Calibration of Laser Triangulating Cameras in Small Fields of View
Calibration of Laser Triangulating Cameras in Small Fields of View

Examensarbete utfört i Bildbehandling
vid Tekniska högskolan vid Linköpings universitet
av

Daniel Rydström

LiTH-ISY-EX--13/4669--SE

Handledare:  Ander Moe
            SICK IVP
            Hannes Ovrén
            ISY, Linköpings Universitet

Examinator:  Maria Magnusson
            ISY, Linköpings Universitet

Linköping, 3 juni 2013
**Avdelning, Institution**  
Division, Department  
Computer Vision Laboratory  
Department of Electrical Engineering  
SE-581 83 Linköping

**Datum**  
Date  
2013-06-03

<table>
<thead>
<tr>
<th>Språk</th>
<th>Rapporttyp</th>
<th>ISBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑ Svenska/Swedish</td>
<td>☑ Licentiatavhandling</td>
<td>—</td>
</tr>
<tr>
<td>☑ Engelska/English</td>
<td>☑ Examensarbete</td>
<td>—</td>
</tr>
<tr>
<td>☐</td>
<td>☑ C-uppsats</td>
<td>—</td>
</tr>
<tr>
<td>☐</td>
<td>☑ D-uppsats</td>
<td>—</td>
</tr>
<tr>
<td>☐</td>
<td>☑ Övrig rapport</td>
<td>—</td>
</tr>
</tbody>
</table>

**Språk**  
Language  
☐ Svenska/Swedish  
☑ Engelska/English  
☐

**Rapporttyp**  
Report category  
☐ Licentiatavhandling  
☑ Examensarbete  
☐ C-uppsats  
☐ D-uppsats  
☐ Övrig rapport  
☐

**URL för elektronisk version**  
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-94210

<table>
<thead>
<tr>
<th>Titel</th>
<th>Författare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalibrering av lasertriangulerande 3D-kamera för användning i små synfält</td>
<td>Daniel Rydström</td>
</tr>
</tbody>
</table>

**Title**  
Calibration of Laser Triangulating Cameras in Small Fields of View

**Författare**  
Author  
Daniel Rydström

**Sammanfattning**  
Abstract

A laser triangulating camera system projects a laser line onto an object to create height curves on the object surface. By moving the object, height curves from different parts of the object can be observed and combined to produce a three dimensional representation of the object. The calibration of such a camera system involves transforming received data to get real world measurements instead of pixel based measurements.

The calibration method presented in this thesis focuses specifically on small fields of view. The goal is to provide an easy to use and robust calibration method that can complement already existing calibration methods. The tool should get as good measurements in metric units as possible, while still keeping complexity and production costs of the calibration object low. The implementation uses only data from the laser plane itself making it usable also in environments where no external light exist.

The proposed implementation utilises a complete scan of a three dimensional calibration object and returns a calibration for three dimensions. The results of the calibration have been evaluated against synthetic and real data.

**Nyckelord**  
Keywords  
Homography, calibration, 3D, laser plane, triangulation, camera, lens distortion, non-linear optimisation
Abstract

A laser triangulating camera system projects a laser line onto an object to create height curves on the object surface. By moving the object, height curves from different parts of the object can be observed and combined to produce a three dimensional representation of the object. The calibration of such a camera system involves transforming received data to get real world measurements instead of pixel based measurements.

The calibration method presented in this thesis focuses specifically on small fields of view. The goal is to provide an easy to use and robust calibration method that can complement already existing calibration methods. The tool should get as good measurements in metric units as possible, while still keeping complexity and production costs of the calibration object low. The implementation uses only data from the laser plane itself making it usable also in environments where no external light exist.

The proposed implementation utilises a complete scan of a three dimensional calibration object and returns a calibration for three dimensions. The results of the calibration have been evaluated against synthetic and real data.
Acknowledgments

I would like to thank SICK IVP for providing such a rewarding master’s thesis project and giving me the opportunity to work for them. I especially thank my supervisor Anders Moe for his understanding and support from the very start to the end.

I also want to thank my parents who has always been supportive of my life choices both in academics and otherwise. Finally I want to thank my fiancée who put up with me during the work on this thesis.

Linköping, June 2013
Daniel Rydström
# Contents

| 1 Introduction | 1 |
| 1.1 Environment | 2 |
| 1.2 Solution Steps | 3 |
| 1.2.1 Object Identification | 3 |
| 1.2.2 Optimisation | 3 |
| 1.3 Purpose | 4 |
| 1.4 Outline | 4 |

| 2 Background Theory | 5 |
| 2.1 Mathematical Definitions | 5 |
| 2.1.1 The Pinhole Camera Model | 6 |
| 2.1.2 Intrinsic Camera Parameters | 7 |
| 2.2 Plane Computation | 8 |
| 2.2.1 Plane from Three Points | 8 |
| 2.2.2 Plane from Least Squares Fitting | 8 |
| 2.3 Computation of Intersection Points | 9 |
| 2.4 Transformation Model | 10 |
| 2.4.1 Object to Laser Plane Coordinates | 10 |
| 2.4.2 Homography | 11 |
| 2.4.3 Lens Distortion | 12 |

| 3 Calibration Method | 15 |
| 3.1 Calibration Object | 15 |
| 3.2 Calibration Process Overview | 16 |
| 3.3 Plane Extraction | 16 |
| 3.3.1 Segmentation | 17 |
| 3.3.2 Labelling | 18 |
| 3.4 Plane Matching | 18 |
| 3.4.1 Plane Computation | 19 |
| 3.4.2 Matching and Corner Extraction | 20 |
| 3.5 Transformation Summary | 23 |
CONTENTS

3.6 Optimisation .................................................. 24
  3.6.1 Create Start Solution ............................... 26
  3.6.2 Geometric Optimisation ......................... 27
  3.6.3 Lens Model Optimisation ....................... 28

4 Experiments .................................................. 31
  4.1 Evaluated Setups ........................................ 31
    4.1.1 Angle Between Camera and Laser Plane .... 31
    4.1.2 Rotation of Object ............................... 31
    4.1.3 Laser Plane Tilt ................................. 32
  4.2 Evaluation Methods .................................... 33
    4.2.1 Resolution X-axis ................................ 33
    4.2.2 Resolution Y-axis ................................ 33
    4.2.3 Resolution Z-axis ................................ 34
    4.2.4 Effects of Skew ................................... 35
    4.2.5 Lens Model Performance ....................... 35
  4.3 Obtaining Data ........................................... 35
    4.3.1 Synthetic Data .................................... 35
    4.3.2 Real Data ......................................... 36
  4.4 Comparison Against Coordinator ................... 36

5 Results ....................................................... 39
  5.1 Synthetic Results Without Lens Distortion .... 39
    5.1.1 Angle Between Camera and Laser Plane .... 39
    5.1.2 Rotation of Object ............................... 40
    5.1.3 Laser Plane Tilt ................................. 40
  5.2 Synthetic Results With Lens Distortion ........ 43
    5.2.1 Angle Between Camera and Laser Plane .... 43
    5.2.2 Rotation of Object ............................... 45
    5.2.3 Laser Plane Tilt ................................. 45
  5.3 Results On Real Data .................................. 49
    5.3.1 Noise and Resolution ............................. 49
    5.3.2 Angle Between Camera and Laser Plane .... 51
    5.3.3 Rotation of Object ............................... 53
    5.3.4 Laser Plane Tilt ................................. 55
    5.3.5 Effects of Skew ................................... 58
    5.3.6 Lens Model Performance ....................... 58

6 Discussion .................................................. 61
  6.1 Method Development ................................... 61
  6.2 Considerations .......................................... 62

7 Conclusions ................................................ 63
  7.1 Summary ................................................ 63
  7.2 Future Work ............................................ 64

A Homography Estimation ........................................ 67
The following conventions are used throughout this thesis.

### General naming convention

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>([ \cdot ])</td>
<td>Homogeneous coordinates</td>
<td></td>
</tr>
<tr>
<td>((\cdot))</td>
<td>Cartesian coordinates</td>
<td></td>
</tr>
<tr>
<td>(A, X, L)</td>
<td>Matrices and multiple vectors stacked as matrices (bold font, large letters)</td>
<td></td>
</tr>
<tr>
<td>(x, y, l)</td>
<td>Vectors (bold font, small letters)</td>
<td></td>
</tr>
<tr>
<td>(S, r)</td>
<td>Scalar values</td>
<td></td>
</tr>
</tbody>
</table>

### Coordinate spaces

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\tilde{Z} = \begin{bmatrix} \tilde{Z}_1 \ \tilde{Z}_2 \ \tilde{Z}_3 \end{bmatrix})</td>
<td>Object coordinate space coordinates</td>
<td>[metric]</td>
</tr>
<tr>
<td>(Z = \begin{bmatrix} Z_1 \ Z_2 \ Z_3 \end{bmatrix})</td>
<td>Object coordinates aligned with the laser plane</td>
<td>[metric]</td>
</tr>
<tr>
<td>(Y = \begin{bmatrix} Y_1 \ Y_2 \end{bmatrix})</td>
<td>Laser plane coordinates</td>
<td>[metric]</td>
</tr>
<tr>
<td>(C = \begin{bmatrix} C_1 \ C_2 \ C_3 \end{bmatrix})</td>
<td>Camera coordinates</td>
<td>[metric]</td>
</tr>
<tr>
<td>(X = \begin{bmatrix} X_1 \ X_2 \end{bmatrix})</td>
<td>Normalised image plane coordinates</td>
<td>[metric]</td>
</tr>
<tr>
<td>(U = \begin{bmatrix} \mu \ v \end{bmatrix})</td>
<td>Real image plane coordinates</td>
<td>[pixels]</td>
</tr>
<tr>
<td>(\tilde{U} = \begin{bmatrix} \tilde{u}_1 \ \tilde{v}_2 \end{bmatrix})</td>
<td>Distorted real image plane coordinates</td>
<td>[pixels]</td>
</tr>
</tbody>
</table>
## Notation

### Operators

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^T, A^T$</td>
<td>Vector or matrix transpose</td>
</tr>
<tr>
<td>$A^{-1}$</td>
<td>Inverse of matrix</td>
</tr>
<tr>
<td>$A^{-T}$</td>
<td>Inverse of the transpose of $A$</td>
</tr>
<tr>
<td>$x \sim y$</td>
<td>Similarity, equal up to an unknown scale factor</td>
</tr>
<tr>
<td>$x = y$</td>
<td>Equality</td>
</tr>
<tr>
<td>$x \times y$</td>
<td>Vector cross product</td>
</tr>
<tr>
<td>$x \cdot y = x^T y$</td>
<td>Vector dot product</td>
</tr>
</tbody>
</table>

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLT</td>
<td>Direct linear transformation</td>
</tr>
<tr>
<td>DOF</td>
<td>Degrees of freedom</td>
</tr>
<tr>
<td>FOV</td>
<td>Field of view</td>
</tr>
<tr>
<td>RANSAC</td>
<td>RANdom SAmple Consensus</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular value decomposition</td>
</tr>
<tr>
<td>COORDINATOR</td>
<td>SICK IVP patented calibration software.</td>
</tr>
</tbody>
</table>
SICK IVP designs and builds cameras for 3D triangulation. The technology used is sheet of light laser triangulation which requires special hardware to perform. The two most essential hardware components are a laser, that projects a line onto what is to be scanned, and a camera, that observes this line from a perspective. In certain areas it provides superior quality and or robustness, compared to competing technologies, such as stereo triangulation.

![System overview](image)

**Figure 1.1: System overview.**

From these systems height maps of objects may be extracted expressed in sensor coordinates. However, the relation to standard metric units is non-trivial. Figuring out this relation is, in essence, what calibrating the system means. This master’s thesis will explore the possibilities of calibrating such a system for small
fields of view (FOV) using a simple calibration object. More precisely, the FOV in which the method primarily should operate is 10-50 mm.

1.1 Environment

As previously mentioned, the technique involves a laser and a camera. The laser projects a line onto the surface of the object which is to be measured. The camera observes where this line appears on the image sensor in real image coordinates. Because the camera observes the laser plane at an angle, the laser line will be reproduced at different heights in the camera sensor depending on the object height. From these sensor measurements the height of the object can be determined. Each such height curve is usually called a slice. A pseudo range image is formed by detecting the laser line in a sequence of images, while moving the object and stacking received slices together. The pseudo range image will then consist of a 2D array with height encoded as the intensity. The rows of this image will correspond to the slices of the scan and the columns the position along the width of the camera sensor. This image expresses the shape of the object in three dimensions, but in non metric units.

The object is scanned by moving it along the axis referred to as $\hat{n}$, a three-dimensional visualisation of the geometry is provided in Figure 1.1. While doing so multiple pictures are taken and corresponding profile curves are extracted at different positions on the object. Scanning this way requires that the object is propagated through the laser plane in a constant direction. However, it is not certain that the laser plane is orthogonal to the direction of movement (see Figure 1.2 for $\hat{n}$ relative the laser plane aligned object coordinates $Z$), nor is it certain that the velocity of the movement is constant. This provides additional challenges to the problem formulation that, unattended, could produce different results when an object is scanned from different directions. This thesis will only consider cases where the propagation speed of the object is constant or a motion encoder is available.

Figure 1.2: Illustration of translation direction and object rotation.

In Figure 1.2 $Z_2$ is orthogonal to the laser plane and $Z_1, Z_2, Z_3$ forms and orthonormal basis. For calibration of the $Z_1$ and $Z_3$ axes only points observed in the laser plane are considered. This gives a homographic relation from the laser plane to the image plane. In certain situations the relative position between the
laser and camera may be calibrated and fixed once and for all in production. For the SICK IVP Ranger series of cameras however, which this thesis will regard, this should in general be customisable. The positioning and orientation of all components in the environment may therefore vary.

1.2 Solution Steps

The complete algorithm consists of several parts. In general terms it can be said to consist of mainly two parts; object identification and optimisation.

1.2.1 Object Identification

Before the points on the object in the pseudo range image can be optimised to fit the ones in world coordinates, defined by the calibration object, it is necessary to find out the position of the object in the scan. This is done in the object identification step. The identification consists of finding where planar surfaces occur in the scan, estimating plane equations for these surfaces and matching the planes to the model of the calibration object. Finally, the points corresponding to the corner points on the ground truth object are found in pseudo range data.

1.2.2 Optimisation

The optimisation problem can be divided into three different parts that are illustrated in Figure 1.3. The first is to determine how a three dimensional object is reproduced in the laser plane. For this a special kind of transformation have to be found. This transformation is determined both from the orientation of the laser plane compared to the propagation of the object and the resolution along the Y-axis.

![Figure 1.3: Perspective distortion, homography mapping, lens distortion](image)

After this it is necessary to find out how the curve seen on the sensor maps to the real curve in the laser plane. A homography is a plane to plane projective transformation that maps straight lines on one plane to straight lines on another plane (further explained in section 2.4.2). This means that a homography accurately represent the transformation between the camera sensor and the laser
plane. In short this step consists of finding a robust estimation of this homography. Finally, in order to get a good measurement, the lens used in the camera must be calibrated. To do this a good enough distortion model should be found and its parameters fitted to the lens. This model can after this be used to undistort the captured images.

Take note that a box shaped object, of the type in Figure 1.3, is unsuitable to use as a calibration object in a real application due to, for example, laser occlusion. It is here used for illustration purposes only.

1.3 Purpose

In this thesis a method for calibrating a sheet of light laser camera setup is to be found. A calibration object with known dimensions is used, but it can be scanned at nearly arbitrary rotation, position and translation direction relative the laser plane coordinate system, seen in Figure 1.2.

