the Matlab Camera Calibration Toolbox [13]
- (b) The pattern used by Zhang [1]
- (c) The pattern used for the Halcon software [14]
- (d) 3D grid as used by Heikkila [10]
- (e) 3D grid with lines as used by Jong-Eun Haa & Dong-Joong Kangb [15]

A Step-by-Step Guide for Calibration

The rest of this chapter will deal with all aspects of camera and lens calibration and image undistortion in a chronological way. Calibration consists of the following steps, with step 3 in accordance with Zhang:

1. Preparation of a calibration pattern (typically black squares on a white background) and taking pictures of it with the system to be calibrated
2. Extraction of the keypoints for every image, i.e., the extraction of characteristic points in the image that can be matched with a model of the calibration pattern
 - (a) Identification of the keypoints (determination the location of the keypoints, e.g., centroids or corners of the black squares)
 - (b) Ordering of the keypoints, to make sure the list of keypoint coordinates has the same order as the list in the model definition)
3. Estimation of the camera and lens distortion parameters from a set of images
 - (a) Calculation of the Homography for every images. The homography is a transformation matrix to describe the Model \rightarrow Image transformation (§2.1.2, Equation 2.9)

- (b) Estimation of the camera intrinsic parameters (the common part in all images) and extrinsic parameters (rotation, translation)
- (c) Calculation of the radial distortion parameters based on all images. This is implemented by a least squares optimisation to determine which distortion parameters describe the transformation Model \rightarrow Image best
- (d) Refinement of all estimates. Because the radial distortion parameters are estimated upon estimations of in- and extrinsic parameters that are not corrected for this distortion, an (iterative) refinement should be carried out.

After the calibration is done and the camera and lens parameters are known, images can be undistorted. The undistortion procedure is described in §2.5.

2.2 Calibration Pattern & Image Acquisition

When using clearly separated objects like in Figures 2.5(b) and (c) it is easier to find the keypoints than when using a checkerboard-like model (Figure 2.5(a)). A pattern with objects ordered in equally spaced rows and columns is preferred for convenient ordering. Although Zhang uses the corners of the black squares as keypoints, here the centroids of the objects will be used. A detailed explanation and a justification for this choice can be found in §B.1³.

The theoretical minimum number of images required is 3, but it is preferred to take 10-20 images depending on the image quality. It is important to keep the optical system fixed and equal to the settings that will be used later on, when images need to be undistorted. This is because the parameters estimated in the calibration process will be different for different lenses and camera settings such as focus distance and aperture.

2.3 Keypoint Extraction

Finding the keypoints actually consists of two steps: identifying their locations and placing them in the right order. It is important to have the order of the keypoints list (e.g. bottom to top, then left to right) equal to the order of the model, because otherwise a least squares fit of (distorted) model points to the keypoints would not make any sense.

2.3.1 Keypoint Identification

The first step in identification of the keypoints is identification of the objects. Objects can be identified by thresholding (§A.2) and labelling (§A.3) the image. From the labelled objects, the keypoints can be derived easily.

³ The MATLAB implementation can deal with either corner or centroid keypoints.

Thresholding the images is not trivial. Several problems occur, varying from inhomogeneous lighting to coma effects caused by the laser focus lens and internal reflections originating from the semi transparent mirror. Multiple thresholding algorithms have been implemented to suit the needs for different kinds of images. Appendices C.1 and C.2 deal with these algorithms and their suitability for different image types.

When a black/white image (Figure 2.6(b)) is available, each set of interconnected black pixels should be identified as an object, and the centroid of this object should be determined. When labelling the image it is important to keep in mind that only the *calibration* objects should be labelled. Therefore, after thresholding all objects that are not calibration objects must be removed (Figure 2.6(c)). Most of these unwanted objects can be ruled out by examining their size (often they originate from noise and thus are much smaller) or position (the image edges are usually a source of unwanted objects). The minimal size for an object to be recognised as a calibration object is set depending on the image size, the number of objects and the number of black pixels in the image.

After all calibration objects have been identified, the keypoints can be calculated. For the centroids this normally means that all x-coordinates of the pixels that are forming the object are averaged and the same is done for all y-coordinates (Figure 2.6(d)), however in Appendix C.3 a more advanced method is proposed.

The corner keypoints are determined using the following strategy: the first keypoint is the position of the pixel which is the furthest away from the centroid. The second keypoint is the position of the pixel second furthest away from the centroid, ignoring all pixels within a given distance⁴ from the keypoint already found. This process goes on until all corners of the object are found. With the correct setting of the distance for ignoring pixels, this method appears quite robust.

2.3.2 Keypoint Ordering

It is important to have a convention for the order of the keypoints. How they are ordered is not important, as long as the model and the extracted points can be matched. We decided to order first from low to high vertical coordinates and then from low to high horizontal coordinates. This does not necessarily match with bottom-top and x and y coordinates, because images pixels are often described in a differently positioned coordinate system.

Although we humans can find the right order of points with ease, it is quite difficult to make this clear to a computer in an unambiguous way. First of all the computer must know which objects are on the corners of our calibration pattern. The first step is to determine the so-called convex Hull, which finds the outermost objects in the set, i.e. all points that would be touched by a rubber band wrapped around the set, as depicted in Figure 2.7(a). To find out which of these points

⁴ This distance is calculated from the minimum object size, which in its turn is based on the number of black pixels in the thresholded image and the number of objects that should be detected. The value for this distance can be overruled by setting it manually.

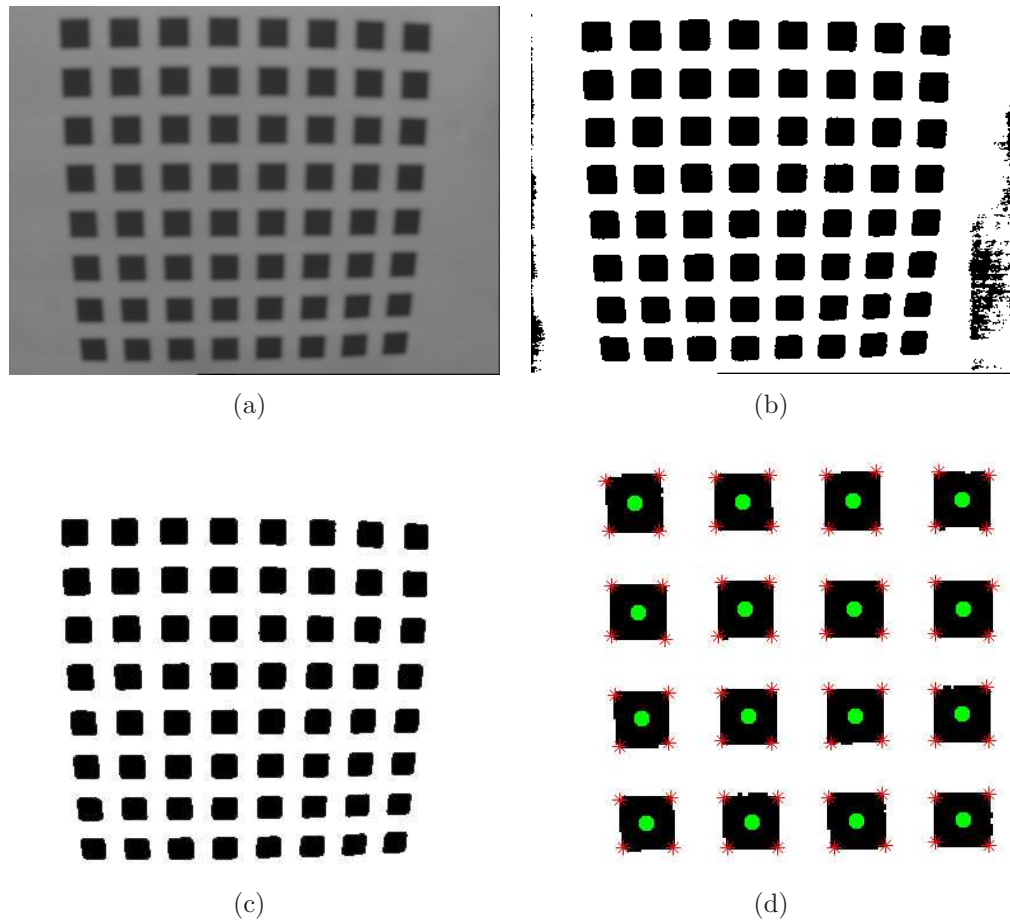


Figure 2.6: Identification of keypoints:

- (a) The original image
- (b) The thresholded image
- (c) The thresholded image, cleaned from unwanted objects
- (d) The extracted keypoints projected onto the thresholded and cleaned image (cut-out)

are on the corners, the angles between the lines with neighbouring points are investigated. The four largest angles (closest to 90 degrees) are assumed to correspond to the corner objects.

Now we know all corner objects, we can select the two with the lowest horizontal coordinate and from that selection we pick the one with the lowest vertical coordinate (see also Figure 2.7). This is our starting point.

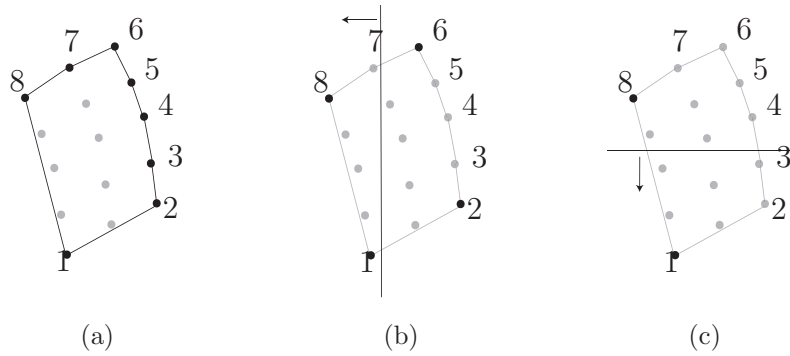


Figure 2.7: The convex hull transform is used to determine the starting point for keypoint ordering

- (a) The convex hull selects all points touched when a rubber band would be wrapped around a set. The corner objects can be found by examining the angles the interconnecting lines make: the lines 8–1 and 1–2 make an angle of approximately 90° and the lines 2–3 and 3–4 make an angle of almost 0° , so 1 is a corner object and 3 is not.
- (b) Of the 4 corner objects, the ones with lowest horizontal index are taken (8 and 1)
- (c) Of the 2 remaining corner objects, the one with lowest vertical index is taken (1)

Finding the next object works as following (see also Figures 2.8 and 2.9):

1. Find the 5 objects closest to the current one (3 when starting a new column)
2. Of these 5, select those that have a higher vertical coordinate than the current object
3. Of these, select the 2 leftmost objects
4. Determine the angle between the vectors from the current object to the 2 candidates
5. If the angle is small (Figure 2.8(c)), take the closest object, else take the leftmost (Figure 2.9(c))

When the last object in the column has been found, the first object in the next column is found in a similar way and the procedure is repeated. All objects in columns that have already been processed are ignored when finding new objects. The final ordering is depicted in Figure 2.10(a).

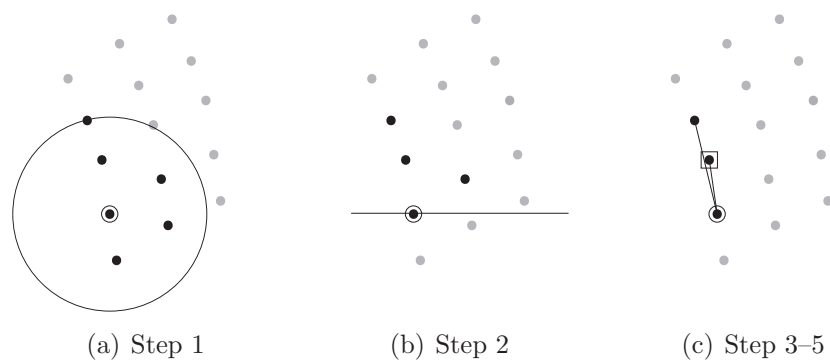


Figure 2.8: Determining which is the next keypoint, scenario 1. The encircled point is the current point, the point with the square will be the next point.

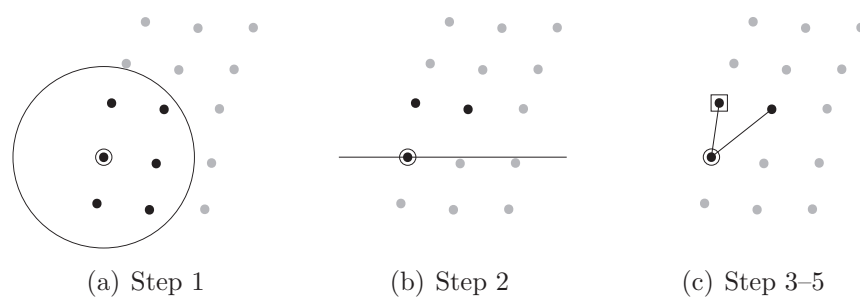
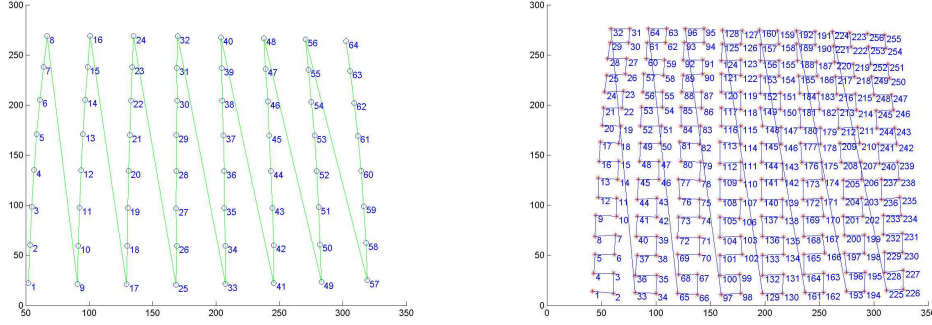


Figure 2.9: Determining which is the next keypoint, scenario 2. The encircled point is the current point, the point with the square will be the next point.

For the ordering of corner keypoints the convex Hull is used again, as it always orders points counter-clockwise. The only thing that remains to be done is to find out which of the four points comes first. Of the two points with the lowest horizontal coordinate, this will always be the one with the lowest vertical coordinate. Figure 2.10(b) shows the resulting order.



(a) Order of the centroid keypoints, regardless of object shape. (b) Order of the corner keypoints, for square objects.

Figure 2.10: Pattern keypoint order.

2.4 Camera and Lens Distortion Estimation

To be able to estimate the lens distortions, the perspective distortions must be eliminated. Therefore the first step is to find a linear transformation matrix that matches the extracted sets of keypoints to the calibration model. From these homography matrices (one for each image), the intrinsic (camera specific) and extrinsic (orientation specific) parameters can be extracted. Then the radial distortions can be estimated.

Both model points \mathbf{M} and the keypoints \mathbf{m} are normalised to improve matrix conditions. The model points are transformed such that their centre of gravity is at the origin and their mean distance to the origin is $\sqrt{2}$, so a typical point will look like $[\mathbf{x}, \mathbf{y}] = [1, 1]$. The keypoints are normalised in almost the same way, but their origin and scaling are determined upon all pixels in an image, instead of on the keypoints themselves. This is because the image provides constant and easily reproducible factors, so the scaling is the same in all images and can be re-calculated any time, as long as the image dimensions are known.

2.4.1 Finding the Homography

Recalling Equation 2.9 from §2.1.2 we know that in the ideal case

$$s\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathbf{M}} \quad \text{with:} \quad \mathbf{H} = \mathbf{A}\mathbf{F}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}], \quad (2.10)$$

holds. In practice there are inaccuracies in the keypoint extraction and due to radial distortions, so a least squares fit will be used to determine \mathbf{H} . Because

it contains both the intrinsic and extrinsic parameters, a homography must be estimated for each image of the model plane.

Equation 2.10 can also be written as

$$su = \bar{h}_1 \tilde{\mathbf{M}} \quad (2.11)$$

$$sv = \bar{h}_2 \tilde{\mathbf{M}} \quad (2.12)$$

$$s = \bar{h}_3 \tilde{\mathbf{M}}, \quad (2.13)$$

or

$$0 = \bar{h}_1 \tilde{\mathbf{M}} - u \bar{h}_3 \tilde{\mathbf{M}} \quad (2.14)$$

$$0 = \bar{h}_2 \tilde{\mathbf{M}} - v \bar{h}_3 \tilde{\mathbf{M}}, \quad (2.15)$$

$$(2.16)$$

which makes it easier to put in a matrix-vector form

$$\underbrace{\begin{bmatrix} \tilde{\mathbf{M}}^T & \mathbf{0} & -u_n \tilde{\mathbf{M}}^T \\ \mathbf{0} & \tilde{\mathbf{M}}^T & -v_n \tilde{\mathbf{M}}^T \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \bar{\mathbf{h}}_1^T / \eta \\ \bar{\mathbf{h}}_2^T / \eta \\ \bar{\mathbf{h}}_3^T \end{bmatrix}}_{\mathbf{x}} = \mathbf{0} \quad (2.17)$$

where:

$$sc = \frac{\sqrt{2}}{n} \sum_{i=1}^n \|\mathbf{m}_i\|$$

$$u_n = \frac{u}{\eta}, \quad v_n = \frac{v}{\eta}$$

to determine a least squares solution for \mathbf{H} , which is needed as an initial guess for the non-linear least squares optimisation. The solution \mathbf{x} in the least squares sense is known to be the eigenvector corresponding to the smallest eigenvalue of $\mathbf{L}^T \mathbf{L}$. This calculation has been optimised using the extra normalisation factor η .

Using the initial guess as starting point a non-linear optimisation is performed: $\min_H \sum_i \|\tilde{\mathbf{m}} - \mathbf{H} \tilde{\mathbf{M}} / s\|^2$ and an accurate homography is found for each image. The homography is always normalised afterward such that $h_{33} = 1$ because the scaling-factor s is present in Equation 2.10 anyway.

2.4.2 Camera Intrinsic & Extrinsic Parameter Estimation

The homographies should be split into two parts: the intrinsic parameters \mathbf{A} and f – the common part in all images – and the extrinsic parameters \mathbf{R} and \mathbf{t} , which are different for every image. Because the homography has 8 degrees of freedom (due to its normalisation) and translation and rotation only take $2 \times 3 = 6$ parameters, there should be 2 constraints. These can be found in the orthonormality property of \mathbf{r}_1 and \mathbf{r}_2 .

Using the fact that⁵

$$[\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = \lambda \mathbf{A} \mathbf{F} [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}], \quad (2.18)$$

the orthonormality of the rotation matrix can be rewritten to:

$$\begin{aligned} \mathbf{r}_1^T \mathbf{r}_2 &= 0 \Rightarrow \mathbf{h}_1^T (\mathbf{A} \mathbf{F})^{-T} \frac{1}{\lambda} \frac{1}{\lambda} (\mathbf{A} \mathbf{F})^{-1} \mathbf{h}_2 = 0, \\ \mathbf{r}_1^T \mathbf{r}_1 &= 1 \Rightarrow \mathbf{h}_1^T (\mathbf{A} \mathbf{F})^{-T} (\mathbf{A} \mathbf{F})^{-1} \mathbf{h}_1 = \lambda^2, \\ \mathbf{r}_2^T \mathbf{r}_2 &= 1 \Rightarrow \mathbf{h}_1^T (\mathbf{A} \mathbf{F})^{-T} (\mathbf{A} \mathbf{F})^{-1} \mathbf{h}_2 = \lambda^2, \end{aligned} \quad (2.19)$$

so the following must hold:

$$\begin{aligned} \mathbf{h}_1^T (\mathbf{A} \mathbf{F})^{-T} (\mathbf{A} \mathbf{F})^{-1} \mathbf{h}_2 &= 0, \\ \mathbf{h}_1^T (\mathbf{A} \mathbf{F})^{-T} (\mathbf{A} \mathbf{F})^{-1} \mathbf{h}_1 &= \mathbf{h}_2^T (\mathbf{A} \mathbf{F})^{-T} (\mathbf{A} \mathbf{F})^{-1} \mathbf{h}_2. \end{aligned} \quad (2.20)$$

The fact that there are only 2 constraints per image, means that at least 3 images are needed to determine the 5 degrees of freedom⁶ in $\mathbf{A} \mathbf{F}$. Now let

$$\begin{aligned} \mathbf{B} &= (\mathbf{A} \mathbf{F})^{-T} (\mathbf{A} \mathbf{F})^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{f^2 \alpha^2} & -\frac{\gamma}{f^2 \alpha^2 \beta} & \frac{\gamma v_0 - \beta u_0}{f^2 \alpha^2 \beta} \\ -\frac{\gamma}{f^2 \alpha^2 \beta} & \frac{\gamma^2}{f^2 \alpha^2 \beta^2} + \frac{1}{f^2 \beta^2} & -\frac{\gamma(\gamma v_0 - \beta u_0)}{f^2 \alpha^2 \beta^2} - \frac{v_0}{f^2 \beta^2} \\ \frac{\gamma v_0 - \beta u_0}{f^2 \alpha^2 \beta} & -\frac{\gamma(\gamma v_0 - \beta u_0)}{f^2 \alpha^2 \beta^2} - \frac{v_0}{f^2 \beta^2} & \frac{(\gamma v_0 - \beta u_0)^2}{f^2 \alpha^2 \beta^2} + \frac{v_0^2}{f^2 \beta^2} + 1 \end{bmatrix}, \end{aligned} \quad (2.21)$$

then, to write the orthonormality equations (Equation 2.20) in a homogeneous matrix-vector form, we write

$$\mathbf{v}_{ij}^T \mathbf{b} = \mathbf{h}_i^T \mathbf{B} \mathbf{h}_j, \quad (2.22)$$

with:

$$\mathbf{b} = [B_{11} \quad B_{12} \quad B_{22} \quad B_{13} \quad B_{23} \quad B_{33}]^T, \quad (2.23)$$

and thus:

$$\mathbf{v}_{ij} = \begin{bmatrix} h_{i1} h_{j1} \\ h_{i1} h_{j2} + h_{i2} h_{j1} \\ h_{i2} h_{j2} \\ h_{i3} h_{j1} + h_{i1} h_{j3} \\ h_{i3} h_{j2} + h_{i2} h_{j3} \\ h_{i3} h_{j3} \end{bmatrix}. \quad (2.24)$$

⁵ The λ was introduced because \mathbf{H} has been defined upon a scale factor.

⁶ These 5 degrees of freedom are the parameters $\alpha, \beta, \gamma, u_0, v_0$ and f with the normalisation constraint on α, β, γ stated in Equation 2.4

This way we obtain the constraints in matrix-vector form for every image:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0}. \quad (2.25)$$

When we do this for all of the n images, we get a matrix \mathbf{V} with dimensions $2n \times 6$. \mathbf{b} can be solved up to a scale factor λ by calculating the eigenvector corresponding to the smallest eigenvalue of $\mathbf{V}^T \mathbf{V}$. From this vector \mathbf{b} we can extract the parameters in \mathbf{AF} as following:

$$\begin{aligned} v_0 &= (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2), \\ \Lambda &= B_{33} - (B_{13}^2 + v_0 (B_{12}B_{13} - B_{11}B_{23})) / B_{11}, \\ \alpha f &= \sqrt{\Lambda / B_{11}}, \\ \beta f &= \sqrt{\Lambda B_{11} / (B_{11}B_{22} - B_{12}^2)}, \\ \gamma f &= -B_{12} (\alpha f)^2 \beta f / \Lambda, \\ u_0 &= \gamma v_0 / \beta - B_{13} (\alpha f)^2 / \Lambda, \\ f &= \frac{1}{\sqrt{2}} \left\| \begin{bmatrix} \alpha f & \gamma f \\ 0 & \beta f \end{bmatrix} \right\|_{fro}. \end{aligned} \quad (2.26)$$

Once the parameters contained in \mathbf{AF} are known, the extrinsic parameters \mathbf{R} and \mathbf{t} can be calculated from Equation 2.18 using

$$\begin{aligned} \lambda &= 1 / \|(\mathbf{AF})^{-1} \mathbf{h}_1\| = 1 / \|(\mathbf{AF})^{-1} \mathbf{h}_2\| \\ \mathbf{r}_1 &= \lambda (\mathbf{AF})^{-1} \mathbf{h}_1 \\ \mathbf{r}_2 &= \lambda (\mathbf{AF})^{-1} \mathbf{h}_2 \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \lambda (\mathbf{AF})^{-1} \mathbf{h}_3. \end{aligned} \quad (2.27)$$

However the obtained rotation matrix $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]$ is not guaranteed to be orthonormal, due to noise on the keypoint positions. As Zhang states in Appendix C of [1], the singular value decomposition $\mathbf{USV}^T = \mathbf{R}$ can be used to fit a proper rotation matrix \mathbf{R}_{new} to the calculated \mathbf{R} . According to Zhang this new rotation matrix would be $\mathbf{R}_{\text{new}} = \mathbf{UV}^T$, but actually the determinant of \mathbf{R}_{new} should still be checked, because if it is -1 the matrix is mirroring as well. In that case $\mathbf{R}_{\text{new}} = -\mathbf{UV}^T$ should be used.

2.4.3 Radial Distortion Parameter Estimation

Most lens distortions are dominated by radial distortions and especially by the first order radial distortion. In literature often the first two radial distortions [1, 16, 17] and sometimes also two tangential distortion components are estimated [18]. An implementation was made for estimation of either 2 (radial) or 4 (2 radial and 2 tangential) distortion parameters.

