

EINDHOVEN UNIVERSITY OF TECHNOLOGY

# Calibration of laser-triangulation based vision systems under motion

An automated method using sloped rigid artifacts fabricated through additive manufacturing

*A report submitted in fulfilment of the requirements for the completion of the Systems and Control Master (45 ECTS) Thesis Project . The project was performed at the University of Waterloo within the Multi-Scale Additive Manufacturing Lab between 17/09/2018 and 30/04/2019 .*

**Author:**

G.J.J. van Houtum

IDnr: 0872274

**University of Waterloo**

Department of Mechanical and Mechatronics Engineering

Multi-Scale Additive Manufacturing Lab

Supervisor: M.L. Vlasca, PhD

**Eindhoven University of Technology**

Department of Mechanical Engineering

Control Systems Technology

Supervisor: Dr.ir. L.F.P. Etman

Report number: CST 2019.059

Master program: Systems and Control

Tilburg, Netherlands, June 2019



## Abstract

Process control of deposition geometry is difficult to achieve in additive manufacturing processes. A laser-triangulation based vision system could provide a low-cost solution for feedback in closed loop control or serve as a monitoring tool for identification of defects during the build process of parts.

Calibration of the vision system is needed since coordinates in images from the camera do not relate to real world coordinates. Accurate and expensive artifacts with known dimensions are often used for calibration. The material and reflective properties of these artifacts are however not representative of the printed material. The field-of-view of the camera can be adjusted to scan larger parts. Calibration with a single artifact for different zoom levels of the camera possibly leads to less accurate calibration since the artifact could only cover a small part of the field-of-view.

A single sloped artifact positioned at different angular rotations relative to the scanning direction provides a solution for different zoom levels of the camera. If images are acquired at several artifact cross sections, the pattern of the artifact will move vertically and horizontally in the field-of-view of the camera. This spread allows for calibration with higher accuracy. The artifact is printed by the additive manufacturing process itself. Calibration is therefore performed on an artifact with the same material as the printed part such that there is no difference in reflectiveness. Furthermore, it provides a low-cost and flexible solution for any additive manufacturing process.

This research provides a calibration method for rigid sloped artifacts fabricated through additive manufacturing. Knowledge of the inclination and angular orientation of the artifact is not necessary for calibration. Only the relative coordinates of the cross section pattern in the artifact are needed. An adaptive laser line centerline and feature point extraction algorithm is developed for calibration point extraction which does not rely on any fixed parameter. Therefore the algorithm is adaptive to changes in material reflectiveness. Simulation shows that a sloped artifact leads to more accurate calibration than an artifact without a slope. An experiment is conducted where a sloped artifact is printed using a Stratasys J750. The calibration result shows that with a 95% confidence the real world measurement errors fall between  $0.0002 \pm 0.008$  and  $0.0001 \pm 0.009$  millimeters in the  $x$  and  $y$  direction respectively for an approximate field-of-view of 7 millimeters.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Process control in additive manufacturing . . . . .	1
1.2	Vision based deposition measurement . . . . .	4
1.3	Problem statement . . . . .	6
1.4	Research objective . . . . .	8
1.5	Report outline . . . . .	8
<b>2</b>	<b>Laser plane projection model</b>	<b>9</b>
2.1	The pinhole camera model . . . . .	9
2.2	Extrinsic view . . . . .	10
2.3	Sensor intrinsics . . . . .	12
2.4	Lens distortion . . . . .	12
2.5	Projection chain . . . . .	14
<b>3</b>	<b>Calibration artifact</b>	<b>15</b>
3.1	Artifact projection . . . . .	15
3.2	Line laser centerline extraction . . . . .	17
3.3	Calibration point extraction . . . . .	20
<b>4</b>	<b>Camera calibration</b>	<b>23</b>
4.1	Multi-image direct linear transform . . . . .	23
4.2	Artifact slope and angle estimation . . . . .	25
4.3	Iterative linear parameter estimation . . . . .	26
4.4	Lens distortion estimation . . . . .	26
4.5	Nonlinear refinement . . . . .	27
<b>5</b>	<b>Results and Discussion</b>	<b>29</b>
5.1	Accuracy and precision evaluation . . . . .	29
5.2	Simulation . . . . .	32
5.3	Experiment . . . . .	36
<b>6</b>	<b>Conclusions and future work</b>	<b>39</b>
	<b>Bibliography</b>	<b>42</b>



# 1 Introduction

Additive Manufacturing (AM) has the ability to produce more complex products than conventional manufacturing processes [1]. Soft material-based AM processes are typically divided into material jetting and material extrusion. The former uses drop-on-demand technology where the nozzle dispenses droplets of material, layer by layer. The latter is referring to the process where the material is extruded through a nozzle on a build platform. Dimensional accuracy, repeatability and short production time are however challenging to achieve within AM processes, in particular extrusion based processes. If the aforementioned problems could be overcome, AM methods could have a better adoption and acceptance in the manufacturing industry [2].

## 1.1 Process control in additive manufacturing

Figure 1.1 shows a typical part production chain for extrusion based processes [3]. First the part is designed using CAD software generating and coded into a so called STL file. This file is read by an application which slices the 3D model into layers resulting in a G-code script. The G-code contains instructions that are provided to the machine controller of the AM machine describing where to move, how fast to move, and what path to follow. This script is uploaded to, and executed by, the AM system.

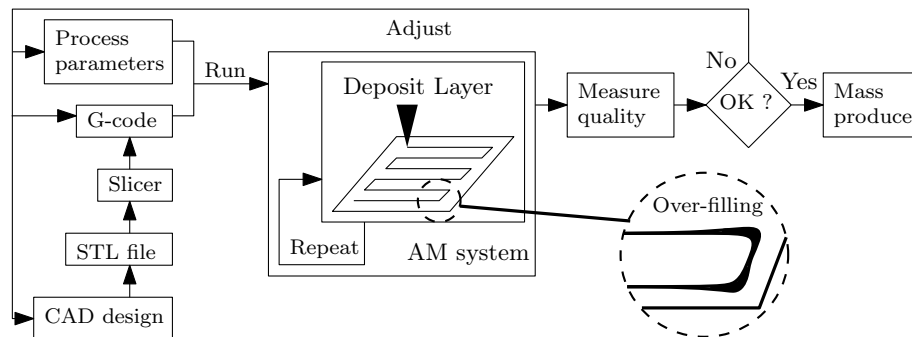


Figure 1.1: Inter-part feedback control: Build quality optimization by producing parts, measuring quality and adjusting process parameters in an iterative way.

### 1.1.1 Inter-part feedback control

Process parameters influence the operation of the deposition in the AM system. For extrusion based processes the temperature of the extruded material and extrusion rate are common parameters. Typically, no in-process geometric measurement of the deposition is used as feedback to optimize process parameters during the build. Instead, constant values for the parameter settings are used throughout the build while desirable part dimensions and properties are found by measuring part quality characteristics such as surface quality and mechanical strength [4]. As shown in Figure 1.1, the process settings and the G-code are provided to the AM system to produce a part. Then, based upon measurements of the printed part, a decision is made to either re-iterate and adapt (process parameters, CAD design and G-code)

or accept the quality and deploy for mass-production. This process, in this report referred to as inter-part feedback control, uses fixed process settings throughout the build. Problems such as over- and under-filling at off-nominal system states compromise deposition dimensional accuracy. For extrusion based processes, gantry velocity is typically reduced when approaching a corner and increased when leaving. A fixed extrusion rate will as a consequence result in overfilling (see Figure 1.1 ) in the corners as shown in [5].

### 1.1.2 Intra-layer feedback control

The implementation of a real-time process parameter feedback controller could improve dimensional accuracy and result in higher 3D deposition quality and geometric fidelity as described in [6]. This method, in this report referred to as intra-layer feedback control, is shown in Figure 1.2. Intra-layer feedback control differs from inter-part feedback control in the sense that process parameters are measured and adjusted in-process. For extrusion based processes extrusion rate and material temperature are often controlled. The volumetric nozzle flow is adjusted dependent on gantry velocity to compensate for over-filling in off-nominal states, such as corners, while maintaining a constant material temperature.

