

Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

**A calibration method for laser-triangulating 3D
cameras**

Examensarbete utfört i Bildbehandling
vid Tekniska högskolan i Linköping
av

Robert Andersson

LITH-ISY-EX--08/4144--SE

Linköping 2008



Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

A calibration method for laser-triangulating 3D cameras

Examensarbete utfört i Bildbehandling
vid Tekniska högskolan i Linköping
av


Robert Andersson

LITH-ISY-EX--08/4144--SE

Handledare: **Henrik Turbell**
SICK|IVP
Björn Johansson
ISY, Linköpings universitet

Examinator: **Klas Nordberg**
ISY, Linköpings universitet

Linköping, 01 December, 2008

	Avdelning, Institution Division, Department Computer Vision Laboratory Department of Electrical Engineering Linköpings universitet SE-581 83 Linköping, Sweden		Datum Date 2008-12-001
	Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____	ISBN _____ ISRN LITH-ISY-EX--08/4144--SE Serietitel och serienummer ISSN Title of series, numbering _____
URL för elektronisk version http://www.cvl.isy.liu.se http://urn.kb.se/resolve?urn=nbn:se:liu:diva-15735			
Titel Title En kalibreringsmetod för lasertriangulerande 3D-kameror A calibration method for laser-triangulating 3D cameras			
Författare Robert Andersson Author			
Sammanfattning Abstract <p>A laser-triangulating range camera uses a laser plane to light an object. If the position of the laser relative to the camera as well as certain properties of the camera is known, it is possible to calculate the coordinates for all points along the profile of the object. If either the object or the camera and laser has a known motion, it is possible to combine several measurements to get a three-dimensional view of the object.</p> <p>Camera calibration is the process of finding the properties of the camera and enough information about the setup so that the desired coordinates can be calculated. Several methods for camera calibration exist, but this thesis proposes a new method that has the advantages that the objects needed are relatively inexpensive and that only objects in the laser plane need to be observed. Each part of the method is given a thorough description. Several mathematical derivations have also been added as appendices for completeness.</p> <p>The proposed method is tested using both synthetic and real data. The results show that the method is suitable even when high accuracy is needed. A few suggestions are also made about how the method can be improved further.</p>			
Nyckelord Keywords camera calibration, range camera, laser triangulation, lens distortion, homography			

Abstract

A laser-triangulating range camera uses a laser plane to light an object. If the position of the laser relative to the camera as well as certain properties of the camera is known, it is possible to calculate the coordinates for all points along the profile of the object. If either the object or the camera and laser has a known motion, it is possible to combine several measurements to get a three-dimensional view of the object.

Camera calibration is the process of finding the properties of the camera and enough information about the setup so that the desired coordinates can be calculated. Several methods for camera calibration exist, but this thesis proposes a new method that has the advantages that the objects needed are relatively inexpensive and that only objects in the laser plane need to be observed. Each part of the method is given a thorough description. Several mathematical derivations have also been added as appendices for completeness.

The proposed method is tested using both synthetic and real data. The results show that the method is suitable even when high accuracy is needed. A few suggestions are also made about how the method can be improved further.

Acknowledgments

I would like to express my gratitude to everybody at SICK|IVP, especially Henrik Turbell who, as my supervisor, guided my work and provided much valuable input. I would also like to thank my second supervisor Björn Johansson and my examiner Klas Nordberg at ISY, Linköping University, for providing feedback on my work and taking the time to investigate possible alternative solutions.

Contents

1	Introduction	1
1.1	Background	2
1.2	Purpose	3
1.3	Outline	3
1.4	Nomenclature	3
2	Basic theory	5
2.1	Mathematical background	5
2.1.1	The pinhole camera	5
2.1.2	Homogeneous coordinates	6
2.2	Projection chain	7
2.2.1	Perspective projection	7
2.2.2	Lens distortion	9
2.2.3	Intrinsic parameters	10
2.3	3D cameras and laser triangulation	11
2.4	Camera calibration	12
2.4.1	Zhang's method	12
2.4.2	Lined-based correction of radial lens distortion	13
2.4.3	Direct Linear Transformation	13
3	Calibration method	15
3.1	Lens distortion	17
3.2	Homography	21
3.2.1	Single pose	21
3.2.2	Multiple poses	24
3.2.3	Arbitrary rotation	25
3.3	Method summary	29
4	Experiments	31
4.1	Lens distortion	31
4.1.1	Synthetic data	31
4.1.2	Real data	32
4.2	Homography estimation	33
4.2.1	Synthetic data	35
4.2.2	Real data	36

5	Conclusions and Future Work	39
5.1	Conclusions	39
5.2	Future work	40
	Bibliography	43
A	The lens distortion model's independence on scale and translation	45
A.1	Scale	45
A.2	Translation	46
B	Calculating angles from line distances	48
C	Least squares estimate of two parallel lines	51
D	Error map generation	54
D.1	Finding a vector from a set of projections	56
D.2	Using weighted projections	58

Chapter 1

Introduction

A regular camera measures the intensity of each pixel on the image sensor. These intensity values correspond to certain points in the world. However, the positions of these points are not known.

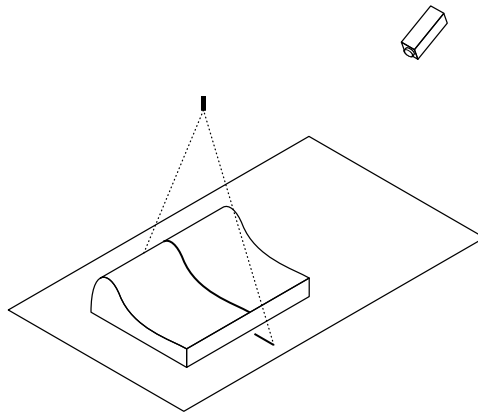


Figure 1.1. Camera setup

A range camera measures the distance to objects. Several techniques exist for measuring distance.[7] The type of range cameras discussed in this thesis use a laser with a fixed position and orientation relative to the camera. The laser projects a sheet of light (from now on referred to as a laser plane) rather than a normal laser beam. The intersection between a measured object and the laser plane is a curve that is visible to the camera. The shape of this curve is used to determine the height of a cross section of the object (referred to as a profile). As the object moves through the laser plane, several profiles are taken and eventually a complete distance map or a 3D model can be generated.

1.1 Background

For an arbitrary setup it is not possible to translate the profile observed by the camera into real world coordinates. There are several reasons why this is not possible:

- **Unknown scale:** There is no way to know the scale of the coordinate system. The object in the view may be large and far away or small and close to the camera.
- **Unknown perspective distortion:** Some parts of the laser plane are closer to the camera than the others. The part of the laser plane observed by the camera is therefore not a rectangle but a trapezium (see figure 1.2).
- **Unknown lens distortion:** The lens used in the camera is not perfect but introduces further distortions. A line in the laser plane is generally not projected to a line on the camera sensor (see figure 1.3).

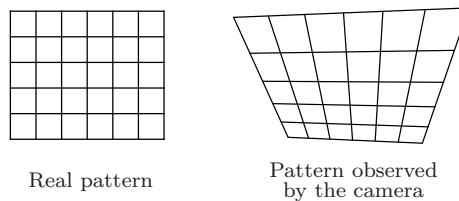


Figure 1.2. Perspective distortion

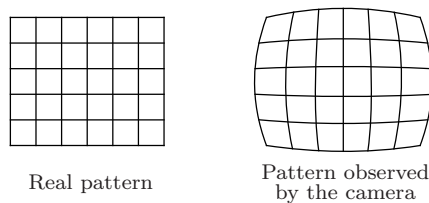


Figure 1.3. Lens distortion

The process of finding a way to translate sensor coordinates into world coordinates is called calibration. There are several techniques for calibrating range cameras. The most naive (but also the most general) approach is to make many observations of an object with known positions to build a translation table. The main limitation with this approach is that it is difficult and expensive to build a rig that can place an object at known positions with good enough precision.

1.2 Purpose

The purpose of this thesis is to construct, implement and evaluate a calibration method that requires only an object with known shape, not known position or rotation.

The goal is to satisfy the following requirements:

- It should handle all unknowns (scale, perspective transformation and lens distortion) described in the previous section.
- Low cost
- Simple to use
- Only objects in the laser plane can be observed

The motivation behind the last requirement is that sometimes the range cameras are equipped with an optical filter that will make anything not lit by the laser very hard to see. It is therefore desirable that only the profile of the laser is used in the calibration method.

A requirement on accuracy has deliberately been left out of the list above. While accuracy is very important it is hard to set constraints on accuracy. Instead the purpose of this project is to investigate the accuracy of the method chosen.

1.3 Outline

In **chapter 2** we give an introduction to the basic theory needed to comprehend the rest of the material in this thesis. In **chapter 3** we describe the chosen calibration method along with the mathematics necessary to use it. **Chapter 4** contains a description of the experiments performed and the results from these experiments. Finally in **chapter 5** we draw some conclusions and suggest areas where further work can be done.

1.4 Nomenclature

The following conventions are used throughout this thesis.

- | | |
|------------------------------|--|
| [] | Homogeneous coordinates |
| () | Cartesian coordinates |
| a , M | Vectors and matrices (bold font) |
| $\mathbf{x} \sim \mathbf{y}$ | Similarity. \mathbf{x} is equal to \mathbf{y} up to a non-zero scale factor. |

$\mathbf{x} \times \mathbf{y}$	Vector cross product	
$\mathbf{x} \cdot \mathbf{y}$	Dot product	
\mathbf{x}^T	The transpose of the vector \mathbf{x}	
\mathbf{A}^{-1}	The inverse of the matrix \mathbf{A}	
\mathbf{A}^{-T}	The inverse of the transpose of \mathbf{A}	
$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$	Object plane/laser plane coordinates	[m]
$\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \end{bmatrix}$	Translated and rotated object plane/laser plane coordinates	[m]
$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$	Image plane coordinates	[m]
$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix}$	Distorted image plane coordinates	[m]
$\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}$	Sensor coordinates	[pixels]

Chapter 2

Basic theory

In order to construct a camera calibration method it is important to first understand the basics of how an object is projected onto the image sensor inside the camera. In this chapter we give an introduction to that process. We also present relevant existing methods.

2.1 Mathematical background

This section discusses some of the mathematical background needed before moving on to describing the complete projection process.

2.1.1 The pinhole camera

The simplest approximation of a camera is a pinhole camera. In this model the lens is substituted for an infinitesimally small hole located in the focal plane through which all light must pass. The image of the depicted object is projected onto the image plane.

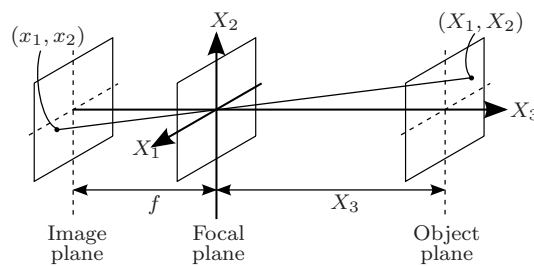


Figure 2.1. The pinhole camera model. f is the focal length. X_3 is the principal axis.

The 2-D image coordinates for a point in the image plane relates to the 3-D

world coordinates as

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = -\frac{f}{X_3} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}. \quad (2.1)$$

Looking at this equation, three things are obvious:

- A point in the image does not correspond to a unique point in the world as there are three unknowns on the right hand side. In figure 2.1 it is shown that a point in the image corresponds to a straight line in the world.
- Objects further away from the camera (larger X_3) are depicted smaller on the image plane. This is the cause of perspective distortion.
- The focal length of the camera, f , determines the scale of the depicted objects.

2.1.2 Homogeneous coordinates

In geometry we usually work with points in Euclidean space (\mathbb{R}^n) using normal Cartesian coordinates. However, in order to easily represent such things as projection and translation using matrix multiplications we need to change the representation of a point in space. By working in projective space (\mathbb{P}^n) instead, that can be accomplished. The coordinates that represent points in the projective space are called homogeneous coordinates. The homogeneous coordinates extend the Cartesian by adding an extra coordinate. The simplest case is when we have a 1-D Euclidean space. If we have the Cartesian coordinate $\mathbf{y} = (y_1)$ the corresponding homogeneous coordinates are

$$\mathbf{x} = \begin{bmatrix} x_1 \\ w \end{bmatrix} = t \begin{bmatrix} y_1 \\ 1 \end{bmatrix}.$$

Note that even though the homogeneous coordinates are a 2-vector they represent a point in the one-dimensional projective space. Throughout this thesis homogeneous coordinates will be written in square brackets and Cartesian coordinates in parentheses. The homogeneous coordinates have some unusual properties:

- Two vectors that only differ in a nonzero scale (t) are considered equivalent. The equivalence is denoted by \sim .
- A special case is a homogeneous vector with $w = 0$. They are considered to correspond to infinite Cartesian coordinates.

Figure 2.2 illustrates the relationship between the homogeneous and Cartesian coordinates. From the figure we can see that the Cartesian coordinates correspond to the intersection between the line $w = 1$ and another line that passes through the homogeneous point and the origin. All homogeneous points that lie on the same line through the origin are equivalent. It is also evident that a line through a homogeneous point with $w = 0$ will never intersect the $w = 1$ line. However as w approaches 0 the intersection point approaches infinity in the x_1 direction.