For undistortion only the non-magnifying camera matrix \mathbf{A} and the distortion factors are needed. Therefore the zooming \mathbf{F} and extrinsic transformations (rotation and translation) should already be applied to the model points before the distortions are added. The distorted image points are therefore calculated from the normalised model keypoints in camera coordinates $[\mathbf{x}, \mathbf{y}, \mathbf{z}]_{c,n}^T$, pre-multiplied with the zooming matrix. The radial distortions can be described using⁷⁸

$$\check{u} = u + (u - u_0) \left[\kappa_1 (\mathbf{x}^2 + \mathbf{y}^2) + \kappa_2 (\mathbf{x}^2 + \mathbf{y}^2)^2 \right] \quad (2.28)$$

$$\check{v} = v + (v - v_0) \left[\kappa_1 (\mathbf{x}^2 + \mathbf{y}^2) + \kappa_2 (\mathbf{x}^2 + \mathbf{y}^2)^2 \right], \quad (2.29)$$

where κ_1 and κ_2 are the radial distortion parameters. Now for every keypoint in every image we can write

$$\begin{bmatrix} (u - u_0)(\mathbf{x}^2 + \mathbf{y}^2) & (u - u_0)(\mathbf{x}^2 + \mathbf{y}^2)^2 \\ (v - v_0)(\mathbf{x}^2 + \mathbf{y}^2) & (v - v_0)(\mathbf{x}^2 + \mathbf{y}^2)^2 \end{bmatrix} \begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix} = \begin{bmatrix} \check{u} - u \\ \check{v} - v \end{bmatrix}. \quad (2.30)$$

If we stack all these equations we can find the values for κ_1 and κ_2 in the least-squares sense using the pseudoinverse (see also §B.2).

2.4.4 Solution Refinement

All parameters have been determined now. However, the homography, rotation matrix and translation vector and camera intrinsic parameters were determined under the assumption that there were no distortions. This means that these estimates are inaccurate and thus the calculated radial distortions will be as well. An iterative process could lead to an optimal estimation of all parameters, but according to Zhang, a non-linear least-squares optimisation taking into account all parameters at the same time performs better.

The optimised parameter values will be calculated by the minimisation of

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{m}_{ij} - \check{\mathbf{m}}(\mathbf{A}, \kappa_1, \kappa_2, \mathbf{R}_i, \mathbf{t}_i), \mathbf{M}_j\|^2 \quad (2.31)$$

using the Levenberg-Marquardt routine[19], with the previously calculated parameter values as initial guess. To reduce the number of parameters to optimise and to make sure an unconstrained optimisation can be carried out, the rotation matrices will be exchanged by Rodrigues parameters. Rodrigues' formula (§B.3) can be used to describe a rotation with the absolute minimum of 3 parameters and provides easy conversion from and to the original 3×3 rotation matrix.

2.5 Images Undistortion

Most procedures to undo images from their distortions somehow use the inverse of the distortion function. For the distortion functions used here (Equation 2.29)

⁷ For readability the subscripts and pre-multiplications with f were left out.

⁸ The derivation of these formulas can be found in §B.2.

according to literature this would mean that an iterative process is needed. Other researches have focussed on developing distortion functions that can easily be inverted[17].

All these efforts seem unnecessary if we realise that we can just distort an ‘empty’ destination image, see where the pixels would go, and extract the image data at these positions from the (distorted) source image. Actually the functions used for translating model points in camera coordinates to distorted image points (Eqs. 2.28 and 2.29) are used again, but now with the coordinates of all pixels instead of the keypoints. This procedure is illustrated by figure 2.11.

This newly developed technique is practical, accurate and fast, but for a real-time implementation things can be speeded up even more. A file with undistortion parameter values is read upon the program start-up and a lookup-table is created defining for every point in the undistorted image, from which pixel in the distorted image the intensity should be taken. In real-time, the only thing that needs to be done is to apply the lookup-table to transform the image pixels, which is very fast. Interpolation could be done, but this will slow down the undistortion process.

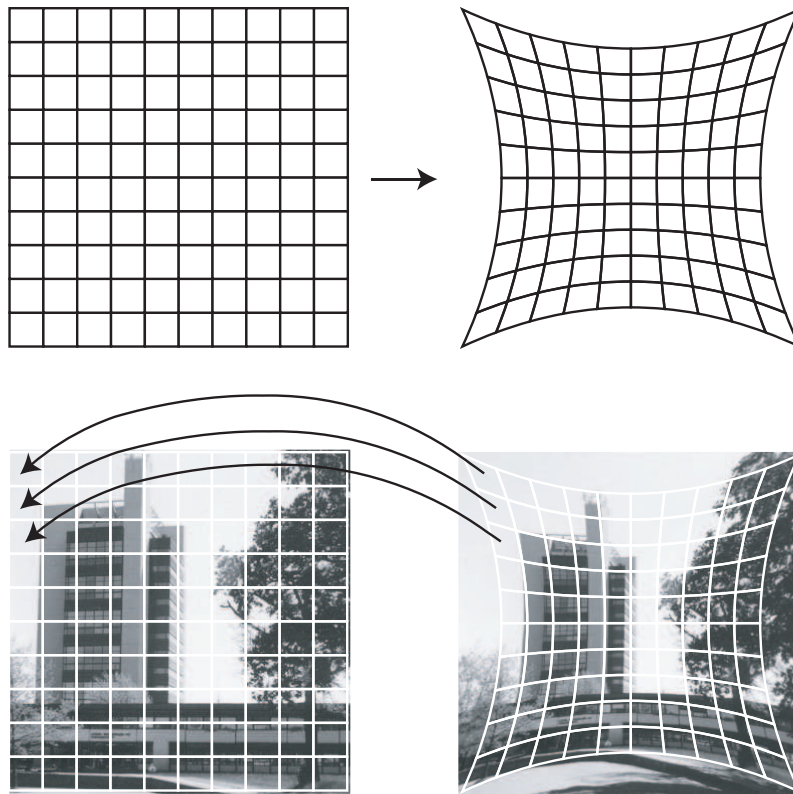


Figure 2.11: First the destination pixel coordinates are distorted, then the intensity of every destination pixel is determined by looking at the intensity at the distorted pixel position in the distorted source image.

Chapter 3

Camera Calibration Experiments

A number of experiments have been carried out to test the camera calibration strategy discussed in Chapter 2. The image processing and keypoint extraction appeared to be quite sensitive to the quality of the images provided. Both ‘good’ and ‘bad’ quality images¹ were used in separate tests to verify performance on the one hand and practical applicability on the other hand.

3.1 Introduction

Many series of images have been captured and processed in various ways. In the evaluation of the results several methods were used to measure the performance of the estimated parameters in undistortion. Analysis techniques have been developed and settings have been tuned during the testing phase. Where possible, old results have been re-analysed with new settings to improve comparability.

3.1.1 Overview of Experiments

The experiments covered in this chapter are:

- Experiments on keypoint extraction, in §3.2
- Calibration based on simulated data sets and images, in §3.3
- A comparison with the results obtained by Zhang, in §3.4
- Calibration of a photo camera, in §3.5
- Calibration of the optical system in the laser welding head, in §3.6

The tests in general were carried out in the following way:

1. 10–25 Images are taken and their keypoints are extracted
2. Image subsets are made (usually consisting of 4 images)

¹ ‘Good’ images are sharp and have a high contrast between the pattern and the background

3. All subsets are fed to the calibration algorithm (when many subsets are available, the total number is mostly limited to 2000–3000 sets)
4. The results are processed:
 - (a) The average number of refinement iterations is calculated per image
 - (b) The average and standard deviation of all subset results are calculated for every estimated parameter
 - (c) The average and standard deviation of the subsets with lowest RMS (see §B.4) are calculated for every estimated parameter
 - (d) The averaged results for *all* and for the *lowest RMS* subsets are saved separately for use with the undistortion algorithm.

3.1.2 Performance Standards

Several performance standards have been applied to measure the quality of a parameter set resulting from the calibration routine. Where possible the outcomes were related to known reference values, otherwise a visual or mathematical analysis was made of images undistorted with the calculated parameters.

Histograms

The results of all subsets are compared with each other by calculating the averages and standard deviation over the subsets for every parameter (§3.1.1). To visualise these results, histograms have been plot. The example in Figure 3.1 shows a histogram for a set results like [2, 1, 3, 2]. The resulting values are on the horizontal axis and their frequency of occurrence is on the vertical axis. This means that the bar at 2 has height 2, because it is twice in the set of results, whereas the bars at 1 and 3 have height 1.

In the histograms throughout this chapter, the vertical axis has been normalised. This has been done because the different optimisations have different numbers of subsets. The ideal histogram is supposed to show a sharp peak (with normalised height 1) at the actual parameter value.

Analysis of Undistorted Images

In the end the estimated parameters should result in a proper undistortion of images. To test this the original images were undistorted using the parameters found by calibration. Keypoints were extracted from the undistorted images and horizontal and vertical lines were fit through the keypoints. The residual of the fit was used as a measure to compare the quality of different estimates.

Also some laser line projections on a flat surface were captured. Using photo-editing software the lines were interrupted approximately halfway. Straight lines were fit through both parts before and after undistortion. As in reality the projected line is straight, no angle should be existing between the two detected lines. Therefore, the angle found from the images is a measure for the quality of undistortion.

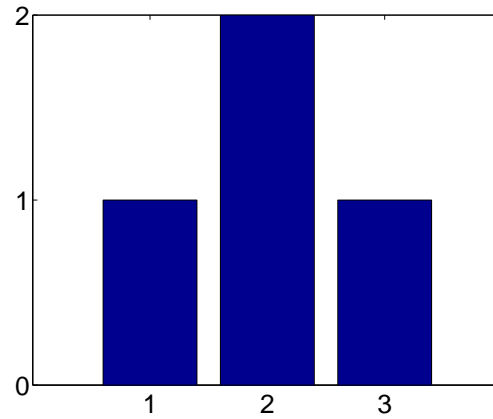


Figure 3.1: Example histogram of a set experiments resulting in a value 1 once, 2 twice and 3 once.

Note:

Most values in this chapter are the result of a normalised parameter optimisation. This means that the values do not immediately represent a physical quantity. The values are used in this normalised form for undistortion as well, which makes a comparison of these values more sensible than one of the un-normalised values.

3.2 Keypoint Extraction

Several thresholding algorithms were implemented to be able to process images of varying quality. Most of these are result of testing different normalisation and thresholding strategies on images that were not thresholded satisfactory with any of the existing algorithms. Appendix C contains detailed information about the algorithms and their applicability.

In general it can be stated that only for qualitatively good images thresholding can be accurate enough to extract meaningful corner keypoints. This is because the corners are far more sensitive to inaccurate thresholding than centroids, as explained in §B.1.

For sharp images the keypoint extraction from a thresholded image works very good, even for corners. Figure 3.2(a) shows that on the images used by Zhang the proposed thresholding and keypoint extraction algorithms even perform better than the algorithms he uses (which have not been published).

The images taken with the welding head however, suffer from the large aberrations of the laser focus lens, which makes it very difficult to determine a proper threshold (Figure 3.2(b)). This almost automatically results in badly determined positions for the corner keypoints (Figure 3.2(c)). For centroid keypoints this is less an issue, however a weighed centroid determination like proposed in §C.3 is recommended.

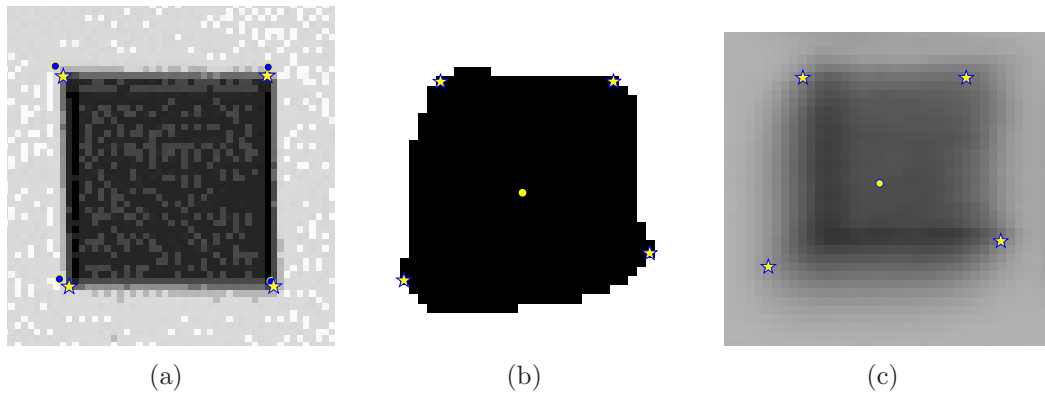


Figure 3.2: Determination of corner keypoints:

- (a) cut-out of an image by Zhang, keypoints extracted after dynamic thresholding II (§C.2.2) depicted as stars, Zhang’s data (with software by Brian Guenter) as dots.
- (b) cut-out of a thresholded image taken using the robot head, it is difficult to indicate where the corners are
- (c) projection of the keypoints determined in (b) in the original image, determining a proper threshold level is difficult, but critical

Labelling Problems

Some problems occurred due to labelling (§A.3) too many or too few objects. Objects connected to the border of the image and objects smaller than a certain size are neglected automatically, but sometimes an extra opening operation (erosion followed by dilation, see also §A.4) was needed to make sure that abusively thresholded parts in the background got connected to the edge. If calibration objects would have been poorly separated, a closing operation (dilation followed by erosion) could have been applied.

3.3 Simulations

For testing purposes several simulations have been carried out. Sets with keypoints have been generated and images have been generated artificially to provide a controlled testing environment with known camera and lens errors and known rotations and translations. The details of the generation of these datasets and images can be found in §D.2.3.

3.3.1 Simulation Data

Non-normalised Data

One test was performed with non-normalised data. The average calculation time for a set of 4 images is 1 minute, of which the major part is consumed by the parameter refinement. The refinement took 185 iterations on average, which is a huge number compared to the 25 iterations (taking 4.4 seconds) which generally

suffice when the parameters are normalised.

Estimates are very inaccurate and the spread in parameters found by analysis of different sets is large. When only the 10% results with the lowest RMS values are considered the results are quite well.

No Radial Distortions

Another test was carried out without radial distortions. A little random noise (about 0.1% of the image width) was added to the keypoint position data to prevent singularities in the calculation process. The parameter refinement converged in only 10 iterations, taking about 1.8 second per set. It could be expected that this converges faster than the other normalised tests (25 iterations), because the parameters estimated using the homography do not suffer from inaccuracies due to radial distortions, so they are already quite accurate.

The standard deviation of the parameters calculated from different sets of images are a lot smaller than in the other tests. The parameters are estimated accurately. Only the optical centre $[u_0, v_0]$ is less accurate, probably because the radial distortion enhances the effect of its location.

Different Noise Levels

Tests with increasing noise levels show an increasing variance in the results of the different image sets, as displayed in Figure 3.3. Especially the radial distortions appear difficult to estimate. The value for the first radial distortion parameter κ_1 is actually always wrong, probably because it is so small and its effect is overshadowed by the second order radial distortion.

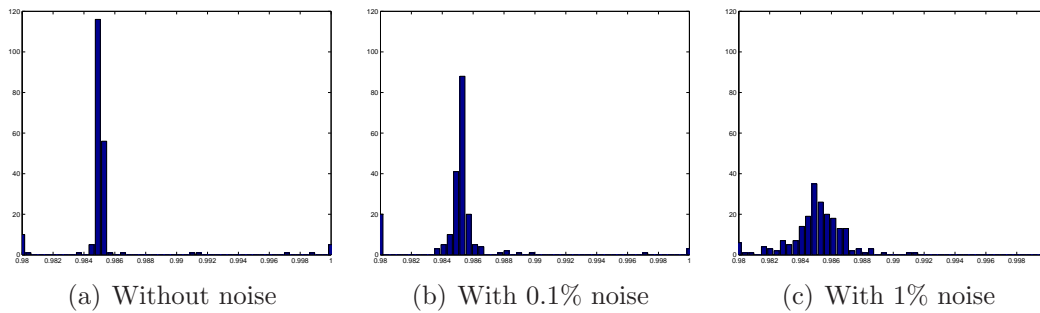


Figure 3.3: Histograms for the variable α .

Conclusions

- Normalisation saves a lot of time and improves accuracy,
- Taking the results from images sets with the 10% lowest RMS values highly improves accuracy,
- Noise and radial distortions do influence convergence, but results are still meaningful.

3.3.2 Simulation Images

Keypoint Ordering

From all generated test images, only for one the extracted keypoints could not be ordered properly. This image had huge radial and perspective distortions, which are very unlikely to occur in practice. Aside from this single image, the ordering algorithm has proven to be considerably robust.

Varying the Number Of Images In a Subset and the Maximum Rotation Angles

Figures 3.4 and 3.5 show some of the results from 4 different tests. Figure 3.4 contains results from parameter estimations where images with small rotation angles (up to 5°) were simulated. This means that perspective distortions are relatively small. For the results in the left column subsets of 4 images were used to base the parameter estimation upon, for the right column subsets of 9 images were used. For Figure 3.5 large rotation angles (up to 45°) were simulated. The images at the left are for estimations with subsets of 4 again and the images at the right for subsets of 9 images.

The first row of histograms in both figures shows the RMS value, which is a measure for the quality of the estimation. As a lower RMS indicates a higher quality, the estimates of the experiments with small angles seem to be better than those of the experiments with large angles. In all cases two or more peaks are visible. This indicates that there are local optima present in the optimisation space. For the other parameter histograms, peaks mainly incorporating results from sets in the low-RMS peak are indicated with an arrow whenever appropriate.

For images with small rotations, it appears to be beneficial to use more images in a subset. The parameter estimations are clearly more stable when subsets of 9 images are used, then when 4 images are used. When only results with a low RMS are taken into account, the number of usable results is higher and the histograms show sharper peaks.

An Accurate Estimation of f

A problem however, is that the factor f is estimated totally wrong. As listed in §D.2.3, the value should be 40, whereas it is estimated somewhere between 5 and 20. This is probably due to the fact that the difference between the factor f and the translation from and to the pattern Δz (which is in \mathbf{t}) have almost the same effect on an image. The only way to distinguish between them is by the perspective distortions, which are very small when only small rotations are present. Numeric results confirm the strong coupling of the estimated values of f and Δz .

For undistortion no accurate estimation of f is needed, as it is just a scaling from the model to the image. Actually f was extracted from the camera matrix $\mathbf{A}\mathbf{F}$ to prevent images from zooming when being undistorted. This means that for undistortion only the parameters in the normalised matrix \mathbf{A} and distortion parameters κ_1 and κ_2 are important. A proper estimate of f however, could be

used to relate image coordinates in pixels to real world coordinates in millimetres. Therefore simulations using images with large rotations have been done to try and find a more accurate value for f .

Figures 3.5(i) and (j) indeed show that with larger rotation angles (and thus larger perspective deformations) lead to a better estimate for f . Except for small improvements of u_0 and v_0 , which can be estimated from both radial distortion effects and perspective deformations, the other parameters however are estimated a lot worse: the histograms for α for instance (Figures 3.5(c) and (d)) are wide and the main peaks are at about 1.1 whereas they should be at 1.0094.

From the sharp peaks in Figures 3.5(e)–(h), one may think that the results for κ_1 and κ_2 have improved, but the opposite is true as the peaks are positioned near 0 whereas they should be at 0.015 and $1 \cdot 10^{-3}$ for κ_1 and κ_2 respectively. Only few results (the ones with low RMS) have an accurate value.

The Influence of Thresholding

The histograms in Figure 3.6 show the performance of some different thresholding techniques (these are discussed in detail in Appendix C). For these experiments images without radial distortions, but with blurring and light spots were used².

Figure 3.6(a) shows that the (centroid) keypoints found from the static thresholding algorithm are not very consistent within an image, as the RMS values are quite high. The fact that the peak is relatively narrow indicates that the errors made in the different images are almost the same. The dynamic thresholding algorithms (Figures 3.6(b) and (c)) perform quite a bit better for some images, but the difference between images is larger. Weighed keypoint determination (Figure 3.6(d)) appears to find the most accurate positions for the keypoints for blurred images with non-homogeneous lighting.

Estimating Tangential Distortions

A test with tangential distortions, shows that it is very difficult to properly estimate the parameters when tangential distortions are present. The fact that the results are so poor, even with a perfectly matching model, is not very promising for real-life tests.

Conclusions

- Undistortion of images works, as Figure 3.7 shows,
- When using images with small rotation angles, using subsets of 9 instead of 4 images for parameter estimation gives more results that can be considered accurate,
- For an accurate estimation of the factor f large rotations are needed,

² Images with radial distortion show the same tendency, although the RMS values are considerably higher ($8 \cdot 10^{-3}$ instead of $7 \cdot 10^{-3}$) Images without blurring result in slightly lower RMS values.

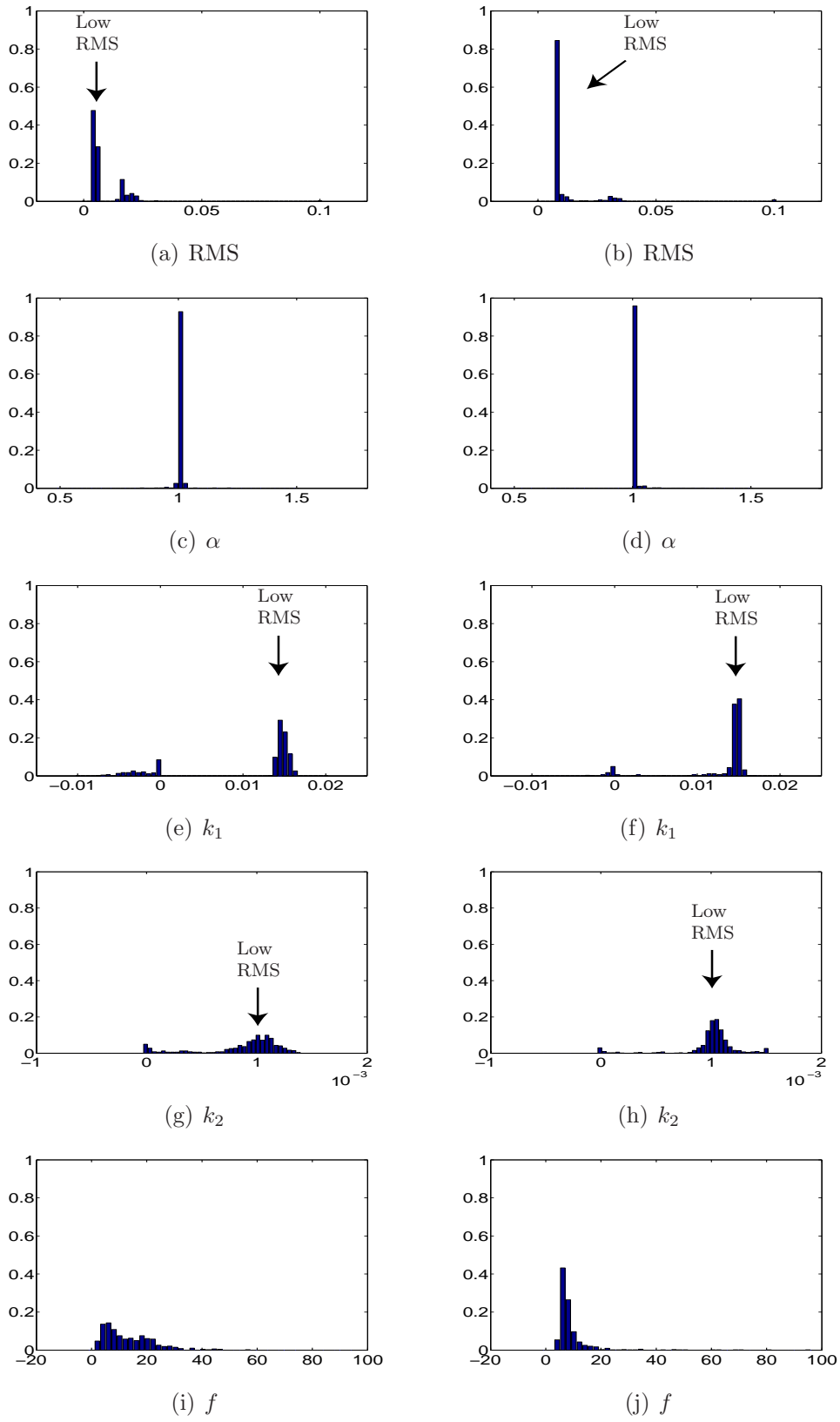


Figure 3.4: Overview of results for simulated images with relatively small rotations
(c) (e) (g) (a) and (i) are results for subsets of 4 images,
(d) (f) (h) (b) and (j) are results for subsets of 9 images.

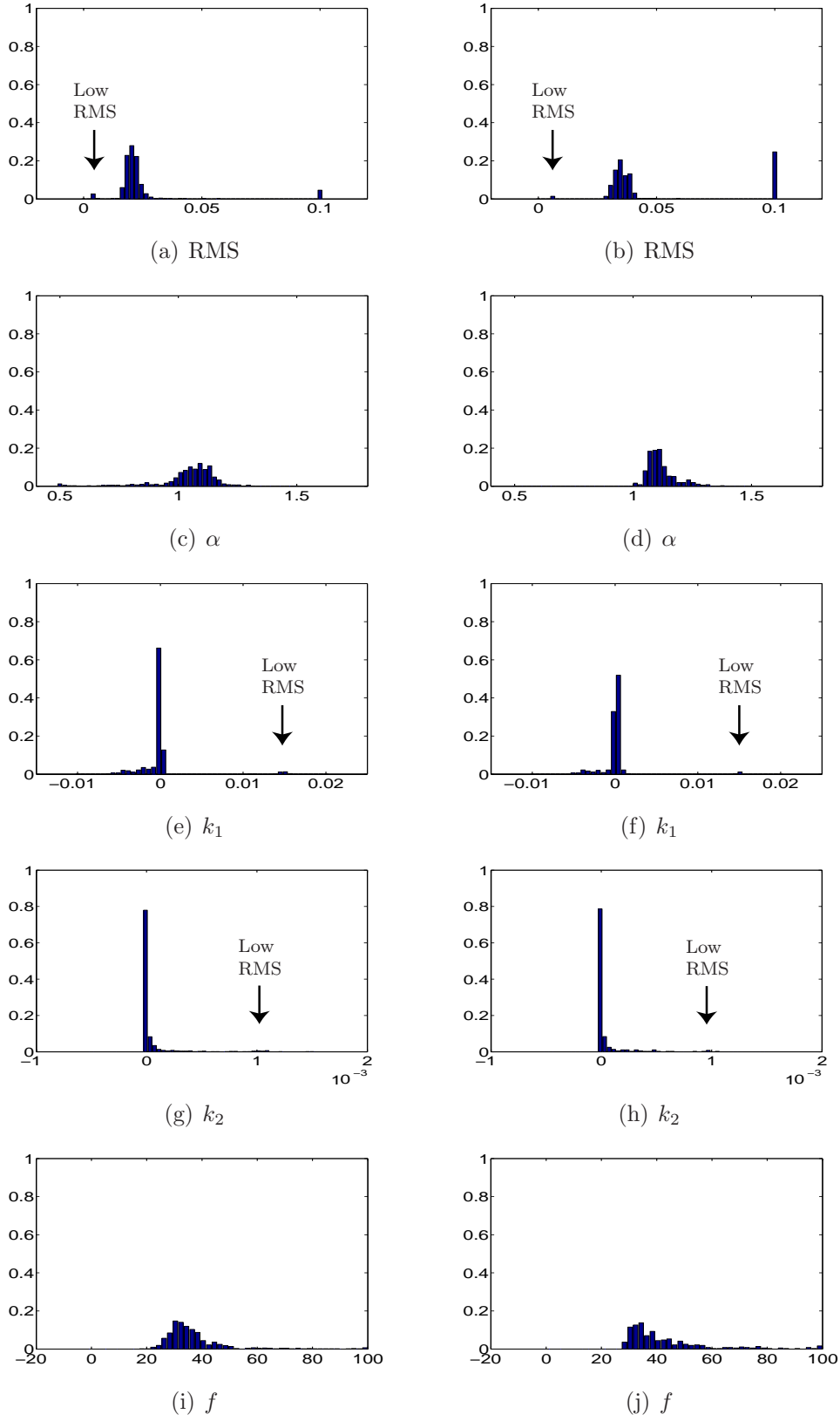


Figure 3.5: Overview of results for simulated images with relatively big rotations

(c) (e) (g) (a) and (i) are results for subsets of 4 images,

(d) (f) (h) (b) and (j) are results for subsets of 9 images.