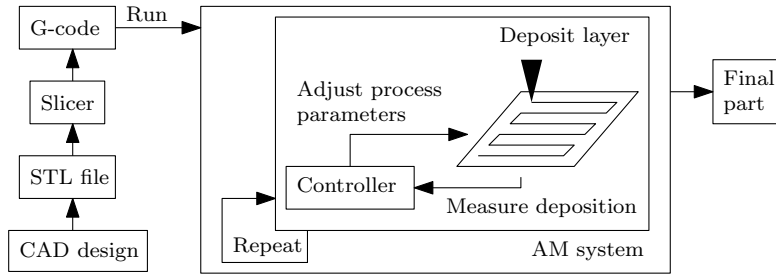


Figure 1.2: Intra-layer feedback control: Build quality optimization through measurement of deposition size and real-time adjustment of process parameters.

Direct control of deposition geometry based upon measurements of deposition size is difficult to achieve for intra-layer based control. Throughout the build process deposition direction is changed and a sensor measuring deposition size would require omni-directional measurement of the deposited material as shown in Figure 1.3. Furthermore, overlap of depositions could lead to difficulties in measurement. Although additional motion control systems could be implemented to achieve tracking it would lead to added complexity and cost.

### 1.1.3 Inter-layer feedback control

Extrusion based AM systems are relatively slow processes in terms of build time. The transient dynamics of the systems which influence process parameters are considered to be much faster than gantry movement. Assuming that these systems operate in steady-state is therefore considered a reasonable approximation. This assumption allows for the development of a steady-state response model describing the link between process parameter value and their influence on the geometry of the deposited material.

In [7] a deposition geometry control method is described on an inter-layer bases using a steady-state model. Inter-layer feedback control is in this report referred to as the method where process settings are adapted for the upcoming layer based upon measurements of the last layer geometry. This way, inaccuracies in the last



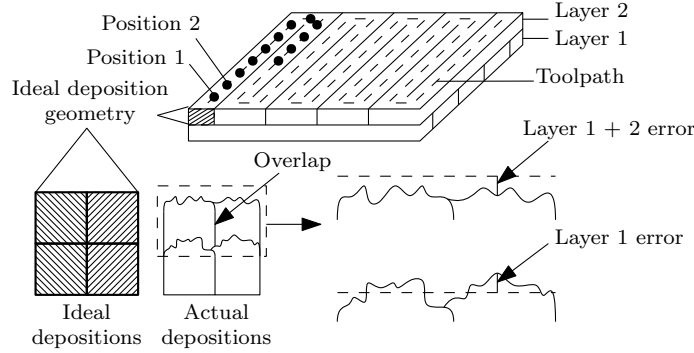


Figure 1.3: Deposition errors: Actual depositions deviate from ideal deposition due to overlap and noise. Errors in previous layers influence future layers.

layer can be compensated for in the next layer. A build process using inter-layer control is shown in Figure 1.4. G-code from a CAD design is fed into a controller. This controller then provides the G-code and process parameters for the upcoming layer to the system. A layer is then deposited and measured such that the controller can compensate for errors in the upcoming layer with new process parameters. This process is repeated until the part is finished.

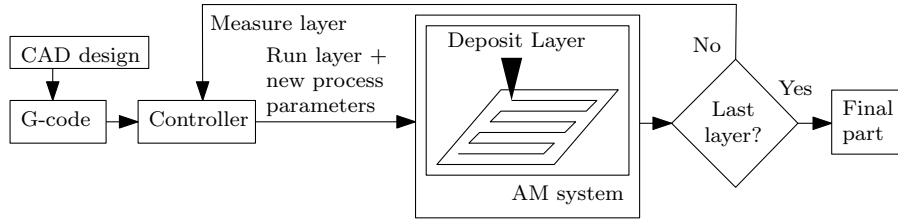


Figure 1.4: Inter-layer feedback control: Build quality optimization through adjustment of process parameters for each layer based upon measurements of the last layer.

A deposition geometry feedback sensor for inter-layer control would not require additional motion control systems for deposition tracking. Instead the sensor could be mounted in a fixed way on the AM system to scan the layer surface. Therefore, less complicated systems would be required for operation compared to intra-layer control. Inter-layer control still requires the AM system to stop between layers and run the toolpath again for measurement. Furthermore, every layer new process parameters need to be provided to the system. Build time is an important factor in AM and inter-layer control would lead to additional time due to the measurement and adaption stages between each layer.

#### 1.1.4 Open-loop control

Production of parts without any form of feedback on what is actually deposited is in this report referred to as open-loop control. Open-loop control is expected to underperform compared to intra- or inter-layer feedback control due to the lack of feedback. Compared to inter-part control it could however still improve dimensional accuracy. A system response model describing the relation between process parameter value and deposition geometry could be used to predict process parameter values for each position in the toolpath of the build and compensate for off-nominal system states such as corners as depicted in Figure 1.5.

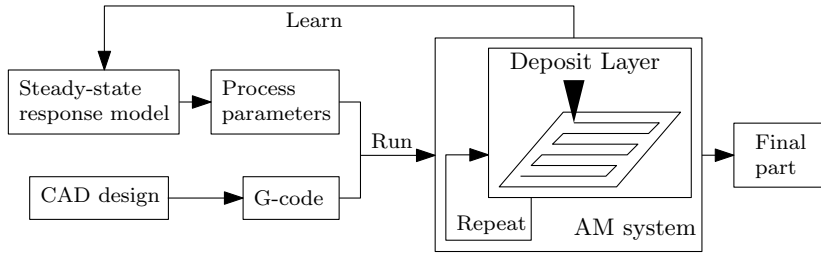


Figure 1.5: Open-loop control: A steady-state process model is used to compensate for over-and under-filling at off-nominal system states.

In [8] an analytical model of an extrusion process is developed and compared to experimental data. The discrepancy between the numerical and experimental results exemplifies the difficulty in modelling the AM extrusion process based upon first principles. A black-box empirical response model based upon experimental data could lead to more accurate results. Operational system intrinsic settings might affect the response of the system. For extrusion based processes a slight variation in viscosity or nozzle geometry may lead to significant change in the deposition geometry while using the same process parameters. Furthermore, external influences such as temperature and vibration might also affect the performance of the AM system. Ideally an empirical model of the system response is determined using recently acquired experimental data to overcome the aforementioned problems.

## 1.2 Vision based deposition measurement

Vision based measurement for process control in AM could provide a low-cost and simple solution for feedback. Implementation for intra-layer control would be difficult for tracking reasons. However, vision based measurement in inter-layer control is feasible by mounting the vision system in a fixed way to the existing AM system. Besides feedback, such a vision system could also be used for process monitoring to track deposition quality and identify defects. Furthermore, in open-loop control it could be used for empirical system response modelling before every build by scanning deposition size for a multitude of process parameters to develop a steady-state system model.

Measurement of deposition dimensions is possible through the implementation of a laser-triangulation based vision system as proposed in [9]. Figure 1.6 shows such a vision system which includes a line laser module and a camera positioned in a triangular setup. The module projects a high intensity line shaped laser on the surface of the object to be measured. The geometry of the object in the laser plane is exposed to the camera, due to the angle between the laser plane and the camera, by the high intensity profile of the laser line.

Coordinates in the laser plane will however appear in perspective on the images created by the camera. The line laser on top of the object, see Figure 1.6, is closer to the camera than the laser line on the print bed. While the object is square in reality, it will appear as a trapezium in the images. Furthermore, lens optics will distort the coordinates. Therefore, image coordinates are not representative of real-world coordinates in the laser plane.