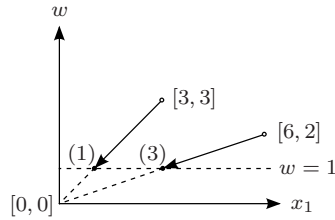


Figure 2.2. Relation between Cartesian (parentheses) and homogeneous coordinates (square brackets)

Homogeneous coordinates are easily generalized into higher dimensions. The corresponding homogeneous coordinates for two-dimensional Cartesian coordinates $\mathbf{y} = (y_1, y_2)^T$ are

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ w \end{bmatrix} = t \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix}.$$

Using homogeneous coordinates it is possible to write the relation between world coordinates and image coordinates in equation 2.1 using matrix notation.

$$\begin{bmatrix} x_1 \\ x_2 \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/f \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3/f \end{bmatrix} \sim \begin{bmatrix} fX_1/X_3 \\ fX_2/X_3 \\ 1 \end{bmatrix} \quad (2.2)$$

Note that this is a mapping from a three-dimensional Euclidean space (\mathbb{R}^3) to a two-dimensional projective space (\mathbb{P}^2).

Homogeneous coordinates are described briefly and used extensively by Hartley and Zisserman[3]. A more in-depth description is given by Bloomenthal[1].

2.2 Projection chain

The projection of depicted objects onto the image sensor can mathematically be divided into several steps, shown in figure 2.3. Each step is described in this section.

2.2.1 Perspective projection

In the previous section it was shown that the pinhole camera projection in section 2.1.1 could be written as a matrix multiplication using homogeneous coordinates. However, this case is unrealistically simple because the object and image plane are parallel. In real applications an object plane can have an arbitrary position and rotation.

If we introduce a 2-D coordinate system on the object plane it is possible to find a matrix that maps points in the object plane to points in the image

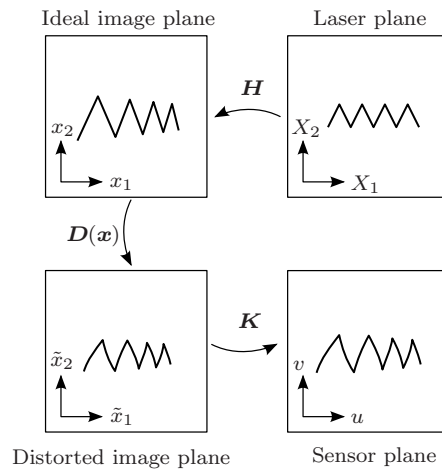


Figure 2.3. All parts of the projection chain

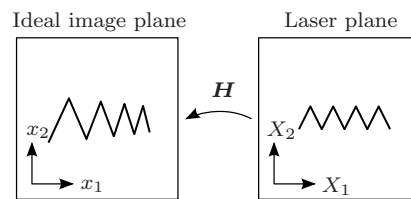


Figure 2.4. Perspective projection

plane. The mapping is a projectivity and it can be represented by a three by three matrix called a homography matrix.[3] Note how this is different from the previous section as the mapping is between two two-dimensional projective spaces. The homography matrix is usually denoted \mathbf{H} and has the following structure:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}. \quad (2.3)$$

Object plane coordinates (\mathbf{X}) are transformed into image plane coordinates (\mathbf{x}) by multiplying with the homography matrix.

$$\mathbf{x} \sim \mathbf{H}\mathbf{X} \quad (2.4)$$

Because we are working with homogeneous coordinates, the scale of the resulting coordinates are not important. It is therefore apparent that only the ratios of the individual matrix elements in \mathbf{H} are important and the scale can be chosen arbitrarily. Left are eight degrees of freedom that describe how points on the object plane are mapped to the image plane. To be able to compare different homography matrices they need to be normalized so that their parameters have the same scale. This can be done by assuming that $h_{33} = 1$ and multiply the matrix accordingly. It is not always safe to assume that, however. In particular if the origin of the object plane is mapped to infinite coordinates in the sensor plane h_{33} should be zero. For the setups discussed in this thesis that should not happen. The origin of the object plane is assumed to be in the camera's field of view.

2.2.2 Lens distortion

As briefly mentioned in section 1.1 the lens in the camera generally introduces further distortions. They arise from the fact that the lens does not work like a simple pinhole. In theory the lens distortion can be an arbitrary mapping. In practice, however, it has been shown that a function that for each point scales the radius from the distortion center according to a polynomial works well.[2] Ma et al[8] suggest several distortion functions and compare their accuracy. Of particular interest for this thesis is a family of functions first proposed in the book *Manual of Photogrammetry*[11], also described by Heikkilä[5]. Heikkilä describes an undistortion function that maps distorted coordinates ($\tilde{\mathbf{x}}$) to undistorted (\mathbf{x}):

$$\mathbf{x} = \tilde{\mathbf{x}} + \mathcal{F}_D(\tilde{\mathbf{x}}, \boldsymbol{\delta}) \quad (2.5)$$

where

$$\mathcal{F}_D(\tilde{\mathbf{x}}, \boldsymbol{\delta}) = \begin{bmatrix} \hat{x}_1(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ + (2p_1 \hat{x}_1 \hat{x}_2 + p_2(r^2 + 2\hat{x}_1^2))(1 + p_3 r^2 + \dots) \\ \hat{x}_2(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ + (p_1(r^2 + 2\hat{x}_2^2) + 2p_2 \hat{x}_1 \hat{x}_2)(1 + p_3 r^2 + \dots) \end{bmatrix}, \quad (2.6)$$

$$\hat{x}_1 = \tilde{x}_1 - \tilde{x}_{1C}, \quad \hat{x}_2 = \tilde{x}_2 - \tilde{x}_{2C}, \quad r = \sqrt{\hat{x}_1^2 + \hat{x}_2^2}, \quad (2.7)$$

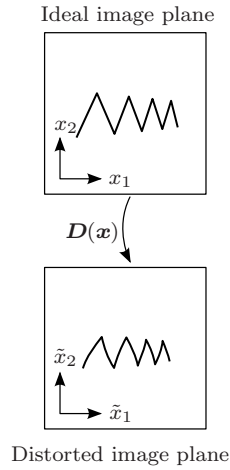


Figure 2.5. Lens distortion

and $\delta = [k_1, k_2, \dots, p_1, p_2, \dots, \tilde{x}_{1C}, \tilde{x}_{2C}]^T$. \tilde{x}_{1C} and \tilde{x}_{2C} define the distortion center. The coefficients k_1, k_2, \dots describe how the distortion changes with the distance from the distortion center. p_1, p_2, \dots describe the tangential or decentering part of the distortion that is dependent on the direction from the distortion center as well as the distance.[5] This part is usually substantially lower in magnitude than the purely radial part.[2]

2.2.3 Intrinsic parameters

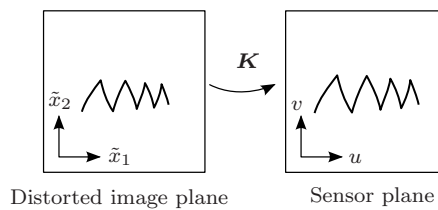


Figure 2.6. Intrinsic parameters

After the perspective projection and the lens distortion a few parameters still affect the image. They describe the shape and position of the sensor in the camera.

They can be arranged in a matrix as

$$\mathbf{K} = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

where $\mathbf{u} \sim \mathbf{K}\tilde{\mathbf{x}}$. α_u and α_v define the scale in the u and v directions. They translate the image plane units (m) into sensor units (pixels). s is a skew that is non-zero only if the sensor rows and columns are not perpendicular, or when taking a picture of a picture.[3] u_0 and v_0 are the coordinates of the principal point. Usually the principal point is near the center of the sensor, but it might be a bit offset due to sensor placement. If the sensor is not parallel to the image plane the matrix \mathbf{K} will be a new planar homography. Those cases are not covered further in this thesis.

2.3 3D cameras and laser triangulation

Figure 1.1 in chapter 1 shows what a typical setup looks like when using a laser triangulating 3D camera. A profile captured by the camera using that setup will look something like figure 2.7. It is apparent that the height of the curve in the profile corresponds to the height of the object.

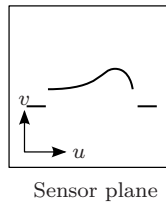


Figure 2.7. Camera profile

If all the parts of the projection chain described in section 2.2 are known it is possible to simply do the inverse of the steps in the chain to get the coordinates in the laser plane. That is

$$\mathbf{X} \sim \mathbf{H}^{-1}\mathbf{D}^{-1}(\mathbf{K}^{-1}\mathbf{u}) \quad (2.9)$$

where \mathbf{D}^{-1} is the inverse of the lens distortion function \mathbf{D} and $\mathbf{u} = (u, v)^T$. Because \mathbf{H} , \mathbf{D} and \mathbf{K} are generally not known we need to calibrate the camera before using it.

2.4 Camera calibration

In the previous section it was concluded that the transformations in the projection chain need to be known in order to calculate the object plane coordinates for features in an image. Several methods for camera calibration have been proposed. Some methods require three-dimensional calibration objects, such as two or three orthogonal planes, or multiple poses of a simpler object with a known movement.[12] Several methods disregard the lens distortion. That allows much faster algorithms but with a trade-off in accuracy. Since the purpose is to have as good accuracy as possible such methods are not suitable. Furthermore, since the calibration only needs to be done once for a particular camera setup computation speed is not a major concern.

2.4.1 Zhang's method

Zhang[12] presented a new method that only requires a two-dimensional pattern, such as a checkerboard, in two or more unknown positions. It calculates all parts of the projection chain, including lens distortion.

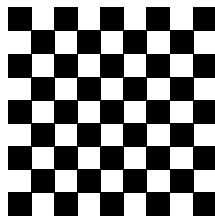


Figure 2.8. A pattern suitable for Zhang's calibration method

For each position a set of interest points are found in the pattern. Using the known relative distance between those points a separate homography \mathbf{H} and the intrinsic parameters \mathbf{K} are calculated explicitly. Next, the lens distortion parameters are estimated using a linear least-squares approach. Finally, all parameters (\mathbf{H} , \mathbf{K} and \mathbf{D}) are refined in a non-linear minimization step. Note that although the distortion model is similar to the one described in section 2.2.2 it is different in that it only handles radial distortion.

The main advantage of Zhang's method is that the calibration object can be constructed by printing the pattern using a laser printer and attaching it to any flat surface. The method fulfills all requirements stated in section 1.2 but one. It requires that objects that are not in the laser plane be observed. Even if it was possible to construct a calibration object that, when inserted into the laser plane, generated a pattern suitable for the method it would still not be possible to use the method, since it requires at least two poses that are not parallel (all patterns in the laser plane are parallel to each other).[12]

2.4.2 Lined-based correction of radial lens distortion

Prescott and McLean[10] proposed a method for calculating radial lens distortion using lines. The method automatically finds line segments in an image that are relatively straight. Using non-linear optimization, parameters that model the distortion are found. The distortion model is similar to the model described in section 2.2.2. However, like in Zhang's method, only the radial part of the distortion model is used.

While finding only the lens distortion is not enough for our problem, the method is still of interest. In the next chapter it is shown how a similar method can be used to estimate the lens distortion before the other parts of the projection chain are known.

2.4.3 Direct Linear Transformation

Another method of interest is the Direct Linear Transformation (DLT) method. It is a straightforward method for determining the homography matrix (\mathbf{H}) given corresponding points in the object plane (\mathbf{X}) and the ideal image plane (\mathbf{x}). While these point correspondences are not known in our case the method is still of use as will be shown in the next chapter. A brief description of the method is given here. Further details are given by Hartley and Zisserman[3] from which the following description has been borrowed. It is presented here with adapted notation.

We know that the homography matrix maps points on the object plane to the ideal image plane:

$$\mathbf{x} \sim \mathbf{H}\mathbf{X}. \quad (2.10)$$

This similarity basically states that the two vectors are parallel but can have different length. The condition that they should be parallel can be expressed by the vector cross product. Equation (2.10) can be rewritten as

$$\mathbf{x} \times \mathbf{H}\mathbf{X} = \mathbf{0}. \quad (2.11)$$

If we denote the j -th row of \mathbf{H} by \mathbf{h}_j^T , we can write

$$\mathbf{H}\mathbf{X} = \begin{bmatrix} \mathbf{h}_1^T \mathbf{X} \\ \mathbf{h}_2^T \mathbf{X} \\ \mathbf{h}_3^T \mathbf{X} \end{bmatrix}. \quad (2.12)$$

If we write $\mathbf{x} = [x_1, x_2, x_3]^T$ the cross product can be written as

$$\mathbf{x} \times \mathbf{H}\mathbf{X} = \begin{pmatrix} x_2 \mathbf{h}_3^T \mathbf{X} - x_3 \mathbf{h}_2^T \mathbf{X} \\ x_3 \mathbf{h}_1^T \mathbf{X} - x_1 \mathbf{h}_3^T \mathbf{X} \\ x_1 \mathbf{h}_2^T \mathbf{X} - x_2 \mathbf{h}_1^T \mathbf{X} \end{pmatrix}. \quad (2.13)$$

We can represent that in matrix form and rewrite (2.11) as

$$\begin{pmatrix} \mathbf{0}^T & -x_3 \mathbf{X}^T & x_2 \mathbf{X}^T \\ x_3 \mathbf{X}^T & \mathbf{0}^T & -x_1 \mathbf{X}^T \\ -x_2 \mathbf{X}^T & x_1 \mathbf{X}^T & \mathbf{0}^T \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = \mathbf{0}. \quad (2.14)$$

The matrix in this equation is a 3×9 matrix which we denote \mathbf{A}_i . The vector on the left hand side is a 9-vector that we denote \mathbf{h} . Since only two rows of \mathbf{A}_i are linearly independent the third row can be omitted.[3] The equation is now

$$\begin{pmatrix} \mathbf{0}^T & -x_3\mathbf{X}^T & x_2\mathbf{X}^T \\ x_3\mathbf{X}^T & \mathbf{0}^T & -x_1\mathbf{X}^T \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = \mathbf{0}. \quad (2.15)$$

We can see that we have two independent equations but eight degrees of freedom in \mathbf{h} . To solve \mathbf{h} completely we need at least four corresponding points where no three points are collinear. We stack n matrices \mathbf{A}_i on top of each other to form a $2n \times 9$ matrix called \mathbf{A} . We now have the following equation:

$$\mathbf{A}\mathbf{h} = \mathbf{0} \quad (2.16)$$

Any attempt at finding \mathbf{h} by multiplying with the inverse (or pseudo-inverse) of \mathbf{A} will only yield the trivial solution $\mathbf{h} = \mathbf{0}$. To find a nonzero solution we need to obtain the Singular Value Decomposition (SVD) of \mathbf{A} . The singular vector corresponding to the smallest singular value is the solution \mathbf{h} . Once \mathbf{h} is found the homography \mathbf{H} is obtained by rearranging \mathbf{h} . More information on SVD can be found in the appendix of Multiple View Geometry by Hartley and Zisserman.[3] Hartley and Zisserman also propose an extension to the DLT algorithm where the data is normalized. The effect of the normalization is that the algorithm becomes much more stable to noise in the measurements. They write:

“Data normalization is an essential step in the DLT algorithm. It must not be considered optional.”