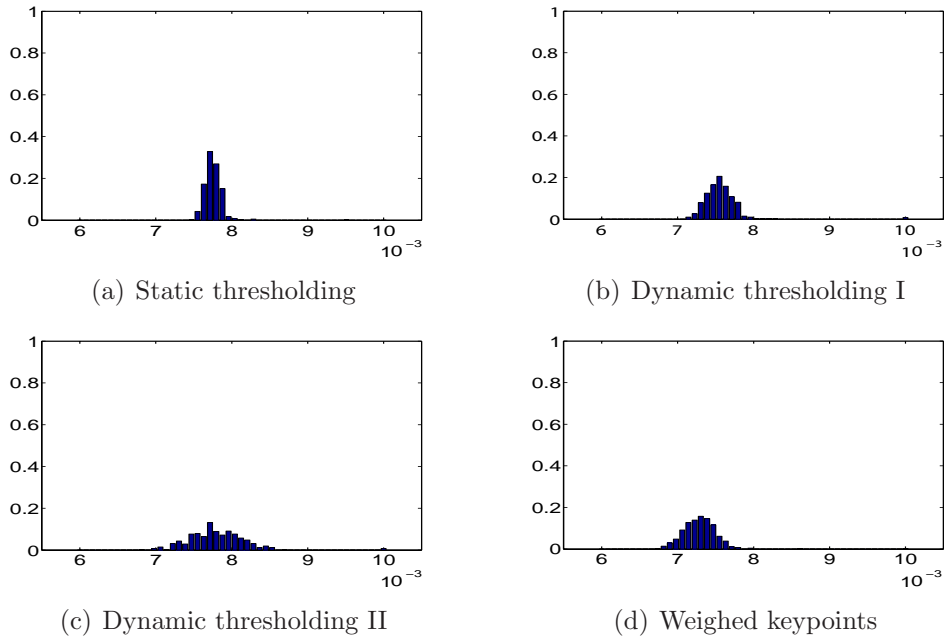


Figure 3.6: Overview of the RMS values after parameter estimation based upon 4 simulated images without radial distortions, but with blurring and lights.

- Large rotations have a bad influence on the general quality of the parameter fits,
- A poorly estimated value for f does not influence undistortion, as it will be compensated for by a poor estimate of Δz ,
- The conclusion of the previous test, that taking the only results from image sets with the lowest RMS values highly improves accuracy, has been confirmed, as even in the test with big rotations the sets with low RMS provide good results,
- For blurred images with non-homogeneous lighting, the weighed keypoint determination gives the best keypoint locations,
- Estimation of tangential distortions should be refrained from, as calibration in that case does not provide meaningful values.

3.4 Zhang's Test Images

The images and extracted corner keypoints Zhang uses to obtain the results presented in his paper[1] are available from his website. These files have been used to verify the results of the MATLAB implementation with the ones stated in the paper.

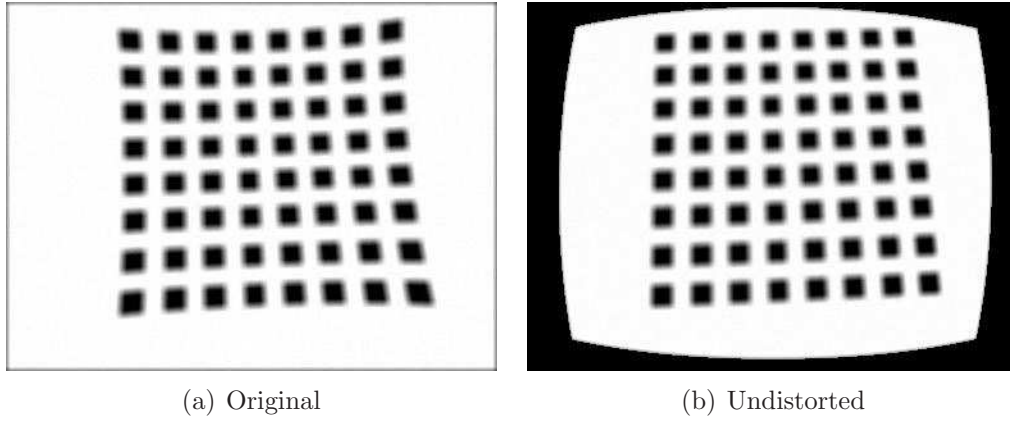


Figure 3.7: Undistortion of a simulated image.

3.4.1 Zhang's Test Data

The keypoint data provided by Zhang were fed to the MATLAB camera calibration program. To compare the output of the program (Table 3.1) to the results Zhang presents in his paper (Table 3.2), the values in Table 3.1 were un-normalised and compensated for taking the factor f out of the camera matrix \mathbf{AF} .

quadruple	(1234)	(1235)	(1245)	(1345)	(2345)	mean	st.dev.
α	832.289	836.402	836.380	834.087	836.525	835.137	1.890
β	832.289	836.402	836.547	834.253	836.525	835.203	1.896
γ	0.250	0.167	0.167	0.250	0.167	0.200	0.045
u_0	304.924	303.869	302.662	304.175	303.013	303.729	0.908
v_0	204.864	204.100	208.503	205.690	205.354	205.702	1.676
κ_1	-0.246	-0.221	-0.227	-0.223	-0.222	-0.228	0.010
κ_2	0.263	0.179	0.179	0.177	0.179	0.196	0.038
RMS	0.719	0.719	0.535	0.719	0.673	0.673	0.079

Table 3.1: Parameter estimates for several subsets of 4 images out of 5 using the MATLAB camera calibration program.

Table 3.3 shows the difference in the results, where 1.00 means that there is no difference and 2.00 would mean that the value found using the MATLAB implementation is twice as big as the value Zhang reports. Although most values are almost equal, some differences are remarkable.

The differences in the variable γ – which is the skew factor in the images – can be explained by the fact that the standard deviations are about a quarter of the mean values (see Tables 3.1 and 3.2), which indicates that it is hard to estimate this variable correctly. The facts that the skew is very small (less than 0.01°) and that perspective may result in a similar effect may play a role in this.

The difference in the RMS (Root Mean Square) values is hard to explain. Because the factor is always about 2 some difference in the calculation may exist³. On the other hand it could also mean that the MATLAB optimisation did

³ A factor $\sqrt{2}$ could possibly be explained by Zhang taking the errors in u and v separately,

quadruple	(1234)	(1235)	(1245)	(1345)	(2345)	mean	st.dev.
α	831.810	832.090	837.530	829.690	833.140	832.850	2.900
β	831.820	832.100	837.530	829.910	833.110	832.900	2.840
γ	0.287	0.107	0.061	0.136	0.110	0.140	0.086
u_0	304.530	304.320	304.570	303.950	303.530	304.180	0.440
v_0	206.790	206.230	207.300	207.160	206.330	206.760	0.480
κ_1	-0.229	-0.228	-0.230	-0.227	-0.229	-0.229	0.001
κ_2	0.195	0.191	0.193	0.179	0.190	0.190	0.006
RMS	0.361	0.357	0.262	0.358	0.334	0.334	0.040

Table 3.2: Parameter estimates obtained by Zhang.

quadruple	(1234)	(1235)	(1245)	(1345)	(2345)	mean	st.dev.
α	1.00	1.01	1.00	1.01	1.00	1.00	0.65
β	1.00	1.01	1.00	1.01	1.00	1.00	0.67
γ	0.87	1.56	2.74	1.84	1.53	1.43	0.53
u_0	1.00	1.00	0.99	1.00	1.00	1.00	2.06
v_0	0.99	0.99	1.01	0.99	1.00	0.99	3.49
κ_1	1.07	0.97	0.99	0.98	0.97	1.00	10.37
κ_2	1.35	0.94	0.93	0.99	0.94	1.03	6.32
RMS	1.99	2.01	2.04	2.01	2.01	2.01	1.99

Table 3.3: Results of the MATLAB implementation divided by the results listed in Zhang's paper.

not converge properly, however stricter convergence criteria did not make much difference.

The difference in standard deviations is remarkable, as the first 3 parameters, α , β and γ are twice as stable, but the other parameters are varying more. The most likely cause is the fact that the factor f was taken out of the camera matrix \mathbf{AF} . A more sophisticated convergence criterion for the parameter refinement (§2.4.4) might improve the reliability of the estimates⁴.

Conclusions

- The MATLAB implementation works properly,
- Taking the zoom factor f out of the camera matrix \mathbf{AF} slightly affects the standard deviation in results from different image sets.

while in the MATLAB implementation the error *distance* was used.

⁴ For u_0 and v_0 the difference in definition (with respect to the upper left corner for Zhang and to the image centre for the MATLAB implementation) may also influence accuracy and convergence criteria.

3.4.2 Zhang's Test Images

Using the images Zhang provides means that the keypoints should still be detected. Because the images provided are of good quality, there should not be much difficulty in finding the keypoints. As mentioned in §3.2 the detected corners appear to be even more accurate than Zhang's data when judged by eye.

To have a measure for the undistortion quality, horizontal and vertical lines were fit through extracted keypoints. The lower the norm of residuals summed over all fitted lines, the smaller the distortions are considered to be. A compensation was done for the fact that calculations with corner keypoints will have twice as many lines as those with centroid keypoints and that there will be twice as many points on each line.

The undistorted images have a total residual of about 300 pixels when corner keypoints are fitted and about 150 pixels when centroids are fitted. This difference can be explained by the fact that corners are harder to detect accurately as discussed in §B.1. Another point is that the centroids are an average over the whole object and thus take less extreme values than the corner keypoints.

After undistortion the residuals are around 160 and 30 pixels for measurements using corners and centroids respectively. Whether the undistortion itself is based on centroid or corner keypoints does not make a big difference, though generally the undistortion using centroid keypoints is slightly better. The results are in line with those obtained after undistortion using Zhang's parameters.

Conclusions

- Keypoint extraction is accurate,
- Undistortion does improve the straightness of lines in images,
- The undistortion results obtained with the MATLAB implementation are as accurate as those obtained by Zhang.

3.5 Photo Camera Calibration

A semi-professional digital photo camera (for specifications see §D.3) has also been put to the test. Because the lenses used for photo cameras are optimised for imaging, images are much clearer than those taken using the welding head. Moreover, the high resolution and large field-of-depth result in much better images and keypoints are much easier to detect. A drawback for undistortion however, is that the radial distortions are already very small.

Memory Problems

The first problem encountered in keypoint extraction and calibration of the photo camera images was the fact that the 6 mega pixel images require a lot of memory for storage. Especially when multiple copies are used for image processing 'out of memory' errors are likely to occur. The current implementation has been

improved on this issue, but for larger images or computers with little memory problems may still occur.

Varying the Number Of Images In a Subset and the Maximum Rotation Angles

Figure 3.8 shows some calibration results. The histograms at the left are from calibration of photos with small rotations and 9 images per subset. The histograms at the right are for large angles and 4 images per subset. These two calibrations were chosen because they provide the most insightful results.

The parameter α is clearly estimated more stable when larger rotation angles and less images per set are used (Figure 3.8(b)). This is typical for all camera parameters and especially for f (Figure 3.8(f)). The radial distortions however are clearly estimated more robust when smaller angles and more photos are used (Figure 3.8(c)).

A comparison of distorted and undistorted images⁵ shows that undistortion using the parameters obtained from images with small rotations even has a negative influence. Also when the radial distortions are set to zero, the results are worse than the original images. Parameters obtained from images with large rotation angles result in good undistortions, which shows the importance of properly estimated camera parameters.

Estimating Tangential Distortions

Estimations of the tangential distortions have been done as well. Their influence appears to be negligible compared to the radial distortions, which is fully in line with literature. Actually the negative influence the estimation of these two extra parameters has on the convergence in general is worse than the good they do in undistortion.

The Effects of Distortions

The graph in Figure 3.9(a) shows the amount of radial distortion present for every point along the diagonal of the image⁶. It shows that in the centre of the images the distortions are negligible, whereas the corners of the image moved almost 15 pixels (which would only be 0.75 mm when printed as a normal photo).

⁵ Lines were fitted both horizontally and vertically through centroid keypoints extracted from the images. The residual of the fit is normative.

⁶ The results are based on the calibration with small rotation angles and 9 images per subset.

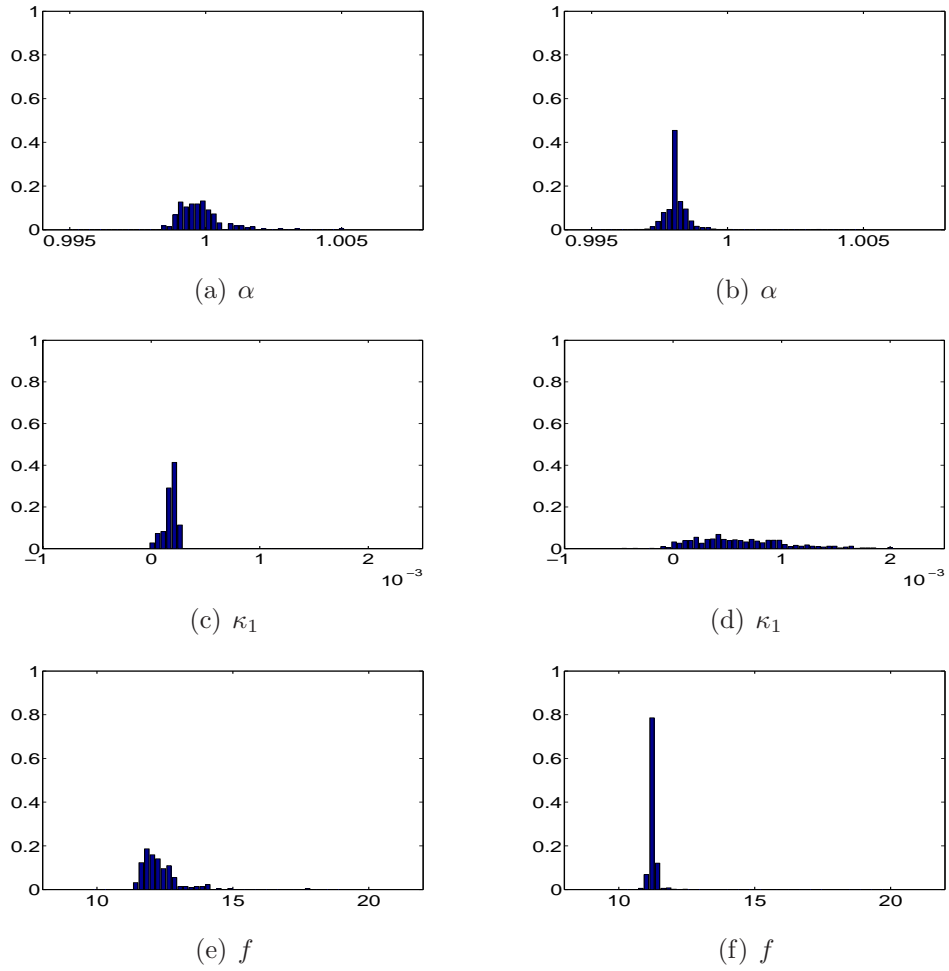
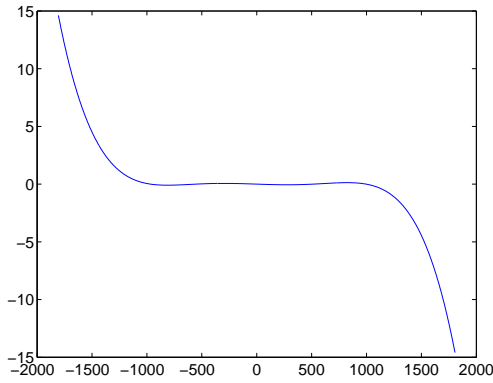
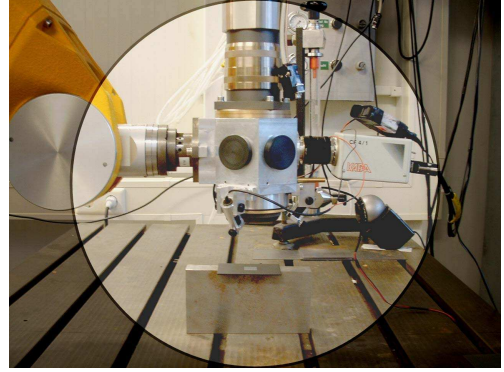


Figure 3.8: Overview of some calibration results for photo camera images:

- (a) (c) and (e) with relatively small rotations and 9 images per set,
- (b) (d) and (f) with relatively large rotations and 4 images per set.



(a) Shift of pixel data due to radial distortions. The shift in pixels is plot against the distance from the image centre of a 3008×2000 pixel image.



(b) The area with negligible radial distortions is highlighted. These are all pixels within a radius of circa 1000 pixels from the image centre, as shown in (a)

Figure 3.9: Radial distortions in images from a semi-professional photo camera.

Conclusions

- The conclusion from the simulations, that using subsets of 9 instead of 4 images for parameter estimation gives more results that can be considered accurate when using images with small rotation angles, is confirmed,
- The conclusion from the simulations, that for an accurate estimation of f large rotations are needed is confirmed,
- For estimating the camera parameters few images with large rotations are preferred above many images with small rotations
- For estimating the radial distortions many images with small rotations are highly preferred,
- The distortions in a semi-professional photo camera are very small indeed

3.6 Welding Head Video Camera Calibration

Due to the bad imaging properties of the high power focus lens, the images shot with the welding head have large distortions and suffer from severe aberrations. Furthermore the lens system has a very limited field of depth, which results in even unsharper images when the calibration pattern is rotated over large angles. This makes calibration upon these images a true challenge.

Several calibration patterns have been produced for testing purposes (see §D.2). In the end, the laser-engraved patterns with 8×8 objects, version 2 and the one with 12×12 objects appeared to have the best performance. This is mainly because the separation of objects is easier than for the patterns with a smaller object spacing.

The Effect of the Image Resolution

One test has been carried out to see whether there is a difference in results obtained from full-resolution or half-resolution images. Each time two images were taken immediately after each other, one using frame option provided by the frame grabber (§D.1.3) and one using the field option. Both sets of images have been processed in the same way. No distinct differences could be found in the results.

Imaginary Calibration Results

In some of the earlier tests (with 8×8 objects and small spacing) imaginary calibration results showed up⁷. This is probably due to badly conditioned matrices, caused by highly correlated or contradicting input data⁸. Imaginary results are being detected before the parameter refinement and the calibration will be terminated for the subset in question.

A remarkable improvement of the overall result can be obtained by leaving images out that often occur in sets with imaginary results. Figure 3.10 shows that the images that show up in imaginary results mainly show up in the results with high RMS. This means that these images somehow clearly have a bad influence on the whole fit. Not only the RMS value, but also especially the first radial distortion parameter κ_1 and the factor f show this dependency. To be able to discard bad data when reviewing the results, it is important to check how often an image shows up in imaginary results.

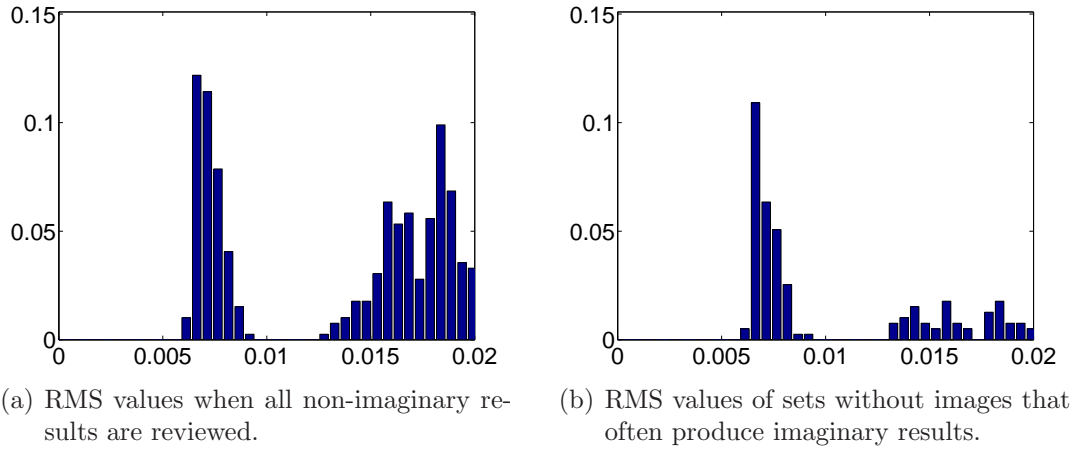


Figure 3.10: The images that often occur in sets with imaginary results also have a bad influence on the other sets they are in.

⁷ The imaginary numbers originate in the calculation of αf and βf in Equation 2.26

⁸ Imaginary results could also be obtained using Zhang's keypoint data, when one data set was used multiple times as input for a single calibration.

The Effect of Large Rotations

Only when an accurate estimate of the factor f is needed, pictures of the calibration pattern should be taken under large rotations. In all other cases these large rotations have a negative influence on the parameter estimation, due to the small depth of field. In case of big rotations, the best results were obtained by using weighed keypoints and about 10 images per subset⁹ (Figure 3.11). The radial distortion parameters however, are completely lost in the estimation.

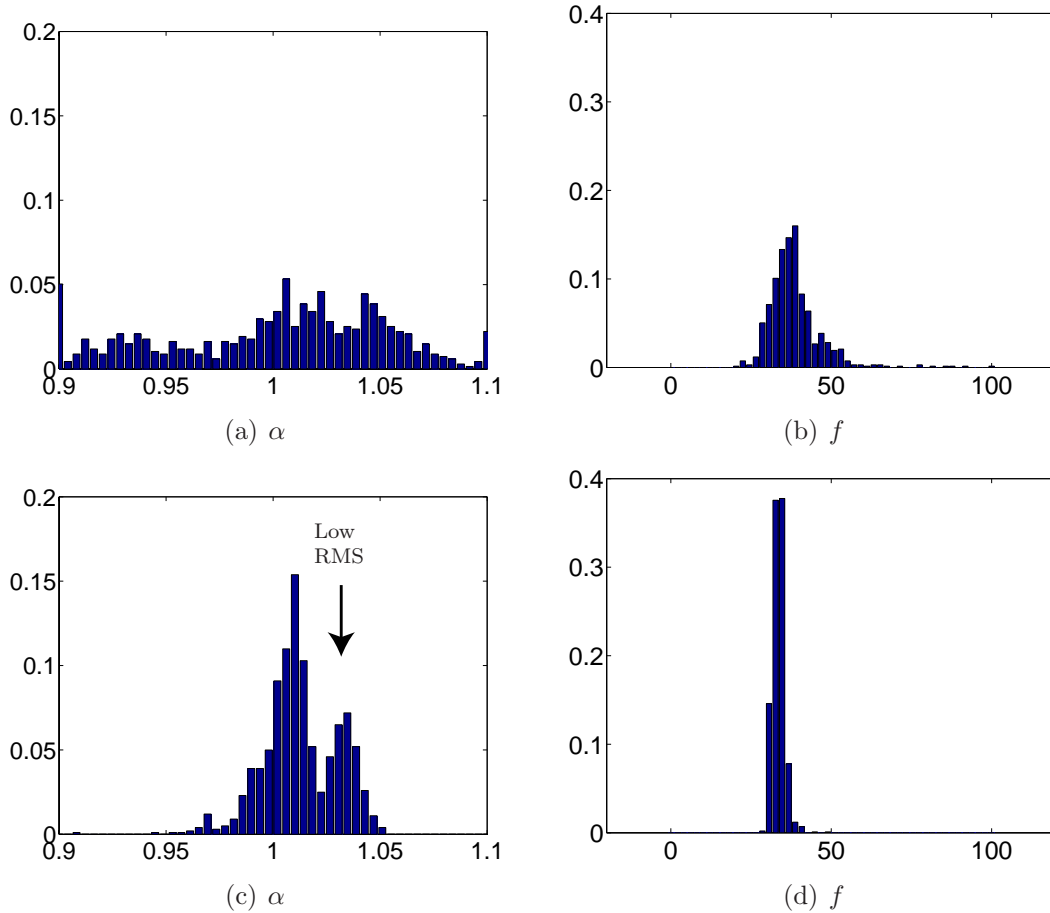


Figure 3.11: Comparison of results for images with large rotations:
 (a) and (b) with dynamic thresholding II and 4 images per subset,
 (c) and (d) with weighed keypoints and 10 images per subset.

The Effect of Large Subsets

It is remarkable that using more images per subset gives a better result for images with large rotations, because in the simulation tests (§3.3.2) this hardly made any difference. The large amount of noise that is present in the images taken using the welding head is supposed to play a role here, as the effect of noise on the parameters can be cancelled out only when enough data is present.