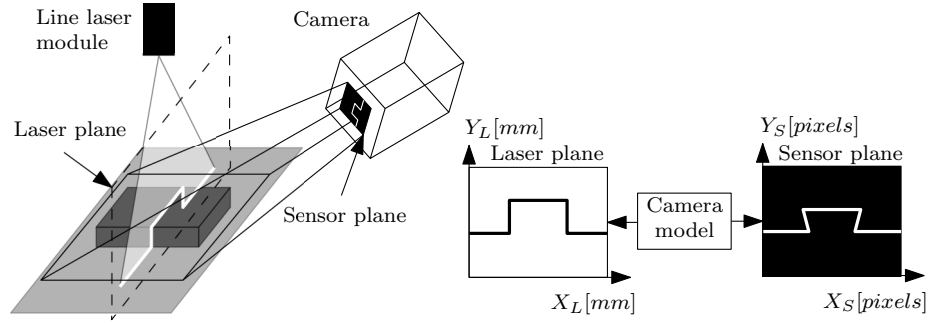


Figure 1.6: Laser-triangulation based vision: Measurement of objects through images taken by a camera positioned angular to the laser plane such that object geometry can be detected.

### 1.2.1 Camera calibration

Calibration of the vision system is needed. A camera model needs to be development such that coordinates in the image plane can be transformed to coordinates in the laser plane as shown in Figure 1.6. Camera calibration methods can be divided in artifact based calibration and auto-calibration as decribed in [10]. The former, uses artifacts with known dimensions. The latter calibrates by extracting features from an unknown scene.

The available methods for laser line based camera calibration using known artifacts have recently been reviewed in [11]. The artifacts used for calibration are either 2D or 3D objects which are manufactured to a high precision. The manufacturing methods for the artifacts are typically CNC machining, grinding and photo-etching/printing. Calibration accuracy is affected by the accuracy of the calibration artifact and its reflective properties. A artifact that absorbes most of the laser line will be hard to observe by the camera. A highly reflective artifact might lead to saturation of the pixels in the camera.

For laser-traingulation based vision, accurate measurement of objects in the laser-plane is essential. The laser plane is fixed relative to the camera which means that a calibration method is needed that can observe the artifact from the laser-plane. 3D Artifacts are particularly of interest because 2D cross sections of the artifact are observed by the laser traingulation based vision system. The pattern of the artifact is observed by the camera and a distorted version of the pattern appears in the images due to the perspective view and lens distortion as mentioned in Section 1.2.

Known coordinates of points in the laser plane and their corresponding coordinates in the images from the camera can be used to calibrate the camera. Finding the corresponding coordinates of points on the pattern in the images is therefore essential. The pattern of the 2D cross sections in the artifact can be designed to include so called features. These features are not affected by the perspective view and lens distortion. A corner in the artifact pattern will for example appear as a corner in the camera image.

Extracting the coordinates of the corners of the pattern in the images and combining them with their corresponding known coordinates in the laser plane pattern leads to a set of point correspondences which can be used to calibrate the vision system. Interesting 3D artifact based calibration methods are described in [12], [13] and [14]. These methods use either a square or triangle pattern artifact surface. The triangular pattern is especially of interest since inclusion of the laser line does not occur as shown in Figure 1.7.

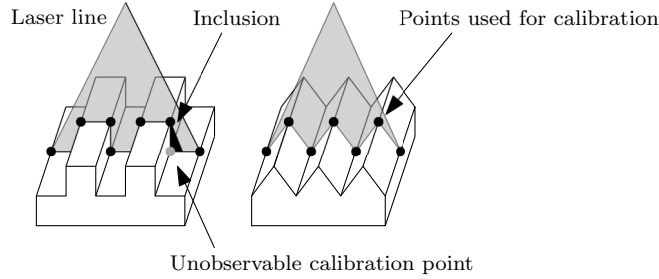


Figure 1.7: Calibration artifacts: Artifacts with square and triangular patterns used for camera calibration.

### 1.2.2 Feature point extraction

A review of common techniques used for feature point extraction in laser lines, as discussed in the previous section, is discussed in [15]. The authors state that the sequence of image processing algorithms needed to extract feature points can be divided into laser line segmentation, centerline extraction, and feature extraction.

The goal of laser line segmentation is to separate the laser line from the image background. Laser line segmentation is often achieved through thresholding methods. As an example, global thresholding is used by [16] where the thresholding value is determined by inspection of the image histogram. In [17], the threshold value is automatically determined by using the average of the maximum and minimum pixel intensity. A commonly used histogram thresholding method based upon maximizing the variance between the fore- and background is Otsu's method [18].

Centerline extraction is used to convert the laser line segment to a pixel position vector representing deposition geometry. A common method [19],[20] defines the centerline position at every column or row as the pixel position with the highest intensity. Other methods [17] and [21] take the maximum pixel difference of the segment edges and set the middle as the laser position. There also exist methods based upon morphology and the watershed algorithm as described in [15].

Feature extraction is application-specific. In [22] the absolute values of the second derivatives of the extracted laser centerline are taken as feature points. Since the shape of the calibration artifact is known on beforehand other methods can be used which exploit this knowledge. In [20] the centerline is segmented into small sections connected to each other using polygonal fitting. If a triangular pattern is used as artifact such a method seems best suitable.

Noise makes it often impossible to detect the laser centerline position for a column or row. Filtering methods are often used to reduce noise. Interpolation with a moving average filter [17] can be used to account for inhomogeneities in the laser line intensity. The distinction between the laser line and background regions is often disturbed. Convolution based filtering [23] with different kernel sizes and shapes are used to smooth the image. In [16] a median filter is used where each pixel value is replaced by the median of neighboring pixels. There are also frequency based filtering methods and wavelet methods such as in [24].

## 1.3 Problem statement

The literature on laser triangulation vision based surface measurement for additive manufacturing processes is currently limited. Furthermore, camera calibration, centerline extraction and feature point extraction are often investigated separately while both influence measurement performance.

Most of the artifact based calibration methods mentioned before rely on accurate and expensive manufactured calibration artifacts. Although a high precision artifact probably leads to more accurate camera calibration it would be interesting to see if calibration could be achieved with reasonable accuracy using a lower precision and less expensive artifacts. Also, for additive manufacturing, the deposited material and their reflective properties change often between prints. Calibration on an artifact that is not representative of the printed material could lead to problems since feature point extraction might be precise and perform well on the artifact but inaccurate and imprecise on the deposited material.

Several camera calibration techniques need the artifact to be positioned at different translated and rotated views. Often this is carried out by manual repositioning of the artifact. A fully automatic method is however preferred where the calibration artifact is fixed in position to reduce human influence in the calibration process. As discussed before, for steady state modelling of the deposition process, used for either open-loop or inter-layer based control, it is interesting to develop a process model before every build due to changing operating conditions. This also holds true for the calibration process. Camera and laser line module position might change and therefore it is preferred to perform a calibration procedure before every build. Doing the repositioning of the artifact manually is therefore not ideal.

Cameras often have the ability to change optical zoom. This way the field-of-view (FOV) of the camera can be adjusted for the size of the printed part by the AM machine. Calibration of the camera for a large FOV would require a larger calibration artifact then for a small FOV. Ideally, only a single artifact is used for either case to cut costs and simplify calibration. Feature points in the artifact could be well visible in a small FOV but only cover a small part of the image in a large FOV as shown in Figure 1.8. Due to the angular view of the laser plane by the camera, perspectivity is more apparent at the edges of the images than the center. Therefore, calibration points should preferably cover the complete FOV to improve calibration accuracy.



Figure 1.8: Camera zoom: The spread of calibration feature points in the field-of-view differs with the level of zoom and could affect calibration accuracy due to the more apparent perspectivity at image edges.