The modified algorithm works as follows:

1. Find transforms (\mathbf{T}_1 and \mathbf{T}_2) for \mathbf{x} and \mathbf{X} that translate their centroids to the origin and scales the data so that the average distance from the origin is $\sqrt{2}$.
2. Apply the DLT algorithm to the normalized data to compute $\hat{\mathbf{H}}$
3. Denormalize $\hat{\mathbf{H}}$ to obtain \mathbf{H} : $\mathbf{H} = \mathbf{T}_1^{-1}\hat{\mathbf{H}}\mathbf{T}_2$

Chapter 3

Calibration method

The basic problem with most existing methods described in section 2.4 is that they require observation of patterns or objects outside the laser plane. As stated in section 1.2, that is not desirable since the cameras sometimes are equipped with bandpass filters that make the objects not lit by the laser hard to see. There are three possible workarounds:

1. Temporarily remove the bandpass filter when calibrating the camera
2. Provide enough light to make the object visible through the bandpass filter
3. Use a method that only needs profiles in the laser plane

Removing the bandpass filter will alter the optical properties of the camera. How much they are altered is not investigated further in this thesis. Using enough light to make the object visible may also be possible but somewhat impractical. However, both these methods have a significant flaw. We need to find the homography between the laser plane and the ideal image plane. Using an approach like Zhang's[12] will require that we place the calibration object exactly where the laser plane will be. As that can be very difficult these methods will probably not have very good accuracy and will definitely not be easy to use.

In this thesis the third option will be investigated. Observing only objects in the laser plane, a 2D to 2D homography and the lens distortion of the camera will be estimated. The proposed calibration method is divided into two parts. First the lens distortion is estimated and then a plane to plane homography is estimated. The advantage of doing it in this order is that the lens distortion will be known and the images can be undistorted before estimating the homography.

It is not perfectly clear that we can estimate the lens distortion before the intrinsic parameters \mathbf{K} . Looking at the projection chain again in figure 3.1 we can see that before trying to undistort the image coordinates using \mathbf{D}^{-1} they should be transformed using \mathbf{K}^{-1} . However, a special circumstance occurs when \mathbf{K} is a shape preserving transformation. If \mathbf{K} preserves shape it can only scale and translate the coordinate system. In appendix A it is proven that the lens

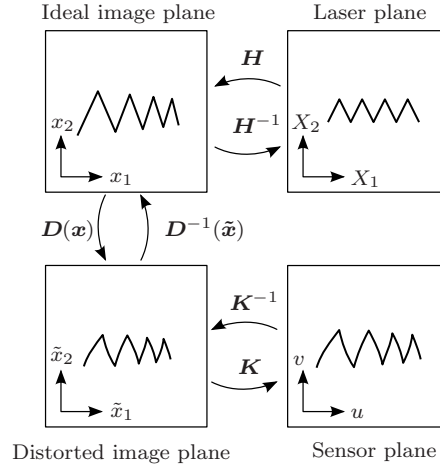


Figure 3.1. Projection chain with inverse functions

distortion model used can model the same distortion with or without \mathbf{K} , using different parameters.

When can we assume that \mathbf{K} is a shape preserving transformation? Looking at the structure of \mathbf{K} (2.8)

$$\mathbf{K} = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

we see that two conditions need to be met.

- The parameter s must be zero.
- The parameters α_u and α_v must be equal.

The first condition ($s = 0$) means that the rows and columns of the sensor chip must be orthogonal, which they are in most normal cameras.[3] The second condition ($\alpha_u = \alpha_v$) means that the pixels on the sensor chip must be square. That condition is not true for all sensor chips. In our case the cameras used have square pixels so the condition is fulfilled. The remaining properties of \mathbf{K} can now be incorporated into \mathbf{H} . \mathbf{H} already has parameters that scale and translate the coordinates, so no additional degrees of freedom are added. The altered projection chain is shown in figure 3.2.

A couple of things are worth noting:

- The homography and the lens distortion will not have the same parameters as if \mathbf{K} was used. For simplicity we keep the names \mathbf{H} and \mathbf{D} .

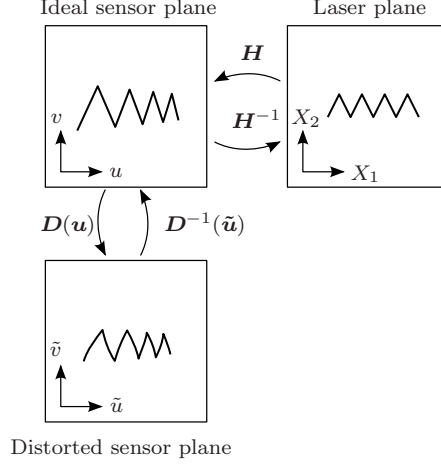


Figure 3.2. Projection chain without K

- The homography now maps object plane coordinates directly to sensor coordinates. The lens distortion is a transform in the sensor plane rather than the image plane.

3.1 Lens distortion

To find the function that undistorts the distorted coordinates, a number of profiles of straight lines are collected. The straight lines are generated by inserting a flat surface into the laser plane (see figure 3.3). Note that, due to the lens distortion, the straight lines will generally not be straight when viewed by the camera, as shown in figure 3.4. The goal is to find a function that makes these observed lines as straight as possible. In order to get good accuracy from the estimation, the inserted plane is moved and rotated for each profile.

The formula in equation (2.6) is used to describe the distortion. Two parameters are used to model the radial distortion and two to model the decentering (or tangential) distortion. In addition, two parameters are used to specify the distortion center. The formula transforms distorted coordinates to undistorted coordinates. It is given here with notation adapted to the rest of the thesis.

$$\mathbf{u} = \tilde{\mathbf{u}} + \mathcal{F}_D(\tilde{\mathbf{u}}, \boldsymbol{\delta}) \quad (3.1)$$

$$\mathcal{F}_D(\tilde{\mathbf{u}}, \boldsymbol{\delta}) = \begin{bmatrix} \hat{u}(k_1 r^2 + k_2 r^4) \\ + (2p_1 \hat{u} \hat{v} + p_2 (r^2 + 2\hat{u}^2)) \\ \hat{v}(k_1 r^2 + k_2 r^4) \\ + (p_1 (r^2 + 2\hat{v}^2) + 2p_2 \hat{u} \hat{v}) \end{bmatrix} \quad (3.2)$$

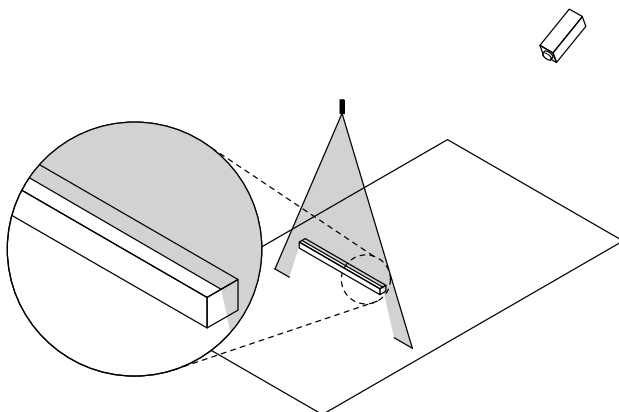


Figure 3.3. Calibration object for estimating the lens distortion

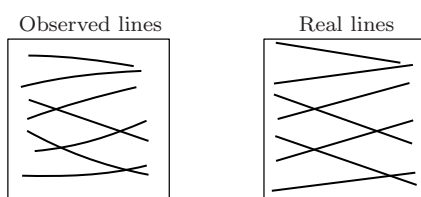


Figure 3.4. Example of lens distortion calibration data

where

$$\hat{u} = \tilde{u} - \tilde{u}_0, \quad \hat{v} = \tilde{v} - \tilde{v}_0, \quad r = \sqrt{\hat{u}^2 + \hat{v}^2}, \quad (3.3)$$

and $\boldsymbol{\delta} = [k_1, k_2, p_1, p_2, \tilde{u}_0, \tilde{v}_0]^T$. \tilde{u}_0 and \tilde{v}_0 define the distortion center in sensor coordinates (pixels). \mathcal{F}_D works in the opposite direction of \mathbf{D} :

$$\mathbf{D}^{-1}(\tilde{\mathbf{u}}) = \tilde{\mathbf{u}} + \mathcal{F}_D(\tilde{\mathbf{u}}, \boldsymbol{\delta}) \quad (3.4)$$

There exist no algebraic methods for calculating the parameters $\boldsymbol{\delta}$. Instead a numerical minimization method is used to find the parameters that best describe the actual distortion of a set of calibration coordinates.

Numerical minimization algorithms such as the Nelder-Mead[9] simplex algorithm works by finding parameters that minimize a cost function. There are several ways of constructing such a cost function. In some cases the calibration coordinates are known in both the ideal and distorted sensor plane (see figure 3.2). A simple approach is then to undistort the observed points given the parameters $\boldsymbol{\delta}$ and compare each calculated point to the corresponding real coordinates. A sum of the squared distances between the calculated points and the real ones can be used as a cost function. In our case the ideal sensor coordinates are unknown for two reasons. Firstly the homography is unknown, so given a set of coordinates in the laser plane there is no way of knowing how these points are projected onto the ideal sensor plane. Secondly the coordinates in the laser plane are unknown in the first place.

All we know about the homography is that it is a linear mapping. This means that straight lines in the laser plane are mapped to straight lines in the ideal sensor plane. This knowledge is enough to make it possible to construct a cost function given only a set of distorted coordinates and the fact that they are on a straight line in the laser plane. The distorted coordinates are undistorted according to $\boldsymbol{\delta}$. The cost is determined by how much the points deviate from a straight line after the undistortion.

How much a set of points deviates from a straight line can be measured in different ways. One way is to use the errors when trying to fit a straight line to the points. This measure has the disadvantage that parameters that place all points in a very small area after the undistortion will have a low cost no matter how well they describe a straight line. Figure 3.5 illustrates the problem.



Figure 3.5. A bad set of points with low cost when fitting a straight line

The way we chose shares some aspects with principal components analysis (PCA, also called Karhunen-Loève transformation) as it uses the eigenvalues of the covariance matrix to determine the cost. The largest eigenvalue, λ_1 , is the variance in the principal direction (the direction with the largest variance) and the smallest eigenvalue, λ_2 , is the variance in the direction orthogonal to the principal direction.[7]

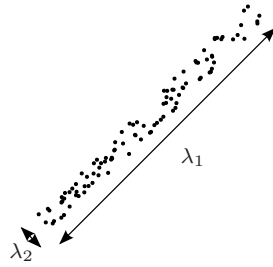


Figure 3.6. Eigenvalues of a general set of points

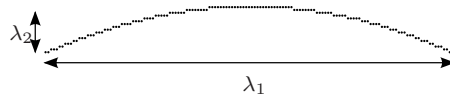


Figure 3.7. Eigenvalues of a curved line

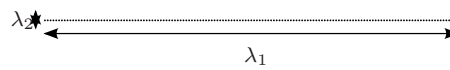


Figure 3.8. Eigenvalues of a straight line

If the points are on a straight line λ_1 will be large while λ_2 will be very small (0 in the ideal case). See figures 3.7 and 3.8 for an illustration. The cost function therefore uses the following cost:

$$\frac{\lambda_2}{\lambda_1} \quad (3.5)$$

This cost has the advantage that if the points are in a small area (both λ_1 and λ_2 are small), the cost will be relatively large (1 is the largest cost possible). The total cost for a set of parameters is the sum of the cost for each line undistorted using that set of parameters.

For more information on PCA see for instance Neural Networks by S. Haykin[4].

3.2 Homography

Now that the lens distortion is estimated it is possible to estimate the plane to plane homography (\mathbf{H}) that maps points in the laser plane (\mathbf{X}) to points in the ideal sensor plane (\mathbf{u}):

$$\mathbf{u} \sim \mathbf{H}\mathbf{X} \quad (3.6)$$

This section describe how \mathbf{H} can be estimated. The section is divided into several subsections. First we describe the basic method of finding \mathbf{H} given a single pose of a calibration object. Next we show a simple way of getting better results using multiple poses. Finally we give a description of how arbitrary rotations of the calibration object can be handled.