⁹ The size of the subset had a bigger influence than the use of weighed keypoints.

For images with small rotations, using larger subsets does provide slightly more stably estimated camera parameters (Figure 3.12(e)). However, the fact that information about the radial distortions is completely lost (Figure 3.12(f)) should be reason enough not to use subsets of many images. Instead of larger sets, weighed keypoints should be used as this also improves stability and even slightly increases the number of sets that will be found to have a low RMS value (Figure 3.12(c) and (d)).

Estimating Tangential Distortions

Although in literature tangential distortions are usually neglected, some tests have been performed with the estimation of both radial and tangential distortions. In all tests it appeared that the estimation of tangential distortions has a negative influence on the convergence for all other parameters. No clear results could be obtained, although the other parameters were not estimated as poorly as in the simulated images test (§3.3.2). It is therefore recommended not to try and fit tangential distortions, unless one is sure that they are present in the images and that they cannot be neglected.

Different Numbers of Objects in the Pattern

A few tests were performed with the pattern with 12×12 objects. Results in general looked similar to the results obtained with the 8×8 objects pattern. For a more thorough investigation, images of the 12×12 pattern should be taken under exactly the same angles and from exactly the same distances as the 8×8 pattern.

Conclusions

- The difference between using full or half-resolution images is negligible,
- Leaving out images that often occur in sets with imaginary results improves the overall results,
- Large rotations are especially beneficial for an accurate determination of the factor f (which in fact is not needed for undistortion),
- If large rotations are used, the number of images in each subset should be high (e.g. 10) otherwise smaller sets (e.g. 4) are preferable,
- If large rotations are used, generally the estimated radial distortion parameters are inaccurate,
- Estimation of tangential distortions should not be done, as it has a very bad influence on the estimation of other parameters,
- Whether a pattern with 8×8 objects or one with 12×12 objects is used does not make much of a difference.

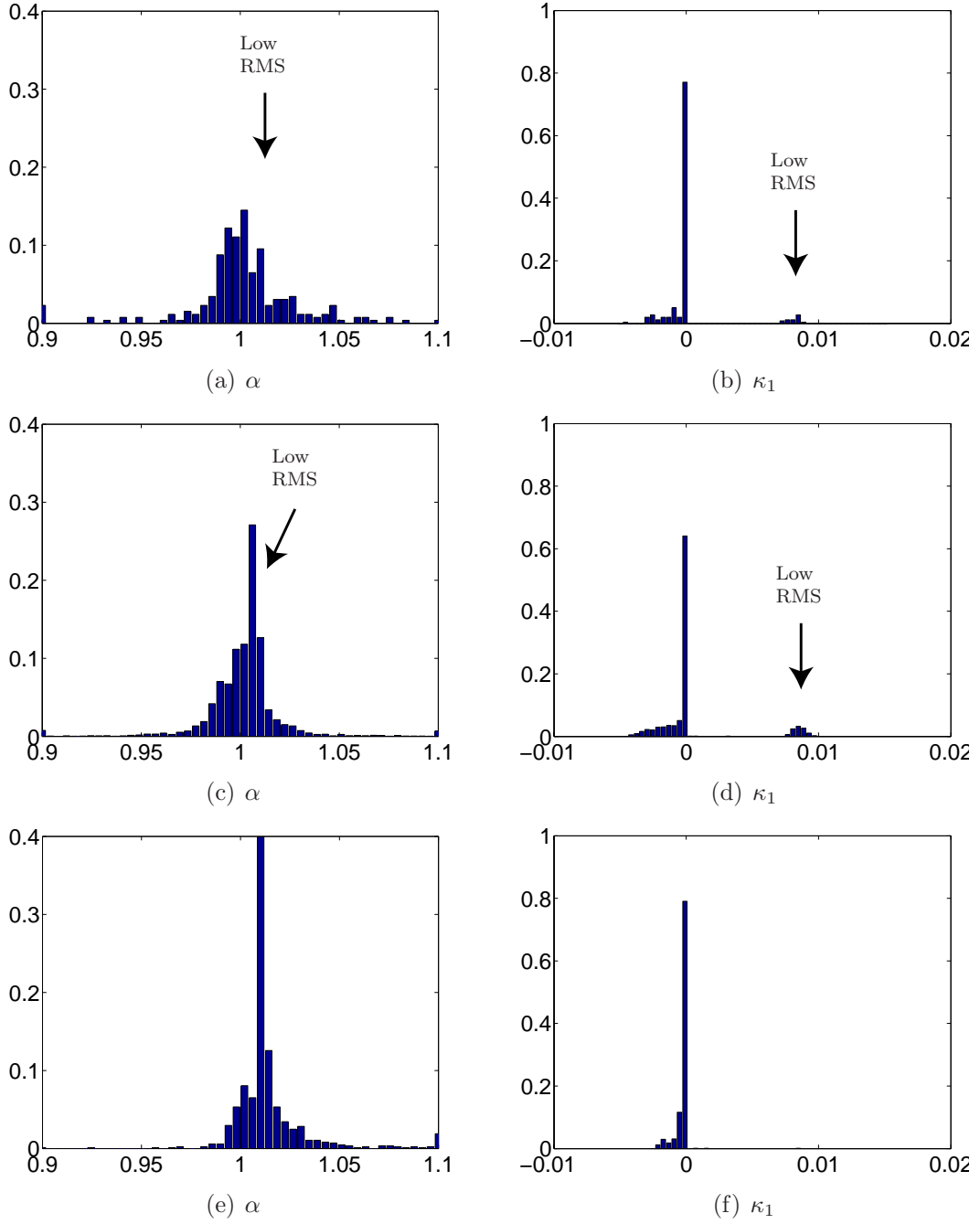


Figure 3.12: Comparison of results for images with small rotations:
 (a) and (b) with dynamic thresholding II and 4 images per subset,
 (c) and (d) with weighed keypoints and 4 images per subset,
 (e) and (f) with weighed keypoints and 10 images per subset.

3.6.1 A Final Calibration Result

After all these tests and analysis of all results, a final parameter estimation should be presented for undistortion of the camera images taken with the laser welding head. Three parameter sets were selected, based mainly on the histogram representation of the results.

For all images the laser-engraved pattern with 8×8 objects and relatively large object spacings was used. For parameter set A the pattern was rotated over small angles (up to circa 10°). For the other two sets, B and C, the pattern was rotated over slightly bigger, but still not extreme angles for some images (up to circa 25°). All keypoints were extracted using the weighed keypoint method. The calibrations resulting in A and B were based on subsets of 4 images, while 12 images per subset were used for to obtain C.

Like for the other experiments, the performance of the parameters is measured using the techniques described in §3.1.2.

The histograms for set C clearly show the sharpest peaks, as the standard deviations are much smaller than those found for A and B (see Table 3.4). Small standard deviations however, do not necessarily mean that the estimated values are correct.

This can be seen when the images of straight line projections are undistorted. When undistorted properly, two fitted line parts should be exactly parallel. The results in Table 3.4 show that the angles between the fitted parts have decreased considerably after undistortion¹⁰. Parameter set A is clearly optimal for this test.

When a set of 16 calibration pattern images is undistorted, keypoints are extracted from the undistorted images and lines are fit through the keypoints, again undistortion makes a big difference and again set A appears to have the best parameters.

Calibration	Distorted	A	B	C
Number of objects	–	8×8	8×8	8×8
Rotations	–	Small	Medium	Medium
Lowest RMS	–	0.003	0.004	0.007
Standard deviation α	–	0.029	0.041	0.005
Standard deviation κ_1	–	0.003	0.003	0.001
Standard deviation f	–	26.050	21.358	2.772
Angle between fitted parts (image 1)	2.47	0.11	0.40	0.38
Angle between fitted parts (image 2)	3.38	0.58	0.79	0.71
Keypoint fitting residual (mean)	115.96	32.48	37.71	35.56

Table 3.4: Comparison of the three candidate parameter sets, based on their performance in several tests.

The final parameters describing the camera and lens distortions are presented in Table 3.5. When the shift of pixels in an image according to these parameters is examined (Figure 3.13), it can be seen that there is virtually no place in the image

¹⁰ The position of the interruption that was made in the line moved 1–5 pixels in these tests, this is equivalent to 50–300 μm .

with negligible distortions. The maximum distortion would be over 5 mm when printed as a photograph! This is a huge difference with the distortions present in the photo camera images (Figure 3.9(a)), which would have a maximum of about 0.75 mm in a photograph.

	Mean	St.dev	St.dev/Mean
α	1.0081	0.0009	0.0009
β	0.9919	0.0009	0.0009
γ	0.0004	0.0014	3.3016
u_0	0.1119	0.1208	1.0792
v_0	0.0136	0.0427	3.1340
κ_1	0.0084	0.001	0.1149
κ_2	0.0001	0.0001	0.7550
f	22.8433	11.0555	0.484
RMS	0.0042	0.0008	0.1949

Table 3.5: The parameters of set A (values for sets with low RMS). These parameters appear to describe the camera and lens distortions best.

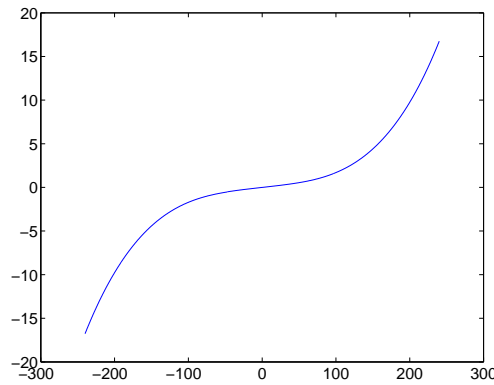


Figure 3.13: Shift of pixel data due to radial distortions. The shift in pixels is plot against the distance from the image centre of a 384×288 pixel image.

Conclusions

- Good undistortion results can be obtained,
- The set with the sharpest peaks in the histograms is not necessarily the set with the best undistortion parameters.

3.7 Guidelines for Taking Good Calibration Images

Some guidelines for taking useful calibration images can be given, however a bit of luck is still part of the deal. It is therefore recommended to take an overabundance of images and review multiple subsets. Some of the guidelines are taken from the *Halcon Application Guide*¹¹ [14].

- The calibration pattern should be clear and have high contrast
- The illumination of the pattern should be as homogeneous as possible
- The camera settings must be fixed
- The whole field of view should be covered with multiple images, i.e, the calibration pattern should be placed in all areas of the field of view at least once
- The pattern should cover the major part of the image.
- Images should be taken from different distances
- Images should be taken from different angles
- Large rotations ($>10^\circ$) should only be used if the images are sufficiently sharp
- The pattern should be as much in focus as possible
- The images should contain as little noise as possible
- The pattern should be completely inside the image.
- All objects in the images should be clearly separated from one another
- The images provided to the calibration program must show dark calibration objects on a light background

¹¹ The guidelines are listed in the part called ‘2 Application Note on 3D Machine Vision’, section 3.1.2

Chapter 4

Seam Detection

To make a good weld, it is important that the exact trajectory of the seam is known to the robot. Because the positioning of subsequent workpieces and their seam trajectories can vary, the seam must be detected again for every workpiece. Also real time correction of errors introduced by robot dynamics makes fast and accurate seam detection a necessity.

4.1 Structured Light & Triangulation

A common way to gain depth information from an image is to use structured light and the triangulation principle [20–22]. Like explained in the introduction (§1.1 and Figure 4.1(a)), laser diodes are used to project a pattern of lines onto the surface of the workpiece. The line is watched by a camera which, because of the angle between the optical axis and the laser diode, sees a deformed (broken) line where it hits an overlap joint. Although the information in the camera image is only 2D, we can solve for the 3D location of the breaking point (i.e., the seam) because we know the line exists only at the intersection of the workpiece and the laser light plane. This process, shown in Figure 4.1(b), is called triangulation.

Although one line projection is sufficient to determine the 3D position of the seam, multiple lines are required to obtain the orientation of the seam. Several setups can be used, like circles [23], two crossing lines [21] or a triangle of three lines. Laser diodes projecting a circle are relatively expensive and are not practical to use for seam inspection as well. The idea of using three instead of two lines originates in the fact that sharp angles in the seam can be detected easier this way. Because the seam is always surrounded by all lines, it will always cause at least two deformations in the image. A drawback of two or three crossing laser light lines however, is that image processing is far more difficult.

In the ‘Integrated welding head’ project initially three lines were used, for which image processing took too much time¹. Therefore the current research focusses on two alternatives: a setup using only one line projection and a setup using two crossing lines. Both setups have been implemented and tested.

¹ In the current implementation a Radon transform is used to identify the positions and orientations of the lines. This method is known to be time consuming, but few other options are available.

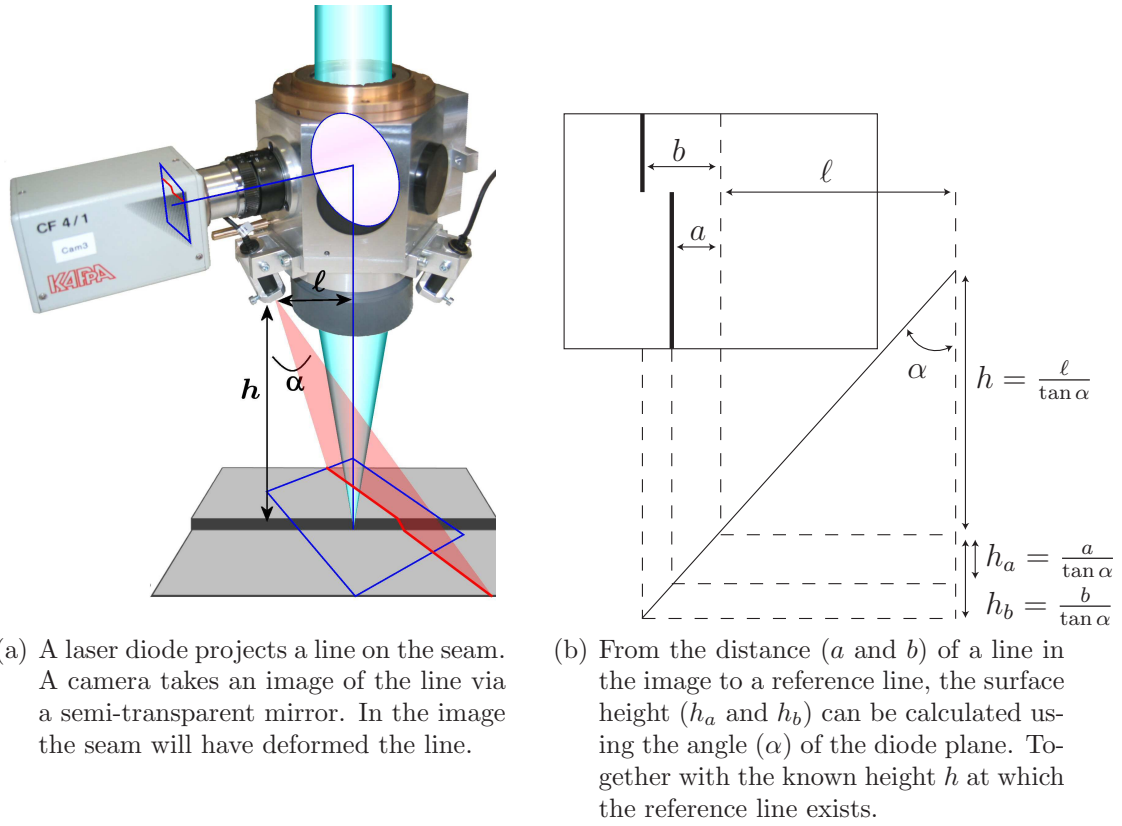


Figure 4.1: The principle of triangulation

4.1.1 Line Deformations

Figure 4.2 illustrates the effects of translations and rotations of the workpiece on the image when using structured light. It is important to grasp the idea that the position and/or orientation of the line in the image will change when the workpiece is moved relative to the welding head. The position and orientation of lines in the camera images therefore contain information about the pose of the welding head in relation to the workpiece.

The simplest movement to explain, and the main idea of triangulation, is the movement from and to the head in vertical direction (Figure 4.2(a)). When the workpiece is moved down, the intersection of the laser light plane with the workpiece will move to the left.

Figure 4.2(b) shows that the same happens when rotating the workpiece around its y-axis. The distance over which the line moves depends on the rotation angle and the distance between the original line and the rotation axis. The rotation around the y-axis and movement in z-direction have the same effect, which means that it is impossible to distinguish between them when only one line is available.

Figure 4.2(c) shows that a rotation around the x-axis results in a rotation of the line in the image. This is because one side of the workpiece (the closest in this case) goes down – so the line projection will move to the left as we saw in the case of the z movement – and the other side is going up, which makes the line

projection move to the right. The fact that this effect is easy to distinguish from the other effects, means that it could be beneficial to use a second laser diode, which is mounted to the head under a 90° angle to the first one. This way the rotation around y-axis can be determined using the second diode.

A movement in x-direction (along the seam) will not show up in the image at all for straight seams (Figure 4.2(e)). A movement of the workpiece in y-direction on the other hand, causes the seam point to be at another position along the laser line (Figure 4.2(d)). A rotation around the z-axis may change the detected location of the seam along the laser line as well (Figure 4.2(f)).

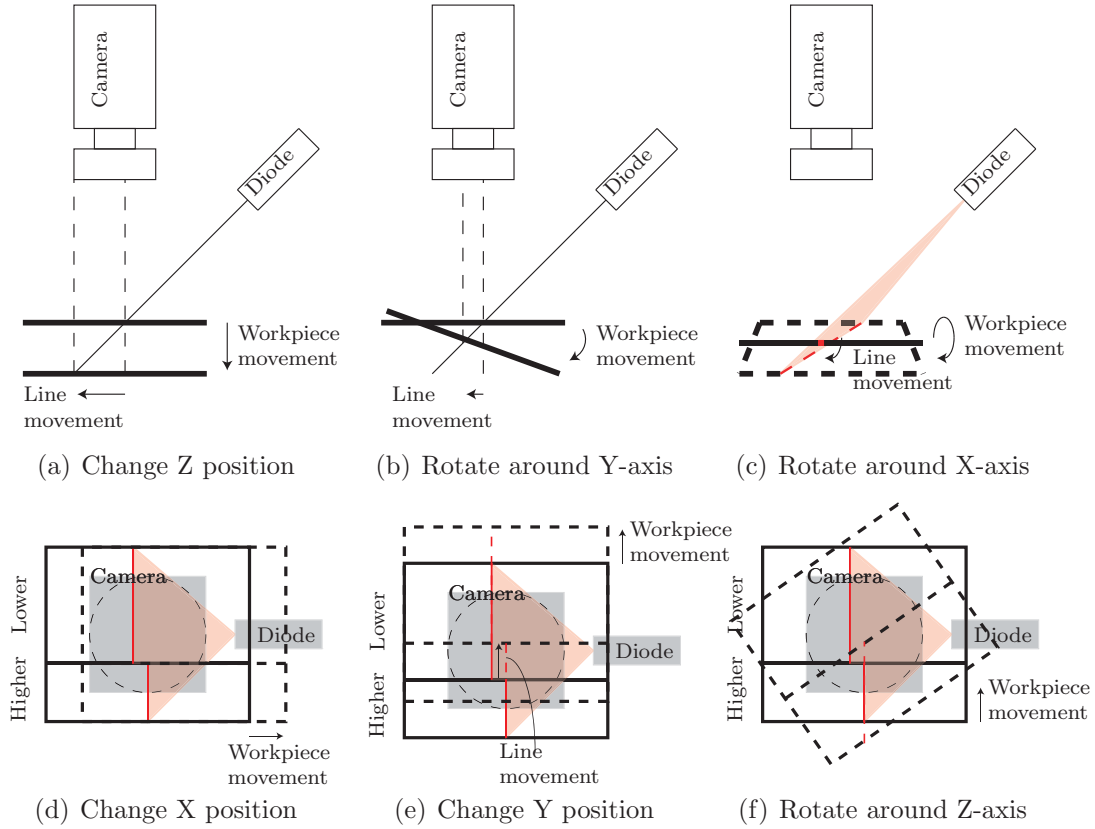


Figure 4.2: Influence of translations and rotations of the workpiece on the line projection in the image.

4.1.2 Calculating the Seam Position

The measurements in the image are done with respect to the reference line and the *Sensor Tool Centre Point* (Sensor TCP). This is the point on the reference line closest to the *Laser Tool Centre Point*, which in its turn is the position of the welding laser spot in the image. Figure 4.3(a) illustrates these relations.

The position of the seam in the image is defined as the point right in between the two endpoints of the line parts. The endpoints are the positions where the last white pixel straight under the fitted line is. Which 2 of the 4 endpoints should be chosen is determined using their mutual distances. The set with the

endpoints closest to each other and belonging to two different lines is used to determine the seam location.

Once the location of the seam in the image is known, its real position can be calculated. The Δx and Δy positions to the Sensor TCP are calculated directly from Δu and Δv in the image, using a pixels-to-millimetres scaling factor as calibrated for according to §4.4.1. From the distance² δ to the reference line, the z position of the seam can be calculated via triangulation.

4.1.3 Calculating the Workpiece Orientation (Single Line)

In the single-line case there is little information to base rotation estimations upon. Seam tracking software may be able to estimate rotations from subsequent position data [24, 25].

However, even with one line a rotation correction can be calculated from the image, as the rotation α of the new line with respect to the reference line (see Figure 4.3(b)) indicates a rotation around the x and/or y -axis. More precisely, it shows a rotation around the axis perpendicular to the reference line, in the working plane and through the point where the new line crosses the reference line. The real rotation angle θ around this axis can be calculated as following:

$$\begin{aligned} \frac{d}{\ell} &= \tan(\alpha) \\ \frac{d}{\Delta z} &= \tan(\phi) \\ \theta = \frac{\Delta z}{\ell} &= \arctan\left(\frac{\tan \alpha}{\tan \phi}\right), \end{aligned} \tag{4.1}$$

with ϕ the angle between the laser light plane and the vertical. Using quaternions [26, 27] or Rodrigues' formula (§B.3) the axis and angle can be converted to a rotation matrix, from which rotations around the x , y and z axes can be extracted³. As the full rotation matrix is built from

$$\begin{aligned} R_x(\alpha) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \\ R_y(\beta) &= \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \\ R_z(\gamma) &= \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{4.2}$$

² See §B.5 for the calculation method.

³ The reason that the rotation transformation can be written in xyz -coordinates this easy, is because the robot and software first deal with translations and then with rotations. This way the rotation axis will always pass through the Sensor TCP, because the height difference is already compensated for.

the arcsine of elements $R_{2,3}$, $R_{3,1}$ and $R_{1,2}$ will result in the rotations around the x , y and z axis respectively. These rotations can be sent to the seam tracking program.

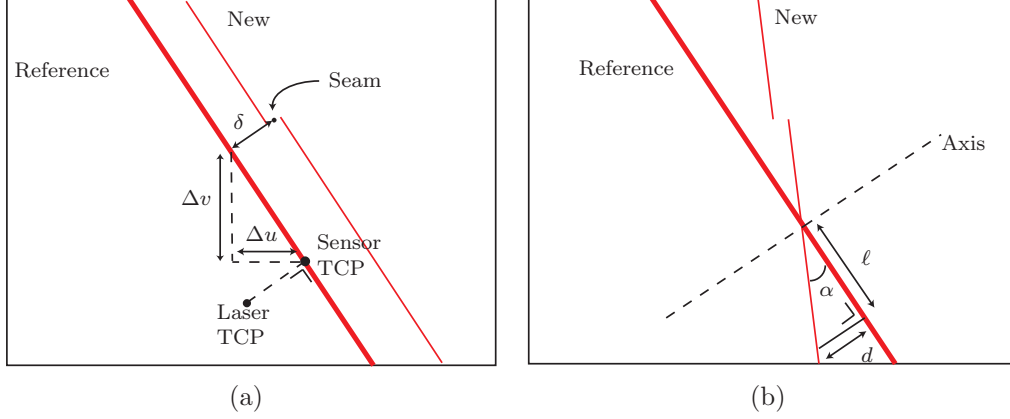


Figure 4.3: A rotation of the workpiece can cause the laser line to rotate in the image.

4.1.4 Calculating the Workpiece Orientation (Crossing Lines)

When two lines are available, the orientation of the workpiece can be fully determined. Of course the line parts used for this calculation should be in the same plane. Therefore the two line parts forming the cross are used for this purpose, rather than two arbitrary line parts.

Two points on each of the 2 line parts are taken and their 3D location is determined just like the seam position was determined in §4.1.2. Using these points, a direction vector can be calculated for each line part. These vectors, \mathbf{v}_1 and \mathbf{v}_2 , will exist at the workpiece plane. Using the outer product a third vector can be calculated:

$$\mathbf{n} = \begin{bmatrix} \mathbf{n}_x \\ \mathbf{n}_y \\ \mathbf{n}_z \end{bmatrix} = \frac{\mathbf{v}_1 \times \mathbf{v}_2}{\|\mathbf{v}_1 \times \mathbf{v}_2\|}. \quad (4.3)$$

This vector \mathbf{n} is perpendicular to the surface, pointing either upward or downward. From this vector the rotations between the head and the workpiece around the x and y -axis can be determined. It is important to keep in mind that the order of successive rotations influences the end-result. When \mathbf{n} is rotated with respect to the tool z -axis, the head can be placed perpendicular to the workpiece using a rotation around the y -axis of $\alpha = -\arctan(\mathbf{n}_x/\mathbf{n}_z)$. The rotation around the x -axis should then be calculated as $\beta = -\arctan(\mathbf{n}'_y/\mathbf{n}'_z)$, with \mathbf{n}' the vector \mathbf{n} rotated around the y -axis over an angle α .

Alternatively, the two vectors \mathbf{v}_1 and \mathbf{v}_2 could be used to span a plane. Then the dihedral angle between this plane and the plane spanned by the reference lines (typically the plane $z = 0$) can be calculated. Using the intersection line

of these planes and the dihedral angle, rotations around the x , y and z axes can be calculated like described in §4.1.3. This method was finally implemented, because a part of the code needed was already present.