The calibration method is needed to make calibration possible in small FOV where the method previously used is unsuitable. The previous method uses measurements in the laser plane that are obtained by manual movement of a calibration object by an operator. When scaling this solution down, three main problems arise. The calibration object has to be positioned very carefully to get a good calibration, which is hard to do by hand. Because of its design the calibration object cannot be manufactured at this small scale without defects. At this scale it is common with enclosed camera systems, it might not be possible to access the laser plane to perform the calibration.

The conditions for the calibration method is that the calibration object is low cost, simple to use and possible to use with the laser as the only light source. It should also handle all unknowns in the surrounding geometry (perspective transformation and lens distortion).

The implementation is, because of these properties, suitable for calibration of systems for inspection of electrical components and other small items. However, it should also be possible to extend the method to other FOV.

1.4 Outline

The remainder of the report is divided into six parts excluding appendices; Background theory, Calibration Method, Experiments, Results, Discussion and Conclusions.

The first chapter holds fundamental background theory needed to understand the rest of the report, mainly related to camera representation, homogeneous coordinates and the transformations implemented. The second chapter describes the proposed calibration method in detail and specifies all design choices made. The third chapter defines the experiments conducted to evaluate the system and the fourth chapter presents the results from these. Before coming to the final chapter, a chapter is left for general discussion. In the final chapter the results are discussed and conclusions are made.
This section introduces basic mathematical concepts and models onto which this thesis work is built. The focus is on how everything works and is connected, compared to sections further into this document that focuses on the implementation and how the application was constructed.

### 2.1 Mathematical Definitions

When using transformations and projections it is almost always convenient to make use of projective spaces. What extending to use projective spaces essentially means, for the purpose of this thesis, is that coordinates are extended from a normal representations in Cartesian coordinates to homogeneous coordinates by adding an extra component to the coordinate. This as in

\[
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2 \\
s
\end{bmatrix} = s \begin{bmatrix}
\tilde{x}_1/s \\
\tilde{x}_2/s \\
1
\end{bmatrix} = s \begin{bmatrix}
x_1 \\
x_2 \\
1
\end{bmatrix} \sim \begin{bmatrix}
x_1 \\
x_2 \\
1
\end{bmatrix}
\]

for the case with the two-dimensional Cartesian coordinate \((x_1 x_2)^T\). The same extension is used in the three-dimensional case. The \(s\) added to the definition of the homogeneous coordinates represents an unknown scale factor. In projective spaces homogeneous coordinates that only differ by a constant factor are considered equal in a special sense, denoted by the similarity sign \((\sim)\).

Apart from giving the coordinates another component this representation provides a way of effectively representing points at infinity (points with \(s\) equal to zero) and also simplifies transformations. For example, it is possible to integrate a translation into the same matrix as a rotation or another transformation. In the following 2D example the transformed coordinate \(x_{ct}\) of a Cartesian coordinate
\( \mathbf{x}_c \) is given by

\[
\mathbf{x}_{ct} = \mathbf{A}_c \mathbf{x}_c + \mathbf{t}_c = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}.
\] (2.2)

Rewriting using homogeneous coordinates and extending the surrounding transformation to the correct dimension in the following way yields the shorter and more compact notation

\[
\mathbf{x}_t = \mathbf{A}_h \mathbf{x} + \mathbf{t}_h = \begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ 0 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = \mathbf{A} \mathbf{x}.
\] (2.3)

The concepts briefly introduced here are more thoroughly explained by Bloomenthal[2] and Hartley[5] in their respective works.

In this thesis report coordinates printed using square brackets ("[ · ]") represent homogeneous coordinates and coordinates inside regular parentheses ("( · )") are Cartesian coordinates. In general homogeneous coordinates are normalised to have the last component equal to one. If nothing else is mentioned, it is assumed that the homogeneous coordinates have been normalised, and therefore the explicit writing of the final component is omitted when it is unimportant for the surrounding context.

### 2.1.1 The Pinhole Camera Model

There are more than one model for how a camera obtains its image, just as there are different kinds of cameras. For the most common cameras however, the normal approach is to model the camera as an ideal pinhole camera and afterwards compensate for any deviations from this model (see for example Heikkilä[7]).

The ideal pinhole camera consists of two essential parts; the aperture, that lets a small amount of light enter the camera, and an image plane, onto which the light that enters the camera is captured. From the aperture to the normalised image plane a line can be drawn that is orthogonal to the image plane. The point at which this line connects with the image plane is known as the principal point. The distance from the principal point to the aperture is known as the focal length \( f \) of the pinhole camera.

In Figure 2.1 the model is presented visually. The optical, or principal, axis is in this image shown as \( C_3 \). Apart from the camera model in itself a point in 3D inside the laser plane is shown mapped to the normalised image plane. By observing this model it is possible to understand that there are multiple points in the world that maps to the same point in the plane. Or more precisely, all points along the line from the normalised image point \( [X_1, X_2]^T \) through the aperture is projected onto the same image point \( [X_1, X_2]^T \). In the special case this thesis considers, only world points that lie in the laser plane are observed, but the laser plane and the normalised image plane are in general not parallel.

Furthermore, the scaling of real world objects depend on the size of \( f \) and the distance from the object to the aperture. However, in this thesis normalised image coordinates are assumed, and this means that \( f \) is set to one. The projection
of a 3D point to the ideal image plane can in total be described by

\[
\begin{pmatrix}
X_1 \\
X_2
\end{pmatrix} = -\frac{1}{C_3} \begin{pmatrix}
C_1 \\
C_2
\end{pmatrix} = -f \begin{pmatrix}
C_1 \\
C_2
\end{pmatrix}.
\] (2.4)

Different values of the focal length \( f \) will be compensated for by using the intrinsic camera parameters.

### 2.1.2 Intrinsic Camera Parameters

The intrinsic camera parameters are parameters that describe the image formation inside the camera. These parameters are used to transform from the ideal image plane of the pinhole camera model (metric) to real image coordinates (pixels). A matrix is normally constructed as

\[
K = \begin{bmatrix}
\alpha & \gamma & u_0 \\
0 & \beta & v_0 \\
0 & 0 & 1
\end{bmatrix}.
\] (2.5)

In this version of \( K \), \( \alpha = f m_u \) and \( \beta = f m_v \) describe the scaling in the \( u \) and \( v \) direction of the sensor and the focal length \( f \) of the camera. The skew of the sensor is determined by the factor \( \gamma \). The remaining \( u_0 \) and \( v_0 \) hold information about the where the optical axis crosses the image plane in image coordinates.

For the SICK IVP Ranger, this transformation can be shown to be a shape preserving transformation. In the Ranger camera the pixels are quadratic and placed in an orthogonal grid, which is why these parameters may be set to fixed values as

\[
K = \begin{bmatrix}
\alpha & 0 & u_0 \\
0 & \alpha & v_0 \\
0 & 0 & 1
\end{bmatrix}.
\] (2.6)

The remaining components of the \( K \)-matrix now constitutes a scaling and a translation, also known as shape preserving transformations.
In general the transformation from normalised image plane coordinates \( \mathbf{X} \) (metric) to observed real image plane coordinates \( \mathbf{U} \) (pixels) is performed by

\[
\begin{bmatrix}
u \\ v \\ 1
\end{bmatrix} =
\begin{bmatrix}
\alpha & 0 & u_0 \\
0 & \alpha & v_0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
1
\end{bmatrix} = \mathbf{KX},
\]

(2.7)

where the real image coordinates in pixels is found in \( \mathbf{U} = (u, v)^T \).

## 2.2 Plane Computation

When computing equations for planes two methods have been used. The first approach uses only the minimal number of points needed, three, and finds the plane that contains all of these points. The second finds the optimal plane in a least squares sense when more than three points are considered.

### 2.2.1 Plane from Three Points

With three selected points \( \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \) the plane equation is obtained by first constructing two vectors in the plane

\[
\mathbf{u} = \mathbf{x}_1 - \mathbf{x}_2, \quad \mathbf{v} = \mathbf{x}_3 - \mathbf{x}_2.
\]

(2.8)

The normal is then found by taking the cross product of these vectors. In this example the normal is also normalised to unit length.

\[
\mathbf{n} = \frac{\mathbf{u} \times \mathbf{v}}{\|\mathbf{u} \times \mathbf{v}\|}.
\]

(2.9)

The requirement for this to hold is that the points are non-collinear. From the normal of a plane the plane equation for a point \( \mathbf{x}_i = (x_i, y_i, z_i)^T \) can be written as

\[
\hat{n}_1 x_i + \hat{n}_2 y_i + \hat{n}_3 z_i = D
\]

(2.10)

or by using homogeneous coordinates \( \mathbf{x} = [x_i, y_i, z_i]^T \)

\[
\mathbf{P} \mathbf{x} = \begin{bmatrix}
\hat{n}_1 & \hat{n}_2 & \hat{n}_3 & -D
\end{bmatrix} \mathbf{x} = 0.
\]

(2.11)

Simply inserting one of the original points gives the constant \( D \) and the plane coefficients \( \mathbf{P} \) are determined.

### 2.2.2 Plane from Least Squares Fitting

An alternative to the geometrical solution above is to find the eigenvectors for a set of points \( \mathbf{p}_i, i = 1, ..., N \), considered to lie in the plane (a minimum of three points is assumed). Form the matrix holding all \( N \) point coordinates

\[
\mathbf{X} = \begin{bmatrix}
| & | & \cdots & | \\
\mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_N \\
| & | & \cdots & |
\end{bmatrix}.
\]

(2.12)
Form the covariance matrix by removing the mean point coordinate \( p_c \) and performing an inner matrix multiplication

\[
C = [X - p_c][X - p_c]^T.
\]  

(2.13)

The eigenvector corresponding to the smallest eigenvalue of \( C \) will be the normal of the plane in least squares sense. What is actually received when performing these computations is a new orthogonal coordinate system of three vectors with the maximum variance of the point coordinates aligned with the eigenvector corresponding to the largest eigenvalue. The direction corresponding to the smallest eigenvalue is the direction with the lowest variance, and thus, it is orthogonal to a plane approximately containing all points. This method of finding a new base of vectors is also known as principal component analysis[6]. Given that the points largely lie in the same plane, the final component \( D \) of the plane equation in (2.10) and (2.11) can be found by taking

\[
D = \text{mean}(\bar{n}^T X),
\]  

(2.14)

or

\[
D = \text{median}(\bar{n}^T X),
\]  

(2.15)

where \( X \) is the set of all 3D points approximately in the plane. One exception, to when this method is needed to determine \( D \), is when at least one point is known to lie perfectly in the plane. In this case, it is enough to use only one of the points known to be in the plane, to find \( D \). In this thesis (2.15) is used instead of the true least squares (2.14), because it is more robust to noisy data. For each plane the coefficients are saved as in (2.11).

### 2.3 Computation of Intersection Points

Given a set of plane equations corresponding to planes that intersect at a point, it is possible to find the point of intersection. If the normals of the planes are stacked in a matrix in the following way

\[
A = \begin{bmatrix}
- & n_1 & - \\
- & n_2 & - \\
\vdots & \vdots & \vdots \\
- & n_N & - 
\end{bmatrix},
\]  

(2.16)

a few things can be said about \( A \) before an intersection point is found. The case we assume to have is when the rank of \( A \) is three. If the rank of \( A \) is three there exists an intersection point for the planes as the matrix defines three distinct planes.

Should the rank be one, all plane equations are coplanar. This means that they approximately represent the same plane and that for this combination of planes an infinite number of points could be found on this plane. If the rank instead is two there are two distinct planes resulting in an infinite number of solutions.
along the intersection line of these two planes. Hence, if the rank is below three, less than three planes are distinct, and there is no point of intersection to find. In case of \( N > 3 \) the planes in the set generally could connect at more than one point. However, for the cases considered in this thesis, the planes should theoretically intersect at only one point, allowing for interpolation in situations where that is not completely true in reality.

The remaining part of the plane equations are the distance to origin. The standard formulation of the plane equations can be written on matrix form using

\[
\mathbf{A} \mathbf{p} = \mathbf{D},
\]

where \( \mathbf{D} = (D_1, D_2, ..., D_N)^T \) hold the distance measurement for the corresponding plane and \( \mathbf{p} \) is a point approximately in all planes.

Solving for the rank three case simply entails computing

\[
\mathbf{p} = \mathbf{A}^{-1} \mathbf{D},
\]

or for cases where \( N > 3 \)

\[
\mathbf{p} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{D},
\]

receiving the intersection point coordinate in \( \mathbf{p} \). In the implementation proposed in this thesis, all sets of planes are considered to only connect at one point and thus the computation of (2.18) and (2.19) holds.

2.4 Transformation Model

In this section the transformation model used in this thesis to transform between points in world coordinates and points in the pseudo range image is presented in detail.

At first it is necessary to model the mapping from the real world coordinates to coordinates in the laser plane. The coordinates observed in the laser plane is then mapped to the real image plane using a homography. Lens distortion is finally added in the real image plane and the final transformation from the world to the pseudo range image is known. Computing the inverse solves the calibration problem for the used configuration.

2.4.1 Object to Laser Plane Coordinates

How an object is represented in the laser plane can be explained using a skew transformation. The problem is complicated by the fact that the calibration object could be translated and rotated arbitrarily. As only the object size and dimensions are predetermined, there is no way to know the position or orientation of the object beforehand. The object pose is modelled using

\[
\mathbf{Z} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tilde{\mathbf{Z}},
\]
where $R$ is a three-dimensional rotation consisting of a yaw and pitch, $t = d \cdot \hat{n}$ a vector representing the distance orthogonal to the laser plane, $\hat{Z}$ hold the calibration object coordinates and $Z$ contain the coordinates in a coordinate system aligned with the laser plane.

It is only necessary to allow two DOF (degrees of freedom) for $R$ as the final component, the roll, can be handled by the homography presented in the next section 2.4.2. Note that neither $R$ nor $t$ are necessary in the final calibration, they are only needed in order to be able to complete the calibration process. However the knowledge of these variables could of course be utilised to visualise data better.

A skew transformation dependant on the translation direction of the object relative the laser plane accurately describes the final mapping to the laser plane. In general two sets of data are obtained from one homogeneous 3D point $Z$ and the normalised translation direction of the object in laser plane coordinates, $\hat{n} = (n_x, n_y, n_z)^T$ (see Figure 1.2).

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -n_x/n_y & 0 & 0 \\ 0 & -n_y/n_y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ 1 \end{bmatrix} = PZ,$$  \tag{2.21}

$$S = \frac{1}{n_y r} Z_2 = T_s Z_2,$$  \tag{2.22}

where $Y$ is a point in the laser plane and $S$ the distance from the point to the laser plane in the received pseudo range image from the uncalibrated setup. The variable $r$ is in this case related to how often an image is taken (either related to ticks from an encoder or time instances). In this thesis data will be captured with an encoder and therefore the variable $r$ will hold the estimated length of one such tick.

### 2.4.2 Homography

A homography is a projective mapping from one plane to another, that maps points and lines on a plane to corresponding points and lines on another plane. A homography can be estimated from either point or line correspondences or a combination of both. It is described by

$$U \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ 1 \end{bmatrix} = HY,$$  \tag{2.23}

$$L_y \sim H^T L_u,$$  \tag{2.24}

where $H$ is the homography matrix, $Y$ hold points in the laser plane and $U$ hold points in the real image plane. In (2.24) $L_y, L_u$ are the corresponding lines in the same planes, note that the dual relation is used for lines compared to points.
The mapping from the laser plane $[Y_1, Y_2]^T$ to the normalised image plane $[X_1, X_2]^T$ constitutes a homography. Because $K$ in (2.6) also can be generalised as a homography, the combined mapping from the laser plane $[Y_1, Y_2]^T$ to the real image plane $(u, v)^T$, also constitutes a homography. Hence, one homography like the one in (2.23) replaces $K$ in (2.6) and the homography from the laser plane to the normalised image plane.