Laser centerline and feature point extraction are preferred to operate in an automated way as well. Most feature point extraction methods rely on algorithms which use predefined parameters for operation such as the kernel size for smoothing operations and the threshold value for laser line segmentation. Deposition of different materials with unknown reflectivity asks for a feature point extraction method which does not rely on any fixed parameter. Furthermore, it should be able to operate under pixel saturation, noise and changing reflectivity of the material.

To summarize, a calibration method is needed using only a single inexpensive artifact which can be used for different zooming levels of the camera such that parts of different size can be measured. A feature point extraction method needs to be developed which is insensitive to changing operating conditions such as material reflectivity and pixel saturation such that it can be used for multiple materials. The whole calibration procedure and, after calibration, measurement of the printed part should be fully automated.

## 1.4 Research objective

In this research a camera calibration method is developed for laser-triangulation based vision systems using a sloped artifact fabricated by the additive manufacturing process itself. The method allows for calibration where the artifact is positioned at an unknown angle to cover the field-of-view of the observing camera to improve calibration accuracy.

The designed artifact has a triangular pattern which changes height due to a slope in the artifact. This way, the artifact pattern and feature points will translate from the bottom to the top in the FOV of the camera if scanned at different locations of the artifact. Angular orientation of the artifact relative to the scanning direction can be applied when the camera is set for a bigger field-of-view as shown in Figure 1.9.

Instead of using an accurate and expensive artifact made from a material which is not representative of the printed material, the artifact is produced by the AM system itself. This way, the centerline extraction algorithm will perform the same on the calibration artifact as on the printed part.

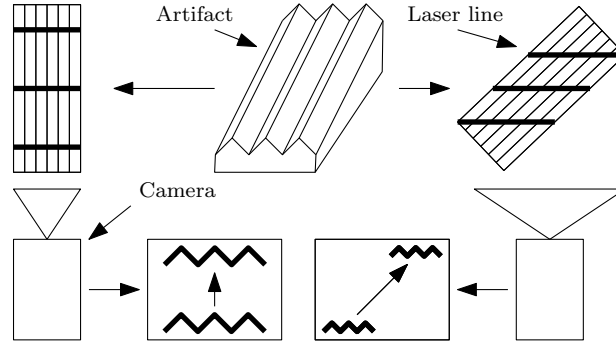


Figure 1.9: Sloped artifact: An artifact with a slope which is positioned at an angle relative to the scanning direction of the vision system will produce calibration points across the field-of-view of the camera which could enhance calibration accuracy.

## 1.5 Report outline

In chapter 2 the analytical camera model used for calibration is discussed. The mapping from points in the laser-plane to their corresponding points in the image-plane is explained. Chapter 3 describes the artifact design and the algorithms developed for centerline and feature point extraction. The calibration method is described in Chapter 4 and simulated as well as experimental results are presented in Chapter 5. The last chapter presents the conclusions of this work and the recommendations for future research.

## 2 Laser plane projection model

This chapter describes the projection model which maps point coordinates in the plane of the laser to pixel coordinates in the images acquired from the camera. Section 2.1 describes the pinhole camera model, which maps 3D points in the camera coordinate system to 2D coordinates on the plane of the camera sensor. Section 2.2 describes how coordinates in the plane of the line laser are transformed to 3D camera coordinates. Section 2.3 presents how camera intrinsics transform 2D sensor plane coordinates to pixel valued coordinates in the images acquired from the camera. Section 2.4 explains the distortion present due to lens optics which distorts the points in the 2D sensor plane. All transformations are combined in Section 2.5 to describe the projection model between image and laser-plane coordinates. An inverse model is derived which is able to map image coordinates to real-world coordinates in the laser plane. Estimating the parameters of this model is the primary goal of camera calibration which is discussed in Chapter 4. The notation as well as all transformations in the first three sections are analogous to the description in [25].

### 2.1 The pinhole camera model

The pinhole camera model is used to describe the projection of coordinates in the 3D camera coordinate system  $(O_C, X_C, Y_C, Z_C)$  to 2D camera sensor plane coordinates  $(O_S, X_S, Y_S)$  and is visualized in Figure 2.1. The sensor plane  $(O_S, X_S, Y_S)$  is positioned in parallel with the  $(O_C, X_C, Y_C)$  plane of the 3D camera coordinate system at a focal distance  $f$  collinear with the  $Z_C$  axis. The center of the sensor plane  $O_S$  lies on the  $Z_C$  axis. Points on the same ray going through the optical center  $O_C$  are projected onto the same coordinates of the sensor plane. The point  $\mathbf{x}_c = (x_c, y_c, z_c)^T$  is seen as the point  $\mathbf{x}_s = (x_s, y_s)^T$  on the laser plane and any point inbetween those points on the same ray will be projected to the same point  $\mathbf{x}_s$ .

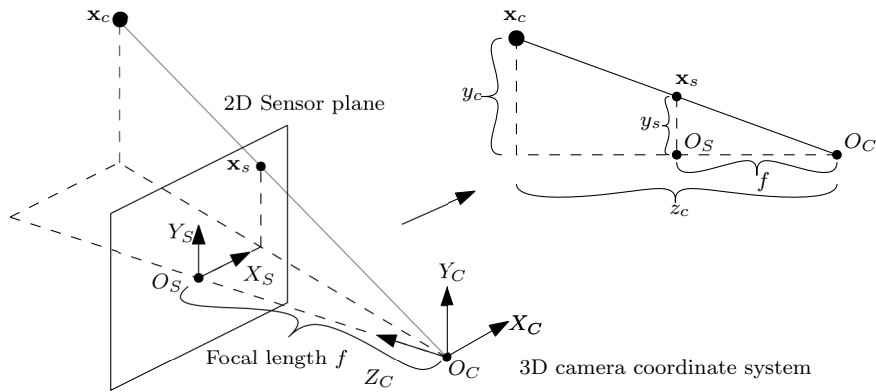


Figure 2.1: Pinhole camera model: A point  $\mathbf{x}_c = (x_c, y_c, z_c)^T$  in 3D camera coordinate system  $(O_C, X_C, Y_C, Z_C)$  is projected to a point  $\mathbf{x}_s = (x_s, y_s)^T$  on the 2D sensor plane  $(O_S, X_S, Y_S)$  dependent on focal length  $f$ .

Using triangle similarity as shown in Figure 2.1, every point in the camera coordinate system is related to a point in the sensor plane by:

$$\frac{y_c}{z_c} = \frac{y_s}{f} \rightarrow y_s = \frac{f}{z_c} y_c \quad \frac{x_c}{z_c} = \frac{x_s}{f} \rightarrow x_s = \frac{f}{z_c} x_c \quad (2.1)$$

In matrix form this reduces to:

$$\mathbf{x}_s = \begin{pmatrix} x_s \\ y_s \end{pmatrix} = \frac{f}{z_c} \begin{pmatrix} x_c \\ y_c \end{pmatrix} \quad (2.2)$$

Equation 2.2 describes the relation between the two coordinate systems in Cartesian coordinates and is a nonlinear relationship. This perspective transformation can be transformed to a linear relationship with the use of *homogenous coordinates* such that

$$\underbrace{\begin{pmatrix} f x_c \\ f y_c \\ z_c \end{pmatrix}}_{\hat{\mathbf{x}}_s} = \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{M}_f} \underbrace{\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}}_{\hat{\mathbf{x}}_c} \quad (2.3)$$

where  $\hat{\mathbf{x}}_s$  and  $\hat{\mathbf{x}}_c$  are the homogenous equivalents of  $\mathbf{x}_s$  and  $\mathbf{x}_c$  and  $\mathbf{M}_f$  is the perspective transformation matrix. Conversion of points in homogenous( $\hat{\mathbf{x}}$ ) to Cartesian( $\mathbf{x}$ ) coordinates is possible by dividing all entries of the vector by the value in the last row followed by erasing the last row. The inverse conversion can be obtained by simply adding a row with a one. Converting to homogenous coordinates is invariant to scale as the coordinate vector can be multiplied with any scalar. For the pinhole camera model this results in:

$$\mathbf{x}_s = \begin{pmatrix} x_s \\ y_s \end{pmatrix} = \frac{f}{z_c} \begin{pmatrix} x_c \\ y_c \end{pmatrix} \equiv \begin{pmatrix} f x_c / z_c \\ f y_c / z_c \\ 1 \end{pmatrix} \equiv \begin{pmatrix} f x_c \\ f y_c \\ z_c \end{pmatrix} = \hat{\mathbf{x}}_s \quad (2.4)$$

$$\mathbf{x}_c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \equiv \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \hat{\mathbf{x}}_c \quad (2.5)$$