4.2 Image Processing

Regardless of the amount of lines projected or their shapes, some basic image processing must be done. The first step in image processing will be to undistort the camera image using the strategy discussed in §2.5. Then a threshold can be applied to separate the lines from the background. Determining a proper threshold here is less difficult than in the case of the calibration pattern, because the laser diodes generally provide a clearly distinguishable line projection.

4.2.1 Seam Detection with a Single Line

Once the image is thresholded, the line can be detected. It is important to keep in mind that the line may be split into two parts by a seam. Several strategies for the detection of lines in the image are possible:

1. fitting a line through all white pixels
2. labelling all objects, finding the objects larger than a certain number of pixels and fitting lines through all pixels of these objects
3. performing a Radon transform

Option 1 is the simplest, but it would fit only one line. This way only a gap joint – where the two line parts are collinear – could be detected. Another problem is that this method is sensitive to noise (white pixels in the background).

Option 2 is more sophisticated, as it is able to detect two lines if a seam is present and it ignores noise pixels. A drawback is that labelling the image first takes a little more time. Problems may occur here when the threshold value is set too high and the line is split in multiple (too) small parts.

Option 3 makes use of a sophisticated mathematical technique to determine image intensity in several radial directions at several positions. The highest intensities in the radon map will correspond to the slope and position of the lines. This option is quite robust, but takes a lot of computation time. This makes it almost impossible to use in real-time.

Considering this, the decision has been made to use option 1 when it is known that only one line is present and the image is of good quality. This is for instance the case when a sensor calibration is done to determine the reference lines and projection angles. In the other cases option 2 will be used.

4.2.2 Seam Detection with Crossing Lines

Of the options mentioned for the single line case, the Radon transform is the only one that also works without any changes for multiple, crossing lines. However

the fact that it is too slow for real-time applications makes it necessary to find other ways.

For 2 crossing lines, after undistortion and thresholding (Figure 4.4(a)), the following procedure is proposed:

1. all objects in the image are labelled and the large ones are filter out (Figure 4.4(b))
2. every object is scanned row by row and the number of white parts in each row is counted, if the pattern 2–1–2 occurs the object will be a cross (Figure 4.4(c))
3. all pixels in the crossing region are made black, this way the 1 cross object will be split to 4 lines
4. 1st order polynomials (straight lines) are fit through these 4 lines and are matched 2 by 2 on their parameters, to find out which belong together in one line. Parts belonging together are given the same label (Figure 4.4(d))

Now 2, 3 or 4 objects should exist. 2 in case there is no seam, 3 in case the seam breaks one of the lines and 4 if the seam goes through both lines.

5. 1st order polynomials are fit through all objects (Figure 4.4(e))
6. if 3 or 4 objects exist, the polynomials resulting from the same line projection are matched based on their parameters
7. for further processing (such as calculating the seam position) the sets determined here can be used as if two single lines were detected.

Another possibility would be to use laser diodes with different colours, as colours can be easily separated in image processing. This however would be expensive and also an expensive colour video camera would be needed. Switching lines on and off is not considered an option because the movement in between both images calls for synchronisation. The hardware to synchronise the diodes with the camera was not available within our group at the time.

4.3 Robot Coordinate Systems

In §2.1.2 a ‘world’/model coordinate system was used. This coordinate system however, is used to relate the observed scene to the captured image and is thus fixed to the welding head. When talking in terms of robot manipulation, this would be called the *tool frame*. The tool frame moves with the tool and is located at a convenient place for describing tool movements, called the tool centre point or TCP (see Figure 4.5). The TCP is typically located at the focussing position of the welding laser.

Other frames defined for robot manipulation are the *world frame* and the *flange frame*. The world frame is fixed to the robot pedestal and thus to the rest of the world. It does not move when the robot moves. The flange frame is located at the tip of the robot, where the tool is connected. The position and orientation of the flange frame are calculated by the robot software.

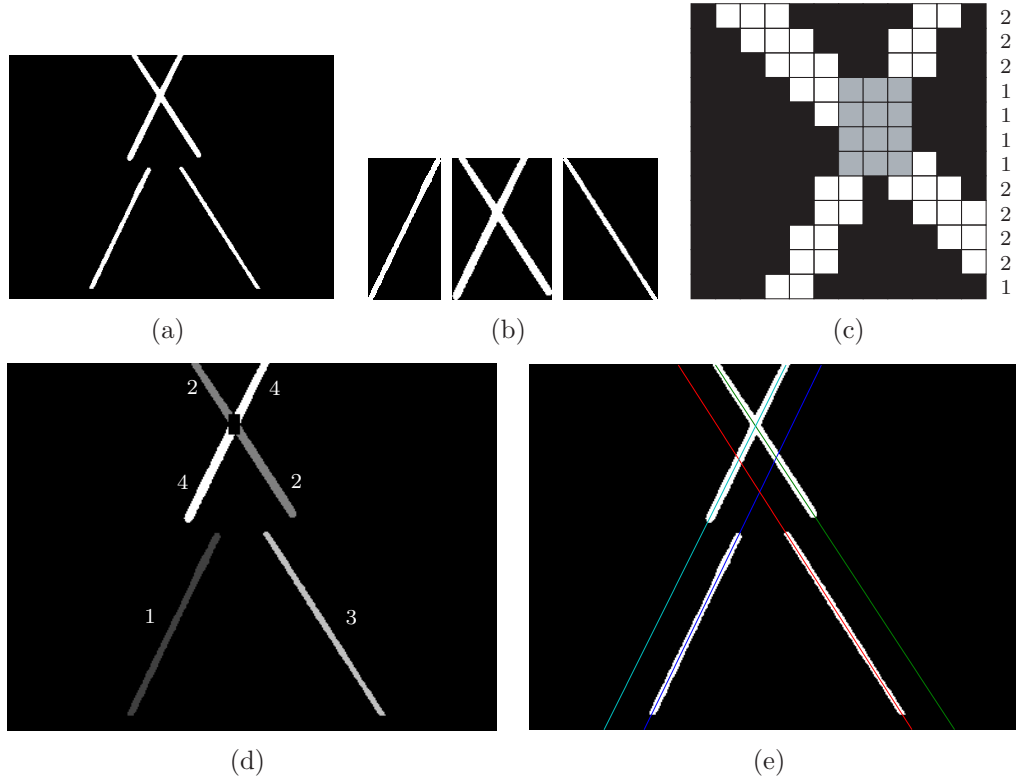


Figure 4.4: Fitting crossing lines:

- (a) The undistorted image with the crossing lines and breaks from the seam.
- (b) The objects extracted from the image
- (c) By counting the number of white parts per row, a cross can be identified. The area marked gray should be made black to cut the cross object into 4 lines.
- (d) In the new image the cross is split into lines, parts belonging to the same line are given the same object number.
- (e) Line fits are computed for the remaining 4 objects.

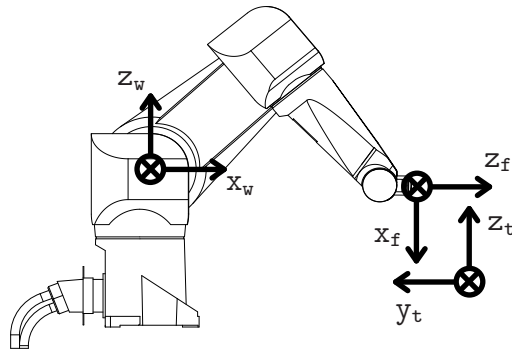


Figure 4.5: Definition of the world (w), flange (f) and tool (t) coordinates.

4.4 Real-World Coordinate Calibration

The previous sections mainly dealt with the principles of triangulation and seam detection. To get meaningful values for the seam location detected in the image, all image coordinates must be translated to real-world coordinates. This translation contains a scaling from pixels to millimetres, aligns the image axes with the movement axes and determines the focal point of the welding laser (tool centre point or TCP)⁴. In the last calibration step the projection angles of the laser diodes are determined and for each diode a reference line is stored. All measurements in the image will be related to these reference lines and the height calculations (triangulation) will make use of the projection angles.

For all these calibration steps the head must be mounted to the robot and positioned perpendicular above a flat surface. An initial, approximate tool definition is obtained using a ruler. After each calibration step the tool-definition in the program is updated to match the newly calculated parameters. All images taken are considered to be undistorted automatically.

Getting the Head Perpendicular to a Plane

It should be possible to calculate the orientation of a calibration pattern as used in Chapter 2 from an image once the camera and lens parameters are known. One possibility would be to extract the keypoints from the undistorted image, compensate for normalisation and scaling factors and to find a homography. The camera extrinsic parameters (rotation and translation) should be easy to determine then. Another possibility is to use epipolar geometry[7] to recover the rotation matrix and translation vector.

Both strategies have been tried, but did not provide accurate results. This is probably due to poorly estimated parameters (especially the scaling factor f). The head was produced with reasonable accuracy and the robot and work-table were aligned very precisely. Therefore the head is considered to be perpendicular to the table if the robot flange is positioned with its z-axis horizontally and its x-axis vertically, downward (see also Figure 4.5).

4.4.1 Scaling Factor and Rotation

The first calibration step defines a pixel-to-millimetre scaling factor and a rotation around the vertical. It is important to do this first, because the pixel-to-millimetre scaling is needed in the next calibration steps.

A pinhole with a bright spot underneath was used to provide a clear, easy identifiable point (marker), fixed in space⁵. The camera must be focussed on the marker, which means that the camera lens must focus at infinity, as the laser focus lens focusses a bundle of perpendicular rays onto the welding spot. The

⁴ A more thorough and probably more accurate method was developed by J. van Tienhoven[28], however, his work is classified and therefore cannot be used in the current project.

⁵ Any other small object, well identifiable in the camera image (i.e. remaining the only white object after thresholding) can be used as well.

head is moved over the marker and images are taken in a raster pattern⁶. Several images are taken at every position in the raster and the detected coordinates of the marker centroid are averaged.

The scaling factor is calculated as the distances between neighbouring points, divided by the accompanying robot movements. The average scaling factors u/x and v/y are initially calculated separately, but finally their average is used for calculations. The difference between the scaling in x and y direction should be minimal, due to the fact that the camera and lens calibration corrected for non-squareness of pixels (by the α and β factors).

To determine the rotation of the head/camera around the optical axis, lines are fit through the raster of detected centroids both horizontally and vertically. The vertical lines are fit with their u and v coordinates exchanged for higher accuracy. The rotation θ is calculated from all⁷ n lines $v = au + b$ by $\theta = \frac{1}{n} \sum_{i=1}^n \tan(a_i)$.

Note:

The scaling factor will not be constant in height. Although it is not within the scope of the current research, this problem will be discussed briefly.

The view angle of the camera causes the field of view to be larger further away from the camera. The scaling factor will therefore be determined at several heights in the focal range of the camera. The welding laser focal point, pilot laser focal point and calibration point are considered to be all within this range.

For triangulation a look-up table will be constructed with scaling factors for every pixel in the image, depending on its distance to a reference line (and thus its height). For every reference line such a look-up table would be needed. In the determination of the distance between the reference line and the point, as described in §4.1.2 and §B.5, the mean of the reference scaling and the scaling at the point location will be used. This implies that the scaling is assumed linear with the height.

4.4.2 X-Y Position of the TCP

The tool centre point (TCP) is the focal point for the welding laser. For calibration a pilot laser is used, which has a slightly different focus position, but the difference is known and can thus be compensated for.

To determine the TCP the pilot laser projects a spot on a flat surface, perpendicular to the optical axis. The position of the spot in the image is then related to the position of the initially estimated z -axis in the image. To determine the position of the initially estimated z -axis, the same marker as in §4.4.1 is used: the pinhole with the bright spot underneath.

⁶ This may seem like making a pattern for camera and lens calibration (§2.2) and actually this raster data could be used instead of one image in the camera and lens calibration procedure.

⁷ Also here the rotations according to rows and columns are calculated separately initially, if the difference is too big a warning is issued.

The head is rotated around the initially estimated z-axis. The position of the marker in the image is captured. Several rotations and captures are carried out. If the initially estimated axis does not coincide with the spot, the spot will describe a circular motion in the image (see Figure 4.6). The centre of this circle defines the u and v location of the initially estimated z-axis in the image, which can be used to calculate the translation to the real TCP.

Two methods for determination of the circle centre from a set of points are discussed in §B.6. One has a geometrical approach and uses pairs of points with a known mutual rotation angle, the other is a linear least squares fitting procedure. Comparative test results are presented in §5.5.2.

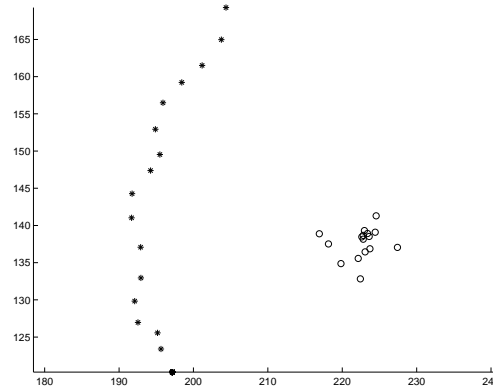


Figure 4.6: When rotating around an estimated axis, the pilot laser spot – indicated with * – will describe a circular motion. From these points estimates of the axis position can be made – indicated with o.

4.4.3 Z Position of the TCP

The vertical location of the TCP (i.e., the focal point for the welding laser) is determined using the pinhole with the bright spot underneath again. When rotating around the y-axis, the position of the marker should not change. A movement of the marker location in x-direction indicates that the initially estimated y-axis is too high or too low, as can be seen in Figure 4.7. The correction Δz to be made can be calculated from the rotation angle α and the movement of the marker δ which is obtained from the movement in pixels using the millimetre-to-pixel scaling factor.

Once the vertical position of the marker is known with respect to the robot flange, the vertical position of the TCP can be found by making a series of spot welds at increasing or decreasing heights. The vertical translation between the calibration height and the height where the smallest spot was produced is considered to be the TCP.

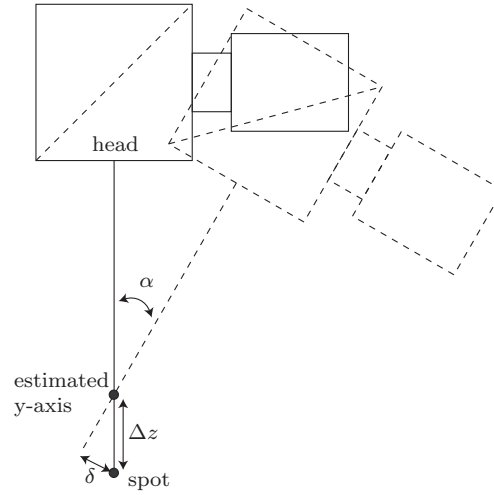


Figure 4.7: If the estimated y-axis is not at the marker, the marker will move in x-direction in the image when rotating around the y-axis.

4.4.4 Laser Diode Calibration

The calibration of the laser diodes is relatively easy. The head is positioned perpendicular to a flat surface and moved up and down with one laser diode switched on at a time. At several heights an image is captured and a line is fit through all white pixels. The height differences and the distance between the lines in the image are related to each other and the projection angle can be found. The difference in scaling factors (see the note in §4.4.1) should be taken into account here for a more accurate estimation.

The parameters of the line which is captured at the focus-height (when the TCP is at the surface) are saved as a reference. All z-coordinate estimates of the lines and points detected will be related to this line.

Chapter 5

Seam Tracking Experiments

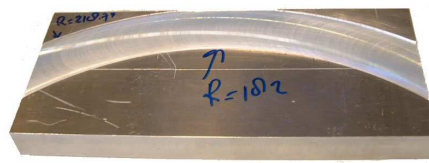
Calibration, seam detection and seam tracking experiments have been performed. The seam detection tests were performed to obtain insight in the measuring accuracy and the difference that undistortion of the images makes. The seam tracking experiments were used as a ‘proof of concept’. Although some specific tests were done, calibration of the welding head was mainly a necessity, because the seam tracking software needs to know according to what reference point the data is provided. Good seam tracking results therefore prove a (sufficiently) accurate tool calibration.

5.1 Test Objects

Seam tracking tests have been performed using the three different test objects shown in Figure 5.1. The first one is a straight seam, the second seam describes a circular path and the third seam is on a double curved surface. All seams are 1 mm thick and of overlap type.



(a) Straight seam test object.



(b) Curved seam test object.



(c) Double curved seam test object.

Figure 5.1: Test objects used for the seam tracking experiments.

5.2 Single Line Seam Detection Experiments

One problem that will always exist for triangulation using structured light is the fact that shadows occur. This may cause the detected seam to be slightly shifted from the real seam position. The direction from which the structured light is projected has some influence on this shift, which should be kept in mind when analysing seam detection results. None of the known commercially available products compensates for this, which indicates that this is probably not a big issue in practical applications.

A benefit of the shadow is that the line projection is broken clearer, which makes it easier to detect. The main problems during detection were caused by poor image quality. Due to the fact that real-time image processing should be fast, a static (manually set) thresholding value is used. This however makes the system very sensitive to different illumination conditions, reflections etcetera. It is therefore proposed to develop a more robust, intelligent thresholding algorithm.

Accuracy Measurements for a Straight Seam

For accuracy measurements the welding head was attached to a XY-manipulator that uses linear motors and moves with high precision ($<10\ \mu\text{m}$). The head was moved in a linear motion over the straight and curved seam test objects and the seam position detected by the software was saved every 2.00 mm. This was done for several seam orientations, always once with the image undistortion turned on and once with image undistortion off. For the sake of completeness it should be mentioned that the estimated z-position may be scaled wrongly, because the diode projection angles were not calibrated. Trends in z-errors however remain valid.

Three dimensional lines were fit through the measurement data. The deviations of the measurements from the fitted line are shown in Figure 5.2. The measurements were performed with the straight seam test object under a rotation, but still perpendicular to the head, and with an inclination respectively.

When comparing Figures 5.2(a) and (b), it can be seen that accuracies without undistortion are worse, especially for the x-positions higher than 70 mm. This is what could be expected, because for the higher x-values, the seam point came closer to the edge of the image (where distortions are large). For the Figures 5.2(c) and (d) the improvement is mainly present at lower x-positions, because the laser line projection is close to the image edge at that moment.

In both cases the over all fitting accuracy of the measurements from undistorted images is better. Because the undistorted images are expected to give the ‘real’ seam position, and strengthened in this belief by the fitting accuracies, the errors measurements from the distorted images are related to the line fitted through the undistorted data. These deviations are shown in Figure 5.3. It is clear that the errors, ranging from 200–500 μm cannot be neglected for laser welding applications.

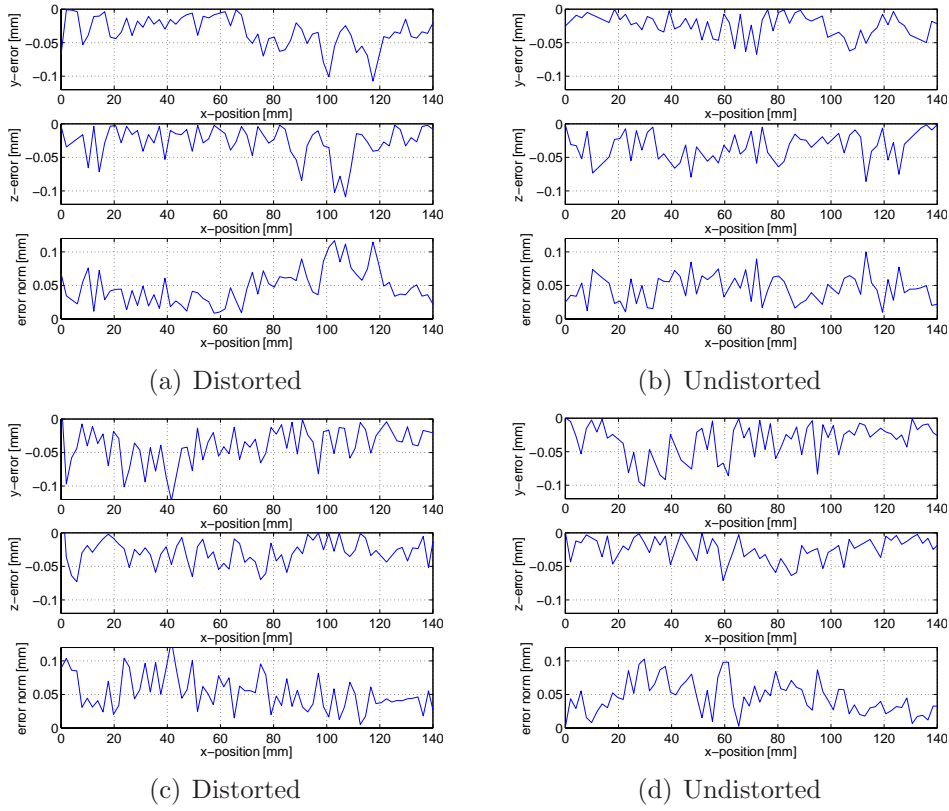


Figure 5.2: Accuracies from measurements on the straight seam test object:

- (a) The seam was rotated around the z-axis; based on distorted images,
- (b) The seam was rotated around the z-axis; based on undistorted images,
- (c) The seam was tilted over the y-axis; based on distorted images,
- (d) The seam was tilted over the y-axis; based on undistorted images,

Accuracy Measurements for a Curved Seam

As Figure 5.4 shows, undistortion results in 2–3 times higher accuracies for the circular seam. The error in y-direction is a little smaller, but the accuracy in z-direction has improved drastically. Also here it should be stressed that the estimated z-position may be scaled wrongly, because the diode projection angles were not calibrated. Trends in z-errors however remain valid.

5.3 Crossing Lines Seam Detection Experiments

A problem with the current implementation of the crossing-lines detection, is that the seam should never show up at the crossing point, as the software cannot cope with this situation. This problem is something to investigate further, as for most applications it will be sufficient to place the crossing at some place in the image where the seam is unlikely to get during tracking. Another possibility would be to detect this situation and neglect all pixels within a certain radius from the seam, such that just 4 separated objects are found.

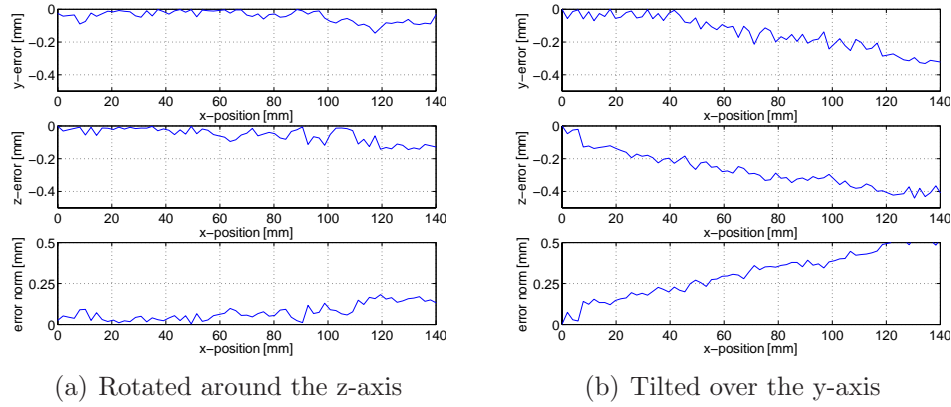


Figure 5.3: Distance between the points measured from distorted images, with respect to a line fit through the undistorted measurements.

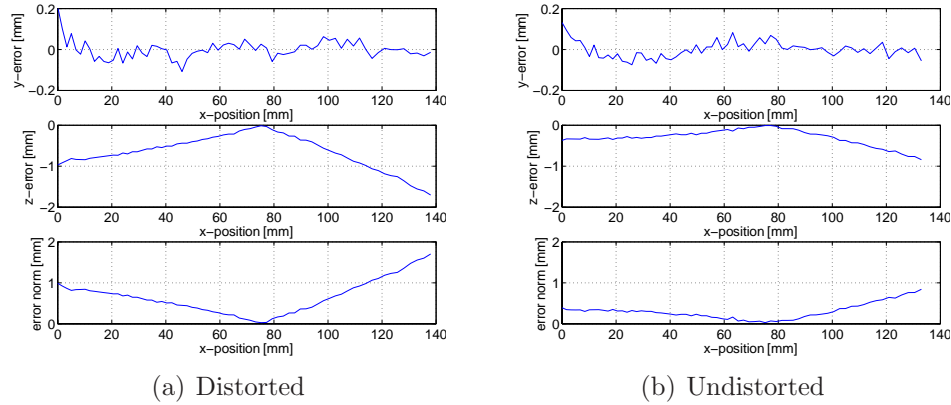


Figure 5.4: Accuracies from measurements on the curved seam test object

Accuracy Measurements

The accuracies of estimated seam position will be the same as for the single line, therefore only a test for the accuracy of orientation estimates has been performed.

The estimation of workpiece rotations was tested by rotating a small test seam over small angles. The results have an accuracy within $\pm 0.5^\circ$. This is mainly due to noise in the image and different thresholding of pixels because of slightly changing intensities. This would mean that higher accuracies could be obtained by using a higher resolution camera and a more sophisticated thresholding algorithm.

If the seam is positioned close to the crossing of the line projections, accuracy is lost fast: errors of 3° are quite common when the seam is less than circa 10 pixels removed from the crossing.

5.4 Seam Tracking Experiments

Seam tracking experiments for both the single line sensor and the crossing lines sensor have been performed with all three test objects mentioned in §5.1. For the single line case the position and orientation data were calculated using the methods described in §4.1.2 and §4.1.3. The data were sent to the seam tracking program 24-Laser (based on [24]), which was set to use information from subsequent measurements to improve the rotation estimates. For the crossing-lines, all translations and rotations could be estimated accurately and rotation estimation by the seam tracking program was turned off.

All seam trajectories provided could be tracked properly. The main problem occurring during tracking was a poor thresholding of the images, resulting in bad estimates. A manual correction of the thresholding value was therefore needed from time to time. When problem is dealt with, it is proven that both concepts (one or two line projections) work.

5.5 Real-World Coordinate Calibration Experiments

Some small tests were performed to verify the calibration procedure. The most important conclusions can be drawn from the ‘Scaling Factor and Rotation’ tests and the comparison of circle fitting algorithms. These two items will therefore be discussed in this section.

The fact that seam tracking was successful (§5.4), shows that the calibration was done properly. However, according to the test some improvements could still be made.