One way of determining $H$ is by direct linear transformation also known as DLT. In this thesis the DLT is used to determine the homography from points. The process involved is explained in further detail in Appendix A. The homography can only be determined up to an unknown scale factor and therefore have 8 degrees of freedom (DOF). Because of this it is enough to measure a minimum of eight points (four in each corresponding plane) to successfully determine a homography. For more information on homographies see Hartley [5].

2.4.3 Lens Distortion

Normally lens distortion would be removed in the normalised image plane before the use of $K$, but after the use of a general homography $H'$. Because $K$ is shape preserving it is allowed to move $K$ to the other side of the lens distortion model, placing it next to $H'$. This is what makes it possible to combine $K$ and $H'$, even when a lens model is used. As straight lines transformed by $KH' = H$ are projected onto other straight lines rectifying lines in the real image plane will result in compensating the true distortion. This allows for lens distortion compensation directly in the real image plane instead of in the normalised image plane.

The chosen distortion model is the same as in Anderssons thesis [1] and many others [7], [11]. This model has proven sufficient and easy enough to implement. In this presentation of the model, $u$ and $v$ represent a pixel coordinate in the real image plane, and boldface $u$ holds both $u$ and $v$.\[
\begin{align*}
    \mathbf{u} &= \mathbf{\tilde{u}} + \mathcal{F}_D(\mathbf{\tilde{u}}, \delta), \\
    \mathcal{F}_D(\mathbf{\tilde{u}}, \delta) &= \left[ \mathbf{\hat{u}}(k_1 r^2 + k_2 r^4) + (2p_1 \mathbf{\hat{u}} \mathbf{\hat{v}} + p_2 (r^2 + 2\mathbf{\hat{u}}^2)) \right] \\
    \mathbf{\hat{u}} &= (\mathbf{\hat{u}}, \mathbf{\hat{v}}) - (\mathbf{\tilde{u}}_c, \mathbf{\tilde{v}}_c) \quad \text{and} \quad r = \sqrt{\mathbf{\hat{u}}^2 + \mathbf{\hat{v}}^2}. \quad (2.25)
\end{align*}
\]

where $(\mathbf{\hat{u}}, \mathbf{\hat{v}}) = (\mathbf{\tilde{u}}, \mathbf{\tilde{v}}) - (\mathbf{\tilde{u}}_c, \mathbf{\tilde{v}}_c)$ and $r = \sqrt{\mathbf{\hat{u}}^2 + \mathbf{\hat{v}}^2}$. The parameters that need to be determined to correct for the distortion are contained in $\delta = (\mathbf{\tilde{u}}_c, \mathbf{\tilde{v}}_c, k_1, k_2, p_1, p_2)$.

The transformation in (2.25) takes a distorted image point, $\mathbf{\tilde{u}}$, and transforms it into an undistorted point, $u$. The introduced variables $\mathbf{\tilde{u}}_c$, $\mathbf{\tilde{v}}_c$ represent the distortion center in the image, $k_1$, $k_2$ model the radial distortion dependent on distance to this center, and $p_1$, $p_2$ model the tangential distortion also dependant on distance to the distortion center.

In (2.26) a total of six parameters are used. In testing it was observed that the parameters $p_1$ and $p_2$ usually were very small and in addition could cause problems in finding an optimal homography (as these parameters can model an unwanted skew in the image plane). To counteract this $p_1$ and $p_2$ were disabled. They have still been used for constructing synthetic data and can be enabled if
the application so demands. The reduced lens distortion model consist of

\[
F_D(\tilde{u}, \delta) = \begin{bmatrix}
\hat{u}(k_1 r^2 + k_2 r^4) \\
\hat{v}(k_1 r^2 + k_2 r^4)
\end{bmatrix},
\]

(2.27)

where the used parameters are the same as in the full lens distortion model presented in (2.26). This is also the same lens model parameters that are used by Zhang in his famous calibration method [13].

The writing of the distortion transformation is shortened by the following notation

\[
\tilde{U} \sim D(U),
\]

(2.28)

where the real image plane coordinate \( U \) is distorted to \( \tilde{U} \). Hence, \( D(\cdot) \) denotes the inverse of (2.25) using the modified version of the lens model in (2.27).
The proposed solution to the calibration problem is to use a 3D scan of a known object and from that determine all the necessary measurements needed to accurately calibrate the camera. This method of calibration is referred to as a scanning method.

In this implementation the object is assumed to move through the laser field in a constant direction. In most applications the velocity is also constant. Because of this the solution proposed in this thesis only takes into account situations were there is a constant propagation speed or a movement encoder is available.

### 3.1 Calibration Object

The object used for calibration has been selected in regards to how well it is possible to identify its features, how easy it is to manufacture and how much information can be determined from a minimum number of scans. In the case of the selected object only one scan should be sufficient.

The object is composed of multiple planar surfaces. The advantage of using planar surfaces compared to feature points is that there are multiple measurements that can be used to robustly estimate a plane for each surface. A robust estimation of the planes allow for high precision detection and computation of the intersection points for every group of at least three surfaces.

The object chosen in this thesis consists of eleven detectable planes from which eleven corner points can be extracted. The corner points are placed at different positions along all axes, providing samples of data with known position relative each other in full 3D. For the height component three different levels are available; base plane, mid and top level (see Figure 3.1 and Appendix B). Apart from being at different heights the points have been positioned in such a way that the planar surfaces between them are not too steep. This in order to get enough
reflected laser light back to the camera from all parts of the object. In addition to even reflectivity, the object was also designed to allow computation of lens distortion parameters, by providing measurements in most parts of the FOV.

### 3.2 Calibration Process Overview

The calibration process is in this section split into three problem areas (see Figure 3.2). The first problem is to find the areas in the pseudo range image (see section 1.1) that represent planes. To do this a 2D segmentation method have been chosen for its robustness and simplicity. When the planar areas are known, plane equations can be formed using a least squares approach. The calculated planes in the uncalibrated dataset must then be matched to the planes on the calibration object. When this is done, the algorithm knows the location of points on the calibration object in scanned range data, and the optimisation to solve the geometric relations can commence.

![Figure 3.2: Calibration Overview.](image)

### 3.3 Plane Extraction

Plane detection is the part of the algorithm that detects where the planes on the calibration object are positioned in the uncalibrated range image and what equations they have. The results from these computations are the center of mass.
for each plane and the plane equations computed directly from the uncalibrated range image.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{plane_detection_process.png}
\caption{Plane detection process.}
\end{figure}

### 3.3.1 Segmentation

Planes are detected using 2D spatial filtering[4] of the range image. Simply consider the range value in the same way as an intensity value in an ordinary image. The key to detecting the planar areas is finding the edges between them. In essence this is done by computing the second order derivatives of the range image. The received data is not completely free from noise and because of this the method applied must be robust to handle this.

A first attempt was to compute the second order derivatives of the image using a smoothed version of the Sobel filter. In initial testing this unfortunately resulted in noisy images that could not be thresholded in a good way. The solution that is now used, demonstrated in Figure 3.3 and Algorithm 3.1, still involves the pure second order derivative, but computed from the second order derivative of an ideal gauss kernel. See Figure 3.4 for an example of derivation results.

#### 3.1 Algorithm: Obtaining the Binary Image

1. Choose a standard deviation \( \sigma \).
2. The size \( s \) of the derivative filter is determined by \( s = 1 + 2 \left\lceil \sigma \sqrt{2} \ln(100) \right\rceil \)
   and a vector \( x \) is constructed according to \( x = -\left\lfloor \frac{s}{2} \right\rfloor, \left\lceil \frac{s}{2} \right\rceil + 1, \ldots, \left\lfloor \frac{s}{2} \right\rfloor \).
3. Define the Gaussian kernel \( K_G = e^{\frac{-x^2}{2\sigma^2}} \).
4. Define the first order derivative of this kernel \( K_{1:der} = \frac{x}{\sigma^2} K_G \).
5. Define the second order derivative of this kernel \( K_{2:der} = \frac{x^2-a^2}{\sigma^4} K_G \).
6. Compute the derivative images from the pseudo range image \( f \) according to; \( f_{xx} = (f * K_G^T) * K_{2:der}, f_{yy} = (f * K_G) * K_{2:der}^T, f_{xy} = (f * K_{1:der}) * K_{1:der}^T \).
7. Compute the combination of all derivatives \( I = \sqrt{f_{xx}^2 + f_{yy}^2 + f_{xy}^2} \).

8. Threshold \( I \) by a factor of the median of \( I \) to receive the binary image.

### 3.3.2 Labelling

The labelling takes the binary image with extracted planar areas and numbers the planes, making them possible to index. The implemented method is a two pass method that first produces a list with more labels than are needed. In the second pass the connected components[4] are found and the image is relabelled. The code for labelling the planar areas was not written as a part of this master’s thesis project and in-house code is used for this process. In Figure 3.4 the full segmentation and labelling process is illustrated.

![Figure 3.4: Segmentation overview.](image)

- **(a)** Original range image obtained from the camera system.
- **(b)** Image from computation of the magnitude of the combined second order derivatives.
- **(c)** Binary image from thresholding the magnitude image with a factor of the median.
- **(d)** Result after labelling. All connected regions have been identified.

### 3.4 Plane Matching

The objective of this segment is to extract the corner points that are needed in order to calibrate the camera setup. This is divided in two modules. First a number of plane equations are determined from pseudo range data. These are then matched to the calibration object model in world coordinates. When the correspondence between planes are known the sought corner points can be obtained.
3.4 Plane Matching

3.4.1 Plane Computation

Before anything else is done the base plane is identified and all planes outside of it are discarded. From the labelled 2D image it is possible to determine which plane is likely to be the base plane simply by measuring the bounding box of each labelled region. To increase the robustness of this method, the area of region bounding boxes are multiplied with the area of the corresponding region, and the results from all planes are placed in a candidate list and sorted. Using the area as well as the bounding box makes the method less likely to pick spurious background segments as base plane.

The last step before choosing a base plane is to make sure that the correct number of planes exist inside the plane of the largest size. If this is not the case the largest plane is removed and the second largest plane is picked from the candidate list and checked. When the largest plane that contains the correct amount of planar areas is found, this is selected and sorted as the base plane. All planar areas detected outside of the base plane are discarded. The identification of the base plane is critical in order for the remainder of the calibration to work.

To estimate plane equations and find the center of mass for the planes, points in 3D are needed. The whole image is reconfigured on the form of a 2D matrix with X, Y and pseudo range coordinates in the first, second and third row respectively.

The center of mass is found by computing the mean of this matrix indexed by each label separately for all planes, but the base plane. The base plane instead gets its center point from the combined set of all other planes. This solves the problem that otherwise would occur when parts of the base plane are occluded.

The plane equations are found by selecting the subset from the 3D point matrix corresponding to a plane, in the same way as for mass calculations, and then by performing the steps explained under section 2.2.2 for each plane separately.

The steps performed in this part are further clarified by Figure 3.5.

**Figure 3.5:** Plane computation.
3.4.2 Matching and Corner Extraction

In order to be able to compute the intersection points of the planes (corner points of the calibration object in the pseudo range image), which are needed to compute the homography, it is necessary to know which planes neighbour each point.

To make the explanation of this process easier to follow we define two sets of $N$ center of mass points $P_w$ and $P_r$, where $P_w$ holds the sorted center points of the calibration object in known metric world coordinates and $P_r$ holds unsorted center points derived from the pseudo range image. The only point that for sure is known to match up in these sets are the last one, known to correspond to the base plane of the calibration object. For short the subsets excluding the last plane will be referred to as $P_{ws}$ and $P_{rs}$ respectively.

From set $P_{ws}$ three points are selected that lie on different positions on the object, making them well defined in relation to each other and the base plane (currently midpoints corresponding to plane 2, 5 and 8 in Appendix B are used). In the implementation proposed, this choice is made once for the whole algorithm, but it could be extended to iteratively change in order to allow for matching even if some planes are not detected. The three mass center points selected from $P_{ws}$ are, together with the base plane mass center, the points that will be matched to some of the center points in $P_r$ and are denoted by $P_{wsb}$. The points that should be matched to $P_{wsb}$ are selected from $P_{rs}$ iteratively. Any set that contains three points from $P_{rs}$ and the base plane of $P_r$ is denoted $P_{rsb}$.

By performing the steps in Figure 3.6 the best correspondences between plane midpoints are found. The transformation $A$ is formed by applying gauss elimination on

$$AP_{wsb} = P_{rsb},$$  

with the current selection of points in $P_{rsb}$. Using this $A$ all points in world coordinates are transformed according to

$$P_w' = AP_w.$$  

A distance map is formed with the distances from one point in the first set, $P_w'$, to each point in the other set, $P_r$, in its columns. The order is given by the selected row index per column. Hence, if the first column, third row is selected the first point in set one corresponds to the third point in set two. It is essential that only one match is selected for each column and row and this puts the constraint on the problem. The best total match is given by the selection that minimises the sum of all distances selected.

The solution to this assignment problem is found using the polynomial time Hungarian algorithm[8]. To compute the Hungarian algorithm external code was used so no complete explanation on the implementation will be presented. In essence the lowest value in each row is subtracted from the other components in the same row. This results in a matrix with zeros in at least one position per row. An attempt is made to select all zeros. If some columns have more than one zero, or equivalently some rows have no zeros the algorithm does the same subtractions with a transposed version of the distance matrix. This will in most situations result in one zero element per row and column. If this is not the case
3.4 Plane Matching

Start

Plane midpoints, equations

Select the solution with the lowest error

Reorder all detected data according to the sort order

Estimate a rotation $R_{est}$ from optimally sorted $P'_f$ and $P_w$

Compute the position of all corner points in uncalibrated data

Continue

Corner points, rotation estimate

Match Loop

Select an untested combination of 3 points belonging to $P_{rs}$ and the point belonging to the baseplane of $P_r$, forming $P_{rsb}$

Find a 3D transformation $A$ from the predefined 4 points $P_{wsb}$ to the selected 4 points $P_{rsb}$

Transform all points in $P_w$ with $A$ and form a $N \times N$ matrix with distances between the two sets $P_r$ and $P'_w = AP_w$

Save solution; transformation, sort order, error given this order

Find the best selection of matches between $P'_w$ and $P_r$ using the Hungarian algorithm

All tested?

Yes

No

Figure 3.6: Plane matching and corner points computation.
a more involved finalisation of the algorithm is necessary, see the original paper by Kuhn[8] for more details.

![Base plane identified](image1)

![Selected matches](image2)

![Sorted labels](image3)

(a) Image showing the remaining planes after the base plane has been identified.

(b) Mid points extracted and matched. The color represents the chosen correspondence. Only the points that were used to estimate $A$ are a perfect match.

(c) Planes sorted to match the order defined on the calibration object. Intersection points for the planes are marked with white crosses.

Figure 3.7: Matching overview.

Take note that the best choice of $A$ is the one that gives a transformation that minimises the total distance between all points, not just the ones selected to estimate $A$. The selected matches and results from matching is shown in Figure 3.7. The points that were used to estimate the match transformation $A$ match up perfectly, while the other points do not. The estimated transformation $A$ will almost never solve the matching perfectly as the full transformation from the pseudo range coordinates to world coordinates is too complicated to solve using one simple linear transformation.