3.2.1 Single pose

To find the homography between the laser plane and the sensor plane we need to generate a set of points in the laser plane. To generate these points we need a more advanced object than the flat surface needed for the lens distortion. We have chosen to use a sawtooth object as shown in figure 3.9. The setup with the calibration object and the laser plane relative to the camera is shown in figure 3.10.

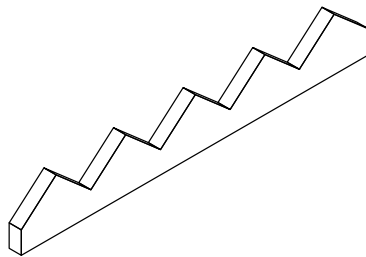


Figure 3.9. Calibration object for estimating the homography

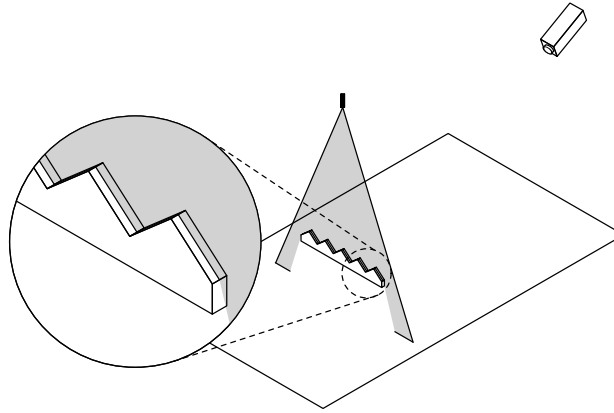


Figure 3.10. Setup for estimating the homography

For now, we assume that the calibration object is parallel to the laser plane. That means that the object can only be rotated around the axis orthogonal to the laser plane. Using the setup in figure 3.10 a captured profile will look like figure 3.11.

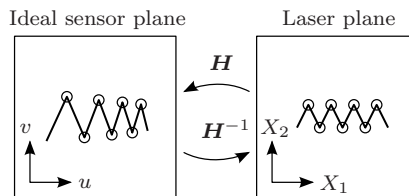


Figure 3.11. A single profile of the calibration object with highlighted peaks and valleys

The captured profile has many points all along the profile. However, only the peaks and valleys (illustrated by small circles in figure 3.11) of the profile can be used to estimate the homography since they are the only points where the corresponding points on the calibration object are known. Note that the position of these points are only known relative to the calibration object itself, not to the laser plane.

In order to get good results, the position of the peaks and valleys in the ideal sensor plane are found by finding the intersections between the lines along the sides of the teeth.

If the exact coordinates (\mathbf{X}) of the peaks and valleys in the laser plane were known, it would be possible to directly calculate \mathbf{H} using the Direct Linear Transformation described in section 2.4.3. However, since the position and rotation of the object is unknown, the exact coordinates for each point can not be determined.

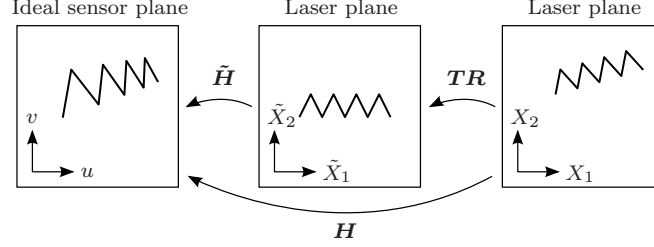


Figure 3.12. Relation between \tilde{H} and H

To overcome this problem we start by assuming a position and a rotation for the object. In our case we assume that the object is placed with the center in the origin of the laser plane and that it is rotated in a fixed way. $\tilde{\mathbf{X}}$ are the coordinates of the points on this object. It is then possible (using the Direct Linear Transformation method) to calculate a homography \tilde{H} that maps the assumed coordinates $\tilde{\mathbf{X}}$ in the laser plane to the measured coordinates \mathbf{u} in the ideal sensor plane.

$$\mathbf{u} \sim \tilde{H}\tilde{\mathbf{X}} \quad (3.7)$$

$\tilde{\mathbf{X}}$ is related to \mathbf{X} in the following way:

$$\tilde{\mathbf{X}} \sim TR\mathbf{X} \quad (3.8)$$

where \mathbf{R} is a matrix that rotates the real object to have the same rotation as the assumed object above and \mathbf{T} is a matrix that translates the rotated object to the origin of the laser plane.

From equations 3.6, 3.7 and 3.8 or simply looking at figure 3.12 we can see that:

$$\mathbf{H} = \tilde{H}TR \quad (3.9)$$

In order to fully determine \mathbf{H} we need to find \mathbf{T} and \mathbf{R} . If the position and orientation of the coordinate system is important, it can be fixed by asking the user to place an object that specifies where the origin is located (\mathbf{X}_o) and another point (\mathbf{X}_r) on the X_1 -axis (where $X_2 = 0$). \mathbf{u}_o is the projection of \mathbf{X}_o onto the ideal sensor plane:

$$\mathbf{u}_o \sim H\mathbf{X}_o \quad (3.10)$$

From equations 3.9 and 3.10 we see that:

$$\mathbf{u}_o \sim \tilde{H}TR\mathbf{X}_o \Rightarrow \tilde{H}^{-1}\mathbf{u}_o \sim TR\mathbf{X}_o \quad (3.11)$$

Since \mathbf{X}_o is the origin both X_1 and X_2 are zero and the rotation matrix \mathbf{R} has no effect. Equation 3.11 can therefore be rewritten as:

$$\tilde{H}^{-1}\mathbf{u}_o \sim T\mathbf{X}_o \quad (3.12)$$

As the only unknown is \mathbf{T} , and \mathbf{T} has only two degrees of freedom, it can be fully determined from equation 3.12. We define \mathbf{d} as:

$$\mathbf{d} \sim \tilde{\mathbf{H}}^{-1} \mathbf{u}_o \quad (3.13)$$

\mathbf{T} will then look like:

$$\mathbf{T} = \left[\begin{array}{cc|c} 1 & 0 & \mathbf{d} \\ 0 & 1 & \\ 0 & 0 & \end{array} \right] \quad (3.14)$$

Similarly if \mathbf{u}_r is the projection of \mathbf{X}_r onto the ideal sensor plane we can conclude that:

$$\tilde{\mathbf{H}}^{-1} \mathbf{u}_r \sim \mathbf{T} \mathbf{R} \mathbf{X}_r \Rightarrow \mathbf{T}^{-1} \tilde{\mathbf{H}}^{-1} \mathbf{u}_r \sim \mathbf{R} \mathbf{X}_r \quad (3.15)$$

All we know about \mathbf{X}_r is that $X_2 = 0$. This is enough since \mathbf{R} has only one degree of freedom. To find \mathbf{R} we need to determine the angle of rotation between the X_1 -axis and the vector

$$\mathbf{r} = \begin{bmatrix} r_{X1} \\ r_{X2} \\ 1 \end{bmatrix} = \mathbf{T}^{-1} \tilde{\mathbf{H}}^{-1} \mathbf{u}_r \quad (3.16)$$

This angle can be found by:

$$\theta = \arctan\left(\frac{r_{X2}}{r_{X1}}\right) \quad (3.17)$$

\mathbf{R} can now be determined:

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

Now that $\tilde{\mathbf{H}}$, \mathbf{T} and \mathbf{R} have been determined, \mathbf{H} can be calculated using equation (3.9):

$$\mathbf{H} = \tilde{\mathbf{H}} \mathbf{T} \mathbf{R}$$

For many applications the exact orientation of the coordinate system is not important and it can be chosen arbitrarily. In the experiments done as a part of this thesis we assume that the origin is in the point of the laser plane that corresponds to the middle column of the bottom row of the sensor and that the \mathbf{X}_1 axis is aligned with the bottom row of the sensor.

3.2.2 Multiple poses

Following the procedure in the last section we can calculate a homography, \mathbf{H} , given a single pose (and possibly some extra information to fix the rotation and translation of the coordinate system in the laser plane). However, using a single pose will likely lead to a bad result because the measurements contain noise. Using

multiple poses to estimate multiple homography matrices is simple but a method for averaging the matrices is needed.

The simplest averaging method is to normalize the matrices to have $h_{33} = 1$ and then construct a new matrix where each element is the average of the element from the given matrices. The problem with such an approach is that the averaging is done over an algebraic value rather than a geometric one. The approach used in this thesis is to synthetically generate at least four points in the laser plane coordinate system and use each given matrix to map them to the sensor plane. Since the different matrices will produce slightly different results each point in the laser plane will map to a point cloud in the sensor plane (see figure 3.13). For each of these point clouds an average point is calculated. A new homography is then calculated that maps the original points in the laser plane to the calculated average points in the sensor plane using the Direct Linear Transformation described in section 2.4.3 (see figure 3.14).

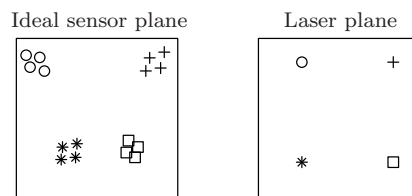


Figure 3.13. Four points mapped to the ideal sensor plane by four different homography matrices

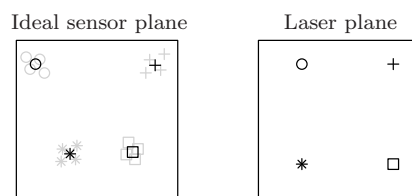


Figure 3.14. The calculated homography maps the points in the laser plane to the average of the point clouds in the ideal sensor plane.

3.2.3 Arbitrary rotation

Previously we have assumed that the calibration object is held parallel to the laser plane. That meant that we knew the distance between the points where the laser plane intersected the peaks and valleys of the object. However, in practice we cannot assume that the object is held exactly parallel to the laser plane. Having

that constraint would seriously complicate the calibration procedure. So what happens if the calibration object is not held parallel to the laser plane? Figure 3.15 illustrates the problem. The distances between the interest points (the peaks and valleys) of the profile is larger when the object is rotated. If we still assume that the object is held parallel to the laser plane, the calculated homography will be incorrect because we assume that the real distance between two points is smaller than it actually is.

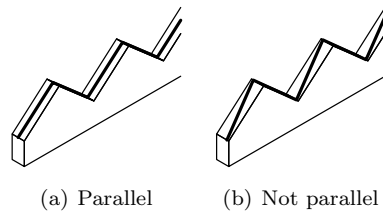


Figure 3.15. When the calibration object is not held parallel to the laser plane the distances between the peaks and valleys increase. The thick black line is the laser line.

We need to find some way to detect at what angles the object is rotated and compensate for it. In this thesis we have chosen to modify the calibration object to allow for us to detect the rotation angles. The modified object is shown in figure 3.16. We have added painted lines on the sides of the teeth. The lines all have known positions and angles. The fact that they are not orthogonal to the sides of object is what allows us to detect the rotation of the object.

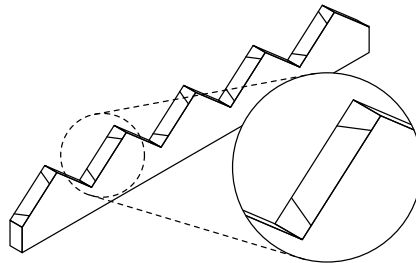


Figure 3.16. Object with tilted lines

The rest of this section describes how the tilted lines allow us to calculate the rotation of the object and how we compensate for the rotation. The intersection between the laser and the tilted lines will show up in the intensity values as they are much darker than the surrounding object. To calculate the rotation of the object we focus on one toothside at a time (shown in figure 3.17).

Along the side of the tooth a new coordinate system (x', y') is introduced (shown in figure 3.18). The points $x'_1-x'_4$ are used to specify the position of the tilted lines. These points are known. The width of the object (w) is also known.

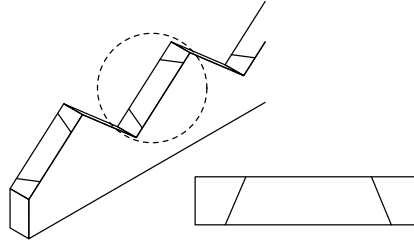


Figure 3.17. To calculate the rotation we focus on one side at a time

The angles α_1 and α_2 describe the angle of the tilted lines. They can be calculated using $x'_1-x'_4$:

$$\alpha_1 = \arctan\left(\frac{x'_1 - x'_2}{w}\right) \tag{3.19}$$

$$\alpha_2 = \arctan\left(\frac{x'_3 - x'_4}{w}\right) \tag{3.20}$$

Using the example in figure 3.18, α_1 will have a negative value while α_2 will be positive.

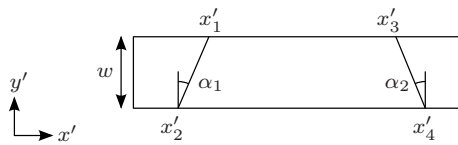


Figure 3.18. A local coordinate system is introduced.

As the laser hits the object two distances (d_1 and d_2 , shown in figure 3.19) are of particular interest. They can be measured using the range data and the intensity values.

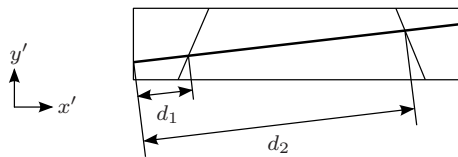


Figure 3.19. The distances d_1 and d_2 can be used to find the rotation. The thick black line is the laser.