5.5.1 Scaling Factor and Rotation

Regarding the scaling factor, a simple test has been done. First the welding head was placed 1 mm higher than the focus position for the marker and a raster of captures was made to calculate the scaling factor. Then the welding head was placed in focus and the scaling factor was determined and finally the it was determined at 1 mm below the focus position. The resulting values for the u/x and v/y factors are listed in Table 5.1. These differences stress the need for a triangulation algorithm that takes the height-dependent scaling factor into account, like suggested in the note in §4.4.1.

	$z = 1$	$z = 0$	$z = -1$
u/x	15.7454	15.6705	15.5836
v/y	15.8467	15.7858	15.7043

Table 5.1: Pixel-to-millimetre scaling factors at different heights.

Another test was performed to see the effect of image undistortion. The scaling factors were calculated once without and once with image undistortion.

From Table 5.2 it is clear that image undistortion compensates for the non-squareness of image pixels.

	Distorted	Undistorted
u/x	16.3321	15.7279
v/y	15.9394	15.8136

Table 5.2: Pixel-to-millimetre scaling factors calculated with and without image undistortion.

5.5.2 X-Y Position of the TCP

Both strategies for finding the rotational axis discussed in §4.4.2 and §B.6 have been tested in a simulation. In order to minimise the effect of robot dynamics, it is desirable keep rotations small, which is expected to have a negative influence on the fitting accuracy. Simulations have therefore been carried out with varying ranges, besides varying numbers of measuring points and varying noise levels on the detected coordinates.

The simulation data were generated at evenly spaced (angular) intervals, for a circle with radius 1 and its centre in $[0,0]$. Normally distributed noise was added to the x and y positions of the generated data points, the mean value (E) was chosen as a percentage of the radius. For each case 1000 simulations were performed, the average distances of the estimated centroids to the real centroids are listed in Table 5.3. The performance of the geometrical approach relative to the fitting algorithm is listed in Table 5.4, where the value .5 means that the error of the geometrical approach is half that of the fitting algorithm and -1 means that the error of the fitting algorithm is half that of the geometrical approach.

LSQ fitting				geometrical approach			
E=2%	50 pts	10 pts	5 pts	E=2%	50 pts	10 pts	5 pts
$0-2\pi$	0.0147	0.0173	0.0204	$0-2\pi$	0.0153	0.0164	0.0184
$0-\pi$	0.0152	0.0220	0.0310	$0-\pi$	0.0181	0.0198	0.0224
$0-\pi/2$	0.0340	0.0586	0.0915	$0-\pi/2$	0.0269	0.0293	0.0333
E=10%	50 pts	10 pts	5 pts	E=10%	50 pts	10 pts	5 pts
$0-2\pi$	0.0744	0.0876	0.1020	$0-2\pi$	0.0760	0.0836	0.0926
$0-\pi$	0.0643	0.1044	0.1494	$0-\pi$	0.0887	0.0958	0.1078
$0-\pi/2$	0.4258	0.3951	0.4558	$0-\pi/2$	0.1360	0.1404	0.1613
E=20%	50 pts	10 pts	5 pts	E=20%	50 pts	10 pts	5 pts
$0-2\pi$	0.1468	0.1761	0.2095	$0-2\pi$	0.1520	0.1667	0.1873
$0-\pi$	0.1424	0.2021	0.2897	$0-\pi$	0.1778	0.1915	0.2244
$0-\pi/2$	0.6073	0.6165	0.6704	$0-\pi/2$	0.2672	0.2871	0.3308

Table 5.3: Accuracy of the estimated circle centre.

Percentage difference			
E=2%	50 pts	10 pts	5 pts
0-2 π	-3.40	5.20	10.29
0- π	-19.08	10.00	27.74
0- $\pi/2$	20.88	50.00	63.61
E=10%	50 pts	10 pts	5 pts
0-2 π	-2.15	4.57	9.22
0- π	-37.95	8.33	27.84
0- $\pi/2$	68.06	64.46	64.61
E=20%	50 pts	10 pts	5 pts
0-2 π	-3.54	5.34	10.60
0- π	-24.86	5.24	22.54
0- $\pi/2$	56.00	53.43	50.66

Table 5.4: Performance improvement of the geometrical approach with respect to the fitting algorithm (positive values mean that the geometrical approach is better)

It appears that the number of points used does not have much influence on the accuracy of either method. The size of the arc used for measurements shows huge influence for the fitting algorithm, and some influence for the geometrical approach. The accuracy of the measurements (the mean noise level on the data) shows quite some influence, but it is more or less the same for both strategies.

When comparing the accuracies of both algorithms, it appears that the currently implemented fitting algorithm only performs better when there are many data points which are spread over a large part of the circle (the negative values in Table 5.4). This means that in practice, for calibration purposes at least¹, the geometrical approach is advantageous.

¹ Note that the geometrical approach needs the rotation angle between two measurements as an input, which makes the method a lot less generally applicable.

Chapter 6

Conclusions & Recommendations

The assignment was to accurately detect the position and orientation of a seam from an image of the workpiece taken via the laser welding optics. This detection should be performed by using structured light: either one line or two crossing lines are projected over the joint. This assignment consists of two big challenges.

6.1 Camera and Lens Calibration

The first challenge is to get an accurate estimate from an image that is heavily distorted by the laser welding optics. A camera and lens calibration procedure and an image undistortion function have been developed to restore proportionality in the image.

The proposed calibration procedure has proven to be flexible and robust for many qualities of input data. The main idea is to take many pictures of a calibration pattern and divide them in subsets. A calibration is performed for every sub set and the results with the lowest fitting residues are selected. Especially for images of poor quality evaluating multiple sets highly increases the probability of finding a good parameter estimation.

Something else that makes parameter estimation upon poor quality images better is a weighed keypoint determination. Due to the fact that a wide area around an object is investigated and can contribute to the keypoint position, blurry edges and large comas are less likely to be neglected in the one case while incorporated in the other.

From experiments it appears that it is difficult to estimate the parameter f , which contains the focal length. This is due to the fact that the focal length and the object distance have a close relation. f can be estimated quite satisfactory when images of the calibration pattern are taken under large angles and when large subsets are used for calibration. In this case other parameters are estimated poorly. Therefore it is a relief that the parameter f is not needed for an accurate undistortion.

Several techniques have been developed analyse the performance of a set of undistortion parameters. Histograms of the calibration results for all subsets give a first impression. The residual of a line fit through keypoints detected in undistorted calibration images will provide a more quantitative measure for

the quality of the undistortion parameters. Finding the difference of angles in an undistorted image with an artificially interrupted straight line projection is another quantitative measure, which is a little closer to the practical application.

After all a calibration of the optical system in the integrated laser welding head appears to be difficult but possible. A set of guidelines for taking proper calibration images should help to obtain good calibration results.

6.2 Seam Detection and Real-World Coordinate Calibration

The second challenge is to detect the position and orientation of a seam from an image. To do this, the information contained in a 2D image must somehow reveal 3D information about the workpiece. Triangulation methods with structured light has been used obtain such information.

The simplest method implemented uses only one line projection of laser light for triangulation. This has the benefit of easy image processing, but can only reveal one rotation parameter for the workpiece. In this case the rotations of the workpiece should be estimated from subsequent images.

A system using two line projections with different angles can offer all three rotation parameters. Such a system would also be more observant with respect to sharp corners in the seam. However, the detection of two crossing lines makes image processing far more complicated. An algorithm has been developed to do seam detection with crossing line projections, but when the seam gets close to the crossing point the detection accuracy and stability of the algorithm cannot be guaranteed.

In general the accuracy of the seam detection depends on two things: the quality of image undistortion and the resolution of the image. In practice however, the static thresholding used in real-time is not sufficiently robust to give a stable result. Threshold values often need to be changed manually to obtain a proper image.

Both methods – the one with one line and the one with the crossing lines – were able to track all test seams. Straight seams with and without an inclination, a curved seam and a seam on a double curved surface were tracked successfully. This also means that the proposed calibration method for the tool gives an accurate estimate of the TCP, as this is essential for proper tracking.

The tool calibration and world coordinate calibration were implemented as an automated process. The influence of the height on the scaling factor is quite big. It is therefore important to determine the scaling factor at several heights and compensate for the differences in the triangulation process.

6.3 Recommendations

Some further developments are recommended (in order of importance):

- Implementation of a height-dependent scaling factor for triangulation (see the note in §4.4.1)
- Development and implementation of a more robust, adaptive thresholding algorithm for thresholding in real-time. Alternatively more powerful structured light diodes could be used.
- Improvement of the robustness of the detection of crossing lines
- Development of a graphical user interface for camera calibration
- Development of a toolbox for analysis of camera calibration results
- Consideration of implementation of the geometric approach (§B.6.1) for finding the X-Y position of the TCP (§4.4.2)
- Consideration of a LSQ approach for determining the scaling factor and rotation in the real world coordinate calibration §4.4.1
- Further research could be done on using a grid of small pinhole-spots as calibration pattern, these spots would be easier to detect (and possibly more accurate) than the current patterns, because of their brightness. In no case should the calibration pattern be made by moving the robot, as this will undoubtedly introduce errors.

Appendices

Appendix A

Morphological Image Processing Functions

Finding the seam or calibration objects in camera images is often an easy job for humans, but a very difficult one for computers. Several mathematical algorithms can be used to find or enhance the important features in an image. The main algorithms used in this thesis will be discussed here. First a short introduction to images and their digital representation will follow. For a wider and more thorough discussion of image processing techniques the reader is referred to [29] and [30]

A.1 Images

Globally three kinds of images can be distinguished: black/white images (also called binary images), grayscale images (also called intensity images) and colour images. All camera images are actually a grid of ‘pixels’, small areas with a certain intensity or colour. Tables A.1–A.3 show examples of the black/white, grayscale and colour image representations. Although these examples show 4×4 pixels, images in general consist of hundreds of thousands or even millions of pixels. The intensity range is not always expressed from 0 to 1, but often from 0 to 255 or even differently.

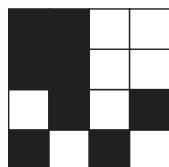
	0	0	1	1
	0	0	1	1
	1	0	1	0
	0	1	0	1

Table A.1: In a binary image pixels can only have values 0 and 1

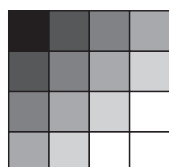
	0	0.2	0.4	0.6
	0.2	0.4	0.6	0.8
	0.4	0.6	0.8	1
	0.6	0.8	1	1

Table A.2: In a grayscale image pixels can have any value between 0 and 1

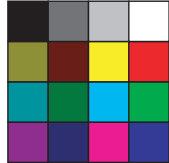
	Red				Green				Blue			
	0	0.3	0.7	1	0	0.3	0.7	1	0	0.3	0.7	1
	0.5	0.5	1	1	0.5	0	1	0	0	0	0	0
	0	0	0	0	0.5	0.5	1	1	0.5	0	1	0
	0.5	0	1	0	0	0	0	0	0.5	0.5	1	1

Table A.3: In a colour image pixels can have any value between 0 and 1 for 3 different layers, like the primary colours Red, Green and Blue.

A.2 Thresholding

Grayscale and colour images can be converted to binary images by thresholding. This means that all pixels with a value smaller than a specified value are turned to black (0) and all pixels with a value larger than the specified value are turned into white (1). Figure A.1 illustrates this process.

Thresholding is a way to highlight the interesting parts of an image. It makes it easier for a computer to handle because the objects present in the picture become more distinct. How the thresholded image looks is to a large extent determined by the choice of the thresholding value. A careful determination of the proper value (which may even change over the pixels in an image) can therefore be a difficult task. Appendix C deals with some specialised thresholding algorithms.

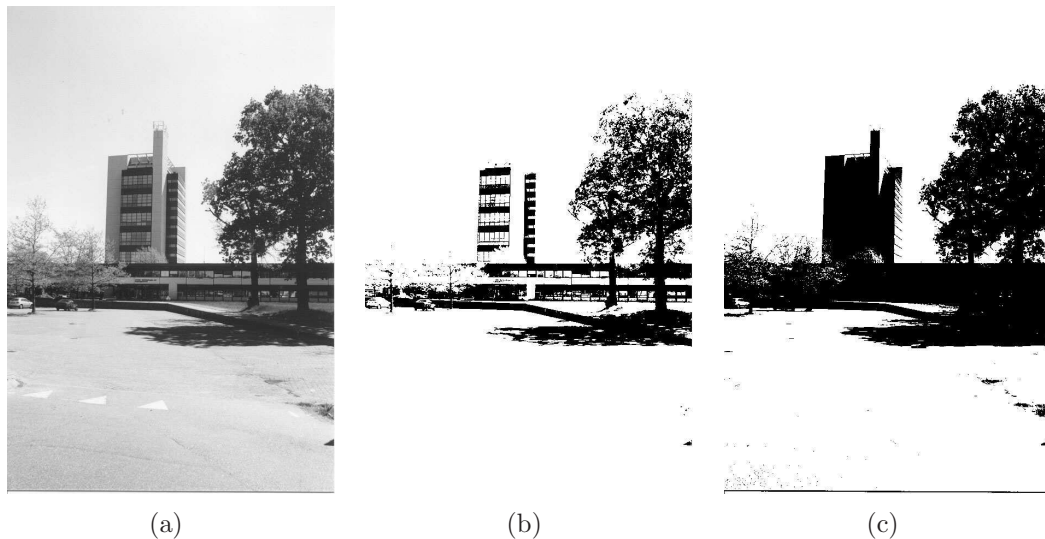


Figure A.1: A grayscale image can be converted to a black/white image by thresholding.

- (a) Original grayscale image.
- (b) Thresholded with a level of 0.3, windows are still recognisable.
- (c) Thresholded with a level of 0.7, the main part of the building and trees become one black area.

A.3 Labelling

The labelling process gives every object (set of interconnected white pixels) in an black/white image an unique number¹. It returns a table which shows for every pixel to which object it belongs, or 0 if it was a black pixel. After grouping pixels together using this labelling procedure, group properties like centroid coordinates, area and bounding box can be determined for every object.

To determine whether two pixels are interconnected, and thus belong to the same object, we can look at the 4 or the 8-connected neighbourhood. The 4-connected neighbourhood only connects to four pixels: up, down, left and right, whereas the 8-connected neighbourhood also connects to the 4 pixels positioned diagonally. This difference is illustrated by the line at the right in Figures A.2(a) and (b).

0	1	1	0	2	0	0	0
1	1	1	0	2	0	0	0
0	1	0	0	0	3	0	0
0	0	0	0	0	3	0	0
0	0	4	0	0	0	5	0
0	0	0	0	0	0	5	0
0	6	6	0	0	0	0	7
0	6	6	0	0	0	0	7

(a) When using a 4-connected neighbourhood diagonally neighbouring pixels are not in the same object (see objects 2,3,5 and 7)

0	1	1	0	2	0	0	0
1	1	1	0	2	0	0	0
0	1	0	0	0	2	0	0
0	0	0	0	0	2	0	0
0	0	3	0	0	0	2	0
0	0	0	0	0	0	2	0
0	4	4	0	0	0	0	2
0	4	4	0	0	0	0	2

(b) When using an 8-connected neighbourhood diagonal neighbours are connected.

Figure A.2: Objects in a black/white image can be identified by a labelling algorithm.

A.4 Erosion & Dilation

The erosion operation removes pixels from the edges of (white) objects (see Figure A.3) by making all pixels neighbouring a black pixels black as well. The 4- or 8-connected neighbourhood definitions discussed in §A.3 can be used to define which pixels are ‘neighbouring’. Another option is to use a structuring element, which is actually just a mask that makes the centre pixel black if one of the pixels within the mask is black. Erosion can be used to remove noise pixels, break ‘bridges’ between white objects and to remove spikes from images.

The counterpart of erosion is dilation, which adds pixels to the edge of objects (see Figure A.4). It can be used to fill small gaps in objects.

¹ If black objects on a white background need to be detected, the image should be inverted first: all pixels with value 1 (white) are given the value 0 (black) and vice versa.

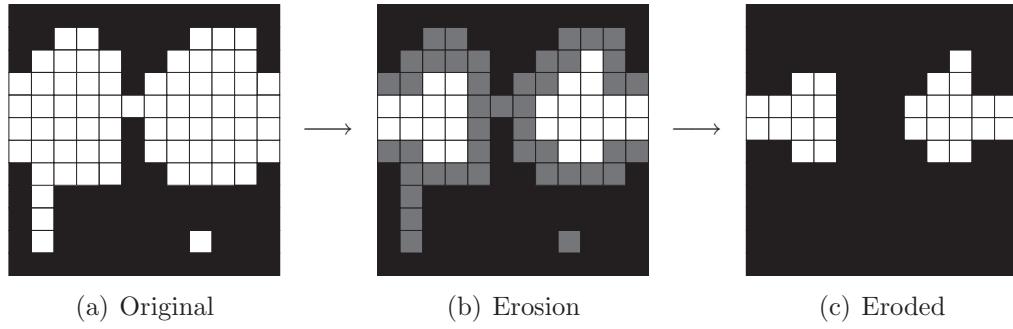


Figure A.3: The standard 8-connected neighbourhood erosion operation makes all pixels that have a black pixel in their neighbourhood black.

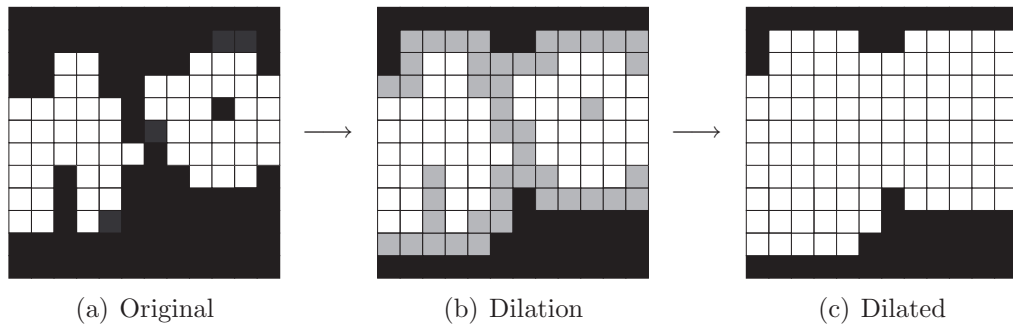


Figure A.4: The standard 8-connected neighbourhood dilation operation makes all pixels that have a white pixel in their neighbourhood white.

When erosion and dilation are combined the opening and closing operations are obtained. Figure A.5 shows that the ‘bridge’ connecting the two objects at the top can be removed using erosion and that the following dilation operation recovers most of the rest of the objects. The object has been opened. Figure A.6 shows the opposite at the bottom: the object is closed.

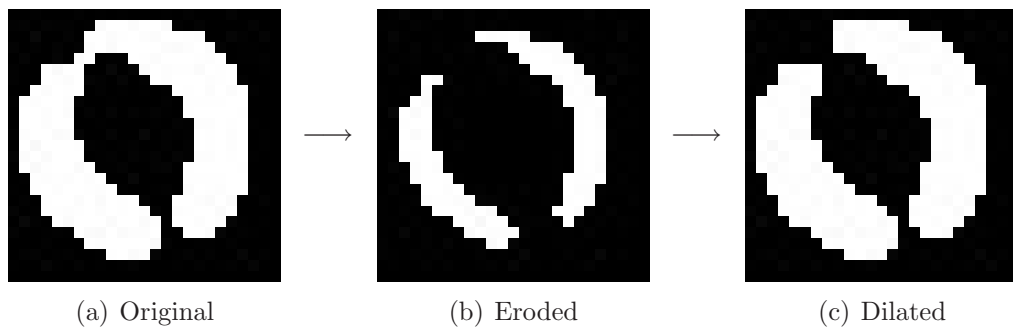


Figure A.5: Opening is the successive application of an erosion and a dilation operation.

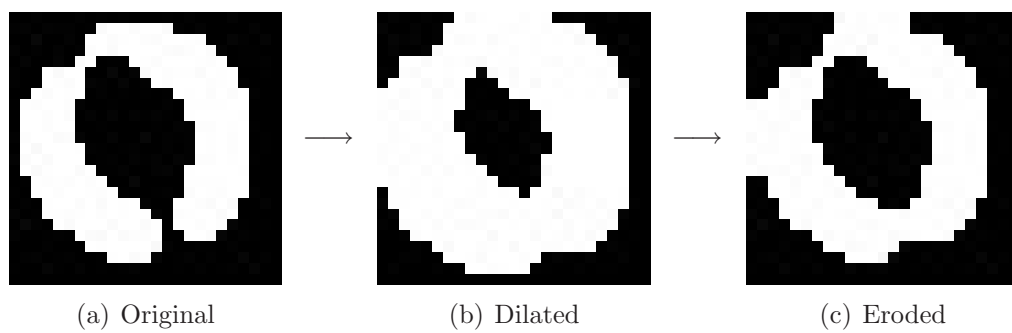


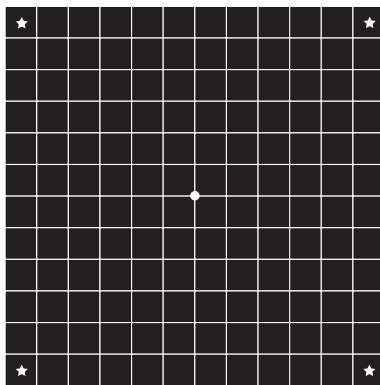
Figure A.6: Closing is the successive application of a dilation and an erosion operation.

Appendix B

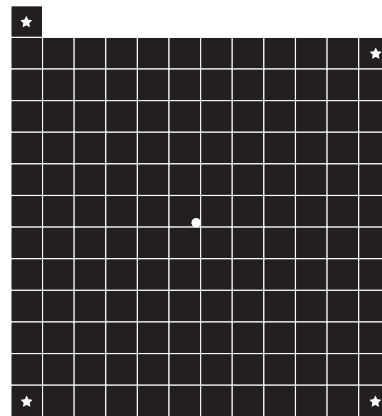
Theory

B.1 Choice of Calibration Objects and Keypoints

Although Zhang [1] uses the corners of the black squares as keypoints, using their centroids is an interesting alternative. Not only remains the option of switching to circles instead of squares, but more importantly, the averaging taking place in the determination of the centroid improves the accuracy of the keypoint location estimation, which is demonstrated in Figure B.1. A disadvantage may be that less keypoints are available, but increasing the number of squares would compensate for that.



(a) In an ideal picture of an object, the keypoints would be found at the stars (corners) or circle (centroid).

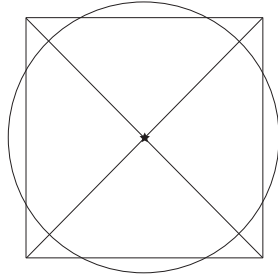


(b) In picture with one extra pixel (due to noise or bad thresholding), the corner keypoint would be shifted a whole pixel, whereas the centroid keypoint only moves slightly.

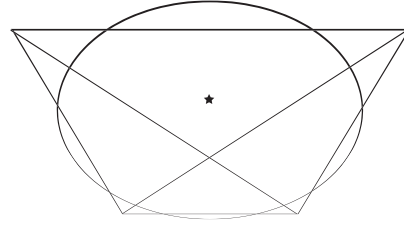
Figure B.1: Comparison of corner and centroid keypoints

A more serious problem with object centroids is that they do not undergo transformations in the same way as points. This means that for instance in a perspective transformation the keypoints detected from the image are not at the positions described by the mathematical deformation of the model points, as pointed out in Figure B.2. This effect can be minimised by using smaller objects

and smaller view-angles.



(a) The centroid of both the circle and the square (which have the same area) is located at the intersection of the diagonals, marked with the star.



(b) When viewed under a large angle, the objects will deform and the transformed centroid position (intersection of diagonals) will no longer match the detected centroid position (star)

Figure B.2: Behaviour of the centroid keypoint on perspective deformation

The problem with the corner keypoints may seem academic and small compared to the problem of shifting centroids. For clear, high resolution images taken under big angles, this is indeed the case, but for the calibration images taken with the integrated welding head, determining the corner of an object is far from evident (as Figure B.3 shows) and due to a small focal depth view angles must be kept small anyway.

Mainly because of easy detection a calibration pattern consisting of rectangles with their centroids as keypoints was chosen. However an algorithm to extract corner keypoints from the images was implemented as well. The choice for squares instead of circles was made based on the fact that with the same object spacing, the squares will have a higher image coverage and thus the best averaging effect on centroids.

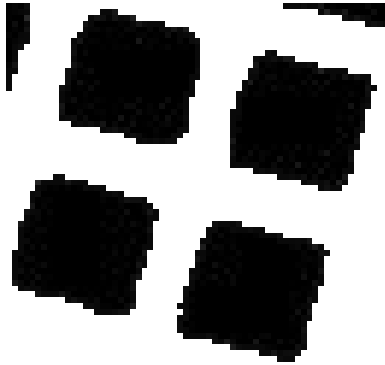


Figure B.3: Determining the position of corners is not always evident.

B.2 Distortion Formulas

Lens distortions are introduced between the scene and the camera. Therefore the rotation and translation of the scene have already occurred, but the camera

transformation has not. This means that the distortion formula should act on the normalised model keypoints in camera coordinates $[\mathbf{x}, \mathbf{y}, \mathbf{z}]_{c,n}^T$, which would be right in between $\mathbf{A}\mathbf{F}$ and $[\mathbf{R} \ \mathbf{t}]$ in Equation 2.9. However, \mathbf{A} was especially normalised because extra zooming of the image in the undistortion process is unwanted. This implies that the distortion is calculated in between \mathbf{A} and $\mathbf{F}[\mathbf{R} \ \mathbf{t}]$ instead¹.