Given these correspondences, a rotation that in least squares sense optimally fits all points to their matches is estimated and saved as a start solution for later optimisations steps. The rotation is found using a SVD approach. First the two matrices

$$C_r = P_r' - \frac{\sum_{i=1}^{N} P_r'(i)}{N},$$  

(3.3)

$$C_w = P_w - \frac{\sum_{i=1}^{N} P_w(i)}{N},$$  

(3.4)
are found, where $P_i''$ hold the sorted center points in the pseudo range image, $i$ is the point index in the corresponding set and hence the subtracted term contains the vector mean of the set. Singular value decomposition is performed on the inner matrix product of $C_r$ and $C_w$ according to

$$[U, S, V] = \text{svd}(C_rC_w^T),$$

where the received matrices $U$ and $V$ contain the left and right singular vectors of the combined matrix $C_rC_w^T$ in their respective columns. If the determinant of $V$ is negative the sign of the final column of $V$ is changed so that the determinant is one. The rotation matrix is finally formed by

$$R_{est} = VU^T.$$  \hspace{1cm} (3.6)

Given that the correspondence between the planes on the detected object and the ground truth object is known it is possible to simply use a look-up table with information about the planes that connect to any given point. This table is presented together with the definition of the calibration object in Appendix B. Given that the planes are known to intersect in a point, the intersection point can be found by using the method described in section 2.3.

### 3.5 Transformation Summary

The image acquisition can be summarised by equations (2.20), (2.21), (2.23), (2.7) and (2.28). From these the following total transformation is produced

$$U \sim D\left(HP\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}\tilde{Z}\right), \hspace{1cm} (3.7)$$

where $\tilde{Z} = [\tilde{Z}_1, \tilde{Z}_2, \tilde{Z}_3]^T$ are homogeneous 3D points that are defined in the calibration object coordinate system and $U = (u, v)^T$ are the pixel coordinates on the sensor (i.e. the real image plane coordinates). As previously mentioned $R$ and $t$ determine the positions and orientation of the calibration object relative the laser plane, $P$ is the skew transformation onto the laser plane, $H$ is the homography from laser plane to real image plane and $D$ the distortion function of the lens.

Also recollect that equation (2.22) provides additional information about the dimension that is lost when transforming using the skew matrix in (3.7). This transformation can be rewritten in the following way

$$S = \begin{bmatrix} 0 & T_s \\ 0 & 0 \end{bmatrix}\begin{bmatrix} R & t \end{bmatrix}\tilde{Z}$$ \hspace{1cm} (3.8)

where $T_s$ is a parameter describing the resolution along the axis perpendicular to the laser plane. This is unaffected by the homography and is used to perform the inversion from laser plane coordinates $Y$ to object coordinates $Z$.

Inverting these transformations together thus constitutes the process that should be performed in order to obtain calibrated data. First consider inverting points in the image up to points in the laser plane

$$Y \sim H^{-1}D^{-1}(U).$$ \hspace{1cm} (3.9)
Scaling $Y$ to normalised homogeneous coordinates and inverting the skew transformation gives the first two components of the laser plane aligned coordinates

$$
\begin{bmatrix}
Z_1 \\
Z_3
\end{bmatrix}
= P^{-1} Y.
\tag{3.10}
$$

Combining the result with $S$ from (3.8) yields the full reconstruction of $Z$ by inserting

$$
\begin{bmatrix}
Z_1 \\
S/T_s \\
Z_3
\end{bmatrix}
= \begin{bmatrix}
Z_1 \\
Z_2 \\
Z_3
\end{bmatrix}
= Z.
\tag{3.11}
$$

The final and optional transformation places the coordinate system in the origin of the calibration object and rotates it to be facing in the direction the calibration object was placed when scanned

$$
\tilde{Z} = \begin{bmatrix}
R \\
0^T \\
1
\end{bmatrix}^{-1} = \begin{bmatrix}
R^T \\
0^T \\
-R^T t
\end{bmatrix} Z.
\tag{3.12}
$$

The problem to optimise the total transformation from world coordinates to the received coordinates in a pseudo range image have 14 DOF (degrees of freedom), when all geometric parameters are included in the problem formulation. These DOF stem from rotation (2 DOF), translation direction (2 DOF), tick resolution (1 DOF), distance to object origin (1 DOF) and a homography (8 DOF). In later stages when the lens distortion model is used an additional 4 DOF are added to the problem formulation, 2 DOF from the distortion center coordinate and 2 DOF from the radial distortion parameters.

### 3.6 Optimisation

Up to this point almost nothing is known about the geometry or homography and all results are given purely by the pseudo range image. The optimisation step is an iterative process that step by step adds complexity to the calibration problem.

Here the transformation described in section 2.4 and 3.5, is used multiple times in order to be able to estimate the homography and to continuously measure the error in world coordinates. Both the optic lens distortion error and the geometric error is minimised by allowing the function *fminsearch* in MATLAB to change the model parameters in both the geometric part of the model and the lens distortion model. The only exception to this is the homography, which at first is recomputed from the other parameters.

The full process of optimising is described in Algorithm 3.2 and visualised in Figure 3.8.
3.6 Optimisation

![Optimisation Diagram]

Figure 3.8: Optimisation Overview.

3.2 Algorithm: Optimisation Stages

1. Set up the parameters needed to perform the optimisation given by what is known from the previous steps of the calibration process.

2. The algorithm is run over only six parameters. Given these parameters it is possible to re-estimate the homography in each iteration which reduces the complexity of the geometric model by eight parameters. This stage uses the geometric error (Section 3.6.2 and Figure 3.9) which yields the optimal calibration in regards to the fit of the calibration object corner points.

3. The lens distortion model is added resulting in a total of ten parameters. Still re-estimating the homography allows for a safer optimisation and faster convergence for the total solution. In this stage the lens error measure is used for optimisation (Section 3.6.3 and Figure 3.10). The result is a solution that optimally fits a sub-selection of all points in the pseudo range image to the plane equations of the calibration object.

4. In the final stage all parameters are let loose including the homography. The total optimisation problem now consists of 18 parameters to optimise. This final stage optimise against a simplified version of the lens error (Section 3.6.3 and Figure 3.11). The reason is that when the homography is independent of the corner points on the calibration object in the pseudo range image, errors in their estimated position are less likely to affect the final calibration.
3.6.1 Create Start Solution

The parameters that have to be set before optimisation are summarised by Table 3.1. Some of the parameters are particularly easy to estimate and some are more difficult.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Rotation matrix describing the rotation of the calibration object.</td>
<td>$[3 \times 3]$</td>
</tr>
<tr>
<td>r</td>
<td>Tick resolution, a scalar that models the sample frequency for each slice.</td>
<td>$[1 \times 1]$</td>
</tr>
<tr>
<td>$\hat{n}$</td>
<td>Normal vector for the motion direction.</td>
<td>$[1 \times 3]$</td>
</tr>
<tr>
<td>d</td>
<td>Distance from laser plane to object origin along $\hat{n}$.</td>
<td>$[1 \times 1]$</td>
</tr>
<tr>
<td>c</td>
<td>Center coordinates of the lens distortion model $[\tilde{u}_k, \tilde{v}_k]$.</td>
<td>$[1 \times 2]$</td>
</tr>
<tr>
<td>k</td>
<td>Radial distortion parameters $[k_1, k_2]$.</td>
<td>$[1 \times 2]$</td>
</tr>
</tbody>
</table>

**Table 3.1:** Overview of the parameters needed to start the optimisation. Parameters below the dashed line belong to the lens model.

The rotation matrix, $R$, is known to only have two DOF. It can rotate the calibration object in the world around the $\tilde{Z}_1$- and $\tilde{Z}_3$-axis (yaw and pitch), but not the $\tilde{Z}_2$-axis (roll). This is because a rotation along this axis can also be described by another rotation around the normal vector $\hat{n}$, that is incorporated in the homography. A full rotation estimate, $R_{\text{est}}$, is provided from the object matching step (see section 3.4.2). When preparing a start solution this rotation is reduced to only incorporate the yaw and pitch angles. This has to be done with some care as for example a roll and pitch of 180° each renders the same result as a yaw of 180°. After the rotation have been decomposed correctly it is recomposed without the roll angle.

For the tick resolution, $r$, the object is assumed to be facing approximately in the motion direction. The resolution is then found by taking the standard deviation of the Y coordinates of the corner points on the calibration object in both the pseudo range image and world coordinates. The quota of world point standard deviation divided by the pseudo range point standard deviation gives an estimate of the compression along the Y-axis. This has proven to be sufficient in all tested cases even with some rotation.

The normal direction was set to face straight forward, $\hat{n} = (0, 1, 0)$. For the tested cases the optimisation converged to a reasonable translation direction from this starting point.

The start value for $d$ is found by finding the position of points six and seven on the calibration object (see Appendix B). In the definition these have the Y-coordinate zero. If the object have been skewed or rotated the midpoint between these still hold a good estimate of the object origin in the pseudo range image. The midpoint coordinate is scaled using the previously estimated tick resolution to get the value in metric units, which is needed in the transformation model.

The lens distortion parameters are among the easiest to initialise. It is assumed that no lens distortion exists. Hence, the values in $k$ are set to zero. The
3.6 Optimisation

Lens distortion center contained within \( c \) are set to the midpoint of the real image plane. This camera have 1536 by 512 pixels in total meaning that the start solution, \( c = (768, 256) \), is suitable.

### 3.6.2 Geometric Optimisation

The geometric distortion is minimised by measuring the euclidean distance between point coordinates in the world (metric units). To do this the corner points of the calibration object that have been detected in the pseudo range image are transformed using the inverse transformation of the model described in section 3.5 and the results are compared to the measurements of the calibration object in real world coordinates.

In Figure 3.9 the geometric error measurement is explained further. Points referred to as world corner points are the points known on the calibration object, in millimetres, and sensor corner points are raw points, found by the intersections of the estimated planes in object matching, in pixels.

![Figure 3.9: Geometric Error Computation.](image)

Apart from these points the current selection of calibration parameters must be known. The error computation assumes both the geometric parameters and the optic parameters to be known. In the optimisation of geometric error only the parameters related to the geometry are optimised, however the lens distortion model is used (if it exists) to improve the transformation, and get better results. In the first step of geometric optimisation the lens distortion is unknown and assumed to not exist. What this means is simply that some of the parameters in
the lens distortion model are zero, the error measurement always does the same computation.

### 3.6.3 Lens Model Optimisation

The lens distortion should model non-linearities that the linear geometric model cannot represent. It is assumed that the majority of these distortion effects occur due to imperfections in the camera lens. What is needed to correct for these effects is a way to measure the distortion in order to minimise it with a lens correction model.

Given that the geometry is known with good enough accuracy it is possible to transform all points on the calibration object in the pseudo range image to points in world coordinate space. The projection of such a point into its corresponding plane in world coordinates is the distance in metric units between the point and the plane, and a measurement of how well the point fits the plane equation. By measuring the error for all planes and their point inliers a total error measurement is constructed. Given that the planes are at the correct position a choice of parameters that minimises this error corresponds to making the surfaces more planar. As surfaces on the calibration object should be perfectly planar and the planes should be positioned correctly after geometrical transformation, this approach estimate parameters that minimises the lens distortion and the total geometric error simultaneously. The computation of this error measure is more clearly explained in Figure 3.10.

Compared to the purely geometrical error measure this measure uses a lot more points on the calibration object. In the implementation evaluated the pseudo range image was down sampled four times prior to the lens error computation. Equivalently to the fraction 1/256 of the total number of points (every 16th along each dimension in the pseudo range image). However, this still corresponds to over 15000 points. This subselection of points is the subselection referred to in Figures 3.10 and 3.11.

In the final part of the optimisation the homography stops being re-estimated from corner points on the calibration object and is instead allowed to vary freely like any other parameter. This means that if there was an error in the position of the detected corner points in the pseudo range image this should not affect the final calibration, given that this error did not force the solution into a local optima. The computations necessary when computing the error in each iteration is reduced to only performing the full inverse transform and measuring the error (see Figure 3.11).
3.6 Optimisation

Geometric and optic calibration parameters from iteration $i$

Start

Rotate and translate world object corner points using $\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$

Skew the points to align them with the laser plane using $P$

Remove lens distortion from a subselection of all points in the pseudo range image using $D^{-1}$

Estimate the homography $H$ from the two point sets

Re-compute object corner points from undistorted data

Perform the full inverse transformation of the subselection of all points seen in the pseudo range image

Measure the error as the average absolute distance between all inverted points and their corresponding planes in metric units

Error measurement in millimetres

End

Figure 3.10: Lens Error Computation.

Geometric and optic calibration parameters from iteration $i$

Start

Perform the full inverse transformation of a subselection of all points seen in the uncalibrated range image.

Measure the error as the average absolute distance between all inverted points and their corresponding planes in metric units.

Error measurement in millimetres

End

Figure 3.11: Lens Error Computation with $H$ as a free parameter.
4 Experiments

4.1 Evaluated Setups

In this section the geometrical configurations that have been evaluated for the reconstruction setup are explained. These variations serve to provide examples on how well the method performs under different conditions.

When evaluating the method the coordinate axes referred to as X, Y and Z correspond to the coordinate system Z in the previous part of the report and is a laser plane oriented coordinate system (see for example 4.3 or 4.1). This renaming of the axes serve to simplify understanding of the results as this is the natural nomenclature for axes in 3D space. Also the presented tables and graphs would not be as visually appealing when using subscripts.

4.1.1 Angle Between Camera and Laser Plane

The three angles between the camera and laser plane have been selected to evaluate how well the calibration functions are $10^\circ$, $30^\circ$ and $45^\circ$ respectively.

When discussing the angle of a camera and laser plane multiple angles may come to mind and there may be cause for confusion. The angle that will be evaluated is the one named $\alpha$ in Figure 4.1. The laser and camera are fixed in reversed ordinary position with the laser at the top and the camera observing from the side.

4.1.2 Rotation of Object

The robustness of the method in regards to how well it can handle careless placement of the calibration object is tested. The geometry of the camera rig will be fixed and a total calibration will be performed and evaluated for different rotations $\theta$ of the calibration object around the Z-axis. The way the object is rotated is
Experiments

Figure 4.1: The angle $\alpha$ between camera and laser plane that is evaluated.

Further illustrated by Figure 4.2. To simulate what occurs in real situations when the calibration object is placed on a conveyor belt the following angles of rotation have been selected.

- $\theta = 0^\circ, \pm 5^\circ, \pm 10^\circ$ (Object facing forward)
- $\theta = 180^\circ, 180^\circ \pm 5^\circ, 180^\circ \pm 10^\circ$ (Reversed operation)

Figure 4.2: The angle $\theta$ that is evaluated for object rotation.

4.1.3 Laser Plane Tilt

In normal situations the laser is tweaked so that the motion direction is orthogonal to this plane. However, there are situations when this is difficult to accomplish or the errors in the measurement induced by a small error in this setting could be very large.

To simulate the effects of a tilted laser plane the normal direction was set to specific angles. For generating synthetic data this meant setting the normal direction parameter for the laser plane correctly for different cases, and for real data the laser was tilted physically using a protractor to measure the angle.

These evaluations was performed with the camera positioned at approximately $30^\circ$ relative the laser plane and with the calibration object facing forward. The
laser plane was tilted forward and backward as well as rotated. The tilt is illustrated by $\beta$ and the rotation by $\gamma$ in Figure 4.3.

![Figure 4.3: Illustration of the laser plane tilt.](image)

The selected angles to use for each these parameters are, $0^\circ$, $\pm 5^\circ$, independently. This shows if a rather large tilt or rotation of the laser can be handled. An even larger tilt or rotation is likely intentional and the algorithm could be tweaked for these situations manually if needed.

## 4.2 Evaluation Methods

The calibration method has been evaluated against both real and synthetic data as well as compared to the already existent COORDINATOR software[1]. Below a description on how the evaluation was performed and how the results were obtained is presented.

### 4.2.1 Resolution X-axis

The precision of X-axis measurements will be estimated from a scan of a saw tooth object (see Figure 4.4). The object has tops 8 mm apart and on average the error stemming from the manufacturing process should be low. The distance between each consecutive top will be measured to give information about how the calibration works in different parts of the image. The resulting error measure is transformed to the real image coordinates where an error map is formed. From all error measurements within the FOV, the mean absolute error is found and used for benchmarking against other configurations.