The angle between the laser and the x' -axis is called θ (see figure 3.20). The point where the laser intersects the left side of the plane (a peak or a valley) is

determined by y'_1 . Neither θ nor y'_1 can be measured directly. They need to be calculated using the known values.

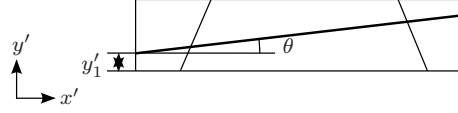


Figure 3.20. Unknowns that have to be calculated

In Appendix B it is shown that d_1 and d_2 depend on θ and y'_1 in the following way:

$$d_1 = \frac{x'_2 \cos(\alpha) - y'_1 \sin(\alpha)}{\cos(\theta - \alpha)} \quad (3.21)$$

$$d_2 = \frac{x'_4 \cos(\alpha) + y'_1 \sin(\alpha)}{\cos(\theta + \alpha)} \quad (3.22)$$

Note that it is assumed that $\alpha_1 = -\alpha_2$. α_1 and α_2 are substituted for α and $-\alpha$ respectively. It is also shown that θ can be calculated using the above equations:

$$\theta = -\text{sgn}(\alpha) \arcsin\left(\frac{x'_2 + x'_4}{R}\right) - \arctan\left(\frac{d_1 + d_2}{(d_1 - d_2) \tan(\alpha)}\right), \quad (3.23)$$

where $R = \sqrt{(d_1 + d_2)^2 + (d_1 - d_2)^2 \tan^2(\alpha)}$. When θ has been calculated, we can find y'_1 by:

$$y'_1 = \frac{x'_2 \cos(\alpha) - d_1 \cos(\theta - \alpha)}{\sin(\alpha)} \quad (3.24)$$

By calculating the value y'_1 for each visible tooth side a number of points where the laser intersects the peaks and valleys is found. Figure 3.21 shows a top view of the object where the calculated intersections with peaks are shown as crosses and the intersections with valleys are shown as circles.

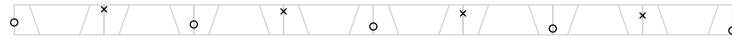


Figure 3.21. Intersections between the laser plane and the peaks and valleys on the calibration object (viewed from above)

In the ideal case with no measurement error the calculated intersections form two parallel straight lines, as shown in figure 3.22. In practice, because of measurement errors, the two sets of points will not describe two parallel straight lines. However, given two sets of points the two lines that fit the data best in a least squares sense can be calculated. How that calculation is performed is shown in appendix C.

When the two lines have been found the rotations around the two axes are easily found by looking at the slope parameter of the lines and the distance between

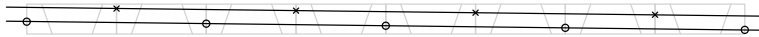


Figure 3.22. Lines through the intersections are parallel

the lines. Once the rotations have been calculated we can adjust the laser plane coordinates used when finding the homography.

One important note that has not yet been discussed is that the method for compensating for an arbitrarily rotated object described in this section does not work if the homography is unknown. The reason is that the distances d_1 and d_2 cannot be determined if the homography is unknown. However, since the impact of a rotated object is relatively small it is possible to calculate the homography matrix first without considering the rotation at all. Since the homography will be reasonably correct, the rotations can be estimated and a new homography can be calculated with compensation for the estimated rotation. The new calculated homography should be more correct than the original. The approach can be iterated until the elements in the homography has converged to fixed values.

3.3 Method summary

Figure 3.23 shows a flowchart model for the basic steps of the calibration method. The steps are briefly described here.

1. The user is asked to provide a number of straight profiles. This is done by holding a flat surface at different positions in the laser plane.
2. The provided profiles are used in a nonlinear estimation method to calculate a set of distortion parameters.
3. The user is asked to provide a number of profiles of the sawtooth shaped calibration object.
4. The distortion parameters calculated in step 2 are used to undistort the profiles of the calibration object.
5. The undistorted profiles are used to calculate the homography matrix.
6. If the homography has converged to a fixed value: Stop. Otherwise, use the calculated homography to estimate the rotation for each profile and use the estimated rotations to calculate a new homography matrix. Repeat as necessary.

In practice the convergence criterion in the last step can be exchanged for a fixed number of iterations in the loop.

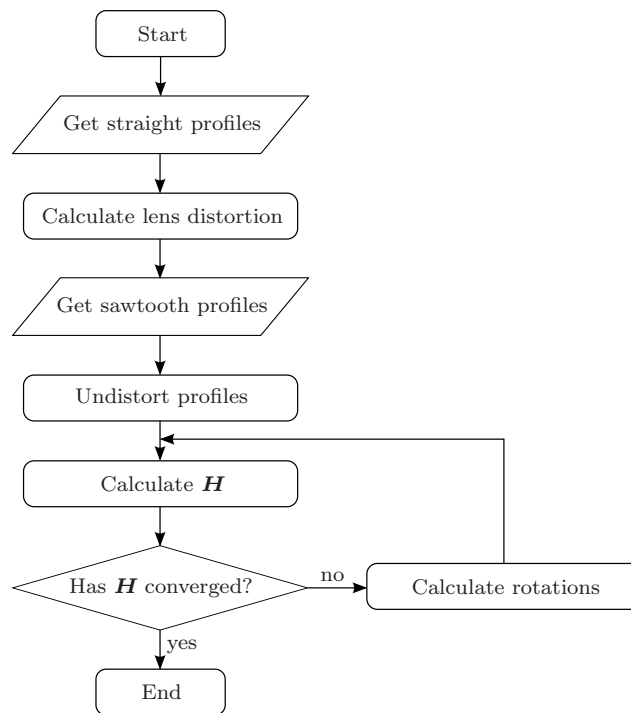


Figure 3.23. Complete calibration method

Chapter 4

Experiments

To test the calibration method the algorithms were implemented in Matlab. A small amount of c code was written to connect Matlab to a range camera. All other code was written in the Matlab language. The nonlinear minimization was implemented by using `fminsearch()` in Matlab, which in turn uses the Nelder-Mead simplex method.[6]

4.1 Lens distortion

To test the calibration method described in section 3.1 two experiments were made. In the first experiment synthetic data was used, while the second experiment used real data from a camera.

4.1.1 Synthetic data

The advantage of using synthetically created data is that the true coordinates of every distorted point is known. When data has been undistorted by the algorithm the result can be compared with the correct values. The disadvantage is that the chosen distortion model will always model the distortions perfectly (since the same model is used in the distortion and the undistortion of data).

The synthetic data is generated by randomly placing a number of lines in the ideal lens plane. These lines represent the intersection between the laser plane and a number of flat surfaces. To simulate that the detection of the laser profile in the camera is not perfect a certain amount of Gaussian noise is added to each point on the line. All lines are then distorted using a fixed set of distortion parameters.

The subsequent step uses the method described in section 3.1 to estimate the distortion parameters using nothing but the distorted lines. Once a set of estimated parameters is found, the distorted lines (distorted with the real parameters) are undistorted (using the estimated parameters). The result is compared to the original data and the mean error is calculated. To get a good estimate of the actual systematic error, the Gaussian noise is only added when calculating the

distortion parameters, not when calculating the error. Figure 4.1 shows how the error depends on the number of profiles used at different distortion levels.

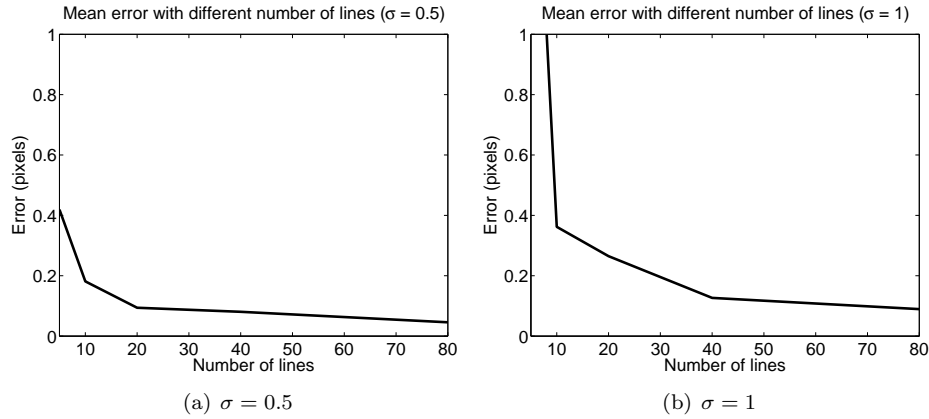


Figure 4.1. Residual error for different noise levels (σ is the standard deviation)

4.1.2 Real data

The method was tested with real data using five different lenses. With each lens 100 profiles were captured. These profiles were then used to calculate the distortion parameters in the same way as when using synthetic data.

The big difference between synthetic data and real data is that with real data there is no obvious way to calculate the residual error. The approach used here is to try to fit a straight line to each profile and calculate the average distance between the points and the fitted line. Note that this will generally underestimate the error as it is assumed that the position of the fitted line is correct. Lens distortion affects all points on a pose and the position of the fitted line will therefore generally not be the correct one. This is especially true when estimating the original errors. However, when calculating the residual errors, much of the distortion has been removed and therefore the fitted line is more correct. If the calculated error is considered a straightness factor instead of an absolute error, it will still be a useful metric for determining the performance of the algorithm.

Besides calculating a single mean error for the whole image plane, an error map is also generated that shows roughly where in the image the errors are located. See appendix D for a detailed description.

Table 4.1 shows the calculated mean deviation from a straight line before and after compensating for lens distortion.

Figures 4.2-4.6 show roughly where the errors are located on the image plane. Note that these error may both be from error in the distortion model and from noisy measurements. Some parts of the image have no distortion data available

(mostly near the edges). That is because there was not enough data available to estimate the error in these parts.

Lens	Original error	Residual error
1	0.609	0.051
2	0.229	0.069
3	0.500	0.044
4	0.201	0.070
5	0.272	0.081

Table 4.1. Original and residual error when estimating lens distortion

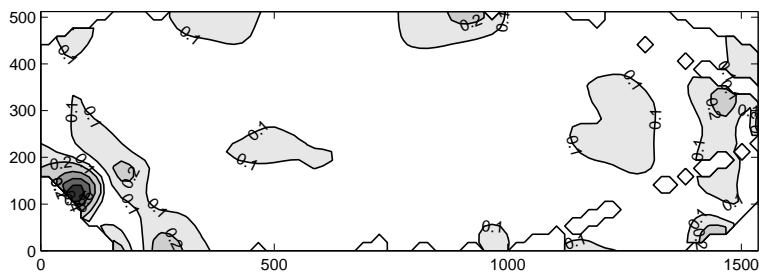


Figure 4.2. Residual error map for lens 1 (errors in pixels)

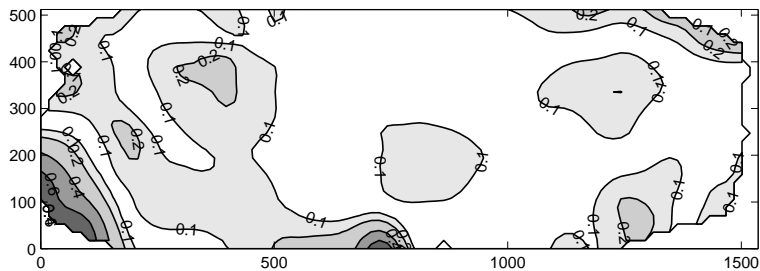


Figure 4.3. Residual error map for lens 2 (errors in pixels)

4.2 Homography estimation

To test the method described in section 3.2 two experiments were made. The method was first tested using synthetic data and then then using real data.

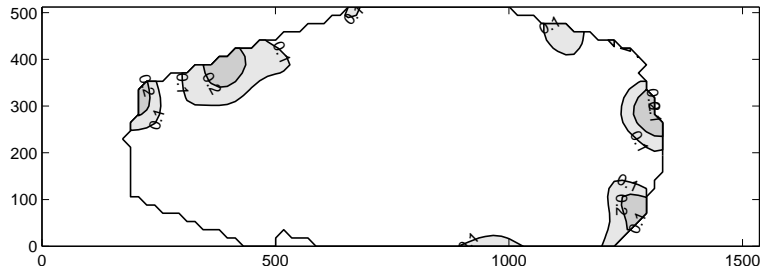


Figure 4.4. Residual error map for lens 3 (errors in pixels)

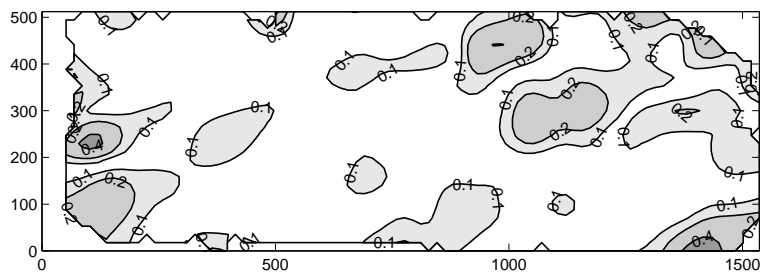


Figure 4.5. Residual error map for lens 4 (errors in pixels)

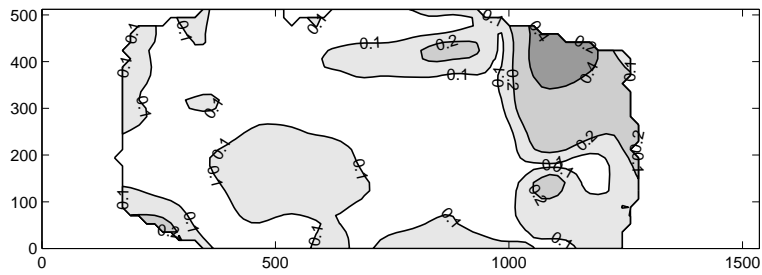


Figure 4.6. Residual error map for lens 5 (errors in pixels)

4.2.1 Synthetic data

Instead of generating a laser profile as in section 4.1 only the coordinates for certain interesting points were calculated. The interesting points are the positions of peaks and valleys in the laser plane and also the position of the intersections between the laser plane and the painted lines on the object. The advantage of doing it this way is that the general algorithm can be tested without first having to implement an algorithm to extract interesting points from a laser profile (such as a corner detector).