To simplify conversion the operator  $\mathcal{H}$  is introduced as:

$$\hat{\mathbf{x}} = \mathcal{H}[\mathbf{x}] \rightarrow \mathbf{x} = \mathcal{H}^{-1}[\hat{\mathbf{x}}] \quad (2.6)$$

Equation 2.3 can therefore be written in Cartesian coordinates using operator  $\mathcal{H}$  as:

$$\mathbf{x}_s = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{M}_f} \mathcal{H}[\mathbf{x}_c] \right] \quad (2.7)$$

## 2.2 Extrinsic view

Camera extrinsics represent the transformations needed to map points in their own 3D world coordinate system ( $O_W, X_W, Y_W, Z_W$ ) to the 3D camera coordinate system ( $O_C, X_C, Y_C, Z_C$ ) as shown in Figure 2.2. A point  $\mathbf{x}_w = (x_w, y_w, z_w)^T$  in 3D world coordinates can be transformed to a point  $\mathbf{x}_c = (x_c, y_c)^T$  in the 3D camera



coordinate system using a rotation matrix  $\mathbf{R}$  and translation vector  $\mathbf{t} = (t_x, t_y, t_z)^T$  such that:

$$\mathbf{x}_c = \mathbf{R}\mathbf{x}_w + \mathbf{t} \quad (2.8)$$

Matrix  $\mathbf{R}$  allows for rotation around all coordinate axis ( $X_W, Y_W, Z_W$ ) and can be modelled as  $\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$  where matrices  $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$  rotate the coordinate system by angles  $\theta_x, \theta_y, \theta_z$  around its corresponding axis such that:

$$\underbrace{\begin{pmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{pmatrix}}_{\mathbf{R}} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{pmatrix}}_{\mathbf{R}_x} \underbrace{\begin{pmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{pmatrix}}_{\mathbf{R}_y} \underbrace{\begin{pmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{R}_z}$$

In homogenous coordinates,  $\mathbf{R}$  and  $\mathbf{t}$  can be combined into a single matrix such that the transformation with the use of operator  $\mathcal{H}$  is written as:

$$\mathbf{x}_c = \mathcal{H}^{-1} \left[ \begin{pmatrix} r_1 & r_4 & r_7 & t_x \\ r_2 & r_5 & r_8 & t_y \\ r_3 & r_6 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathcal{H}[\mathbf{x}_w] \right] \quad (2.9)$$

Laser triangulation based vision as described in Chapter 1 simplifies the 3D world coordinate system ( $O_W, X_W, Y_W, Z_W$ ) to a 2D laser plane coordinate system ( $O_L, X_L, Y_L$ ), as shown in Figure 2.2, since all points observed in the real world lie on the laser plane. A point  $\mathbf{x}_l = (x_l, y_l)^T$  in the laser plane can be seen as a point  $\mathbf{x}_w^0 = (x_w, y_w, 0)^T$  in the 3D world where every point lies on the plane ( $O_W, X_W, Y_W$ ).

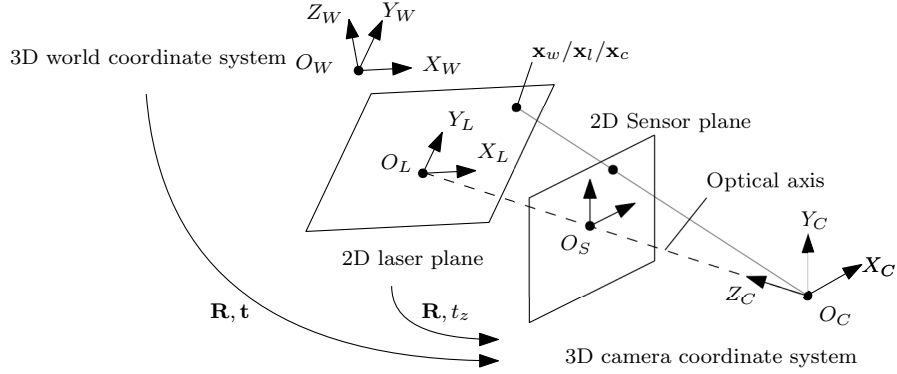


Figure 2.2: Extrinsic view: A point  $\mathbf{x}_w$  in 3D world coordinate system ( $O_W, X_W, Y_W, Z_W$ ) is transformed to a point  $\mathbf{x}_c$  in the 3D camera coordinate system ( $O_C, X_C, Y_C, Z_C$ ) using rotation matrix  $\mathbf{R}$  and translation vector  $\mathbf{t}$ . In laser-triangulation based vision this is reduced to the transformation of a point  $\mathbf{x}_l$  in the laser plane ( $O_L, X_L, Y_L$ ) to the camera coordinate system. If the origin  $O_L$  is defined to lie on the optical axis only translation  $t_z$  is needed.

The origin of the laser plane  $O_L$  can be chosen freely and, as a first deviation from the theory as given in [25], it is possible to define the center of the laser plane  $O_L$  to be aligned with the optical axis such that no translation in  $t_x$  and  $t_y$  is needed. Equation 2.9 can therefore be rewritten as a transformation from a point  $\mathbf{x}_l$  in the laser plane to a 3D point  $\mathbf{x}_c$  in the camera coordinate system using homogenous

coordinate transformation matrix  $\mathbf{M}_e$  as follows:

$$\mathbf{x}_c = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} r_1 & r_4 & r_7 & 0 \\ r_2 & r_5 & r_8 & 0 \\ r_3 & r_6 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{\hat{x}}_w^0} \underbrace{\begin{pmatrix} x_w \\ y_w \\ 0 \\ 1 \end{pmatrix}}_{\mathbf{M}_e} \right] = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} r_1 & r_4 & 0 \\ r_2 & r_5 & 0 \\ r_3 & r_6 & t_z \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_e} \mathcal{H}[\mathbf{x}_l] \right] \quad (2.10)$$

### 2.3 Sensor intrinsics

Coordinate transformation of points in the sensor plane to image plane coordinates in pixels is subject to sensor intrinsic parameters which is visualized in Figure 2.3. The transformation of a point  $\mathbf{x}_s$  in the sensor plane ( $O_S, X_S, Y_S$ ) to a point  $\mathbf{x}_i$  in the image plane ( $O_I, X_I, Y_I$ ) is described by affine transformation matrix  $\mathbf{S}$  and translation vector  $\mathbf{t}$  as:

$$\mathbf{x}_i = \underbrace{\begin{pmatrix} s_x & \beta \\ 0 & s_y \end{pmatrix}}_{\mathbf{S}} \mathbf{x}_s + \underbrace{\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}}_{\mathbf{t}} \quad (2.11)$$

where  $s_x$  and  $s_y$  are scale factors to transform real world sensor coordinates to pixel valued image coordinates. Non square pixels are modelled if  $s_x \neq s_y$  and a single factor  $s = s_x = s_y$  can be used if pixels are square. Variable  $\beta$  is used to model skewed pixels as shown in Figure 2.3. The offset between the origin  $O_S$  of the laser plane and the origin  $O_I$  of the image plane is compensated for by translations  $x_0$  and  $y_0$ . Cartesian transformation components  $\mathbf{S}$  and  $\mathbf{t}$  can be combined into transformation matrix  $\mathbf{M}_i$  for homogenous coordinates. Equation 2.11 can therefore be written as:

$$\mathbf{x}_i = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} s_x & \beta & x_0 \\ 0 & s_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_i} \mathcal{H}[\mathbf{x}_s] \right] \quad (2.12)$$