### 4.2.2 Resolution Y-axis

The Y-axis is evaluated in a similar way as the X-axis. The saw tooth object is rotated $90^\circ$ and aligned orthogonal to the laser plane (see Figure 4.4). Only one level along the Z-axis will be evaluated as the number of scans needed otherwise would be too many. The average absolute error within the FOV is still used for benchmarking.
4.2.3 Resolution Z-axis

The accuracy of the resolution along the Z-axis is measured by scanning the same planar surface at different heights in the laser field. For synthetic data these planes are generated by constructing a shape similar to a staircase in real world coordinates and transforming it into sensor coordinates. The staircase is positioned such that for any slice in the scan one plane would fill the whole width of the camera sensor. This allows to measure the distance between each step individually for every column of the sensor. For all pixels in each plane a plane equation is also found and the distances between neighbouring planes are computed.

For the real data case it was hard to find an object with known dimensions to use. Because of this the saw tooth object was used once again. By putting the object at an angle of approximately 45°, and scanning it for different positions along X in the laser field, planar areas at different heights were observed (see Figure 4.5). From these areas, plane equations were formed, and the orthogonal distance between the planes were compared. Unfortunately this does not allow for a perfect measurement of the Z-axis, as the two other axes are involved.

The first scan, or the first step of the staircase, was positioned so that the surface was just about visible on the camera sensor. The evaluation object was then adjusted until measurements from the whole sensor had been taken.

For benchmarking the orthogonal distances between the found plane equations inside the FOV are used. These distances are a form of averaged errors for the plane in total. By using distances between all planes data is collected to form a sparse error map of almost the whole camera sensor. The total mean and standard deviation for the calibration error related to these planes are computed and data is presented in graphs and tables.
4.3 Obtaining Data

The data used for calibration is obtained differently for synthetic data compared to real data. In Figure 4.6 an example of how the data that was used for calibration is presented and below follows a short explanation on how data gathering was conducted.

4.3.1 Synthetic Data

To obtain synthetic data as close to real data as possible, points are generated in the world coordinate system and are then transformed using the projection pipeline defined in section 3.5. From these points, and their known connection
to a plane, a range image can be interpolated. After interpolating a range image, lens distortion is added. The evaluation data used for one of the datasets is presented in Figure 4.7.

Parameters used to transform data are set to reasonable values by observing results from a calibration of real data. The lens distortion parameters are received from a calibration with the COORDINATOR software.

### 4.3.2 Real Data

Real data is obtained by as accurately as possible adjusting the camera and laser to the desired configuration and then scanning. After all data for one configuration have been obtained the setup is adjusted. It is important that all tests are completed before changing anything as it will be impossible to reconstruct the exact same configuration. In Figure 4.8 some real verification data is presented.

### 4.4 Comparison Against Coordinator

Where it is applicable the setup will also be calibrated using the COORDINATOR. The calibration received is then evaluated and presented together with the results from the proposed method. No synthetic testing on the COORDINATOR software will be performed and hence no comparisons to synthetic data will be used.

Furthermore, the COORDINATOR only calibrates the X- and Z-axis making a comparison for the Y-axis impossible.
(a) Verification data used for X-axis. Several offsets along the X-axis are used to get more data for each pixel in the FOV.

(b) Verification data for the Y- and Z-axis. Y is a saw tooth object of infinite width and Z is a tilted plane.

Figure 4.7: Verification data example for all axes separately. Images presented are distorted using a reasonable projection pipeline and lens distortion. The distance between each top of the teeth is 8 mm in in calibrated coordinate space.
(a) Verification data used for X-axis. Multiple scans with offsets along the X-axis are used to get more data for each pixel in the fov.

(b) Verification data for the Y-axis. Data is sparse with about five scans per evaluation manually merged to one range image.

(c) Verification data for the Z-axis. Planes exist in multiple scans and only the areas that have well defined planes are used for evaluation.

Figure 4.8: Verification data example for all axes separately. Images presented are received from scanning real data.
The results section presents the data associated with the tests introduced in chapter 4. The first part deals with very simple synthetic data without any distortion stemming from a lens distortion model. This enables the possibility to verify that the transformation works as intended and that the optimal solution can be found given a very good lens or previously estimated lens distortion parameters.

The second part adds lens distortion to synthetic data. This gives the opportunity to, among other things, see if the correct lens model parameters are found.

For the third part the calibration is performed on real data with lens distortion. This is the real test to see how good results are achievable in reality.

For all test there exists graphs and tables with data summarising the results of the calibration. All measurements are presented in millimetres.

### 5.1 Synthetic Results Without Lens Distortion

In the first series of tests no lens distortion was added when constructing synthetic data. Only the corner points on the calibration object were optimised against their known position. This means that the sum of euclidean distances for eleven points was used. The limit for the calibration to stop was when either the total error was less than or the error changed by less than $10^{-10}$ mm. In this regard errors received from evaluation with a magnitude of less than or close to this limit should be considered very good.

#### 5.1.1 Angle Between Camera and Laser Plane

In this synthetic test the graphs in Figure 5.1 show that the calibration have converged to good solutions for all cases. There are small differences in the actual performance results, but they are likely due to round off errors.
Even if data is generated from ideal points data have been interpolated to construct an image. The result of this is that there are small errors, but those errors have no real importance.

![Figure 5.1: Graphical visualisation of the results from calibration on synthetic data, dependent on the angle between camera and laser plane.](image)

In Table 5.1 it is shown that there are marginal differences between each of the tested calibration situations. There are no reasons to believe that the algorithm in itself have any limitations in regards to this change in configuration.

<table>
<thead>
<tr>
<th>Angle α (°)</th>
<th>Mean Abs. Error (mm)</th>
<th>Std. of Error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>10°</td>
<td>8.092e-015</td>
<td>4.948e-014</td>
</tr>
<tr>
<td>30°</td>
<td>1.602e-014</td>
<td>1.868e-014</td>
</tr>
<tr>
<td>45°</td>
<td>1.54e-014</td>
<td>2.453e-014</td>
</tr>
</tbody>
</table>

Table 5.1: Numerical results from calibration on synthetic data, dependent the angle between the camera and the laser plane.

### 5.1.2 Rotation of Object

For all evaluated rotation angles, of the object around its own Z-axis, the error is less than $10^{-12}$ millimetres. Hence, these errors are acceptable considering the parameters used for optimisation. The results from these tests are summarised by Figure 5.2 and Table 5.2.

No real conclusions can be drawn from these results other than that the calibration works well for all angles.

### 5.1.3 Laser Plane Tilt

For the most part the same can be said for evaluation of the laser plane tilt as for the rotation evaluation. There are no conclusions to be drawn other than that the calibration has converged to a good solution in all cases. In Figure 5.3 and Table 5.3 it is shown that the results are overall very good.
5.1 Synthetic Results Without Lens Distortion

Figure 5.2: Graphical visualisation of the results from calibration on synthetic data, dependent on object rotation around the Z-axis.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Mean Abs. Error (mm)</th>
<th>Std. of Error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>θ</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>-10°</td>
<td>1.193e-014</td>
<td>1.4e-014</td>
</tr>
<tr>
<td>-5°</td>
<td>1.309e-014</td>
<td>3.152e-013</td>
</tr>
<tr>
<td>0°</td>
<td>1.602e-014</td>
<td>1.868e-014</td>
</tr>
<tr>
<td>5°</td>
<td>3.458e-014</td>
<td>7.474e-014</td>
</tr>
<tr>
<td>10°</td>
<td>1.456e-014</td>
<td>3.526e-013</td>
</tr>
<tr>
<td>170°</td>
<td>1.277e-014</td>
<td>9.182e-013</td>
</tr>
<tr>
<td>175°</td>
<td>2.181e-014</td>
<td>8.953e-014</td>
</tr>
<tr>
<td>180°</td>
<td>1.029e-014</td>
<td>3.736e-013</td>
</tr>
<tr>
<td>185°</td>
<td>2.275e-014</td>
<td>1.143e-013</td>
</tr>
</tbody>
</table>

Table 5.2: Numerical results from calibration on synthetic data, dependent on object rotation around the Z-axis.
Figure 5.3: Graphical visualisation of the results from calibration on synthetic data, dependent on tilt of the laser plane relative the motion direction of the calibration object.

Table 5.3: Numerical results from calibration on synthetic data, dependent on the tilt of the laser plane relative the motion direction of the calibration object.
5.2 Synthetic Results With Lens Distortion

Lens distortion is added using the same model as the one that is to be estimated. However, the added distortion is computed using more model parameters than are available in the calibration model. This simulates a real situation closer as it will never be possible to perfectly map the real lens distortion using a model. The results after calibration shows if the used model can converge in an optimal solution during these circumstances.

The distortion parameters used when adding the lens distortion are

$$\delta^T = \begin{bmatrix} \tilde{u}_c \\ \tilde{v}_c \\ k_1 \\ k_2 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 794.168 \\ 298.094 \\ 1.02247e-007 \\ 6.29454e-014 \\ -1.4525e-006 \\ 1.0524e-005 \end{bmatrix}$$

(5.1)

and the parameters estimated in the optimisation are the first four, $\tilde{u}_c$, $\tilde{v}_c$, $k_1$ and $k_2$. The choice was made to not optimise the last two parameters as they can act as a skew and make it harder to estimate the homography. Removing two parameters improves the overall speed of the optimisation.

For these datasets the calibration is run in the way presented in section 3.6. The iterations were allowed to continue until the change in error was less than $10^{-10}$ for all optimisation steps. When the solution converges it is used to transform the dataset and evaluate the precision of the calibration.

5.2.1 Angle Between Camera and Laser Plane

Changing the angle between the camera and laser plane directly affects how many pixels represent a real world unit. If the camera is positioned at a greater angle more pixels will be used to measure the same distance compared to when using a lesser angle. Hence, it is natural that the precision for the Z-axis is better when using great angles.

Even so, for the perfect synthetic case this effect is barely noticeable in the results (see Figure 5.4 and Table 5.4). The axis it should affect most severely is the Z-axis, but even for those results it is hard to see a dramatic difference. It is however understandable, as it in the synthetic case only is a matter of pixel resolution in the generated pseudo range image. There is no noise or other disruptions that will be more pronounced in relation to the resolution for a lesser angle compared to a greater.

The errors when adding the lens distortion is clearly higher than without (Figure 5.4 and Table 5.4). This is natural, especially as the distortion was created with a more complex model than the estimated.

In total the results affected most severely by the addition of lens distortion are the X-axis results. This could be due to measuring an insufficient number of long and vertically oriented lines close to the edges of the FOV. What this means
Figure 5.4: Graphical visualisation of the results from calibration on synthetic data with lens distortion added. Data depends on camera positions relative laser plane.

is that radial effects that distort along the X-axis might not be observed as well as the corresponding distortion along the Z-axis.

<table>
<thead>
<tr>
<th>Angle (°)</th>
<th>Mean Abs. Error (mm)</th>
<th>Std. of Error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>10°</td>
<td>0.0003352</td>
<td>9.333e-005</td>
</tr>
<tr>
<td>30°</td>
<td>0.0003258</td>
<td>9.183e-005</td>
</tr>
<tr>
<td>45°</td>
<td>0.0003417</td>
<td>9.009e-005</td>
</tr>
</tbody>
</table>

Table 5.4: Numerical results from calibration on synthetic data with lens distortion added. Data depends on camera position relative laser plane.

By studying the parameter results from the lens distortion model in Table 5.5 it is hard to see that there are any major errors. However, a small change in any of these parameters have effects on a micrometer scale that are hard to estimate. All in all it seems that the algorithm proposed has found the correct solution in order to remove lens distortion.

<table>
<thead>
<tr>
<th>Angle (°)</th>
<th>Lens Distortion Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$a_c$</td>
</tr>
<tr>
<td>Ground truth</td>
<td>794.168</td>
</tr>
<tr>
<td>10°</td>
<td>794.16775</td>
</tr>
<tr>
<td>30°</td>
<td>794.16851</td>
</tr>
<tr>
<td>45°</td>
<td>794.1685</td>
</tr>
</tbody>
</table>

Table 5.5: Lens distortion parameters received at different positions of the camera relative the laser plane.
5.2 Synthetic Results With Lens Distortion

5.2.2 Rotation of Object

It is easy to see that the error is actually larger when the object is oriented perfectly (see Figure 5.5 and Table 5.6). It could have a rather simple explanation. As the FOV is constant, the result when the object is rotated is that it takes up a larger part of the FOV, allowing for more positions inside the laser plane to be observed. This gives a higher probability of both the geometric and the lens distortion model converging to the correct solution. As can be seen in Table 5.7 the lens distortion parameters have converged well for all solutions, but a small change could have large effects at this scale.

![Graphical visualisation of the results from calibration on synthetic data with lens distortion added. Data depends on object rotation around the Z-axis.](image)

It is interesting to see that in these cases the Y-axis is the axis with the best results, compared to before when it was the worst. The reason for this might be that the lens distortion only affects calibration coordinates in the XZ-plane. Looking at the standard deviation of the error further reveals that the Y-axis calibration most likely is more stable. However, as the total calibration links all axis to each other it is not obvious to draw any real conclusions.

5.2.3 Laser Plane Tilt

The main conclusion from these tests is that the calibration fairs about on par with the results from the rotation tests (see Figure 5.6). The resulting error, and
<table>
<thead>
<tr>
<th>Angle θ</th>
<th>Mean Abs. Error (mm)</th>
<th>Std. of Error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>-10°</td>
<td>8.34e-005</td>
<td>8.01e-006</td>
</tr>
<tr>
<td>-5°</td>
<td>0.000122</td>
<td>4.63e-006</td>
</tr>
<tr>
<td>0°</td>
<td>0.000151</td>
<td>2.65e-006</td>
</tr>
<tr>
<td>5°</td>
<td>0.000141</td>
<td>2.9e-006</td>
</tr>
<tr>
<td>10°</td>
<td>7.41e-005</td>
<td>9.61e-006</td>
</tr>
<tr>
<td>170°</td>
<td>6.23e-005</td>
<td>1.05e-005</td>
</tr>
<tr>
<td>175°</td>
<td>0.000132</td>
<td>3.97e-006</td>
</tr>
<tr>
<td>180°</td>
<td>0.000322</td>
<td>6.84e-006</td>
</tr>
<tr>
<td>185°</td>
<td>7.59e-005</td>
<td>9.03e-006</td>
</tr>
<tr>
<td>190°</td>
<td>6.29e-005</td>
<td>9.85e-006</td>
</tr>
</tbody>
</table>

Table 5.6: Numerical results from calibration on synthetic data with lens distortion. Data depends on object rotation around the Z-axis.

<table>
<thead>
<tr>
<th>Angle θ</th>
<th>uc</th>
<th>vc</th>
<th>k1</th>
<th>k2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth</td>
<td>794.168</td>
<td>298.094</td>
<td>1.02247e-007</td>
<td>6.29454e-014</td>
</tr>
<tr>
<td>-10°</td>
<td>794.17096</td>
<td>298.09003</td>
<td>1.022621e-007</td>
<td>6.293641e-014</td>
</tr>
<tr>
<td>-5°</td>
<td>794.16919</td>
<td>298.10094</td>
<td>1.0227021e-007</td>
<td>6.294132e-014</td>
</tr>
<tr>
<td>0°</td>
<td>794.1682</td>
<td>298.10912</td>
<td>1.0227642e-007</td>
<td>6.2944871e-014</td>
</tr>
<tr>
<td>5°</td>
<td>794.16727</td>
<td>298.10594</td>
<td>1.0227362e-007</td>
<td>6.2944764e-014</td>
</tr>
<tr>
<td>10°</td>
<td>794.16844</td>
<td>298.08743</td>
<td>1.0226044e-007</td>
<td>6.2934454e-014</td>
</tr>
<tr>
<td>170°</td>
<td>794.16884</td>
<td>298.08417</td>
<td>1.0225815e-007</td>
<td>6.293257e-014</td>
</tr>
<tr>
<td>175°</td>
<td>794.16805</td>
<td>298.1034</td>
<td>1.0227191e-007</td>
<td>6.2943037e-014</td>
</tr>
<tr>
<td>180°</td>
<td>794.1648</td>
<td>298.15685</td>
<td>1.023106e-007</td>
<td>6.2970788e-014</td>
</tr>
<tr>
<td>185°</td>
<td>794.16815</td>
<td>298.08805</td>
<td>1.0226093e-007</td>
<td>6.2934688e-014</td>
</tr>
<tr>
<td>190°</td>
<td>794.17017</td>
<td>298.08435</td>
<td>1.0225806e-007</td>
<td>6.2933666e-014</td>
</tr>
</tbody>
</table>

Table 5.7: Lens distortion parameters received from calibration of a rotated calibration object.
the standard deviation of the same, has the same order of magnitude as when evaluating the rotation.