Each pose is generated by randomly placing and rotating the object. The rotations around the X_1 and X_3 axes are calculated so that all peaks and valleys intersect the laser plane.

After a number of poses have been generated, the coordinates are mapped onto the image plane by using a fixed homography matrix that has been constructed to represent a reasonable camera setup. The main algorithm (described in section 3.2) uses the image plane coordinates and a description of the original object to estimate the homography matrix used.

To calculate how well the algorithm can estimate the homography matrix a grid of coordinates in the laser plane are mapped to the image plane by the original homography matrix and mapped back by the inverse of the estimated matrix. The result is compared to the original grid and the max and mean errors are calculated. These errors are calculated using both a single estimation without trying to compensate for rotations around the X_1 and X_3 axes and using an iterative approach that compensates for the rotations.

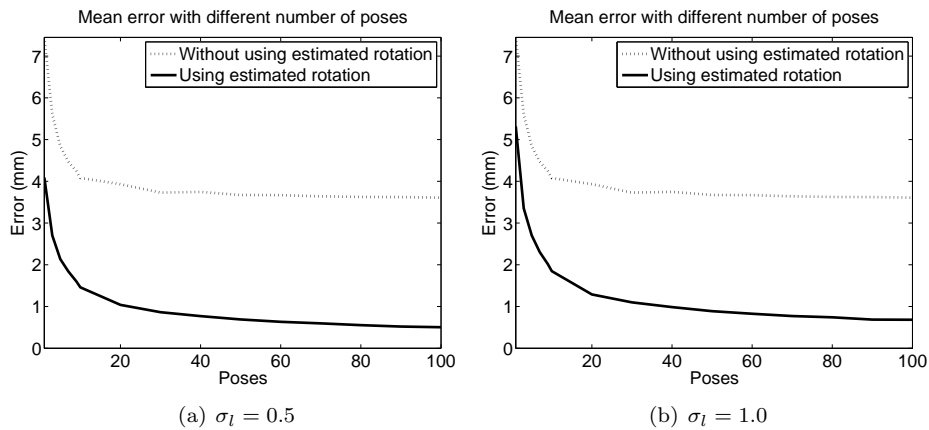


Figure 4.7. Mean error with different number of poses when the object is randomly placed in the laser plane

Figures 4.7 and 4.8 show the error when using different number of poses for the homography estimation. In figure 4.7 the object was placed with random rotations in the laser plane. In figure 4.8 the object was positioned as if it was held perfectly straight in the laser plane. It is important to note that the errors presented here do

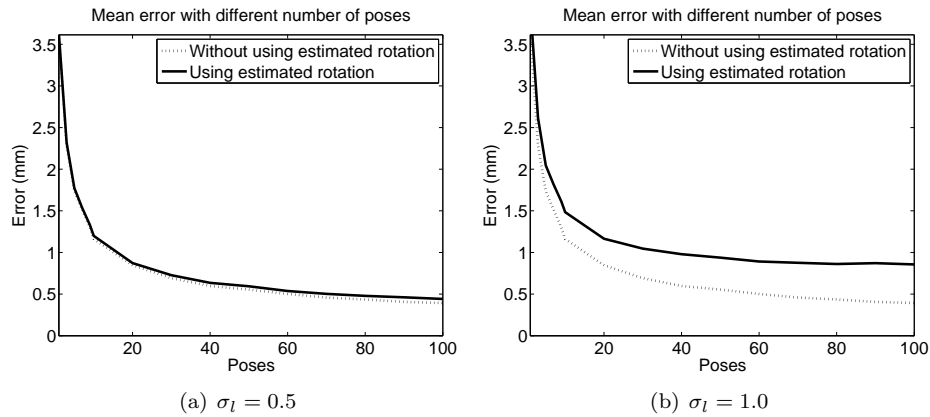


Figure 4.8. Mean error with different number of poses when the object is placed straight in the laser plane

not include the errors from the lens distortion step. The data has intentionally not been distorted using lens distortion in order to be able to evaluate the method used for calculating the homography without getting interference from other distortions. When running the simulation we used a field of view of approximately 1×1 meters. Some distortion was also added. Noise was added to the positions of the peaks and valleys with a standard deviation $\sigma_p = 0.1$ pixels. This noise level is quite low as the expected precision of the peak/valley extraction is high. Furthermore, noise is added to the position of the tilted lines. Two different noise levels ($\sigma_l = 0.5$ and $\sigma_l = 1.0$) were simulated.

4.2.2 Real data

The final experiment uses the entire method described in chapter 3 on real data. To estimate the lens distortion a large number of straight profiles (600) were used. To speed up the computation, only a fraction of the points along each profile (every 20th) were used. Two different homographies were calculated. One where the object was held as straight as possible and one where it was tilted intentionally. The homographies were estimated using 60 profiles.

To evaluate the calibration method a different set of 100 profiles of the saw-tooth calibration object were taken. Using the lens distortion parameters and the homography matrix calculated, the evaluation set was undistorted. Errors were calculated by comparing the distances between peaks and valleys in each undistorted profile to the real distances on the calibration object. The absolute errors for different distances are calculated to show how the error varies with distance. If the homography estimate is wrong, the absolute error should be larger with larger distances.

Figure 4.9 shows the errors for the homography calculated when holding the object as straight as possible. The figure shows the error both with and without

using the tilted lines to calculate the rotations when estimating the homography. The field of view is approximately 35×20 cm.

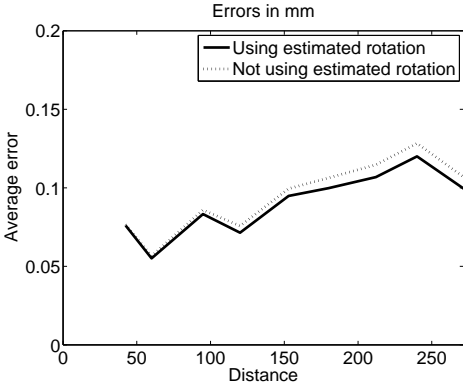


Figure 4.9. Errors when trying to hold the calibration object straight

Figure 4.10 is similar to figure 4.9 but using the homography calculated when the object was intentionally tilted.

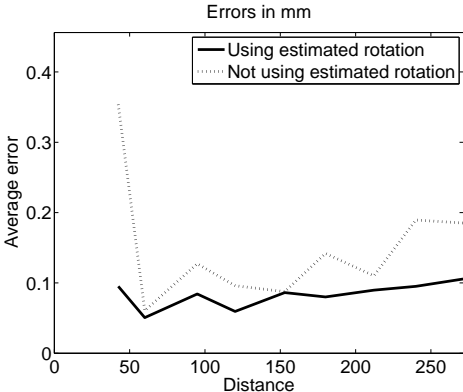


Figure 4.10. Errors when trying to tilt the calibration object intentionally

Chapter 5

Conclusions and Future Work

The purpose of this thesis was to construct a method for calibrating range cameras. A two step method has been constructed, implemented and tested. In this chapter the results are discussed and possible future improvements are suggested.

5.1 Conclusions

The estimated residual errors in section 4.2.2 show that the average residual error is fairly small. The error was roughly 0.1 mm even when measuring distances across almost the whole field of view. The longest distance measured was 270 mm. That means that the average error was as low as 0.04% of the distance or approximately 0.4 pixels with a sensor 1536 pixels wide.

One interesting property shown in figure 4.9 is that the average error remains almost constant at all measured distances. If the errors come from a badly estimated homography the errors would scale linearly with the distance. While the errors for long distances are slightly higher there is no significant difference. Similarly if the problem was that the lens distortion was badly estimated the errors should be larger near the edges. That would result in that long distances would have a significantly larger error than short since distances nearly as wide as the field of view by necessity has its endpoints near the edges.

It is possible that a substantial amount of the residual error comes from the fact that the precision of the calibration object is unknown. It is also possible that the lens introduces distortions in parts of the image that does not follow the lens distortion model. An attempt was made at plotting the errors in different areas of the sensor. However, the error was too noisy and uniformly spread across the entire sensor to provide any useful information.

Figure 4.10 shows that compensating for an arbitrarily rotated object using tilted lines can significantly improve the accuracy of the calibration if the object is not held straight. However, the method used for finding the position of the

tilted lines requires that the camera has a large aperture or long exposure time in order to work. Basically the intensity along the entire laser profile needs to be fully saturated except for the points where the tilted lines intersects the laser line. This situation can probably be improved, as discussed in the next section. Another option is omitting the tilted lines completely. Figure 4.9 shows that when the object is held reasonably straight (as straight as possible when holding it by hand), the tilted lines provide little additional accuracy. If it can be assumed that the object is held straight the calibration method can be simplified by omitting the steps that calculates the rotation of the object and compensates for it. This would probably also make the calibration object cheaper to construct.

One of the requirements was that the method should be simple to use. While there are still some parts that can be simplified, the method is still relatively simple. Compared to some other methods it does not require an expensive calibration rig that can place the calibration object at precise locations. Nor does it require that an object be placed exactly in the laser plane, which a checkerboard method such as Zhang's[12] would require.

Another requirement was that the method should have a low cost. By avoiding an expensive calibration rig, the cost has been kept down significantly. A checkerboard approach would reduce the cost even further. However, there is a trade off between accuracy and cost and, because the cost is still relatively low, we opted for higher accuracy.

5.2 Future work

While this thesis covers the basics of the proposed method, several parts are worth investigating further.

- The method is divided into two steps. The first step estimates the lens distortion and the second step estimates the homography. This distinction is necessary in the algorithm because the lens distortion needs to be known before the homography can be estimated. As presented here, the two steps require separate sets of profiles, first using a flat object and second using a sawtooth-shaped calibration object. It should be possible to use properties on the sawtooth-shaped object as data for the lens distortion estimation as well. The peaks and valleys, for instance, describe straight lines. The sides of the teeth are also straight. If using such data provides enough information to calculate the lens distortion accurately, the calibration procedure would only need one set of training profiles. That would simplify the calibration procedure for the user.
- The method used for finding the tilted lines is rather primitive. There may be two possible solutions to that problem. First it may be possible to rewrite the existing method so that it is less sensitive to the lighting conditions. Second it may be possible to find another type of object where features similar to the tilted lines can be extracted directly from the range data.

- The method for compensating for the rotation requires that the homography is reasonably correct to begin with. If the rotations can be extremely large, that assumption will not hold. Another possible method may be to use a combination of measurements that is constant under a homography. That way the rotation can be calculated before any attempt at finding the homography has been made. It would also eliminate the need for a loop that waits for \mathbf{H} to converge. One such possible combination may be cross ratio of lengths (the ratio of ratio of lengths), as described by Hartley and Zisserman[3]. Whether an explicit expression for the rotation can be found using such a measurement is not known.

Bibliography

- [1] Jules Bloomenthal and Jon Rokne. Homogeneous coordinates. *The Visual Computer*, 11(1):15–26, January 1994.
- [2] F. Devernay and O. D. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001.
- [3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2000.
- [4] Simon S. Haykin. *Neural networks: a comprehensive foundation*. Prentice-Hall, second edition, 1999.
- [5] J. Heikkilä. Geometric camera calibration using circular control points. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, October 2000.
- [6] The Mathworks Inc. Matlab function reference, 2008.
- [7] Bernd Jähne. *Digital Image Processing*. Springer, sixth edition, 2005.
- [8] Lili Ma, Yangquan Chen, and Kevin L. Moore. A family of simplified geometric distortion models for camera calibration. *CoRR*, cs.CV/0308003, 2003. informal publication.
- [9] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.
- [10] B. Prescott and G. F. McLean. Line-based correction of radial lens distortion. *Graphical Models and Image Processing*, 59(1):39–47, January 1997.
- [11] C. C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, fourth edition, 1980.
- [12] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

Appendix A

The lens distortion model's independence on scale and translation

It is not obvious that the lens distortion can be estimated without knowing the other intrinsic parameters in the camera. We need to know that the lens distortion model can model the same distortion (using different parameters) even if the coordinate system is scaled or translated. The lens distortion model was presented in section 2.2.2. It translates distorted coordinates to undistorted coordinates. In section 3.1 we choose to use two parameters to describe the radial lens distortion and two parameters to describe the decentering distortion. Furthermore, two parameters are used to define the distortion center.

$$\mathbf{x} = \tilde{\mathbf{x}} + \mathcal{F}_D(\tilde{\mathbf{x}}, \boldsymbol{\delta}) \quad (\text{A.1})$$

where

$$\mathcal{F}_D(\tilde{\mathbf{x}}, \boldsymbol{\delta}) = \begin{bmatrix} \hat{x}_1(k_1 r^2 + k_2 r^4) + 2p_1 \hat{x}_1 \hat{x}_2 + p_2(r^2 + 2\hat{x}_1^2) \\ \hat{x}_2(k_1 r^2 + k_2 r^4) + p_1(r^2 + 2\hat{x}_2^2) + 2p_2 \hat{x}_1 \hat{x}_2 \end{bmatrix}, \quad (\text{A.2})$$

$$\hat{x}_1 = \tilde{x}_1 - \tilde{x}_{1C}, \quad \hat{x}_2 = \tilde{x}_2 - \tilde{x}_{2C}, \quad r = \sqrt{\hat{x}_1^2 + \hat{x}_2^2}, \quad (\text{A.3})$$

and $\boldsymbol{\delta} = [k_1, k_2, p_1, p_2, \tilde{x}_{1C}, \tilde{x}_{2C}]^T$. \tilde{x}_{1C} and \tilde{x}_{2C} define the distortion center.