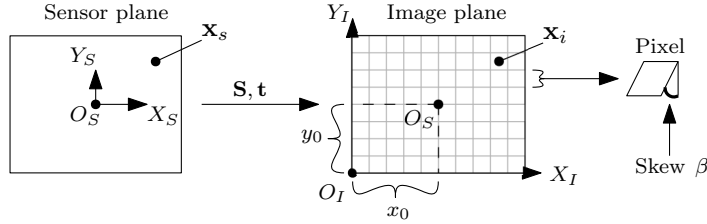


Figure 2.3: Sensor intrinsics: Transformation of points  $\mathbf{x}_s$  in the sensor plane ( $O_S, X_S, Y_S$ ) to points  $\mathbf{x}_i$  in image plane ( $O_I, X_I, Y_I$ ) using matrix  $\mathbf{S}$  and translation vector  $\mathbf{t}$  which model camera intrinsic parameters compensating for sensor scale, principle point offset ( $x_0, y_0$ ) and skewed pixels ( $\beta$ ).

### 2.4 Lens distortion

Ideally, the points observed in the sensor plane follow the pinhole camera model as described in Section 2.1. This model is however neglecting the influence of lens optics positioned in front of the camera's sensor. Lens optics distort as depicted in Figure 2.4 and the most influential sources of disturbance ordered from most to least prominent according to [26] and [25] are:

- *Radial distortion* which is caused by light refractions due to the "fish-eye" effect. Light rays entering the optical lens further from the lens center tend to cause higher distortion resulting in barrel or pincushion distortion.
- *Tangential distortion* which is caused by a non-parallel alignment of the sensor chip with respect to the optical lens.
- *Decentering distortion* which is the result of mis-alignment of the optics with the optical axis.

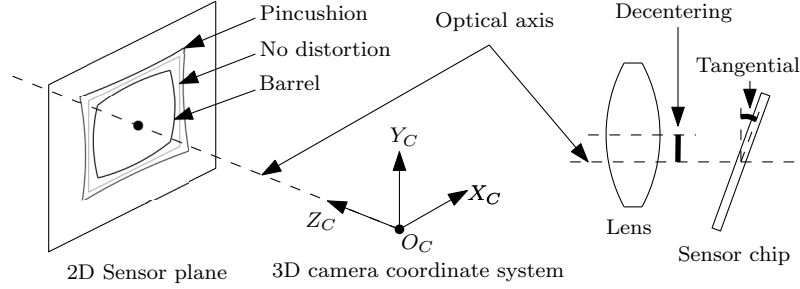


Figure 2.4: Lens distortion: Possible sources of lens distortion. Radial distortion in the form of barrel or pincushion distortion, tangential distortion due to a non-parallel sensor chip and decentering distortion as a result of the mis-alignment of lens optics relative to the optical axis.

Although it is interesting to develop an analytical model  $\mathbf{D} : \mathbf{x}_s \rightarrow \mathbf{x}_{ds}$  describing how (ideal) sensor plane points  $\mathbf{x}_s$  are mapped to (distorted) points  $\mathbf{x}_{ds}$  it is far more important to develop an inverse model  $\mathbf{D}^{-1} : \mathbf{x}_{ds} \rightarrow \mathbf{x}_s$  since the goal is to develop and tune a model which converts pixel coordinates in the image plane to real-world coordinates in the laser plane for measurement operation. In [27] such a model is described which can be written as:

$$\mathbf{x}_s = \mathbf{D}^{-1}(\mathbf{k}_\infty, \mathbf{p}_\infty, \mathbf{c}, \mathbf{x}_{ds}) = \underbrace{\begin{pmatrix} \hat{x}_{ds} \\ \hat{y}_{ds} \end{pmatrix} r_\infty}_{radial} + \underbrace{\begin{pmatrix} 2p_1 \hat{x}_{ds} \hat{y}_{ds} + p_2 (r_{ds}^2 + 2\hat{x}_{ds}^2) \\ 2p_2 \hat{x}_{ds} \hat{y}_{ds} + p_1 (r_{ds}^2 + 2\hat{y}_{ds}^2) \end{pmatrix} t_\infty}_{tangential} \quad (2.13)$$

where

$$\begin{aligned} \mathbf{k}_\infty &= (k_1, k_2, \dots)^T, \quad \mathbf{p}_\infty = (p_1, p_2, \dots)^T, \quad \mathbf{c} = (x_c, y_c)^T, \quad \mathbf{x}_{ds} = (x_{ds}, y_{ds})^T, \\ r_\infty &= (1 + k_1 r_{ds}^2 + k_2 r_{ds}^4 + \dots), \quad t_\infty = (1 + p_3 r_{ds}^2 + \dots), \quad r_{ds} = \sqrt{\hat{x}_{ds}^2 + \hat{y}_{ds}^2}, \\ \hat{x}_{ds} &= x_{ds} - x_c, \quad \hat{y}_{ds} = y_{ds} - y_c \end{aligned}$$

with  $\mathbf{k}_\infty$  and  $\mathbf{p}_\infty$  being infinite dimensional parameter vectors compensating for radial and tangential distortion. Parameter vector  $\mathbf{c}$  compensates for the decentering component.

In [12] it is proven that lens distortion can be corrected for in the image plane instead of the sensor plane by the inverse model if pixels are square and without skew. The inverse model can therefore be used to transform distorted image points  $\mathbf{x}_{di}$  to undistorted image points  $\mathbf{x}_i$  and no further correction is needed in the sensor plane.

$$\mathbf{x}_i = \mathbf{D}^{-1}(\mathbf{k}_\infty, \mathbf{p}_\infty, \mathbf{c}, \mathbf{x}_{di}) \quad (2.14)$$

## 2.5 Projection chain

The projection of points  $\mathbf{x}_l$  in the laser plane ( $O_L, X_L, Y_L$ ) to (distorted) points  $\mathbf{x}_{di}$  in the image plane ( $O_I, X_I, Y_I$ ) can be obtained by combining transformations in the previous sections as:

$$\mathbf{x}_s = \mathcal{H}[\mathbf{M}_f \cdot \mathbf{M}_e \cdot \mathcal{H}[\mathbf{x}_l]], \quad \mathbf{x}_{ds} = \mathbf{D}(\mathbf{x}_s), \quad \mathbf{x}_{di} = \mathcal{H}^{-1}[\mathbf{M}_i \cdot \mathcal{H}[\mathbf{x}_{ds}]]$$

where  $\mathbf{D}$  is the undefined forward distortion model as discussed in Section 2.4. Merged together this leads to:

$$\mathbf{x}_{di} = \mathcal{H}^{-1}[\mathbf{M}_i \cdot \mathcal{H}[\mathbf{D}(\mathcal{H}^{-1}[\mathbf{M}_f \cdot \mathbf{M}_e \cdot \mathcal{H}[\mathbf{x}_l]])]] \quad (2.15)$$

As explained in Section 2.4, the inverse distortion model can be used to undistort in the image plane instead of the sensor plane if pixels are square  $s_x = s_y = s$  and without skew  $\beta = 0$ . Therefore, distortion in the sensor plane by the forward distortion model  $\mathbf{D}$  can be omitted and compensated for by the inverse model  $\mathbf{D}^{-1}$  in the image plane. Additionally it is assumed that there is no offset between the origin  $O_I$  of the image plane and the origin  $O_S$  of the sensor plane such that  $x_0 = y_0 = 0$ . The projection from points  $\mathbf{x}_l$  in the laser plane to points  $\mathbf{x}_i$  in the image plane can then, with  $\mathbf{M}_h = \mathbf{M}_i \mathbf{M}_f \mathbf{M}_e$ , be described as:

$$\mathbf{x}_i = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_i} \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{M}_f} \underbrace{\begin{pmatrix} r_1 & r_4 & 0 \\ r_2 & r_5 & 0 \\ r_3 & r_6 & t_z \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_e} \mathcal{H}[\mathbf{x}_l] \right] \quad (2.16)$$

$$= \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} h_1 & h_4 & 0 \\ h_2 & h_5 & 0 \\ h_3 & h_6 & h_9 \end{pmatrix}}_{\mathbf{M}_h} \mathcal{H}[\mathbf{x}_l] \right] \quad (2.17)$$