![Graphical visualisation of the results from calibration on synthetic data with lens distortion added. Data depends on tilt of the laser plane relative the motion direction of the calibration object.](image)

**Figure 5.6:** Graphical visualisation of the results from calibration on synthetic data with lens distortion added. Data depends on tilt of the laser plane relative the motion direction of the calibration object.

Looking more closely at the data in Table 5.8, the results seem unaffected by changing $\beta$. For synthetic data this change won't affect the reflectivity of the surface and this is to be expected. The situation might be different when using real data.

When $\gamma$ is changed to a positive value the results seem to be better than when instead making a negative change. This may look confusing at first. There are however a few explanations for this. First, and foremost, the calibration object is not placed perfectly in the center of the FOV. Secondly the lens distortion model does not have its center in the center of the FOV. This means that errors in lens distortion parameters will be more or less severe in different parts of the view. When the object is skewed in different ways it will take up different parts of the FOV and therefore marginally change the calibration. The fluctuations observed when changing $\gamma$ could therefore be seen as logical variations and nothing out of the ordinary.
<table>
<thead>
<tr>
<th>Angle</th>
<th>Mean Abs. Error (mm)</th>
<th>Std. of Error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>β</td>
<td>γ</td>
<td>X</td>
</tr>
<tr>
<td>-5.0°</td>
<td>0.0°</td>
<td>0.000107</td>
</tr>
<tr>
<td>-2.5°</td>
<td>0.0°</td>
<td>0.000101</td>
</tr>
<tr>
<td>2.5°</td>
<td>0.0°</td>
<td>0.000116</td>
</tr>
<tr>
<td>5.0°</td>
<td>0.0°</td>
<td>0.000109</td>
</tr>
<tr>
<td>0.0°</td>
<td>-5.0°</td>
<td>9.47e-005</td>
</tr>
<tr>
<td>0.0°</td>
<td>-2.5°</td>
<td>0.000112</td>
</tr>
<tr>
<td>0.0°</td>
<td>2.5°</td>
<td>9.56e-005</td>
</tr>
<tr>
<td>0.0°</td>
<td>5.0°</td>
<td>7.46e-005</td>
</tr>
</tbody>
</table>

**Table 5.8:** Numerical results from calibration on synthetic data with lens distortion. Data varies with laser plane tilt relative the motion direction of the calibration object.

<table>
<thead>
<tr>
<th>β</th>
<th>γ</th>
<th>$\bar{\alpha}_c$</th>
<th>$\bar{\beta}_c$</th>
<th>$k_1$</th>
<th>$k_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth</td>
<td>-5.0°</td>
<td>794.168</td>
<td>298.094</td>
<td>1.02247e-007</td>
<td>6.29454e-014</td>
</tr>
<tr>
<td>-5.0°</td>
<td>0.0°</td>
<td>794.16816</td>
<td>298.09671</td>
<td>1.0226989e-007</td>
<td>6.293101e-014</td>
</tr>
<tr>
<td>-2.5°</td>
<td>0.0°</td>
<td>794.16888</td>
<td>298.09486</td>
<td>1.0226648e-007</td>
<td>6.293611e-014</td>
</tr>
<tr>
<td>2.5°</td>
<td>0.0°</td>
<td>794.16671</td>
<td>298.09891</td>
<td>1.0226997e-007</td>
<td>6.2936966e-014</td>
</tr>
<tr>
<td>5.0°</td>
<td>0.0°</td>
<td>794.16808</td>
<td>298.09623</td>
<td>1.0226822e-007</td>
<td>6.2934944e-014</td>
</tr>
<tr>
<td>0.0°</td>
<td>-5.0°</td>
<td>794.1685</td>
<td>298.09309</td>
<td>1.0226462e-007</td>
<td>6.2937099e-014</td>
</tr>
<tr>
<td>0.0°</td>
<td>-2.5°</td>
<td>794.16804</td>
<td>298.09998</td>
<td>1.0226939e-007</td>
<td>6.2941393e-014</td>
</tr>
<tr>
<td>0.0°</td>
<td>2.5°</td>
<td>794.16926</td>
<td>298.09387</td>
<td>1.0226468e-007</td>
<td>6.2937039e-014</td>
</tr>
<tr>
<td>0.0°</td>
<td>5.0°</td>
<td>794.1705</td>
<td>298.08722</td>
<td>1.0226004e-007</td>
<td>6.2935593e-014</td>
</tr>
</tbody>
</table>

**Table 5.9:** Lens distortion parameters received from calibration when tilting the laser plane.
5.3 Results On Real Data

There are a lot of things that could go wrong when evaluating real data and it is important to observe the possible problem areas. First and foremost the dimensions of the calibration object have not been verified to be correct in an exact way. A calliper was used to measure the object in different directions to verify its size. However, these measurements are not exact enough to match the target of the calibration system.

In addition to this uncertainty, the evaluation is performed using a saw tooth object that is normally used for calibration using the COORDINATOR software. This makes results biased towards the COORDINATOR in case the dimensions are not perfect for either the new calibration object or the saw tooth object.

That being said the results from evaluation on real data are not surprising for the most part. The aim was to get even better precision from this calibration method, but as initial results on a new method the numbers are quite promising.

5.3.1 Noise and Resolution

Before diving into the results two tables with information about the datasets should be presented. When judging the results from the calibration it is important to know what kind of noise is present and what resolution actually is possible to get from the configuration.

The noise levels for different configurations could vary quite a lot both depending on the geometry, but also on ambient light. In an attempt to summarise the noise levels for each data set Table 5.10 was put together. The presented numbers are the mean absolute value of point distances to planes in different parts of the camera sensor, expressed in millimetres. The column of data labelled fov holds measurements taken within the area of the calibration object and the second, called whole sensor, contains the average from all taken measurements.

In most situations it would be very hard to get results that are better than these measurements. However, these noise measurements are computed in relation to planes with normals close to parallel with the Z-axis. Therefore, the noise levels best measure the noise along the Z-axis. It is for this axis that it is most relevant to measure noise, but it means that for any of the two other axes it may be possible to get better results. Most likely it is not the case that the noise is uniform in three dimensions and it would not be surprising if the noise is considerably lower in the X- or Y-direction.

The size of the fov and where it is located on the image sensor also bears relevance to the result of the calibration. In Table 5.11 the data and statistics for each used dataset is presented. The Rows on sensor values are the settings used in the camera when capturing data. For all datasets the full width of 1536 pixels were used an therefore no specification of columns are provided in the same format. The resolution results are based on measurements inside the bounding box of the calibration object. By taking the results of any X- or Z-calibration below and dividing it by the resolution value in this table an approximate result in pixels is received. This is good for relating the results to other fov.
### Table 5.10: Noise level overview for all real datasets.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Noise level (mm)</th>
<th>Whole sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>α fov</td>
<td></td>
</tr>
<tr>
<td>10°</td>
<td>0.043410437 0.16859322</td>
<td></td>
</tr>
<tr>
<td>30°</td>
<td>0.030882571 0.051208742</td>
<td></td>
</tr>
<tr>
<td>45°</td>
<td>0.049582548 0.050077099</td>
<td></td>
</tr>
<tr>
<td></td>
<td>θ fov</td>
<td></td>
</tr>
<tr>
<td>-10°</td>
<td>0.031936023 0.051400194</td>
<td></td>
</tr>
<tr>
<td>-5°</td>
<td>0.030977899 0.051559102</td>
<td></td>
</tr>
<tr>
<td>0°</td>
<td>0.030882571 0.051208742</td>
<td></td>
</tr>
<tr>
<td>5°</td>
<td>0.029724552 0.051064997</td>
<td></td>
</tr>
<tr>
<td>10°</td>
<td>0.029978383 0.051101594</td>
<td></td>
</tr>
<tr>
<td>170°</td>
<td>0.032782773 0.051128793</td>
<td></td>
</tr>
<tr>
<td>175°</td>
<td>0.031439306 0.051225888</td>
<td></td>
</tr>
<tr>
<td>180°</td>
<td>0.03106416 0.051235524</td>
<td></td>
</tr>
<tr>
<td>185°</td>
<td>0.031745511 0.05123382</td>
<td></td>
</tr>
<tr>
<td>190°</td>
<td>0.032047268 0.051179074</td>
<td></td>
</tr>
<tr>
<td></td>
<td>β γ fov</td>
<td></td>
</tr>
<tr>
<td>-5°</td>
<td>0° 0° 0.051527403 0.048924552</td>
<td></td>
</tr>
<tr>
<td>5°</td>
<td>0° 0° 0.033184107 0.046396986</td>
<td></td>
</tr>
<tr>
<td>0°</td>
<td>0° -5° 0.030882571 0.051208742</td>
<td></td>
</tr>
<tr>
<td>10°</td>
<td>0.037073761 0.084029974</td>
<td></td>
</tr>
<tr>
<td>170°</td>
<td>0.037116468 0.084008259</td>
<td></td>
</tr>
<tr>
<td>175°</td>
<td>0.037116468 0.084008259</td>
<td></td>
</tr>
<tr>
<td>180°</td>
<td>0.037258658 0.084004116</td>
<td></td>
</tr>
<tr>
<td>185°</td>
<td>0.037265611 0.084098592</td>
<td></td>
</tr>
<tr>
<td>190°</td>
<td>0.037063333 0.084091719</td>
<td></td>
</tr>
</tbody>
</table>

### Table 5.11: FOV and resolution overview for all datasets.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Rows on sensor (px)</th>
<th>Resolution (mm/px)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>α Row start Row end X Z</td>
<td></td>
</tr>
<tr>
<td>10°</td>
<td>100 196 0.040456213 0.17014871</td>
<td></td>
</tr>
<tr>
<td>30°</td>
<td>280 420 0.037183857 0.08398701</td>
<td></td>
</tr>
<tr>
<td>45°</td>
<td>180 404 0.033803342 0.054340433</td>
<td></td>
</tr>
<tr>
<td></td>
<td>θ Row start Row end X Z</td>
<td></td>
</tr>
<tr>
<td>-10°</td>
<td>280 420 0.036623332 0.084074199</td>
<td></td>
</tr>
<tr>
<td>-5°</td>
<td>280 420 0.03679354 0.084070883</td>
<td></td>
</tr>
<tr>
<td>0°</td>
<td>280 420 0.037183857 0.08398701</td>
<td></td>
</tr>
<tr>
<td>5°</td>
<td>280 420 0.037248735 0.084126802</td>
<td></td>
</tr>
<tr>
<td>10°</td>
<td>280 420 0.037073761 0.084029974</td>
<td></td>
</tr>
<tr>
<td>170°</td>
<td>280 420 0.037116468 0.084008259</td>
<td></td>
</tr>
<tr>
<td>175°</td>
<td>280 420 0.037258658 0.084004116</td>
<td></td>
</tr>
<tr>
<td>180°</td>
<td>280 420 0.03714052 0.084079133</td>
<td></td>
</tr>
<tr>
<td>185°</td>
<td>280 420 0.037265611 0.084098592</td>
<td></td>
</tr>
<tr>
<td>190°</td>
<td>280 420 0.037063333 0.084091719</td>
<td></td>
</tr>
<tr>
<td></td>
<td>β γ Row start Row end X Z</td>
<td></td>
</tr>
<tr>
<td>-5°</td>
<td>0° 0° 200 424 0.034730089 0.084025862</td>
<td></td>
</tr>
<tr>
<td>5°</td>
<td>0° 0° 232 424 0.034686408 0.065904257</td>
<td></td>
</tr>
<tr>
<td>0°</td>
<td>0° -5° 280 420 0.037183857 0.08398701</td>
<td></td>
</tr>
<tr>
<td>0°</td>
<td>5° 220 428 0.036474978 0.080730762</td>
<td></td>
</tr>
</tbody>
</table>
5.3 Results On Real Data

5.3.2 Angle Between Camera and Laser Plane

When changing the perspective between the camera and the laser the resolution, in terms of how many pixels measure one metric unit, changes drastically. This should mean that for a larger angle of $\alpha$ the error in metric units should be lower. However, this is not the case for the calibration performed using the proposed calibration method (see Figure 5.7 and also Tables 5.12 and 5.13 for data related to the figure).

![Graphical visualisation of the results from calibration on real data.](image)

**Figure 5.7:** Graphical visualisation of the results from calibration on real data. Data varies with the angle between the laser plane and the optical axis of the camera. Dashed lines represent results from the COORDINATOR software.

It is obvious that all errors are lower for a 30° angle compared to a 10° angle. A 10° angle can give a larger FOV and allows to measure larger objects. However, it also means that a smaller portion of the FOV have been used for calibration. Furthermore, the difference between height levels in the real image will most definitely be lower. Studying the noise levels in the scanned area it can be concluded that for the whole FOV the noise is considerably higher compared to other configurations (see Table 5.10). In the calibrated part of the FOV it is also higher compared to the scan with 30° angle (see $\theta$ equal to zero in Table 5.10). Compared to the COORDINATOR the results from calibration with the proposed method are slightly better for both axes.

Not surprisingly the results improve for the 30° angle compared to the 10° angle. The results from the COORDINATOR follows, but are still at least ten micrometres worse for both axes.

The results for the 45° angle are surprising. Although the dataset looked good the calibration seems to be worse compared the 30° case. Measurements of noise levels do provide some insight into what causes these bad results. A noise level increase of approximately 20 micrometres helps explain the 25 micrometre decrease in precision for the calibration. One other thing to have in mind is that
### Table 5.12: Mean absolute error from calibration on real data. Data varies with the angle between the laser plane and the optical axis of the camera.

<table>
<thead>
<tr>
<th>Angle α (°)</th>
<th>Mean Abs. Error (mm) X</th>
<th>Mean Abs. Error (mm) Y</th>
<th>Mean Abs. Error (mm) Z</th>
<th>Mean Abs. Error Coordinator (mm) X</th>
<th>Mean Abs. Error Coordinator (mm) Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>10°</td>
<td>0.060791</td>
<td>0.0381568</td>
<td>0.0869811</td>
<td>0.0960611</td>
<td>0.100988</td>
</tr>
<tr>
<td>30°</td>
<td>0.0142002</td>
<td>0.0540934</td>
<td>0.0445437</td>
<td>0.0417027</td>
<td>0.0662991</td>
</tr>
<tr>
<td>45°</td>
<td>0.0193269</td>
<td>0.0756789</td>
<td>0.0989223</td>
<td>0.0796339</td>
<td>0.0596013</td>
</tr>
</tbody>
</table>

### Table 5.13: Standard deviation of the error from calibration on real data. Data varies with the angle between the laser plane and the optical axis of the camera.