A.1 Scale

We assume a coordinate system

$$\tilde{\mathbf{x}}' = \lambda \tilde{\mathbf{x}}, \quad (\text{A.4})$$

46 The lens distortion model's independence on scale and translation

where λ is any non-zero scale factor. Using that coordinate system and introducing new parameters $\boldsymbol{\delta}' = [k'_1, k'_2, p'_1, p'_2, \tilde{x}'_{1C}, \tilde{x}'_{2C}]^T$, the lens distortion function will be

$$\mathcal{F}_D(\tilde{\boldsymbol{x}}, \boldsymbol{\delta}) = \begin{bmatrix} \hat{x}'_1(k'_1 r'^2 + k'_2 r'^4) + 2p'_1 \hat{x}'_1 \hat{x}'_2 + p'_2(r'^2 + 2\hat{x}'_1{}^2) \\ \hat{x}'_2(k'_1 r'^2 + k'_2 r'^4) + p'_1(r'^2 + 2\hat{x}'_2{}^2) + 2p'_2 \hat{x}'_1 \hat{x}'_2 \end{bmatrix}. \quad (\text{A.5})$$

Using (A.4) we can see that

$$\begin{aligned} \hat{x}'_1 &= \lambda \tilde{x}_1 - \lambda \tilde{x}_{1C} = \lambda \hat{x}_1 \\ \hat{x}'_2 &= \lambda \tilde{x}_2 - \lambda \tilde{x}_{2C} = \lambda \hat{x}_2. \end{aligned} \quad (\text{A.6})$$

We can also see that

$$r' = \sqrt{\hat{x}'_1{}^2 + \hat{x}'_2{}^2} = \sqrt{(\lambda \hat{x}_1)^2 + (\lambda \hat{x}_2)^2} = \lambda \sqrt{\hat{x}_1^2 + \hat{x}_2^2} = \lambda r. \quad (\text{A.7})$$

Using those substitutions we can rewrite (A.5) as

$$\mathcal{F}_D(\tilde{\boldsymbol{x}}, \boldsymbol{\delta}) = \begin{bmatrix} \lambda \hat{x}_1(k'_1 \lambda^2 r^2 + k'_2 \lambda^4 r^4) + 2p'_1 \lambda^2 \hat{x}_1 \hat{x}_2 + p'_2(\lambda^2 r^2 + 2\lambda^2 \hat{x}_1^2) \\ \lambda \hat{x}_2(k'_1 \lambda^2 r^2 + k'_2 \lambda^4 r^4) + p'_1(\lambda^2 r^2 + 2\lambda^2 \hat{x}_2^2) + 2p'_2 \lambda^2 \hat{x}_1 \hat{x}_2 \end{bmatrix}. \quad (\text{A.8})$$

Rearranging slightly we get

$$\mathcal{F}_D(\tilde{\boldsymbol{x}}, \boldsymbol{\delta}) = \begin{bmatrix} \hat{x}_1 \underbrace{(k'_1 \lambda^3 r^2 + k'_2 \lambda^5 r^4)}_{k_1} + 2 \underbrace{p'_1 \lambda^2}_{p_1} \hat{x}_1 \hat{x}_2 + \underbrace{p'_2 \lambda^2}_{p_2} (r^2 + 2\hat{x}_1^2) \\ \hat{x}_2 \underbrace{(k'_1 \lambda^3 r^2 + k'_2 \lambda^5 r^4)}_{k_2} + \underbrace{p'_1 \lambda^2}_{p_1} (r^2 + 2\hat{x}_2^2) + 2 \underbrace{p'_2 \lambda^2}_{p_2} \hat{x}_1 \hat{x}_2 \end{bmatrix}. \quad (\text{A.9})$$

Comparing (A.9) with (A.5), only a few things have changed as indicated by the braces in (A.9). We can now find parameters

$$\boldsymbol{\delta}' = \begin{bmatrix} k'_1 \\ k'_2 \\ p'_1 \\ p'_2 \\ \tilde{x}'_{1C} \\ \tilde{x}'_{2C} \end{bmatrix} = \begin{bmatrix} k_1/\lambda^3 \\ k_2/\lambda^5 \\ p_1/\lambda^2 \\ p_2/\lambda^2 \\ \lambda \tilde{x}_{1C} \\ \lambda \tilde{x}_{2C} \end{bmatrix} \quad (\text{A.10})$$

so that the same distortion is modeled in both coordinate systems. That means that we can scale the coordinate system with any non-zero scale factor without losing the ability to model a certain distortion.

A.2 Translation

If the coordinate system is translated as

$$\tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{x}} + \boldsymbol{t}, \quad (\text{A.11})$$

where

$$\boldsymbol{t} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}, \quad (\text{A.12})$$

we can see that choosing $\tilde{x}'_{1C} = \tilde{x}_{1C} + t_1$ and $\tilde{x}'_{2C} = \tilde{x}_{2C} + t_2$ will result in:

$$\hat{x}'_1 = \tilde{x}'_1 - \tilde{x}'_{1C} = (\tilde{x}_1 + t_1) - (\tilde{x}_{1C} + t_1) = \tilde{x}_1 - \tilde{x}_{C1} = \hat{x}_1 \quad (\text{A.13})$$

and

$$\hat{x}'_2 = \tilde{x}'_2 - \tilde{x}'_{2C} = (\tilde{x}_2 + t_2) - (\tilde{x}_{2C} + t_2) = \tilde{x}_2 - \tilde{x}_{C2} = \hat{x}_2. \quad (\text{A.14})$$

Thus, choosing the distortion center in this way, effectively cancels the translation out. That is no surprise, since using the same translation for the distortion center translates it along with the rest of the coordinate system. The same lens distortion can now be modeled before and after the translation.

Appendix B

Calculating angles from line distances

In order to find how θ and y'_1 can be determined from d_1 and d_2 we first do the opposite and find how d_1 and d_2 depends on θ and y'_1 . Using figure B.1 we can determine how d_1 can be described in terms of θ , y'_1 and parameters that we already know. Note that the angle α_1 has a negative value in this figure. From

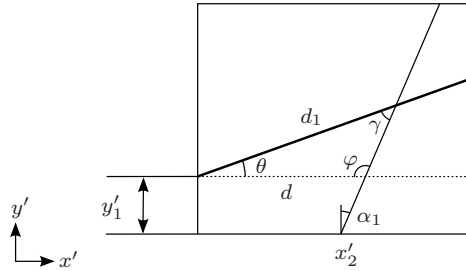


Figure B.1. Left part of figure 3.20

the figure we can directly see that:

$$\varphi = \frac{\pi}{2} - \alpha_1 \quad (\text{B.1})$$

$$\gamma = \frac{\pi}{2} + \alpha_1 - \theta \quad (\text{B.2})$$

The distance d depends on y'_1 :

$$d = x'_2 - y'_1 \tan(\alpha_1) \quad (\text{B.3})$$

The law of sines gives the following equation:

$$\frac{d_1}{\sin(\varphi)} = \frac{d}{\sin(\gamma)}. \quad (\text{B.4})$$

Substituting φ , γ and d for their values in (B.1)-(B.3) gives us

$$\frac{d_1}{\sin\left(\frac{\pi}{2} - \alpha_1\right)} = \frac{x'_2 - y'_1 \tan(\alpha_1)}{\sin\left(\frac{\pi}{2} + \alpha_1 - \theta\right)}. \quad (\text{B.5})$$

The equation can be simplified further by rewriting the sine functions as cosine:

$$\frac{d_1}{\cos(\alpha_1)} = \frac{x'_2 - y'_1 \tan(\alpha_1)}{\cos(\theta - \alpha_1)} \quad (\text{B.6})$$

Finally we multiply both sides by $\cos(\alpha_1)$:

$$d_1 = \frac{(x'_2 - y'_1 \tan(\alpha_1)) \cos(\alpha_1)}{\cos(\theta - \alpha_1)} = \frac{x'_2 \cos(\alpha_1) - y'_1 \sin(\alpha_1)}{\cos(\theta - \alpha_1)} \quad (\text{B.7})$$

Using a very similar procedure we can also see that:

$$d_2 = \frac{x'_4 \cos(\alpha_2) - y'_1 \sin(\alpha_2)}{\cos(\theta - \alpha_2)} \quad (\text{B.8})$$

From now on we will assume that the angles α_1 and α_2 always have the same magnitude but opposite signs. We introduce a new angle $\alpha = \alpha_1 = -\alpha_2$. The equations (B.7) and (B.8) can now be rewritten:

$$d_1 = \frac{x'_2 \cos(\alpha) - y'_1 \sin(\alpha)}{\cos(\theta - \alpha)} \quad (\text{B.9})$$

$$d_2 = \frac{x'_4 \cos(\alpha) + y'_1 \sin(\alpha)}{\cos(\theta + \alpha)} \quad (\text{B.10})$$

From (B.9) we find an expression for y'_1 :

$$y'_1 = \frac{x'_2 \cos(\alpha) - d_1 \cos(\theta - \alpha)}{\sin(\alpha)} \quad (\text{B.11})$$

Equation (B.10) with y'_1 substituted gives us:

$$d_2 = \frac{x'_4 \cos(\alpha) + x'_2 \cos(\alpha) - d_1 \cos(\theta - \alpha)}{\cos(\theta + \alpha)} \quad (\text{B.12})$$

We rearrange the equation slightly to have all θ on the left hand side:

$$d_1 \cos(\theta - \alpha) + d_2 \cos(\theta + \alpha) = (x'_2 + x'_4) \cos(\alpha) \quad (\text{B.13})$$

The cosine addition formula states that:

$$\cos(\theta - \alpha) = \cos(\theta) \cos(\alpha) + \sin(\theta) \sin(\alpha) \quad (\text{B.14})$$

$$\cos(\theta + \alpha) = \cos(\theta) \cos(\alpha) - \sin(\theta) \sin(\alpha) \quad (\text{B.15})$$

Using those substitutions in (B.13) yields:

$$(d_1 + d_2) \cos(\theta) \cos(\alpha) + (d_1 - d_2) \sin(\theta) \sin(\alpha) = (x'_2 + x'_4) \cos(\alpha) \quad (\text{B.16})$$

Dividing both sides by $\cos(\alpha)$ gives:

$$\underbrace{(d_1 + d_2)}_a \cos(\theta) + \underbrace{(d_1 - d_2) \tan(\alpha)}_b \sin(\theta) = x'_2 + x'_4 \quad (\text{B.17})$$

The formula $a \cos(\theta) + b \sin(\theta)$ can be rewritten in harmonic form as:

$$a \cos(\theta) + b \sin(\theta) = R \sin(\theta + \rho), \quad (\text{B.18})$$

where $R = \sqrt{a^2 + b^2}$,

$$\rho = \arctan\left(\frac{a}{b}\right) + \psi \quad (\text{B.19})$$

and

$$\psi = \begin{cases} 0 & \text{if } b \geq 0 \\ \pi & \text{if } b < 0 \end{cases} \quad (\text{B.20})$$

Because $b = (d_1 - d_2) \tan(\alpha)$ and $d_1 - d_2 < 0$, ψ can be rewritten as

$$\psi = \begin{cases} 0 & \text{if } \alpha \leq 0 \\ \pi & \text{if } \alpha > 0 \end{cases} \quad (\text{B.21})$$

Using this method on (B.17) gives us

$$R \sin\left(\theta + \arctan\left(\frac{a}{b}\right) + \psi\right) = x'_2 + x'_4, \quad (\text{B.22})$$

where $R = \sqrt{(d_1 + d_2)^2 + (d_1 - d_2)^2 \tan^2(\alpha)}$. Realizing that $\sin(x + \pi) = -\sin(x)$ we can rewrite this as

$$-\text{sgn}(\alpha) R \sin\left(\theta + \arctan\left(\frac{a}{b}\right)\right) = x'_2 + x'_4. \quad (\text{B.23})$$

Further rearrangements results in:

$$\sin\left(\theta + \arctan\left(\frac{a}{b}\right)\right) = -\text{sgn}(\alpha) \frac{x'_2 + x'_4}{R}. \quad (\text{B.24})$$

To find a formula for θ we use arcsin on both sides:

$$\theta + \arctan\left(\frac{a}{b}\right) = \arcsin\left(-\text{sgn}(\alpha) \frac{x'_2 + x'_4}{R}\right) \quad (\text{B.25})$$

Which finally gives us

$$\theta = -\text{sgn}(\alpha) \arcsin\left(\frac{x'_2 + x'_4}{R}\right) - \arctan\left(\frac{a}{b}\right). \quad (\text{B.26})$$

Appendix C

Least squares estimate of two parallel lines

The least squares estimate is a method for finding parameters that model something optimally in terms of minimizing the sum of squared errors. Normal linear least squares can be used to fit a line to a set of points.