B.2.1 Radial Distortions

The first and second order radial distortions are described by²[1, 18]:

$$\check{\mathbf{x}} = \mathbf{x} + \mathbf{x} \left[\kappa_1 (\mathbf{x}^2 + \mathbf{y}^2) + \kappa_2 (\mathbf{x}^2 + \mathbf{y}^2)^2 \right] \quad (\text{B.1})$$

$$\check{\mathbf{y}} = \mathbf{y} + \mathbf{y} \left[\kappa_1 (\mathbf{x}^2 + \mathbf{y}^2) + \kappa_2 (\mathbf{x}^2 + \mathbf{y}^2)^2 \right], \quad (\text{B.2})$$

where κ_1 and κ_2 are the radial distortion parameters. From Equation 2.2 we can derive that:

$$\begin{aligned} \check{u} &= u_0 + \alpha \check{\mathbf{x}} + \gamma \check{\mathbf{y}} \\ \check{v} &= v_0 + \beta \check{\mathbf{y}}. \end{aligned} \quad (\text{B.3})$$

Substitution of Equations B.1 and B.2 gives:

$$\begin{aligned} \check{u} &= u_0 + \alpha \mathbf{x} + \alpha \mathbf{x} [\dots] + \gamma \mathbf{y} + \gamma \mathbf{y} [\dots] \\ &= \underbrace{u_0 + \alpha \mathbf{x} + \gamma \mathbf{y}}_{=u} + \underbrace{\alpha \mathbf{x} [\dots] + \gamma \mathbf{y} [\dots]}_{=(u-u_0)[\dots]} \\ &= u + (u - u_0) \left[\kappa_1 (\mathbf{x}^2 + \mathbf{y}^2) + \kappa_2 (\mathbf{x}^2 + \mathbf{y}^2)^2 \right], \end{aligned} \quad (\text{B.4})$$

and

$$\begin{aligned} \check{v} &= v_0 + \beta \mathbf{y} + \beta \mathbf{y} [\dots] \\ &= \underbrace{v_0 + \beta \mathbf{y}}_{=v} + \underbrace{\beta \mathbf{y} [\dots]}_{=(v-v_0)[\dots]} \\ &= v + (v - v_0) \left[\kappa_1 (\mathbf{x}^2 + \mathbf{y}^2) + \kappa_2 (\mathbf{x}^2 + \mathbf{y}^2)^2 \right]. \end{aligned} \quad (\text{B.5})$$

To solve these equations they can be put in matrix-vector form:

$$\begin{bmatrix} (u - u_0) (\mathbf{x}^2 + \mathbf{y}^2) & (u - u_0) (\mathbf{x}^2 + \mathbf{y}^2)^2 \\ (v - v_0) (\mathbf{x}^2 + \mathbf{y}^2) & (v - v_0) (\mathbf{x}^2 + \mathbf{y}^2)^2 \end{bmatrix} \begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix} = \begin{bmatrix} \check{u} - u \\ \check{v} - v \end{bmatrix}. \quad (\text{B.6})$$

If these 2 equations for all m points in all n images are stacked we get a $2mn \times 2$ matrix \mathbf{D} times a 2×1 vector $\boldsymbol{\kappa}$ equals a $2mn \times 1$ vector \mathbf{d} . The linear least-squares solution is calculated using the Moore-Penrose matrix inverse (pseudoinverse):

$$\mathbf{k} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{d}. \quad (\text{B.7})$$

¹ This is allowed because f is the same for \mathbf{x} and \mathbf{y} so the effect can be fully compensated for by the values of κ_1 and κ_2 (also in τ_1 and τ_2)

² For readability the subscripts were left out.

B.2.2 Extension with Tangential Distortions

When also tangential distortions are taken into account the distortions are described by[18]:

$$\check{\mathbf{x}} = \mathbf{x} + \mathbf{x} \left[\kappa_1 (\mathbf{x}^2 + \mathbf{y}^2) + \kappa_2 (\mathbf{x}^2 + \mathbf{y}^2)^2 \right] + \tau_1 (3\mathbf{x}^2 + \mathbf{y}^2) + 2\tau_2 \mathbf{x}\mathbf{y} \quad (\text{B.8})$$

$$\check{\mathbf{y}} = \mathbf{y} + \mathbf{y} \left[\kappa_1 (\mathbf{x}^2 + \mathbf{y}^2) + \kappa_2 (\mathbf{x}^2 + \mathbf{y}^2)^2 \right] + 2\tau_1 \mathbf{x}\mathbf{y} + \tau_2 (\mathbf{x}^2 + 3\mathbf{y}^2), \quad (\text{B.9})$$

where κ_1 and κ_2 are the radial distortion parameters and τ_1 and τ_2 are the tangential distortion parameters. Writing Eqs. B.8 and B.9 into image coordinates gives:

$$\check{u} = u + (u - u_0) [\dots] + \alpha\tau_1 (3\mathbf{x}^2 + \mathbf{y}^2) + \alpha 2\tau_2 \mathbf{x}\mathbf{y} + \gamma 2\tau_1 \mathbf{x}\mathbf{y} + \gamma\tau_2 (\mathbf{x}^2 + 3\mathbf{y}^2) \quad (\text{B.10})$$

and:

$$\check{v} = v + (v - v_0) [\dots] + \beta 2\tau_1 \mathbf{x}\mathbf{y} + \beta\tau_2 (\mathbf{x}^2 + 3\mathbf{y}^2) \quad (\text{B.11})$$

In matrix-vector form this is:

$$\begin{bmatrix} (u - u_0) (\mathbf{x}^2 + \mathbf{y}^2) & (u - u_0) (\mathbf{x}^2 + \mathbf{y}^2)^2 \\ (v - v_0) (\mathbf{x}^2 + \mathbf{y}^2) & (v - v_0) (\mathbf{x}^2 + \mathbf{y}^2)^2 \end{bmatrix} \begin{bmatrix} \alpha (3\mathbf{x}^2 + \mathbf{y}^2) + \gamma 2\mathbf{x}\mathbf{y} & \alpha 2\mathbf{x}\mathbf{y} + \gamma (\mathbf{x}^2 + 3\mathbf{y}^2) \\ \beta 2\mathbf{x}\mathbf{y} & \beta (\mathbf{x}^2 + 3\mathbf{y}^2) \end{bmatrix} \begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \check{u} - u \\ \check{v} - v \end{bmatrix}. \quad (\text{B.12})$$

This system can be solved in the same way as B.6, using B.7.

B.3 Rodrigues' Rotation Formula

Rodrigues' rotation formula (Equation B.13) offers an efficient way to express a rotation matrix (Eq. B.14) as a rotation by an angle θ about a fixed axis specified by a unit vector $\vec{\omega} = [\omega_x, \omega_y, \omega_z]$ [31].

$$\mathbf{R} = e^{\mathbf{\Omega}\theta} = \mathbf{I} + \mathbf{\Omega} \sin \theta + \mathbf{\Omega}^2 (1 - \cos \theta), \quad (\text{B.13})$$

$$= \begin{bmatrix} 1 + (1 - \cos \theta) (-\omega_z^2 - \omega_y^2) & -\sin(\theta) \omega_z + (1 - \cos \theta) \omega_y \omega_x & \sin(\theta) \omega_y + (1 - \cos \theta) \omega_z \omega_x \\ \sin(\theta) \omega_z + (1 - \cos \theta) \omega_y \omega_x & 1 + (1 - \cos \theta) (-\omega_z^2 - \omega_x^2) & -\sin(\theta) \omega_x + (1 - \cos \theta) \omega_z \omega_y \\ -\sin(\theta) \omega_y + (1 - \cos \theta) \omega_z \omega_x & \sin(\theta) \omega_x + (1 - \cos \theta) \omega_z \omega_y & 1 + (1 - \cos \theta) (-\omega_y^2 - \omega_x^2) \end{bmatrix}, \quad (\text{B.14})$$

with:

$$\mathbf{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (\text{B.15})$$

The fact that $\vec{\omega}$ is unitary means that the fourth variable θ can be merged in without loss of information, we choose: $\mathbf{w} = \vec{\omega}\theta$. This shows that a regular 3D rotation matrix containing 9 parameters can be written in only 3 independent parameters using Rodrigues' representation. This is especially beneficial when optimising a rotation, because there are less parameters and no mutual constraints.

The Rodrigues parameters $\vec{\omega}$ and θ can easily be calculated using the identity

$$\begin{aligned} \mathbf{R} - \mathbf{R}^T &= \begin{bmatrix} 0 & -2\omega_z \sin \theta & 2\omega_y \sin \theta \\ 2\omega_z \sin \theta & 0 & -2\omega_x \sin \theta \\ -2\omega_y \sin \theta & 2\omega_x \sin \theta & 0 \end{bmatrix} \\ &= 2\mathbf{\Omega} \sin \theta, \end{aligned} \quad (\text{B.16})$$

because we know that $\vec{\omega}$ must be unitary. A small problem may arise here because the arcsin function will return a value for θ between $-\pi/2$ and $\pi/2$, whereas it might as well be $\pi - \theta$. This can be checked using one of the values from

$$\mathbf{R} + \mathbf{R}^T = \begin{bmatrix} 2 + 2(1 - \cos \theta)(-\omega_z^2 - \omega_y^2) & \cdots & \cdots \\ 2(1 - \cos \theta)\omega_y\omega_x & \cdots & \cdots \\ 2(1 - \cos(\theta))\omega_z\omega_x & \cdots & \cdots \end{bmatrix}. \quad (\text{B.17})$$

Once $\vec{\omega}$ and θ are known they can be multiplied to obtain three independent parameters in the form of \mathbf{w} . Extracting $\vec{\omega}$ and θ from \mathbf{w} is easy because $\vec{\omega} = \mathbf{w}/\|\mathbf{w}\|$ and $\theta = \|\mathbf{w}\|$, after which a full rotation matrix can be generated using Equation B.13.

B.4 The Root Mean Square (RMS)

The *root mean square* or RMS is a value that gives an indication of how well the parameters are estimated: a smaller RMS means that the distance between the given data points and the points generated by evaluation of a fitting function are smaller. The RMS is defined as

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2}, \quad (\text{B.18})$$

where x are the data points and \hat{x} are the accompanying estimates.

In the camera and lens calibration procedure the RMS value is used to check the reliability of the estimated parameters. The x in Equation B.18 would be

the keypoints extracted from the images and the \hat{x} would be model points, transformed using the estimated rotation, translation, radial distortion and camera parameters. The more accurate the parameters are estimated, the better the model and transformations will describe the pattern in the image. This implies that \hat{x} matches x better, which results in a smaller RMS value.

B.5 Distance Between a Point and a Line

The distance d between a point $\mathbf{p} = [p, q]$ and a line $y = ax + b$ is defined as the shortest distance between \mathbf{p} and any point on the line. The point $\mathbf{r} = [r, s]$ on the line closest to \mathbf{p} is at the intersection of the line and the line $y = -\frac{1}{a}(x - p) + q$ which is perpendicular to it and also goes through \mathbf{p} . This is illustrated by Figure B.4. The distance can be calculated when the position of \mathbf{r} is known, which can be determined by stating:

$$-\frac{1}{a}(r - p) + q = ar + b,$$

so

$$\begin{aligned} -\frac{1}{a}(r - p) &= ar + b - q \\ (r - p) &= -a(ar + b - q) \\ r + a^2r &= p - ab + aq \\ (1 + a^2)r &= p - ab + aq \\ r &= \frac{p - ab + aq}{1 + a^2} \end{aligned}$$

and with $y = ax + b$:

$$\begin{aligned} s &= ar + b \\ d &= \sqrt{(p - r)^2 + (q - s)^2}. \end{aligned}$$

B.6 Determination of a Circle's Centre From Point Data

In §4.4.2 a rotation around an estimated axis provides point data describing a circle, because the estimated axis and the point that should be rotated around do not coincide. Two algorithms were used to find the centre of the circle that can be drawn through the data. The first one has a geometrical approach and uses pairs of points with a known mutual rotation angle, the second one is a linear least squares fitting procedure.

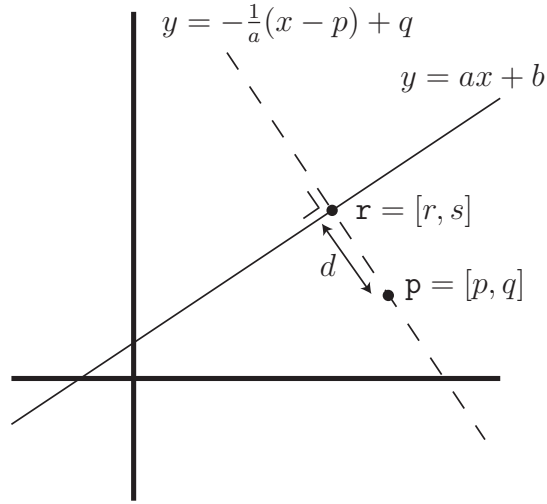


Figure B.4: Calculation of the distance between a point and a line.

B.6.1 A Geometrical Approach

From 2 points $\mathbf{p}_1 = [x_1, y_1]$ and $\mathbf{p}_2 = [x_2, y_2]$ the centre of rotation can be determined as following (see also Figure B.5):

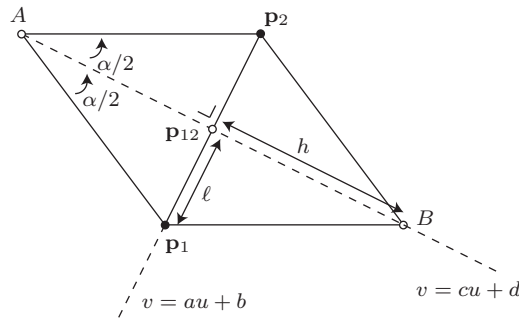


Figure B.5: From 2 points (\mathbf{p}_1 and \mathbf{p}_2) and a known rotation angle α the position of the rotation axis (A) can be calculated.

First define the point \mathbf{p}_{12} as

$$\mathbf{p}_{12} = [x_{12}, y_{12}] \quad (\text{B.19})$$

$$x_{12} = \frac{x_1 + x_2}{2} \quad (\text{B.20})$$

$$y_{12} = \frac{y_1 + y_2}{2}. \quad (\text{B.21})$$

Then the variables ℓ , h and the line parameters a, b, c and d are determined:

$$\ell = \sqrt{(x_{12} - x_1)^2 + (y_{12} - y_1)^2} \quad (\text{B.22})$$

$$h = \frac{\ell}{\tan \alpha/2} \quad (\text{B.23})$$

$$a = \frac{y_2 - y_1}{x_2 - x_1} \quad (\text{B.24})$$

$$b = y_1 - ax_1 \quad (\text{B.25})$$

$$c = -1/a \quad (\text{B.26})$$

$$d = y_{12} - cx_{12}. \quad (\text{B.27})$$

A is at the line $y = cx + d$ at distance h from \mathbf{p}_{12} (just like B) so

$$y_A = cx_A + d \quad (\text{B.28})$$

and

$$h = \sqrt{(x_A - x_{12})^2 + (y_A - y_{12})^2} \quad (\text{B.29})$$

so

$$h = \sqrt{(x_A - x_{12})^2 + (cx_A + d - y_{12})^2} \quad (\text{B.30})$$

$$h^2 = (x_A - x_{12})^2 + (cx_A + d - y_{12})^2 \quad (\text{B.31})$$

$$0 = \underbrace{(1 + c^2)}_q x_A^2 + \underbrace{(-2x_{12} + 2cd - 2cy_{12})}_r x_A + \underbrace{x_{12}^2 + d^2 - 2dy_{12} + y_{12}^2 - h^2}_s. \quad (\text{B.32})$$

This is a quadratic equation in x_A . Solving this equation gives two solutions for x_A , which are actually the x coordinates of A and B :

$$x_A = \frac{-r}{2q} \pm \frac{\sqrt{r^2 - 4qs}}{2q} \quad (\text{B.33})$$

with the accompanying y coordinates

$$y_A = cx_A + d. \quad (\text{B.34})$$

$$(\text{B.35})$$

The problem now is to determine which solution of Equation B.33 represents point A and which represents point B . Because the direction of the rotation α is known, this can be solved for. In Figure B.5 it is immediately clear that A is the point searched for, because the positive (counter clockwise) rotation α turns line $A\text{-}\mathbf{p}_1$ into $A\text{-}\mathbf{p}_2$, whereas a negative (clockwise) rotation would be needed to do this from point B .

To state this mathematically, normal algebra is not sufficient any more. Geometric algebra[32] provides the wedge product, which is very useful in this case.

The wedge product of two 2D vectors is defined as: $\mathbf{x} \wedge \mathbf{y} = x_1y_2 - x_2y_1$. The wedge product of the vectors pointing from A to \mathbf{p}_1 and from A to \mathbf{p}_2 respectively will be positive, and the vectors from point B will have a negative wedge product. Using this fact it can be sorted out which of the calculated points $[x_A, y_A]$ is point A and thus the centre of rotation.

By doing this for multiple images from multiple rotations, a cloud of estimated centres can be obtained. The 'real' centre is considered to be the mean of all these estimates.

B.6.2 A Least Squares Fitting Approach

The method described above is quite laborious. Another possibility is to use a least squares fitting procedure:

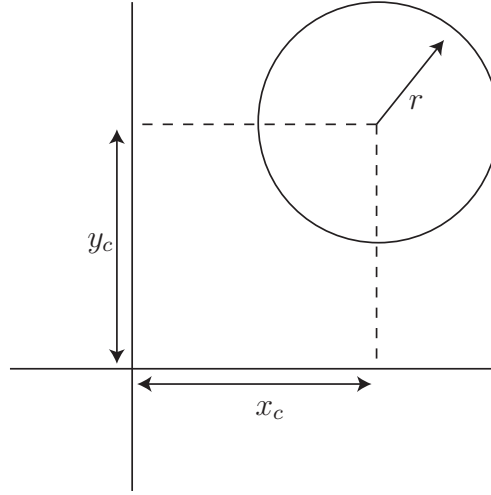


Figure B.6: Definition of a circle.

The points on a circle satisfy the equation (see also Figure B.6)

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad (\text{B.36})$$

and thus

$$x^2 - 2x_c x + x_c^2 + y^2 - 2y_c y + y_c^2 = r^2 \quad (\text{B.37})$$

$$x^2 + y^2 + a_1 x + a_2 y + a_3 = 0, \quad (\text{B.38})$$

with $a_1 = -2x_c$, $a_2 = -2y_c$ and $a_3 = x_c^2 + y_c^2 - r^2$.

This can be written in matrix-vector form as

$$\underbrace{\begin{bmatrix} \mathbf{x} & \mathbf{y} & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}}_{\mathbf{a}} = \underbrace{-(\mathbf{x}^2 + \mathbf{y}^2)}_{\mathbf{b}}, \quad (\text{B.39})$$

where the \mathbf{x} , \mathbf{y} and $\mathbf{1}$ are vectors containing the coordinates of all detected points. The linear least-squares solution is calculated using the Moore-Penrose matrix inverse (pseudoinverse):

$$\mathbf{a} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}. \quad (\text{B.40})$$

The circle centre and radius can be calculated from \mathbf{a} by:

$$x_c = -a_1/2 \quad (\text{B.41})$$

$$y_c = -a_2/2 \quad (\text{B.42})$$

$$r = \sqrt{\frac{a^2 + b^2}{4}} - c. \quad (\text{B.43})$$

Appendix C

Thresholding Algorithms

Several thresholding algorithms were implemented to provide a robust program, able to process a wide range of ‘real life’ calibration images.

C.1 Static Thresholding

Static thresholding means that one thresholding value is determined and used to threshold all pixels in the image. Static thresholding works very good in cases of homogeneous illumination and a homogeneously reflecting fore- and background. In practice the pictures of the paper-printed pattern were thresholded badly and the laser-engraved anodised aluminium pattern pictures were thresholded quite satisfactory.

C.1.1 Black/White Balancing Thresholding

This method determines the average ‘white’ level and the average ‘black’ level to determine a thresholding value. This means that the overall intensity of the image is not important any more, only the weighed average of the 10% darkest and the 10% lightest pixels determines the threshold value. A MATLAB implementation for grayscale image I looks like:

```
x=sort(reshape(I,1,[]));           % sort pixels from black to white
bk=mean(x(1:ceil(length(x)/10))); % average black value (lowest 10%)
wh=mean(x(end-ceil(length(x)/10):end)); % average white value (higher 10%)
level = ((3*bk+wh)/4)/255;         % set the thresholding value
Ibw=im2bw(I,level);                % threshold the image
```

Applicability

This method is the least sophisticated one, which makes it only useful for very plain pictures.

C.1.2 Matlab Thresholding

MATLAB provides two functions to threshold an image in a convenient way. `graythresh` determines a threshold level based on Otsu’s method[33], which

uses clustering and tries to minimize the intraclass variance of the thresholded black and white pixels. Then `im2bw` can be used to threshold the image using the level determined previously. The code for grayscale image `I` would look like:

```
level=graythresh(I); % determine thresholding level
Ibw=im2bw(I,level); % threshold the image
```

Applicability

This method can very well be used for almost all pictures taken from the laser-engraved anodised aluminium pattern, as long as the illumination is reasonably homogeneous.

C.2 Dynamic Thresholding

Dynamic thresholding – also called adaptive thresholding – calculates a threshold value for every pixel, based on the local image properties such as brightness and contrast. The following strategies were inspired by the `THRESHHOLD_IMG.M` MATLAB source code in appendix A.2. of [34].

C.2.1 Dynamic Thresholding I

First of all the canny edge detection algorithm is used to enhance the edges. The results of this edge detection applied to test images was not at all sufficiently accurate, complete and robust to be used without further processing, but it showed a benefit by improving the separation of foreground and background.

```
% enhance the edges in the image to get a better separation
bw = edge(I,'canny'); % finds edges using local intensities
E = I;
E(bw)=0; % set edges black (enhance)
```

The second part of the algorithm consists of determining the local intensity and contrast values of the image. The image is subdivided in blocks, for every block the mean and standard deviation of all pixel intensities are determined. The resulting matrices, with one value for every block, are resized to the original image size using bilinear interpolation.

```
% determine optimal blocksize to use for determination
% of the area to use for mean value calculation
nblocks=10;% use at least nblocks in largest direction
bsiz = bestblk(size(I), max(size(I))/nblocks);% find best block size

% calculate mean for every block
tm = blkproc(E,bsiz,'mean(x(find(x)))');
tm(find(isnan(tm)))=0; % if all pixels in a block are black
```

```

% calculate standard deviation for every block
tsdv = blkproc(I,bsiz,'std(x(find(x)))');
% if homogeneous take full image stdev
tsdv(find(isnan(tsdv))) = std(reshape(I,1,[]));

% scale the found intensity trend back to the original image size
tm = imresize(tm,size(I),'bilinear');
tsdv = imresize(tsdv,size(I),'bilinear');

```

Now for every pixel in the image a local average intensity and standard deviation are present. With these values a lower and higher threshold value are found, which are used to stretch the image. Only the gray values in between the bounds remain gray, outside the bounds they will be turned into black or white. This process is also known as clipping.

```

% set threshold values based on local intensity and standard dev.
Tl = tm - tsdv/2; % lower threshold value
Th = tm + tsdv/2; % higher threshold value

%normalize
G = (I-Tl)./(Th-Tl);
%set all pixels greater than Th to 1
G(find(G > 1)) = 1;
%set all pixels less than Tl to 0
G(find(G < 0)) = 0;

```

Now the image is normalized a static threshold can be applied to get a fully thresholded image. The MATLAB thresholding discussed in §C.1.2 would be most suitable as the amount of black and white pixels that forms the basis for the black/white balancing thresholding algorithm is not meaningful any more after clipping the image.

Applicability

This method works quite well for most pictures, including the ones taken using the paper printed pattern. A drawback is that very small variations in the background are magnified, resulting in a very high noise level on the edges (see Figure C.1(d)). However this noise can often easily be removed by applying the constraint that ‘objects’ are not allowed to be connected to the edges of the image.

C.2.2 Dynamic Thresholding II

In this variant a rough estimation of the foreground and background intensity is made, based upon an initial static (MATLAB) thresholding.

```

level=graythresh(I);          % determine thresholdlevel for bw
I2=im2bw(I,level);           % make black/white

% try to estimate the intensity of the background over the whole image
Id = double(I);
I4=Id;
I4(find(~I2))=mean(Id(:)); % set "black" pixels in I2 to image mean value

```

```

bsiz=ceil(bestblk(size(I))/5);           % determine best blocksize
tm = blkproc(I4,bsiz,'mean(x(find(x)))'); % calculate mean for every region
tm = imresize(tm,size(I),'bilinear');    % resize to original picture size

tsdv = std(Id(:));                       % standard deviation over the whole image

```

When the local intensities of the background (i.e., the areas that became white after the initial thresholding) and the standard deviation in the image are determined, these values can be used to determine a thresholding value for every pixel in the image.

```

% set threshold values based on local intensity and standard deviation
Tl = tm-tsdv;
Th = tm+tsdv;

% normalize
G = (Id-Tl)./(Th-Tl);
% set all pixels greater than Th to 1
G(find(G > 1)) = 1;
% set all pixels less than Tl to 0
G(find(G < 0)) = 0;

```

In this case, only the parts where $G = 0$ are considered. This means that the upper threshold value remains unused.

Applicability

This algorithm works very good for the paper printed pattern pictures, where the background is not plain but shows a clear structure. The magnification of this noise in the background like encountered with the first dynamic thresholding algorithm is no issue here (see Figure C.1(e)). For nice, plain pictures however, the static MATLAB thresholding (§C.1.2) often gives a better result.

C.3 Advanced Keypoint Determination.

An advanced technique using normalisation, thresholding, clipping and weighing was developed to accurately determine the positions of the centroid keypoints. The huge coma effects caused by the laser focus lens make it difficult to define a good threshold value as every object gets a sort of ‘comet tail’. This blurry extension of the objects is sometimes included and other times excluded from the threshold, depending on slight illumination variations. The remedy is to threshold the image first with a too low threshold value, then to determine an area around every object and then to determine the keypoint position using the pixel intensities as a weighing factor.

The first step is to normalize the image, i.e., to undo it from inhomogeneous illumination effects. To do so, the (grayscale) image is divided into blocks of which the local mean intensities are determined. These local intensities are then interpolated to result in a local mean value for every pixel in the original image (Figure C.2(b)). These mean values are subtracted from the original image and the 50% gray level (usually 128) is added (Figure C.2(c)).