<table>
<thead>
<tr>
<th>Angle α (°)</th>
<th>Std. of Error (mm) X</th>
<th>Std. of Error (mm) Y</th>
<th>Std. of Error (mm) Z</th>
<th>Std. of Error Coordinator (mm) X</th>
<th>Std. of Error Coordinator (mm) Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>10°</td>
<td>0.0475807</td>
<td>0.0357378</td>
<td>0.0846747</td>
<td>0.0575965</td>
<td>0.0938856</td>
</tr>
<tr>
<td>30°</td>
<td>0.0101319</td>
<td>0.0456487</td>
<td>0.0518254</td>
<td>0.0514411</td>
<td>0.0630948</td>
</tr>
<tr>
<td>45°</td>
<td>0.0150819</td>
<td>0.144282</td>
<td>0.0751327</td>
<td>0.0481649</td>
<td>0.0666043</td>
</tr>
</tbody>
</table>

When the angle between camera and laser is great, the reflected light from the object will be lower in some areas on the object, forcing the setting of exposure to be more of a compromise in order to detect lines on all areas. This means that the risk of over- or underexposure is higher. The final factor that can affect the results, when evaluating these configuration changes, is that in order to change the camera angle the whole position of the camera have to be changed to keep the object inside the available FOV. This means that different parts of the sensor were used in these comparisons, even if care was taken to position the camera as well as possible.

Looking more closely at the calibration along the Y-axis a trend towards worse results for a larger angle can be seen. In particular for the 45° case, but also in part for the 30° case, this has an explanation. Because the object used for evaluation consist of saw teeth with a basis of 8 and a height of 4 millimetres the angle of the surfaces are actually 45° relative the base plane of the scan. This means that it will be impossible to see the far side of the teeth due to camera occlusion in a perfect 45° case. In this case the camera was not positioned perfectly at 45° and the back surfaces was partially observed and subsequently repaired using median filtering, before estimating the resolution. This of course introduces uncertainty and errors that might not actually be there. For both cases with greater angles, the results of the fact that the teeth are tilted in this way, is worse precision on the far side of each tooth compared to the 10° case.

The lens distortion parameters have converged to values similar to those of the COORDINATOR (see Table 5.14). In general there is not much to conclude other than that the values are similar, as they should be when the same part of the FOV has been used. One thing to remember when comparing the results of this new method to the COORDINATOR is that the COORDINATOR uses the full lens model presented in section 2.4.3. This means that it has two more parameters that interact with the the other parameters. A small difference in the coordinates of the distortion center $\tilde{u}_c, \tilde{v}_c$ or the distortion parameters $k_1, k_2$ could likely be...
explained by this fact.

The most prominent difference is in the 10° case for the new method, where the sign of $k_1$ and $k_2$ appears to have flipped compared to all other solutions. This phenomenon has also been observed in the COORDINATOR in some cases. In general it still results in a good calibration, even if it intuitively seems bad. One explanation to why it occurs for the 10° case could be that a very small part of the sensor is used for such a small angle between camera and laser plane, which makes the lens distortion model hard to determine. This means that there might be lens model parameters that are more different than in other cases, but still solve the problem well inside the FOV.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Lens Distortion Parameters</th>
<th>$\hat{\alpha}$</th>
<th>$\hat{\gamma}$</th>
<th>$k_1$</th>
<th>$k_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10°</td>
<td></td>
<td>727.14121</td>
<td>227.17986</td>
<td>-2.9242106e-008</td>
<td>1.4826188e-013</td>
</tr>
<tr>
<td>30°</td>
<td></td>
<td>745.84141</td>
<td>211.1775</td>
<td>6.6733303e-008</td>
<td>2.4257298e-014</td>
</tr>
<tr>
<td>45°</td>
<td></td>
<td>725.8103</td>
<td>229.1653</td>
<td>9.116058e-008</td>
<td>1.120584e-014</td>
</tr>
<tr>
<td>10°</td>
<td></td>
<td>714.1192</td>
<td>200.602</td>
<td>1.446941e-007</td>
<td>-1.754327e-014</td>
</tr>
<tr>
<td>30°</td>
<td></td>
<td>824.8649</td>
<td>216.8919</td>
<td>6.106972e-008</td>
<td>2.617886e-014</td>
</tr>
<tr>
<td>45°</td>
<td></td>
<td>797.4171</td>
<td>200.7162</td>
<td>7.332356e-008</td>
<td>1.307174e-014</td>
</tr>
</tbody>
</table>

*Table 5.14: Lens distortion parameters received from calibration using different angles between the laser plane and optical axis of the camera. Values below the dashed line correspond to COORDINATOR results.*

### 5.3.3 Rotation of Object

In Figure 5.8 it is clear to see that for the most tested cases the new calibration method renders a better result compared to the COORDINATOR. For this dataset only the yaw angle of the calibration object itself changed and therefore there is only one calibration of the COORDINATOR to compare to.

Interestingly enough the algorithm performs more evenly and gets slightly better results when the calibration object is rotated approximately 180° (this is easier to see in Table 5.15). This may have to do with the reflectivity of a majority of the surfaces being better compared to when scanning in the forward orientation. Given these results the recommendation for use should be to place the object with the pointy end (point one in Appendix B) facing away from the camera. However, these differences are very small and could also be a coincidental.

In general there are no other conclusions to be made other than that the orientation of the object has no serious effect on the results of the calibration. There is one calibration that performed slightly worse than the others, but this is likely an unfortunate coincidence. As mentioned before there are always differences between sets, apart from what is actually tested. In this test it would be more logical if the evaluation results had been symmetric around 0° and 180° rotation. Differences here likely depend on slightly different fill of the FOV due to the rotation and positioning of the object, that also leads to different parts of the lens being used. All evaluation results have been presented as they are without recalibrating certain angles to try and improve the results. This is in order to show the
Figure 5.8: Graphical visualisation of the results from calibration on real data. Data varies with object rotation around the Z-axis. Dashed lines represent results from the COORDINATOR software.
kind of results that can be expected from the algorithm without extra tinkering.

<table>
<thead>
<tr>
<th>Angle θ</th>
<th>Mean Abs. Error (mm)</th>
<th>Std. of Error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>Coordinator</td>
<td>0.0416757</td>
<td>-</td>
</tr>
<tr>
<td>-10°</td>
<td>0.00143865</td>
<td>0.0552752</td>
</tr>
<tr>
<td>-5°</td>
<td>0.02359272</td>
<td>0.05706</td>
</tr>
<tr>
<td>0°</td>
<td>0.01420022</td>
<td>0.0540934</td>
</tr>
<tr>
<td>5°</td>
<td>0.01283632</td>
<td>0.053091</td>
</tr>
<tr>
<td>10°</td>
<td>0.01072762</td>
<td>0.0525721</td>
</tr>
<tr>
<td>170°</td>
<td>0.011266</td>
<td>0.055568</td>
</tr>
<tr>
<td>175°</td>
<td>0.00976544</td>
<td>0.0552561</td>
</tr>
<tr>
<td>180°</td>
<td>0.00974954</td>
<td>0.0541747</td>
</tr>
<tr>
<td>185°</td>
<td>0.0100698</td>
<td>0.0568842</td>
</tr>
<tr>
<td>190°</td>
<td>0.0096378</td>
<td>0.0540858</td>
</tr>
</tbody>
</table>

Table 5.15: Numerical results from calibration on real data. Data varies with object rotation around the Z-axis.

Moving on to the lens distortion parameters, visualised in Table 5.16, it is likely that the COORDINATOR has found another solution than the method presented in this thesis. Keep in mind that the COORDINATOR also makes use of the two extra parameters \( p_1 \), \( p_2 \) and this does not seem all to strange. However, the results won’t tell us which one actually is the better solution in terms of lens model. In this particular case the total calibration received from the proposed method performs better, but what part the lens model played is not clear.

<table>
<thead>
<tr>
<th>Angle θ</th>
<th>( \hat{u}_c )</th>
<th>( \hat{v}_c )</th>
<th>( k_1 )</th>
<th>( k_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinator</td>
<td>824.8649</td>
<td>216.8919</td>
<td>6.106972e-008</td>
<td>2.617886e-014</td>
</tr>
<tr>
<td>-10°</td>
<td>741.75041</td>
<td>229.44299</td>
<td>7.1425245e-008</td>
<td>3.378767e-014</td>
</tr>
<tr>
<td>-5°</td>
<td>730.17248</td>
<td>223.30262</td>
<td>7.2850058e-008</td>
<td>2.3388167e-014</td>
</tr>
<tr>
<td>0°</td>
<td>745.84141</td>
<td>211.1775</td>
<td>6.6733303e-008</td>
<td>2.4257298e-014</td>
</tr>
<tr>
<td>5°</td>
<td>759.5259</td>
<td>206.72301</td>
<td>6.8009694e-008</td>
<td>3.8187266e-014</td>
</tr>
<tr>
<td>10°</td>
<td>792.78662</td>
<td>212.69622</td>
<td>6.0855263e-008</td>
<td>4.1382817e-014</td>
</tr>
<tr>
<td>170°</td>
<td>760.70812</td>
<td>206.53433</td>
<td>5.9917795e-008</td>
<td>3.5149736e-014</td>
</tr>
<tr>
<td>175°</td>
<td>746.60549</td>
<td>210.80784</td>
<td>6.5334385e-008</td>
<td>3.0502825e-014</td>
</tr>
<tr>
<td>180°</td>
<td>745.8475</td>
<td>211.1753</td>
<td>7.4608273e-008</td>
<td>9.2639885e-015</td>
</tr>
<tr>
<td>185°</td>
<td>734.21108</td>
<td>219.14389</td>
<td>7.2408047e-008</td>
<td>1.8541608e-014</td>
</tr>
<tr>
<td>190°</td>
<td>746.98783</td>
<td>210.62928</td>
<td>7.0098318e-008</td>
<td>1.9744617e-014</td>
</tr>
</tbody>
</table>

Table 5.16: Lens distortion parameters received from calibration of a rotated calibration object using real data.

5.3.4 Laser Plane Tilt

When the laser plane is tilted or rotated in relation to the translation direction of the scan a skewing occurs. This means that it is expected that the precisions in general should be worse when the laser plane is non-orthogonal to the direction of motion. The effects of the skew are clearly visible in the scans used for calibration, but the effects are not clearly pronounced in the numerical results, which
suggests that the skewing was handled well (see Figure 5.9 as well as Table 5.17 and 5.18).

<table>
<thead>
<tr>
<th>Angle β</th>
<th>Mean Abs. Error (mm)</th>
<th>Mean Abs. Error Coordinator (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>-5° 0°</td>
<td>0.0439732</td>
<td>0.0590913</td>
</tr>
<tr>
<td>5° 0°</td>
<td>0.0133369</td>
<td>0.0901717</td>
</tr>
<tr>
<td>0° 0°</td>
<td>0.0142002</td>
<td>0.0540934</td>
</tr>
<tr>
<td>0° -5°</td>
<td>0.0258733</td>
<td>0.0344861</td>
</tr>
<tr>
<td>0° 5°</td>
<td>0.0162327</td>
<td>0.0404861</td>
</tr>
</tbody>
</table>

Table 5.17: Numerical results of the mean absolute error from calibration on real data. Data varies with the laser plane tilt.

<table>
<thead>
<tr>
<th>Angle β</th>
<th>Std. of Error (mm)</th>
<th>Std. of Error Coordinator (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>-5° 0°</td>
<td>0.0302133</td>
<td>0.0302113</td>
</tr>
<tr>
<td>5° 0°</td>
<td>0.00946085</td>
<td>0.0500644</td>
</tr>
<tr>
<td>0° 0°</td>
<td>0.0101319</td>
<td>0.0456487</td>
</tr>
<tr>
<td>0° -5°</td>
<td>0.0142189</td>
<td>0.0243551</td>
</tr>
<tr>
<td>0° 5°</td>
<td>0.011534</td>
<td>0.026184</td>
</tr>
</tbody>
</table>

Table 5.18: Numerical results of the standard deviation of the error from calibration on real data. Data varies with the laser plane tilt.

The comparison to the COORDINATOR is in this test perfectly fair, even if that system have no way to model a tilted or rotated laser plane in relation to the direction of motion. The COORDINATOR observes only points in the plane and makes sure that they are in the correct position relative each other. This means that if an object is moved through the laser field at an angle, the dimensions are compressed and stretched in ways that cannot be detected. However, as the results of the calibration are only evaluated inside the laser plane for the X- and Z-axis the calibration received from the COORDINATOR should also solve the problem.

Investigating Figure 5.9 it is possible to conclude that tilting the laser plane back and fourth (varying β) gives worse results either way. This can be related to previous results when the camera was moved in relation to the laser plane (section 5.3.2). In many ways these configurations are similar. Take away the fact that data is skewed, and the same limitations due to compression of the object height and reduced reflection, will play a part. When the laser plane is tilted away from the camera (β positive) the resolution is decreased, but in return the reflection is increased. The opposite holds for adjusting β in the other direction. In general terms it seems that the skew is handled about as well as movement of the camera. This could suggest that it is the aforementioned factors that limit the calibration and not the skew introduced by a motion direction non-orthogonal to the laser plane.

Finally the received lens distortion parameters are presented in Table 5.19. As usual it is hard to draw conclusions from these results. In general it is clear that the COORDINATOR and the new calibration method have found quite different solutions to the problem and some of the solutions are more similar to each other.
5.3 Results On Real Data

Figure 5.9: Graphical visualisation of the results from calibration on real data. Data depends on the laser plane tilt. Dashed lines represent results from the COORDINATOR software.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Lens Distortion Parameters</th>
<th>$eta$</th>
<th>$\gamma$</th>
<th>$\tilde{u}_c$</th>
<th>$\tilde{v}_c$</th>
<th>$k_1$ (e-007)</th>
<th>$k_2$ (e-015)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5°</td>
<td>0°</td>
<td>771.40548</td>
<td>206.11112</td>
<td>1.0028696</td>
<td>6.2029501</td>
<td>206.11112</td>
<td>1.0028696</td>
</tr>
<tr>
<td>5°</td>
<td>0°</td>
<td>767.08657</td>
<td>206.00793</td>
<td>8.3656919</td>
<td>-7.1484886</td>
<td>206.00793</td>
<td>8.3656919</td>
</tr>
<tr>
<td>0°</td>
<td>0°</td>
<td>745.84141</td>
<td>211.1775</td>
<td>6.6733303</td>
<td>2.4257298</td>
<td>211.1775</td>
<td>6.6733303</td>
</tr>
<tr>
<td>0°</td>
<td>-5°</td>
<td>791.7549</td>
<td>212.00272</td>
<td>9.4708416</td>
<td>-2.4462458</td>
<td>212.00272</td>
<td>9.4708416</td>
</tr>
<tr>
<td>0°</td>
<td>5°</td>
<td>727.9108</td>
<td>226.1181</td>
<td>9.4202777</td>
<td>1.0976915</td>
<td>226.1181</td>
<td>9.4202777</td>
</tr>
<tr>
<td>-5°</td>
<td>0°</td>
<td>815.6754</td>
<td>190.5728</td>
<td>6.684884e</td>
<td>2.236566e</td>
<td>190.5728</td>
<td>6.684884e</td>
</tr>
<tr>
<td>5°</td>
<td>0°</td>
<td>817.9817</td>
<td>193.8089</td>
<td>6.608727e</td>
<td>1.226117e</td>
<td>193.8089</td>
<td>6.608727e</td>
</tr>
<tr>
<td>0°</td>
<td>0°</td>
<td>824.8612</td>
<td>216.8919</td>
<td>6.106972e</td>
<td>2.617886e</td>
<td>216.8919</td>
<td>6.106972e</td>
</tr>
<tr>
<td>0°</td>
<td>-5°</td>
<td>826.0201</td>
<td>242.2261</td>
<td>6.611216e</td>
<td>1.582617e</td>
<td>242.2261</td>
<td>6.611216e</td>
</tr>
</tbody>
</table>

Table 5.19: Lens distortion parameters received from calibration when tilting the laser plane. Values below the dashed line are results from calibration using the COORDINATOR software.
than others. The most important to note in these tables is that the distortion centres are about the same for all calibrations, at least using the same method. This means that a reasonable optima was found for all configurations.