Three assumptions are made regarding the inverse distortion model. First it is assumed that the center of distortion is aligned with the optical axis and therefore decentering distortion is assumed to be zero or  $\mathbf{c} = (0, 0)^T$  and can be omitted from the inverse distortion model. The second and third assumptions are that two parameters each for radial and tangential distortion compensation, in the form of parameter vectors  $\mathbf{k} = (k_1, k_2)^T$  and  $\mathbf{p} = (p_1, p_2)^T$ , is adequate. Therefore, by replacing the forward distortion model with the inverse distortion model  $\mathbf{D}^{-1}$ , combined with the previous assumptions, the relation between distorted ( $\mathbf{x}_{di}$ ) and undistorted ( $\mathbf{x}_i$ ) points in the image plane is defined as:

$$\mathbf{x}_i = \mathbf{D}^{-1}(\mathbf{k}, \mathbf{p}, \mathbf{x}_{di}) \quad (2.18)$$

Combined with Equation 2.17 this leads to the following equation describing the relation between points  $\mathbf{x}_l$  in the laser plane and  $\mathbf{x}_{di}$  in the image plane as:

$$\mathbf{D}^{-1}(\mathbf{k}, \mathbf{p}, \mathbf{x}_{di}) = \mathcal{H}^{-1}[\mathbf{M}_h \cdot \mathcal{H}[\mathbf{x}_l]] \quad (2.19)$$

The inverse projection model which maps points  $\mathbf{x}_{di}$  in the image plane to points  $\mathbf{x}_l$  in the laser plane can be derived from Equation 2.19 such that

$$\mathbf{x}_l = \mathcal{H}^{-1}[\mathbf{M}_h^{-1} \cdot \mathcal{H}[\mathbf{D}^{-1}(\mathbf{k}, \mathbf{p}, \mathbf{x}_{di})]] \quad (2.20)$$

or can be defined as an *Image-to-Laser plane Projection* function (*ILP*):

$$\mathbf{x}_l = ILP(\mathbf{h}, \mathbf{k}, \mathbf{p}, \mathbf{x}_{di}) \quad (2.21)$$

with  $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_9)^T$  a parameter vector describing matrix  $\mathbf{M}_h$

### 3 Calibration artifact

The proposed artifact design used for camera calibration is considered in this chapter. First the projection of the artifact onto the laser plane is discussed. Then the developed centerline extraction algorithm is discussed which extracts the laser centerline as projected onto the artifact from the images. At last the calibration feature point extraction algorithm is discussed which identifies the feature points of the artifact in the images using the extracted centerlines.

#### 3.1 Artifact projection

As proposed in Chapter 1 for this research an artifact is used with a slope. When the laser triangulation system is moved over the artifact the projected line laser on the artifact will change position in a vertical direction of the laser plane for every image taken. Rotation of the artifact with respect to the direction of motion of the laser plane will result in a translation of the line laser pattern in a horizontal direction of the laser plane as shown in Figure 1.9. The slope and rotation of the artifact produce line laser patterns across the laser plane and therefore cover a bigger field-of-view of the camera.

The laser plane projection model as explained in Chapter 2 accounts for lens distortion and the perspective view of the camera. Radial and tangential lens distortion as well as the perspective view are more apparant at image edges. The generation of calibration points spread out over the FOV of the camera is therefore thought to improve calibration accuracy.

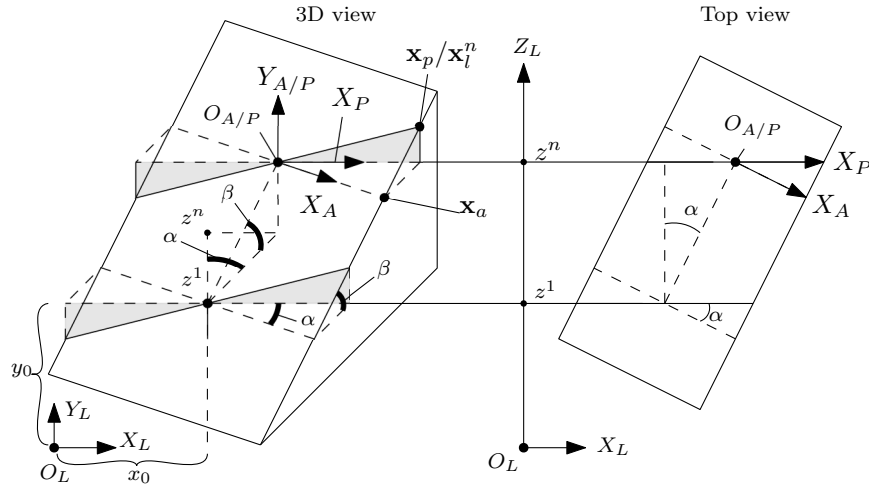


Figure 3.1: Artifact projection: Artifact pattern plane ( $O_A, X_A, Y_A$ ) is defined to exist at every cross section of the artifact. A point  $\mathbf{x}_a$  on that plane is therefore equal at every cross section on the artifact. Parallel pattern plane ( $O_P, X_P, Y_P$ ) is the observed pattern plane in the laser plane ( $O_L, X_L, Y_L$ ) dependent on artifact slope angle  $\beta$  and angular rotation  $\alpha$  such that a point  $\mathbf{x}_a$  is observed as  $\mathbf{x}_p$ . Position of the parallel pattern plane in the laser plane is dependent on initial translation parameters  $x_0, y_0$  and movement of the vision system orthogonal to the laser plane in direction  $Z_L$  such that point  $\mathbf{x}_p$  is transformed to a point  $\mathbf{x}_l^n$  in the laser plane dependent on the position  $z^n$  of image  $n$ .

The exact artifact pattern on the slope of the artifact, such as the patterns as shown in Figure 1.7, is not of interest at this point. The projection of the pattern onto the laser plane can, without loss of generality, be explained by an artifact without any pattern as shown in Figure 3.1. The coordinate system  $(O_A, X_A, Y_A)$ , which is a cross section plane of the artifact exists at every position on the slope of the artifact. Point  $\mathbf{x}_a = (x_a, y_a)^T$  lies on such a cross section as depicted in the figure. If the artifact is rotated relative to the laser plane  $(O_L, X_L, Y_L)$  by angle  $\alpha$  the observed cross section changes and point  $\mathbf{x}_a$  is observed as point  $\mathbf{x}_p = (x_p, y_p)^T$  within coordinate system  $(O_P, X_P, Y_P)$  which is parallel to the laser plane  $(O_L, X_L, Y_L)$ . This transformation is determined based upon the angles  $\alpha$  and  $\beta$  in Figure 3.1 and is in Cartesian coordinates defined as:

$$\mathbf{x}_p = \begin{pmatrix} \cos(\alpha)^{-1} & 0 \\ \tan(\beta) \tan(\alpha) & 1 \end{pmatrix} \mathbf{x}_a \quad (3.1)$$