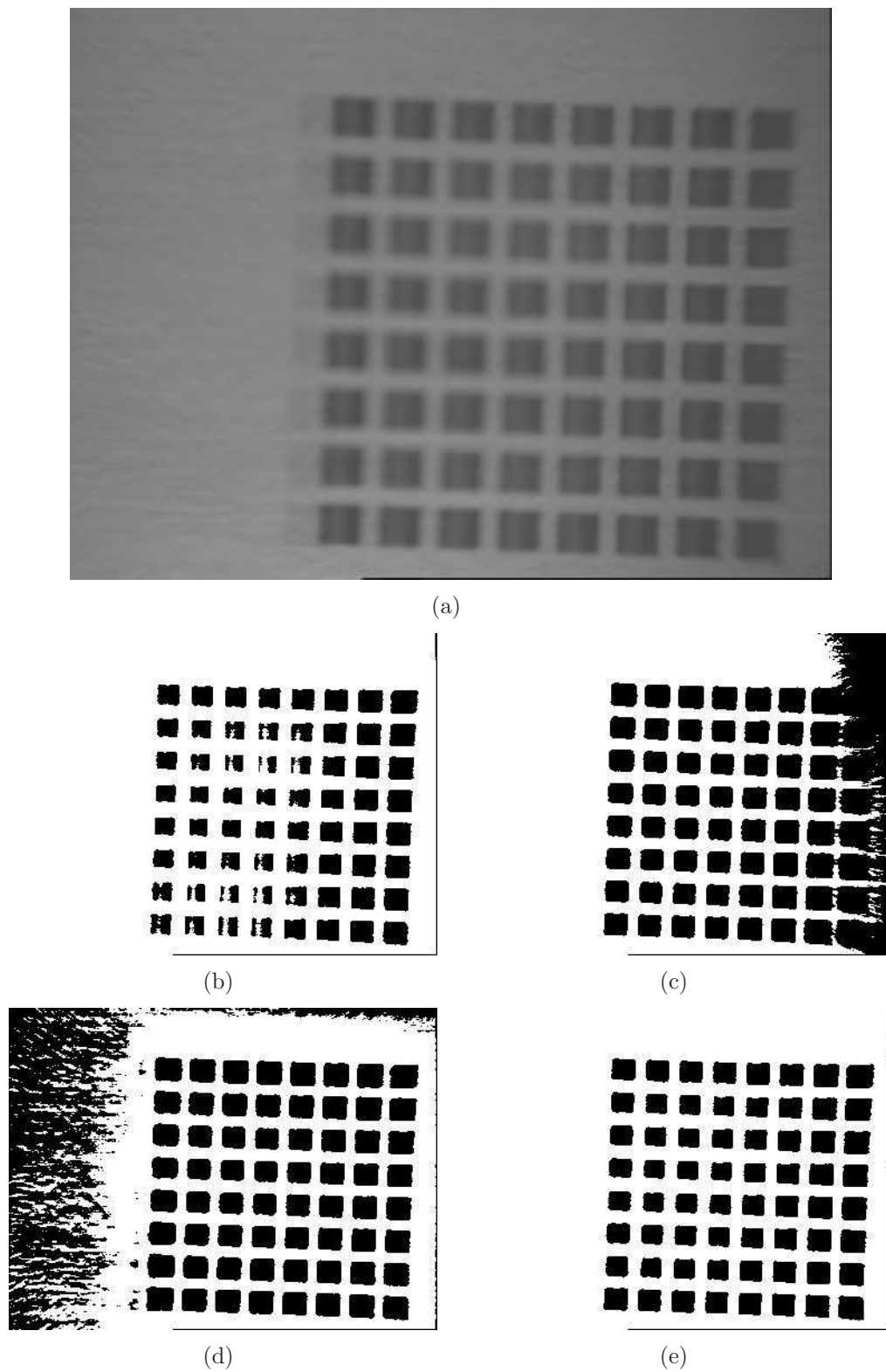


Figure C.1: Comparison of several thresholding algorithms:

- (a) Original image (pattern printed on paper)
- (b) Black/white balancing
- (c) MATLAB thresholding
- (d) Dynamic thresholding I
- (e) Dynamic thresholding II

```

if size(I,3)>1           % if colour image
    I=rgb2gray(I);      % convert to grayscale
end
I=double(I);

blkSize=bestblk(size(I),40);% find best blocksize
% only calculate mean for x-values >0, because x is padded with 0s
Imn=blkproc(I,blkSize,'mean(x(find(x)))');

Imn=Imn-128;
Imn=imresize(Imn,size(I),'bilinear'); % resize
I=I-Imn;                          % normalize original image with mean

```

This method will over-‘correct’ the area with the pattern, because it is relatively dark, but not because of bad illumination. The idea to compensate for this by determining the local mean values of an initially thresholded image has been implemented successfully. However in the end this does not lead to a better thresholding of the pattern. The fact that the local background of the pattern is too light, sort of helps the static MATLAB thresholding applied afterwards to find a good thresholding value.

Due to the light background of the pattern, the rest of the background will some times turn out black after thresholding, so the edges of the image are flood-filled to remove this. After this the objects are labelled (Figure C.2(d)), their sizes are examined and when an object is too small, its pixels are painted white, so it won’t be found in a second labelling step. Of the objects that do meet the size requirements, the centroid position is determined the old way, as a reference.

```

level=graythresh(uint8(I)); % determine threshold level
It=im2bw(uint8(I),level);  % threshold image

% flood-fill edges
[h,w]=size(It);
It(1,:)=0;                 % make top line black
It(:,w)=0;                 % make right line black
It(h,:)=0;                 % make bottom line black
It(:,1)=0;                 % make left line black
It=imfill(It,[1,1]);       % make everything connecting to the edge white

L=bwlabel(~It,4);          % label image

% minimum object size is set according to the image dimensions and pattern
minArea = (size(It,1)*size(It,2)-sum(It(:)))/(2*numRows*numCols);
RP=regionprops(L,'Area', 'Centroid', 'PixelList'); % get object properties
A=cell2mat({RP(:).Area}); % convert to matrix
Q=find(A>minArea);         % see whether objects have a large enough area
Q2=find(A<minArea);        % objects with too small area
printLog(3,['Number of objects > minArea = ' num2str(length(Q))]);
centroids=cell2mat({RP(Q).Centroid}); % list of object centroids

```

```

% make all pixels of objects that were too small white
for i=1:length(Q2)
    It(sub2ind(size(It),RP(Q2(i)).PixelList(:,2),RP(Q2(i)).PixelList(:,1)))=1;
end
L=bwlabel(~It,4); % label again (too small objects are removed now)

```

All objects are now looped over and one by one they are used as a mask for selecting a part of the original image. First the mask is made a bit larger using an erosion algorithm, to incorporate a bit of the (blurry) surroundings of the original object. Now the intensity of the selection is stretched and clipped, such that the part of which we are sure it will belong to the object are black (for the used images this part contains the 200 pixels darkest pixels). Then the pixel coordinates are weighed using the inverse of their intensity, as for dark objects, darker pixels should have a higher weight. This way the new centroid positions are determined. (Figure C.2(e))

```

% calculate area of interest and weighed keypoints
waitbar(1/(max(L(:))+1),wb); % show progress
[X,Y]=meshgrid(1:size(I,2),1:size(I,1)); % make grid with x and y locations
for i=1:max(L(:)) % loop all objects
    se = strel('disk',ceil(sqrt(minArea)/2)); % make filter for erosion
    E = imdilate(L==i,se); % erode image (object neighbourhood mask)
    s=sort(I(find(E==1))); % sort the pixel intensities
    mi=s(ceil(.7*minArea)); % value of the 200th most black pixels
    ma=s(floor(end-.7*minArea)); % value of the 200th most white pixels

    diff=ma-mi; % difference in intensity
    I=(I-mi)*255/diff; % stretch image intensities
    I(find(I<0))=0; % clip the image (lower than 0 = black)
    I(find(I>255))=255; % clip the image (higher than 255 = white)

    W=255-I; % pixel weighing factor (darker is higher)
    pW=X.*W; % weigh x position of the pixel
    meanx(i)=mean(mean((pW(find(E==1)))))/mean(mean(W(find(E==1))));
    pW=Y.*W; % weigh y position of the pixel
    meany(i)=mean(mean((pW(find(E==1)))))/mean(mean(W(find(E==1))));
end

```

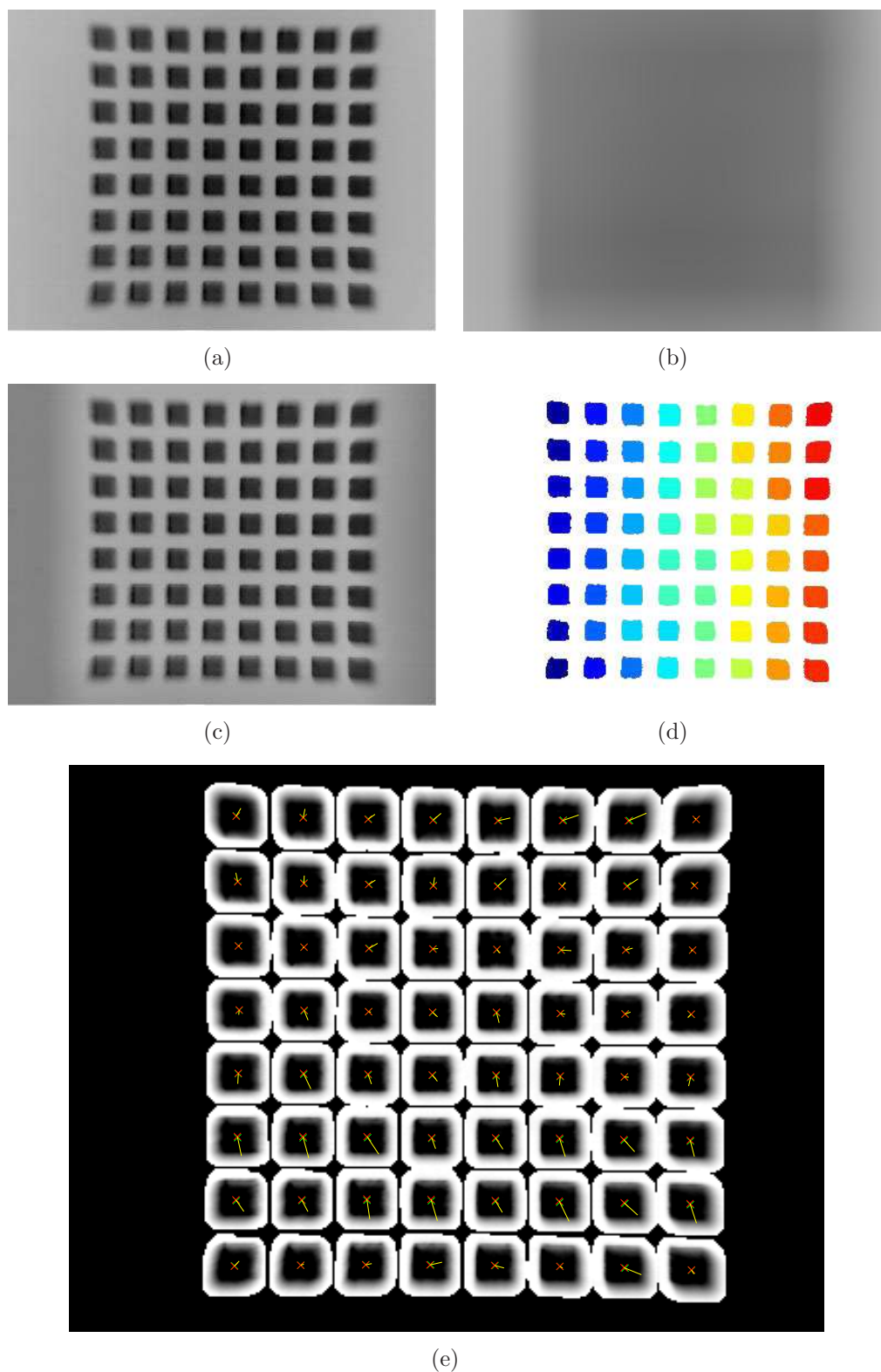



Figure C.2: Normalisation and weighed keypoint determination:

- (a) Original image
- (b) Local intensities, interpolated to the size of the original image
- (c) Normalised image (note the too bright background of the pattern)
- (d) Thresholded image with labelled objects (colour depends on object number)
- (e) Combined masks for all objects, with centroids and the direction in which they moved from the non-weighted centroid (length of lines is magnified $15\times$)

Appendix D

Hardware Specifications

D.1 Welding Head Parameters

D.1.1 Laser Diodes

The specifications of the laser diodes used to project lines at the workpiece are:

type	: Line generator laser diode module LML
wavelength	: 650 nm
output power	: 3 mW
beam	: line, 78°, adjustable focus
operation voltage	: 3 V, DC
operation current	: 45 mA
divergence	: <2.0 mrad
availability	: www.apinex.com

D.1.2 Lenses

All experiments have been carried out using the following lenses:

laser focus lens	: Trumpf 100
focal distance	: 100 mm
camera lens	: Rainbow G25 1.4, C-Mount
focal distance	: 25 mm
aperture	: full (F1.4)

considering the geometry of the welding head and the camera used, this results in a field of view of approximately 18×24 mm. Laser focus lenses of 150 and 200 mm are available as well. Different camera lenses will be used in combination with these, to maintain the field of view.

D.1.3 Camera & Frame Grabber

The specifications of the video camera mounted to the welding head are:

brand : Kappa
type : CF 4/1
chip type : black/white CCD
chip dimensions : 4.8×3.6 mm
maximum full frame frequency : 25 Hz

The specifications of the frame grabber used to read out the analogue video data with the computer are:

brand : Ellips
type : Rio Full
frame dimensions : 768×576 px
field dimensions : 384×288 px
colour depth : 8 bit grayscale
maximum frame frequency : 50 Hz

D.2 Calibration Patterns

The calibration pattern has been produced in several variants. Their properties are listed here.

D.2.1 Paper Patterns

Patterns printed on paper were printed on normal paper using a standard office laser printer on normal paper. They were carefully attached to a flat steel plate using tape, making sure that no wrinkles are present.

objects : squares
number of objects : 8×8
pattern size : 15×15 cm (for use without laser focus lens)
pattern size : 15×15 mm (for use with laser focus lens)
object size : 1.304×1.304 cm (or mm)
object spacing : 1.956 cm (or mm)
object colour : black
background colour : white
base material : paper
production method : printing using a laser printer, best quality

D.2.2 Laser-engraved Patterns

The paper patterns tend to wrinkle after some days, probably due to the humidity of the air. Therefore the pattern has been engraved in a small plate of anodised aluminium as well. The following parameters were used:

objects	: squares
number of objects	: 8×8
pattern size	: 15×15 mm
object size	: 1.304×1.304 mm (version 1)
object spacing	: 1.956 mm (version 1)
object size	: 1×1 mm (version 2)
object spacing	: 2 mm (version 2)
object colour	: gray/white
background colour	: black
base material	: anodised aluminium
production method	: laser engraving using a VectorMark [®] VWS 800
object hatching	: 20 μ crosshatch, 0 and 90°
engraving speed	: 57 mm/s
pulse frequency	: 5957 Hz
pulse width	: unknown (no setting available)
mean power	: 99% of 30 W
repetitions	: 1
lens focal distance	: 163 mm
focus position	: on surface
shielding gas	: none
pressure	: atmospheric
wavelength	: Nd-YAG (1064 nm)

Another pattern was produced using a femto-second laser, which produces even more accurate patterns and has a more user-friendly interface.

objects	: squares
number of objects	: 12×12
pattern size	: 15×15 mm
object size	: 0.652×0.652 mm
object spacing	: 1.304 mm
object colour	: gray/white
background colour	: black
base material	: anodised aluminium
production method	: laser engraving using a Coherent Vitesse 800 + RegA 9000 femto-second laser
object hatching	: 20 μ crosshatch, 0 and 90°
engraving speed	: 300 mm/s
pulse frequency	: 250 kHz
pulse width	: 200·10 ⁻¹² s
mean power	: 1 W
repetitions	: 1
lens focal distance	: 100 mm
focus position	: on surface
shielding gas	: none
pressure	: atmospheric
wavelength	: 800 nm

D.2.3 Simulated Images

Simulated Data Sets

Data sets for simulations are generated by transforming the model centroid keypoint coordinates using a known set of camera and lens parameters and randomly generated rotations and translations. Random noise can be added to the transformed coordinates.

The parameters were set as following:

Parameter	Value
α_0	100
β_0	103
γ_0	1
$u_{0,0}$	5 px
$v_{0,0}$	-3 px
$\kappa_{1,0}$	$1 \cdot 10^{-9}$
$\kappa_{2,0}$	$1 \cdot 10^{-9}$
image size	200×200 px

The image size results in a normalisation factor¹ of $n=54.5656$ and the camera parameters result in a factor $f_0=101.5135$, according to

$$f_0 = \frac{\left\| \begin{pmatrix} \alpha & \gamma \\ 0 & \beta \end{pmatrix} \right\|_{fro}}{\sqrt{2}}. \quad (D.1)$$

Using these parameters the values to be detected by the algorithm become:

Parameter	Value	transformation used
α	0.9851	α_0/f_0
β	1.0146	β_0/f_0
γ	0.0099	γ_0/f_0
u_0	1.7410	$-\left(\frac{\text{image width}}{2} - u_{0,0}\right)/n$
v_0	1.8876	$-\left(\frac{\text{image height}}{2} - v_{0,0}\right)/n$
κ_1	$2.9774 \cdot 10^{-6}$	$n^2 \kappa_{1,0}$
κ_2	0.0089	$n^4 \kappa_{2,0}$
f	1.8604	f_0/n

Simulated Images

The simulated images are created by transforming the model corner keypoint coordinates with a known set of camera and lens parameters and randomly generated rotations and translations. The the transformed model points belonging to the same object are then connected and the resulting shapes are filled with

¹ Normalisation is such that the mean over all pixel-to-image centre distances is $\sqrt{2}$

black. This approach gives a slightly different result from the real situation, because the edges of objects won't be curved now. The resulting plot will be captured to a 384×288 pixel image. Images that cannot be processed because the pattern is not fully within the boundaries are taken out manually.

The parameters the values to be detected by the algorithm are:

Parameter	Value.
α	1.0094
β	0.9905
γ	0.0015
u_0	0.0333
v_0	-0.2645
κ_1	0.0150
κ_2	0.0010
f	40.0

For tests with blurring, the images created are blurred in MATLAB using (for a grayscale image I):

```
psf=fspecial('gaussian',7,10);
v=0.0001;
Iblur=imnoise(imfilter(I,psf),'gaussian',0,v);
```

To add some effect of uneven lighting, the images are edited in Jasc's PAINT SHOP PRO, version 8. In the 'illumination effect' *lights* only one light is used, with the settings:

darkness : 35	horizontal : 96	smoothness : 28
intensity : 40	vertical : 92	cone size : 49
direction : 313	scale : 10	asymmetry : 613

D.3 Photo Camera

The camera and camera settings used are the following:

brand	: Nikon
type	: D70
lens	: AF Micro Nikkor 55mm f/2.8
image size	: 3008×2000 pixels
depth	: 24 Bit
f-number	: F/8
exposure	: 1/125 s
exposure compensation	: +1

References

- [1] Z. Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, Dec 1998. URL <http://research.microsoft.com/~zhang/Calib/>.
- [2] D. Iakovou, R.G.K.M Aarts, and J. Meijer. Integrated sensors for robotic laser welding. In *Proceedings of the third International WLT-conference on Lasers in Manufacturing, Munich, Germany*, pages 121–126. AT-Fachverlag GmbH, Stuttgart, Jun 2005. ISBN 3-00-016402-2.
- [3] D. Iakovou, R.G.K.M Aarts, and J. Meijer. Sensor integration for robotic laser welding processes (paper #2301). In *Proceedings of the International Congress on Applications of Lasers and Electro-Optics (ICALEO), Miami, USA*, 2005.
- [4] Wikipedia, the free encyclopedia. Lens (optics), 2005. URL http://en.wikipedia.org/wiki/Lens_%28optics%29.
- [5] Cartage.org. Lens, lens system and optical aberrations, 2005. URL <http://www.cartage.org.lb/en/themes/Sciences/Physics/Optics/Optical/Lens/Lens.htm>.
- [6] Merriam-Webster Inc. *Merriam-Webster's Collegiate Dictionary*. 2000. URL <http://www.m-w.com/>.
- [7] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition - A Unified Approach*. Computational Imaging and Vision. Kluwer Academic Publishers, 6 edition, 1996. ISBN 0-7923-4199-6.
- [8] R.Y. Tsai. An efficient and accurate camera calibration technique for 3d-machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374. IEEE, 1986.
- [9] B.K.P. Horn. Tsais camera calibration method revisited, 2000. URL http://people.csail.mit.edu/bkph/articles/Tsai_Revisited.pdf.
- [10] *A Four-step Camera Calibration Procedure with Implicit Image Correction*, Washington DC, USA, 1997. IEEE Computer Society. URL [http://www.ee.oulu.fi/~jth/doc/\(ps\)orhttp://www.vision.caltech.edu/bouguetj/calib_doc/papers/heikkila97.pdf](http://www.ee.oulu.fi/~jth/doc/(ps)orhttp://www.vision.caltech.edu/bouguetj/calib_doc/papers/heikkila97.pdf).

- [11] G.G. Savii. Camera calibration using compound genetic-simplex algorithm. *Journal of Optoelectronics and Advanced Materials*, 6(4):1255–1261, 2004. URL http://joam.infim.ro/JOAM/pdf6_4/Savii.pdf.
- [12] H. Zollner and R. Sablatnig. Comparison of methods for geometric camera calibration using planar calibration targets. In W. Burger and J. Scharinger, editors, *Digital Imaging in Media and Education, Proceedings of the 28th Workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR)*, pages 237–244, Hagenberg, Austria, 2004. Schriftenreihe der OCG. URL <http://www.prip.tuwien.ac.at/~sab/papers/oagm04b.pdf>.
- [13] J.Y. Bouguet. Camera calibration toolbox for matlab, 2004. URL http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.
- [14] *Halcon Application Guide 7.02*. MVTec Software GmbH, 2005. URL <http://www.mvtec.com/download/documentation/pdf-7.0/ApplicationGuide.pdf>.
- [15] J.E. Haa and D.J. Kangb. Initialization method for self-calibration using 2-views. *Pattern Recognition*, 38:143–150, 2005.
- [16] X. Jian, A.A. Malcolm, and F. Zhongping. Camera calibration with micron level accuracy. Technical Report AT/01/037/PS, Singapore Institute of Manufacturing Technology, 2001. URL <http://www.simtech.a-star.edu.sg/research/TechnicalReports/TR0090.pdf>.
- [17] L. Ma, Y.Q. Chen, and K.L. Moore. A new analytical radial distortion model for camera calibration. *CoRR: Computer Vision and Pattern Recognition*, 2003. URL http://arxiv.org/PS_cache/cs/pdf/0307/0307046.pdf.
- [18] D.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.
- [19] J.J. Moré. *The Levenberg-Marquardt algorithm: Implementation and theory*. In G.A. Watson, (Ed.) *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, pages 105–116. Springer-Verlag, Berlin, Heidelberg, New York, 1977.
- [20] B. Vodanovic. Structured light tracks seams. *Sensor Review*, 16(1):35–39, 1996. Published by Emerald Group Publishing Limited.
- [21] R. Dillmann, M. Kaiser, F. Wallner, and P. Weckesser. *PRIAMOS: An advanced mobile system for service, inspection, and surveillance tasks*. In T. Kanade, H. Bunke and H. Noltemeier, (Eds.) *Modelling and Planning for Sensor Based Intelligent Robot Systems*, volume 21 of *Series in Machine Perception and Artificial Intelligence*, chapter 11. World Scientific, 1995. ISBN 981-02-2238-6. URL <http://wwwipr.ira.uka.de/de/Publikationen/download.php?id=975595024&filetype=pdf>.

- [22] B. Xu, D.F. Cuminato, and N.M. Keyes. Evaluation of fabric smoothness appearance using a laser profilometer. *Textile Research Journal*, 68 (12):900–906, 1998. URL <http://www.utexas.edu/depts/he/XuPapers/15-WRINKLE-TRJ.pdf>.
- [23] T. Jaeckel, J. Miesner, F. Frischkorn, H. Kreitlow, M.R. Jackson, and R. Parkin. Laser welding seam tracking system. In R.M. Parkin, A. Al-Habaibeh, and M.R. Jackson, editors, *Proceedings of the International Conference on Mechatronics (ICOM03)*, Loughborough, UK, pages 593–598, Jun 2003. ISBN 1-86058-420-9.
- [24] M.W. de Graaf, R.G.K.M. Aarts, J. Meijer, and J.B. Jonker. Modeling the seam teaching process for robotic laser welding. In Paul Drews, editor, *Proceedings of the Mechatronics & Robotics 2004 conference*. APS - European Centre for Mechatronics, September 2004.
- [25] T.W. van der Zee. Modeling the seam teaching process for laser welding applications. Master’s thesis, University of Twente, March 2003. Report no. wa0879.
- [26] J.B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, 1999.
- [27] The matrix and quaternions faq, Nov 2003. URL <http://skal.planet-d.net/demo/matrixfaq.htm#Q53>. Version 1.21.
- [28] J. van Tienhoven. Automatic tool center point calibration for a laser welding robot. Master’s thesis, University of Twente, January 2004. Report no. wa0919, classified.
- [29] J.C. Russ. *The Image Processing Handbook*. CRC Press, 4 edition, 2002. ISBN 0-8493-1142-X.
- [30] F. van der Heijden. *Image Based Measurement Systems - Object recognition and parameter estimation*. John Wiley & Sons, 1994. ISBN 0-471-950629.
- [31] E.W. Weisstein et al. Rodrigues’ rotation formula. From MathWorld—A Wolfram Web Resource, 2005. URL <http://mathworld.wolfram.com/RodriguesRotationFormula.html>.
- [32] D. Hestenes. *Clifford Algebra to Geometric Calculus*. Fundamental theories of physics. Kluwer Academic Publishers, 1997. ISBN 9027725616.
- [33] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.
- [34] C.B. Lee. Radial undistortion and calibration on an image array. Master’s thesis, Massachusetts Institute of Technology, 77 massachusetts avenue, Cambridge, USA, May 2000. URL http://groups.csail.mit.edu/graphics/pubs/thesis_cblee.pdf.