5.3.5 Effects of Skew

When performing a calibration it can be hard to know that the skew compensation is working the way it should. Because these tests are mostly performed inside the laser plane, it is not impossible to find a calibration that returns a small error, but in some regard fails to correctly adjust for skew. In order to make sure that this is not the case, a calibration was performed with the laser plane rotated approximately 5°, all other parameters were set to the standard configuration (30° between camera and laser plane and no rotation of the calibration object). The results of scanning a grid pattern is shown in Figure 5.10a and the result after restoring using the estimated normal direction of the laser plane is displayed in Figure 5.10b.

This is not a scientific evaluation per se, but it shows that there is nothing weird going on. It would be possible to do the same kind of test and then evaluate the results numerically by measuring very accurately in the image. This was not a priority when performing the evaluations.

5.3.6 Lens Model Performance

It is, as stated many times before, very hard to show that a lens model is working other than by showing good results in the metric evaluation. In effect most comments related to the lens models under each test section are a bit vague. In an attempt to visualise the results from two of the calibrations performed during this evaluation a grid pattern was captured in the image plane and rectified using the received lens model from the base configuration with 30° between the laser plane and camera and no rotation for any other component.

The results from this experiment is shown in Figure 5.11 and although it can not show any numerical results in regards to how straight the lines actually are it shows that the lens model is doing what it is supposed to do. Like in the case of skew compensation it would be possible to extract numerical data from this image, but no effort was put into this during the evaluation of this master’s thesis.
5.3 Results On Real Data

Figure 5.10: Visualisation of how skew is compensated.
(a) Original image captured from the camera.

(b) Lens distortion removed using parameters from the method proposed in this master’s thesis.

(c) Lens distortion removed using parameters from the COORDINATOR software.

**Figure 5.11:** Visualisation of how the lens model affects lines in the image plane.
This section is dedicated to comments on the algorithm that are too loosely defined to be placed under conclusions.

### 6.1 Method Development

Before settling on the method for 2D segmentation in section 3.3.1 two other methods were attempted and later discarded. Principal curvature measurements allowed for detection of all planes and development of the algorithm. The drawbacks of this method is that in planar areas, were the normals of the planes face straight up, there is a noticeable ripple in the received image. This results in few points being detected in these planes and sometimes that the planes are missed completely. The second method that was attempted was the computation of the structure tensor both on the original image and a smoothed one. The only one that was usable was the one received from the smoothed image. However, there was no clear advantage in using this method compared to simple filtering, on the contrary the corners of some planes became unsuitably rounded. In comparison to all other attempted methods the selected method is rather robust and gives clean areas independent of where on the object they have been found. Compared to principal curvatures in particular, that was the main contender, the selected method is slightly faster and better in a majority of situations. For an explanation on how principle curvatures work, see either the work in Swedish by Persson[10], or the English introduction to curves and surfaces by Shifrin[12].

In the selected method for extracting planar areas, algorithm 3.1 is used to compute the variable $I$. This way of combining the derivatives is heuristic and have no basis in the literature referenced. An alternative method could be Marr-Hildreths method (see for example Gonzalez[4]), where the laplacian $\nabla f^2 = f_{xx} + f_{yy}$ is computed and analysed. However, the results from these implementations
give a similar result and the method used in this thesis have been proven to work satisfactory.

The algorithm in section 3.4.2 could be said to remind of RANSAC [3], with the selection of points for model estimation and computation of the total error to decide on the optimal sorting. One or both of the point sets could be selected at random, but instead the method now uses one predefined set and performs an extensive search for the other. The matching tests all possible combinations of point selections from \( P_r \) and matches them to the predefined set, before choosing the best match. When only three points are to be chosen from a set of ten the number of possible choices are only 720 and fully manageable. If the problem were to grow it is possible it would be better to use some kind of RANSAC or spatial formulation of the problem.

In the earlier stages of algorithm development another approach to plane detection and computation was attempted. This method was based on RANSAC using 3D plane fitting. In essence three points were selected from pseudo range data and used to estimate a plane, inliers were computed and the plane re-estimated. This was done until the plane equation converged. When a number of planes had been fitted a predetermined number of the largest distinct planes were chosen. The RANSAC method was discarded because it became to complicated in order to, with confidence, detect all planes and not just the same multiple times. It also became unsuitably slow, directly related to the first problem. In addition there are severe problems related to fitting regular plane equations to planes distorted by a non-linear camera lens. The 2D filtering method now implemented always takes the same time to perform, independently of luck or when calibration is started. The chosen method also provides better planar areas and all in all a more robust basis for the rest of the calibration.

### 6.2 Considerations

Unfortunately the evaluation of the calibration was performed using the saw tooth object originally constructed for use with the COORDINATOR software. This object is manufactured in a CNC mill resulting in defects compared to the ideal shape of the object, that makes it less suitable as a evaluation object. Especially evaluations of the Z-axis have to be considered carefully as there is a rounding of the edge in the lower parts of the saw shape. This rounding, or fillet which is the technical term, threatens to introduce an error in the plane fitting for all evaluation results. It is suggested that the Z-axis results are re-estimated using a better evaluation object before drawing any final conclusions about the Z-axis calibration results.

It would be good to as accurately as possible ascertain that the calibration object has the dimensions specified in the design of the object. From the calibration results it seems that the dimensions are rather good, as a calibration using the object renders a fairly low error, when evaluated against another object. Even so it would add to the confidence of the results if both calibration and evaluation object could be measured independently beforehand.
Conclusions

The purpose of this thesis was to construct a calibration method for small fields of view that performs better and is easier to use compared to the COORDINATOR software.

The proposed method uses one scan of a three-dimensional object to optimise a transformation that takes a pseudo range image and transforms it into 3D and metric units. The method has been implemented and tested and in this final chapter some conclusions are presented and future work areas suggested.

7.1 Summary

All in all the calibration is robust and fairs well for all tested configurations. The initial parts with detection, matching and optimisation using control points have proven to be very stable and always resulted in suitable solutions. The method implemented accurately estimates the homography that calibrates the laser plane as well as compensates for lens distortion.

In the used implementation the optimisation step of the calibration is slow, close to being unacceptable. This is the main drawback of the method that otherwise would seem to be superior to the present calibration method.

The precision of the calibrated setup fair well in comparison to the COORDINATOR software providing a superior result for most of the tested configurations. The proposed method allows for estimation of the resolution along the translation direction and compensation of skew effects, things that the COORDINATOR is incapable of handling. Furthermore, one of the most important advantages is that there is no need to move the object manually in the laser field. This means that less is dependant on the person performing the calibration and that the calibration may be performed in tight spaces, still without the need for external light.
In order to get even better results there would most likely be a need to improve the quality of the scans. Either by changing the laser or by controlling the ambient light better. An error level of only 40 to 45 micrometer for the Z-axis while at the same time the noise is just above 30 micrometres must be considered good.

Under the circumstances that was evaluated, it would seem that an average error of about 15 micrometres is reasonable with respect to the X-axis, for close to all calibrations performed. This corresponds to an error of about 0.4 pixels using Table 5.11. Using the same table and taking the standard configuration with a $30^\circ$ angle between the laser plane and the camera the same average error would be 0.6 pixels, provided that the calibration has an average error of about 52 micrometres in world units. In general the Y-axis calibration is about as good or slightly worse compared to the Z-calibration. Not a lot of effort was put into evaluating this axis as it was considered a bonus to some extent.

It is hard to give one good explanation for the remaining error after calibration. One source of this error is definitely the noise in the measurements themselves. As was shown in Table 5.10 the noise have the same order of magnitude as the error itself, it would be impossible to get a calibration that far supersed the noise level. Comparing synthetic data, with lens distortion added, to real data shows the Z-axis going from consistently being better than the X-axis to being worse in real data. This says something about the role of the noise in the measurement. It is also possible that part of the remaining error comes from uncertainties in both the calibration and evaluation objects used.

Some of the demands on the method was to perform the calibration without external light, make use of a simple and small calibration object, estimate the resolution along the Y-axis and to find the tilt of the laser plane, all which have been achieved in the method proposed in this report.

In addition the calibration should be performed in one step. Currently the only thing the user has to provide is one full scan of the calibration object and the calibration method will solve the rest. There also was a requirement on scalability that is likely to be fulfilled as there are no limitations in the method itself, a simple change in the specification of the calibration object dimensions should be enough to adjust for this. Although this was never tested.

### 7.2 Future Work

The implementation presented in this report is a complete calibration method for calibrating a laser triangulating camera system. However, there are still problems that might be interesting to look into.

- The implemented method needs a speed improvement in areas of intense optimisation. In order to achieve this it would be suitable to make use of a better optimisation method. The Nelder-Mead[9], or downward simplex method, implemented by MATLABs fminsearch is old and in comparison to modern methods rather stupid.
A suggestion would be to try and use Levenberg-Marquardt optimisation instead and in that case attempt to calculate the Jacobian matrix with derivatives for each parameter. This would likely increase the rate of convergence dramatically.

- In the parts where plane intersections are computed there seems to be an error in the corner point positions found. Currently all points thought to be part of a plane after 2D segmentation are used for estimation of the plane equation. It is plausible that some amount of these points actually are outliers. In this case possible plane outliers do not affect the final calibration as the corner points of the calibration object are only used initially. However, the rate of convergence would perhaps be higher if the accuracy of the planes, and thus the corner points, could be improved.

One rather simple thing to try would be to re-estimate the planes from the best plane inliers, thus removing the noisiest points when determining plane coefficients.

- To make the method accessible for users it is necessary to package it and automate it in a GUI. Work was started on this using MATLAB as a basis. However, it would probably be better to move over to C/C++ to get better performance and portability.
For estimation of homographies it is often useful to define a vector $\mathbf{h}$ from $\mathbf{H}$ like in

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}, \quad (A.1)$$

$$\mathbf{h} = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}^T. \quad (A.2)$$

### A.1 Homography from Points

For two corresponding points $\mathbf{x}$ and $\mathbf{y}$ reformulate (2.23) to

$$\mathbf{x} \times \mathbf{H} \mathbf{y} = 0. \quad (A.3)$$

By using (A.2) this cross product is equivalent to (A.4). This equation is over-determined, and by dropping row three and assuming normalised homogeneous points the final matrix for one correspondence is found, see (A.5).

$$\begin{bmatrix} \mathbf{\hat{o}}^T & -x_3 \mathbf{y}^T & x_2 \mathbf{y}^T \\ x_3 \mathbf{y}^T & \mathbf{\hat{o}}^T & -x_1 \mathbf{y}^T \\ -x_2 \mathbf{y}^T & x_1 \mathbf{y}^T & \mathbf{\hat{o}}^T \end{bmatrix} \mathbf{h} = 0, \quad (A.4)$$

$$\begin{bmatrix} \mathbf{\hat{o}}^T & -\mathbf{y}^T & x_2 \mathbf{y}^T \\ \mathbf{y}^T & \mathbf{\hat{o}}^T & -x_1 \mathbf{y}^T \end{bmatrix} \mathbf{h} = 0. \quad (A.5)$$

By stacking rows from different point correspondences and then solving the total equation system using singular value decomposition (SVD), the best fit for $\mathbf{h}$ can be found. Finally reshape $\mathbf{h}$ into $\mathbf{H}$ to get the homography, see (A.1) and (A.2).
This solution to the problem is referred to as the direct linear transform, or DLT for short.

Using the SVD for solving these equations is referred to in literature as the homogeneous solution. As an alternative, there also exists an inhomogeneous solution. The inhomogeneous method involves solving the least squares problem subject to an additional constraint on $\mathbf{h}$. To be more exact one element of $\mathbf{h}$ is set to one. This fixes the scaling of $\mathbf{H}$, and as $\mathbf{H}$ only can be determined up to an unknown scale factor this is perfectly reasonable. The rest of the equations may then be solved using, for example Gaussian elimination. However, if the true value of the fixed element is close to zero no scaling of the estimated $\mathbf{H}$ will provide a stable solution (see [5] for details).

### A.2 Normalisation

In order to avoid certain ill-conditioned situations it is good to implement a normalisation of data as a pre-processing step before computing a homography. The most well accepted way of doing this was presented by Hartley [5].

#### A.2.1 Normalisation for Points

For normalisation all points in both images are translated to have the total mean at the origin. After this they are scaled so that the total mean distance from points to the origin equals $\sqrt{2}$. In (A.6) and (A.7) points in image one and two are stored in $\mathbf{X}$, $\mathbf{Y}$ and their corresponding transformations are $\mathbf{N}_x$, $\mathbf{N}_y$ respectively. Note that in this explanation a general homography estimation is considered. In the final implementation, distortion will be considered outside of these equations.

\[
\mathbf{X}' = \mathbf{N}_x \mathbf{X}, \quad \text{(A.6)}
\]
\[
\mathbf{Y}' = \mathbf{N}_y \mathbf{Y}. \quad \text{(A.7)}
\]

The homography $\mathbf{H}'$ mapping $\mathbf{X}'$ and $\mathbf{Y}'$ by $\mathbf{Y}' = \mathbf{H}' \mathbf{X}'$ may then be estimated from at least four point correspondences. The final homography is found by using

\[
\mathbf{H} = \mathbf{N}_y^{-1} \mathbf{H}' \mathbf{N}_x. \quad \text{(A.8)}
\]
Figure B.1: Perspective (ISO)
Figure B.2: Front and back view

Table B.1: Object point coordinates

<table>
<thead>
<tr>
<th>Point</th>
<th>X (mm)</th>
<th>Y (mm)</th>
<th>Z (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>-30</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>-20</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>-20</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>-10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>31</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>-9</td>
<td>31</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>-5</td>
<td>15</td>
<td>10</td>
</tr>
</tbody>
</table>
Table B.2: Plane and point connectivity

<table>
<thead>
<tr>
<th>Plane</th>
<th>Point Neighbours</th>
<th>Point</th>
<th>Intersecting Planes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1, 2, 6</td>
<td>1</td>
<td>1, 7, 8, 11</td>
</tr>
<tr>
<td>2</td>
<td>2, 3, 6</td>
<td>2</td>
<td>1, 2, 11</td>
</tr>
<tr>
<td>3</td>
<td>6, 3, 9, 8</td>
<td>3</td>
<td>2, 3, 4, 11</td>
</tr>
<tr>
<td>4</td>
<td>3, 4, 10, 9</td>
<td>4</td>
<td>4, 5, 6, 11</td>
</tr>
<tr>
<td>5</td>
<td>4, 7, 11, 10</td>
<td>5</td>
<td>6, 7, 11</td>
</tr>
<tr>
<td>6</td>
<td>4, 5, 7</td>
<td>6</td>
<td>1, 2, 3, 8, 9</td>
</tr>
<tr>
<td>7</td>
<td>5, 1, 7</td>
<td>7</td>
<td>5, 6, 7, 8, 9</td>
</tr>
<tr>
<td>8</td>
<td>1, 6, 7</td>
<td>8</td>
<td>3, 9, 10</td>
</tr>
<tr>
<td>9</td>
<td>6, 8, 11, 7</td>
<td>9</td>
<td>3, 4, 10</td>
</tr>
<tr>
<td>10</td>
<td>8, 9, 10, 11</td>
<td>10</td>
<td>4, 5, 10</td>
</tr>
<tr>
<td>11</td>
<td>1, 2, 3, 4, 5</td>
<td>11</td>
<td>5, 9, 10</td>
</tr>
</tbody>
</table>
Bibliography


Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnt som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Daniel Rydström