If each point $\mathbf{x}_i = (x_i, y_i)$, and the fitted line has the form $y = kx + m$, where k and m define the line. The error for each point (the vertical distance between a point and the line) can be written as:

$$r_i = y_i - (kx_i + m) \quad (\text{C.1})$$

If we have n points, the sum of squared errors can be written as:

$$S = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n (y_i - (kx_i + m))^2 \quad (\text{C.2})$$

To find the optimal parameters we set the derivatives of S with respect to the two parameters to zero:

$$\frac{\partial S}{\partial k} = -2 \sum_{i=1}^n (y_i x_i - kx_i^2 - mx_i) = 0 \quad (\text{C.3})$$

$$\frac{\partial S}{\partial m} = -2 \sum_{i=1}^n (y_i - kx_i - m) = 0 \quad (\text{C.4})$$

Rearranging those equations gives us:

$$\sum_{i=1}^n kx_i^2 + \sum_{i=1}^n mx_i = \sum_{i=1}^n y_i x_i \quad (\text{C.5})$$

$$\sum_{i=1}^n kx_i + \sum_{i=1}^n m = \sum_{i=1}^n y_i \quad (\text{C.6})$$

This system of equations can be written in matrix form as:

$$\mathbf{X}^T \mathbf{X} \boldsymbol{\beta} = \mathbf{X}^T \mathbf{y} \quad (\text{C.7})$$

where

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} m \\ k \end{pmatrix} \text{ and } \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

The equation can be solved as

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (\text{C.8})$$

When we have two sets of points A and B and we want to find one line for each set we could use the above method one time for each set of points. However, since we want the two lines to be parallel we need to find both lines at the same time. The two lines can be defined by three parameters k, m_1 and m_2 such that $y = kx + m_1$ for the first line and $y = kx + m_2$ for the second. We now have the error:

$$r_i = \begin{cases} y_i - (kx_i + m_1) & \text{if } i \in A \\ y_i - (kx_i + m_2) & \text{if } i \in B \end{cases} \quad (\text{C.9})$$

The sum of squared errors S can now be written as:

$$S = \sum_{i \in A} r_i^2 + \sum_{i \in B} r_i^2 = \sum_{i \in A} (y_i - (kx_i + m_1))^2 + \sum_{i \in B} (y_i - (kx_i + m_2))^2 \quad (\text{C.10})$$

The derivatives with respect to the three parameters can now be written as:

$$\frac{\partial S}{\partial k} = -2 \sum_{i \in A} (y_i x_i - kx_i^2 - m_1 x_i) - 2 \sum_{i \in B} (y_i x_i - kx_i^2 - m_2 x_i) \quad (\text{C.11})$$

$$\frac{\partial S}{\partial m_1} = -2 \sum_{i \in A} (y_i - kx_i - m_1) \quad (\text{C.12})$$

$$\frac{\partial S}{\partial m_2} = -2 \sum_{i \in B} (y_i - kx_i - m_2) \quad (\text{C.13})$$

Setting them equal to zero and rearranging them gives us the following equations:

$$\sum_{\forall i} kx_i^2 + \sum_{i \in A} m_1 x_i + \sum_{i \in B} m_2 x_i = \sum_{\forall i} y_i x_i \quad (\text{C.14})$$

$$\sum_{i \in A} kx_i + \sum_{i \in A} m_1 = \sum_{i \in A} y_i \quad (\text{C.15})$$

$$\sum_{i \in B} kx_i + \sum_{i \in B} m_2 = \sum_{i \in B} y_i \quad (\text{C.16})$$

The system of equations can be written in matrix form as:

$$\mathbf{X}^T \mathbf{X} \boldsymbol{\beta} = \mathbf{X}^T \mathbf{y} \quad (\text{C.17})$$

where

$$\mathbf{X} = \begin{pmatrix} a_1 & b_1 & x_1 \\ a_2 & b_2 & x_2 \\ \vdots & \vdots & \\ a_n & b_n & x_n \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} m_1 \\ m_2 \\ k \end{pmatrix} \text{ and } \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

The parameters a_i and b_i determine whether the point is in set A or B :

$$a_i = \begin{cases} 1 & \text{if } i \in A \\ 0 & \text{otherwise} \end{cases}, \quad b_i = \begin{cases} 1 & \text{if } i \in B \\ 0 & \text{otherwise} \end{cases}$$

Like before the parameters can be found by:

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \tag{C.18}$$

Appendix D

Error map generation

A problem when using straight lines to estimate the lens distortion is that it is hard to estimate the residual errors that the distortion model does not handle. Ideally it would be nice to know the exact error vector for each part of the sensor plane. While that can not be achieved an attempt is presented here at estimating something similar.

As input data we have a set of lines that have been undistorted using a lens distortion model. We make the assumption that the general position of the line is correct and that any remaining distortions appear only on parts of the line. That means that we can find an error for each part of an input line by comparing it to a straight line fitted by a least squares estimate.

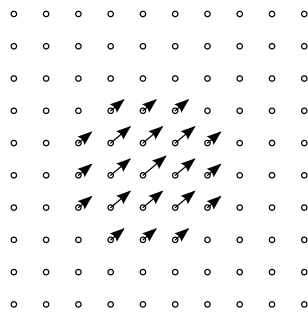


Figure D.1. A field of vectors describing residual distortion

For each fitted line we can find error vectors on the line that describe the error orthogonal to the line. We do not know if there is an error in the direction along the line. Using several lines with different rotations it should be possible to estimate the error vector for a certain point. See figures D.2 and D.3 for an illustration of the problem.

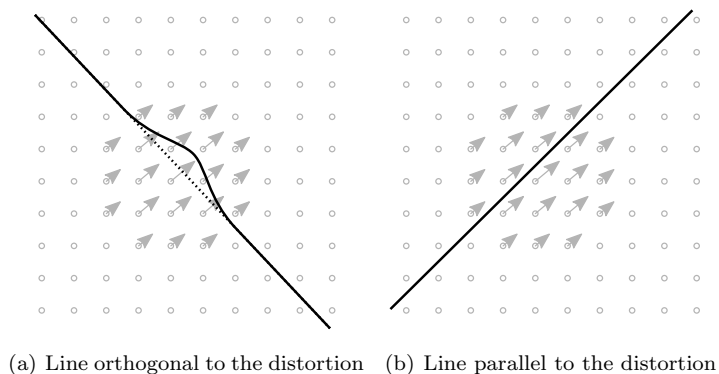


Figure D.2. Lines in different directions have different measurable errors

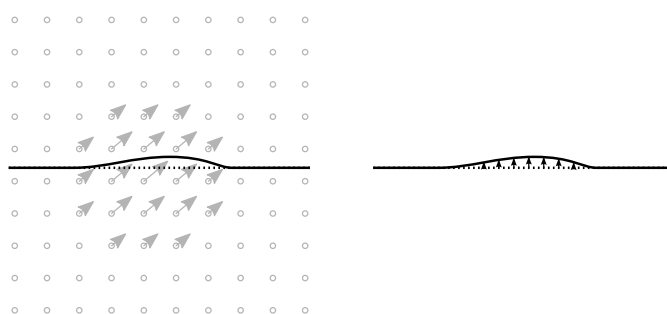


Figure D.3. The error can only be measured in the direction orthogonal to the fitted line.

D.1 Finding a vector from a set of projections

If we have distortion information in several directions for a point (two or more lines intersect at that point) we can calculate the orientation and magnitude of the actual error. The measured error in each direction is represented by a vector (called \mathbf{p}_i in the following section). In the trivial case we have two input vectors that are orthogonal to each other. In that case the actual error (represented by a vector called \mathbf{v}) is the sum of the input vectors. Figure D.4 illustrates that case.

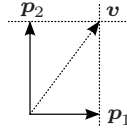


Figure D.4. The original vector when the two known directions are orthogonal

In general, however, the input vectors are not orthogonal and a different method is needed. Rather than simply adding the vectors we need to find the vector \mathbf{v} that, when projected onto the normalized vectors $\hat{\mathbf{p}}_i$, has the same length as the corresponding vectors \mathbf{p}_i . A simple case is when we have just two input vectors, as shown in figure D.5.

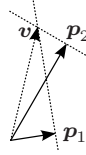


Figure D.5. The original vector with two known directions

The constraints on \mathbf{v} can be written as

$$\mathbf{v} \cdot \hat{\mathbf{p}}_i = |\mathbf{p}_i|. \quad (\text{D.1})$$

Since $\hat{\mathbf{p}}_i = \frac{\mathbf{p}_i}{|\mathbf{p}_i|}$ we can rewrite that as

$$\mathbf{v} \cdot \frac{\mathbf{p}_i}{|\mathbf{p}_i|} = |\mathbf{p}_i| \Leftrightarrow \mathbf{v} \cdot \mathbf{p}_i = |\mathbf{p}_i|^2 = \mathbf{p}_i \cdot \mathbf{p}_i. \quad (\text{D.2})$$

Rewriting the dot products as vector multiplications we get

$$\mathbf{p}_i^T \mathbf{v} = \mathbf{p}_i^T \mathbf{p}_i. \quad (\text{D.3})$$

The equations can be written in matrix form as

$$\mathbf{P}^T \mathbf{v} = \mathbf{b}, \quad (\text{D.4})$$

where

$$\mathbf{P} = \begin{pmatrix} | & | & \cdots & | \\ \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \\ | & | & \cdots & | \end{pmatrix} \quad (\text{D.5})$$

and

$$\mathbf{b} = \begin{pmatrix} \mathbf{p}_1^T \mathbf{p}_1 \\ \mathbf{p}_2^T \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_n^T \mathbf{p}_n \end{pmatrix}. \quad (\text{D.6})$$

If we only have two input vectors the vector \mathbf{v} can be found by

$$\mathbf{v} = \mathbf{P}^{-T} \mathbf{b}. \quad (\text{D.7})$$

In general we will have more than two input vectors and, because of measurement noise, it will not be possible to find a \mathbf{v} that satisfies all constraints. Instead \mathbf{v} will be calculated using a least squares estimate.

The length of the residual error vector for each input vector can be written as

$$\mathbf{r}_i = \mathbf{v} \cdot \hat{\mathbf{p}}_i - \mathbf{p}_i \cdot \hat{\mathbf{p}}_i. \quad (\text{D.8})$$

The sum of squared residuals is

$$\mathcal{S} = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n (\mathbf{v} \cdot \hat{\mathbf{p}}_i - \mathbf{p}_i \cdot \hat{\mathbf{p}}_i)^2. \quad (\text{D.9})$$

To find the optimal \mathbf{v} we set the derivative of \mathcal{S} to zero:

$$\frac{\partial \mathcal{S}}{\partial \mathbf{v}} = -2 \sum_{i=1}^n (\mathbf{v} \cdot \hat{\mathbf{p}}_i - \mathbf{p}_i \cdot \hat{\mathbf{p}}_i) \hat{\mathbf{p}}_i = 0. \quad (\text{D.10})$$

We can rewrite that as

$$\sum_{i=1}^n (\mathbf{v} \cdot \hat{\mathbf{p}}_i) \hat{\mathbf{p}}_i - \mathbf{p}_i = 0. \quad (\text{D.11})$$

And finally

$$\sum_{i=1}^n (\mathbf{v} \cdot \hat{\mathbf{p}}_i) \hat{\mathbf{p}}_i = \sum_{i=1}^n \mathbf{p}_i. \quad (\text{D.12})$$

This system of equations can be written in matrix form as

$$\hat{\mathbf{P}} \hat{\mathbf{P}}^T \mathbf{v} = \mathbf{P} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \quad (\text{D.13})$$

where

$$\hat{\mathbf{P}} = \begin{pmatrix} | & | & \cdots & | \\ \hat{\mathbf{p}}_1 & \hat{\mathbf{p}}_2 & \cdots & \hat{\mathbf{p}}_n \\ | & | & \cdots & | \end{pmatrix} \quad (\text{D.14})$$

and

$$\mathbf{P} = \begin{pmatrix} | & | & \dots & | \\ \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_n \\ | & | & \dots & | \end{pmatrix}. \quad (\text{D.15})$$

We can now find \mathbf{v} by

$$\mathbf{v} = (\hat{\mathbf{P}}\hat{\mathbf{P}}^T)^{-1}\mathbf{P} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \quad (\text{D.16})$$

D.2 Using weighted projections

The previous section discussed how known errors in different directions can be used to find the overall error. In order to use that method the error must be known in several directions for each point in the sensor plane. In practice that means that at least two lines must intersect in each point. That is not a reasonable requirement. Instead, when calculating the error vector known vectors that are close to the point are used. Depending on how much data has been collected different area sizes can be used. However, it is clear that known errors close to the point should have a bigger influence than ones far away. To achieve that we need to add weights to the least squares estimate. That can be done by introducing a new vector \mathbf{w} that contains the weights of all input vectors:

$$\mathbf{w} = (w_1 \quad w_2 \quad \dots \quad w_n), \quad (\text{D.17})$$

where w_i is the weight of \mathbf{p}_i . The formula for \mathbf{v} with support for weights becomes:

$$\mathbf{v} = (\hat{\mathbf{P}} \text{diag}(\mathbf{w}) \hat{\mathbf{P}}^T)^{-1} \mathbf{P} \mathbf{w}^T, \quad (\text{D.18})$$

where $\text{diag}(\mathbf{w})$ generates a matrix with the elements of \mathbf{w} in its diagonal and all other elements are zero.

Using this method, an error vector can be calculated for each point in the sensor plane. To visualize the errors, only the length of the calculated vectors are shown in the generated error maps.