To acquire multiple images at different locations or cross sections of the artifact the vision system is moved in direction  $Z_L$ , as shown in Figure 3.1, which is orthogonal to the laser plane. The transformation from a point  $\mathbf{x}_p$  in coordinate system  $(O_P, X_P, Y_P)$ , which exists at every cross section of the artifact, to the point  $\mathbf{x}_l^n$  in the laser plane coordinate system  $(O_L, X_L, Y_L)$  at position  $z^n$  is defined as:

$$\mathbf{x}_l^n = \mathbf{x}_p + \begin{pmatrix} x_0 + \tan(\alpha)(z^n - z^1) \\ y_0 + \frac{\tan(\beta)}{\cos(\alpha)}(z^n - z^1) \end{pmatrix} \quad (3.2)$$

where  $z^n$  is the position of the laser triangulation system when image  $n$  is taken and  $z^1$  is the position of the first image. The vision system is assumed to move at constant velocity  $V$  and each image acquired from the camera is time-stamped with time  $T^n$ . Therefore the distance travelled between the initial image  $z^1$  and image  $z^n$  is rewritten as:

$$\Delta z^n = z^n - z^1 = V(T^n - T^1) \quad (3.3)$$

Equations 3.1, 3.2 and 3.3 are combined into a transformation which maps points  $\mathbf{x}_a$  in an artifact cross section plane to points  $\mathbf{x}_l^n$  in the laser plane dependent on scan velocity  $V$  and time of image acquisition  $T^n$ .

$$\mathbf{x}_l^n = \underbrace{\begin{pmatrix} a & 0 \\ bc & 1 \end{pmatrix}}_{\mathbf{R}_a} \mathbf{x}_p + \underbrace{\begin{pmatrix} x_0 + c\Delta z^n \\ y_0 + ab\Delta z^n \end{pmatrix}}_{\mathbf{t}_a^n} \quad (3.4)$$

where  $a = \cos(\alpha)^{-1}$ ,  $b = \tan(\beta)$ ,  $c = \tan(\alpha)$  and  $\Delta z^n = V(T^n - T^1)$ . Affine matrix  $\mathbf{R}_a$  rotates the artifact pattern such that it becomes parallel to the laser plane. Translation vector  $\mathbf{t}_a^n$  translates the points to the laser plane coordinate system. In homogenous coordinates, with  $\mathbf{R}_a \equiv \mathbf{M}_r$  and  $\mathbf{t}_a^n \equiv \mathbf{M}_t^n$  this can be written as:

$$\mathbf{x}_l^n = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} 1 & 0 & x_0 + c\Delta z^n \\ 0 & 1 & y_0 + ab\Delta z^n \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_t^n} \underbrace{\begin{pmatrix} a & 0 & 0 \\ bc & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_r} \mathcal{H}[\mathbf{x}_a] \right] \quad (3.5)$$

or as a *Artifact-to-Laser Plane* function (*ALP*):

$$\mathbf{x}_l^n = ALP(\alpha, \beta, V, T^1, T^n, \mathbf{x}_a) \quad (3.6)$$

### 3.2 Line laser centerline extraction

The laser triangulation system, when calibrated, is used for measurement of objects. The laser projects a line on top of the surface of the object and, due to the angular position of the camera relative to the laser plane, the geometry or size of the object in the laser plane can be detected as explained in Chapter 1 and visualized in Figure 1.6. The laser line width is not infinitely small in reality and the intensity of the laser line is higher at the center of the line than the edges. Figure 3.2 shows a more realistic view of the laser line, in the real world as well as in the images acquired by the camera. The goal is to extract the centerline of the laser line for each image as discussed in Section 3.2.2. This centerline will then be used to find calibration points, as discussed in Section 3.3. The centerline, although depicted as being a straight line, follows the shape of the object and can therefore have any position in the image.

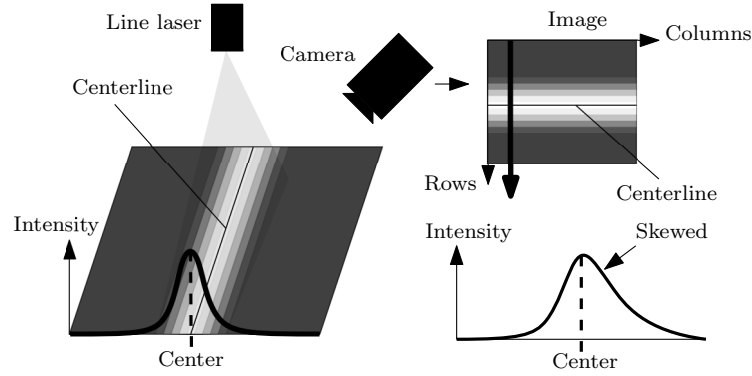


Figure 3.2: Line laser shape: The line laser has highest intensity at the center and reduced intensity at the edges. A centerline extraction algorithm is necessary to extract the center of the line laser. The line laser shape, as seen in an image acquired by the camera, is distorted and can appear skewed due to the perspective view of the camera.

#### 3.2.1 Challenges in center-line extraction

Without any knowledge about the shape of the centerline, the position of the centerline in the images needs to be determined for every column in the image. As summarized in Chapter 1, many centerline extraction algorithms exist. Automation and adaption without any parameters is however not very common. Noise is most often filtered and cut-off frequencies or kernel sizes are chosen to fit a particular use case. The shape of a cross section of the line laser might show similarity with a Bell curve. The corresponding camera image output might however be completely different. Properties affecting the observed laser line shape by the camera are summarized as follows:

- *Material reflectiveness* has an influence on the shape of the laserline. Smooth light glossy materials will reflect more than matte dark rough materials.
- *Pixel saturation* might occur in images of the camera. The intensity pattern of a column of the image might therefore be thresholded for all pixels around and including its maximum.
- *Discretization* of real intensity values in the camera due to the finite pixel bit-depth. Therefore the real intensity values are round-off to a discrete value which introduces an error.

- *Noise* is expected to be present due to a variety of sources. Lens optics impurity of the line laser, local reflectivity changes due to surface roughness and observation error, such as skeweness due to the perspective view as shown in Figure 3.2, are some of the potential causes.

The laser-triangulation based vision system should be flexible in the sense that change of deposition material should not affect operation. Furthermore, it should be able to detect the centerline without knowledge about the material. In 3D printing it is common to use multiple materials in a single build or layer. Therefore the centerline extraction algorithm should be column-wise adaptive to these properties.

Existing centerline detection algorithms which rely on peak detection, filtering and edge detection through differencing are in general not adaptive. Peak detection is impossible in images where saturation is present. Filters can be adaptive in the sense that an algorithm is developed to find a suitable cut-off frequency or kernel size at the cost of additional complexity. Edge detection through differencing is hard in situations with a high signal-to-noise ratio.

### 3.2.2 Column-wise adaptive centerline extraction (CACE)

A new centerline extraction algorithm is therefore developed in this research addressing most of the problems previously discussed. For each column of the image, template matching is conducted. A specific template is constructed for each image column based upon data of that single column only. Therefore, differences in saturation or line width between columns do not affect operation. Cross-correlation is used for template matching to find the position where the template best fits the column data. Subpixel accuracy is achieved with the use of quadratic interpolation of the cross-correlation function around its peak. The algorithm is further explained by the steps below and visualized in Figure 3.3.

For each column  $C$  in an image acquired from the camera:

1. Sort  $C$  lowest at  $i = 0$  to highest at  $i = N$ , to create sorted column  $S$  where:

$$S(i+1) \geq S(i) \quad \forall \quad i \in \{0, \dots, N-1\} \quad (3.7)$$

2. Construct array  $L$  by linear interpolation between  $S(0)$  and  $S(N)$ :

$$L(i) = \frac{S(N) - S(0)}{N}i + S(0) \quad \forall \quad i \in \{0, \dots, N\} \quad (3.8)$$

3. Determine  $i = i_M$  for which the distance between  $S$  and  $L$  is largest:

$$i_M = \operatorname{argmax}(L - S) \quad (3.9)$$

4. Create template  $T$  by concatenation of  $S(i_M, \dots, N)$  and its mirror:

$$T = \{S(i_M), \dots, S(N-1), S(N), S(N-1), \dots, S(i_M)\} \quad (3.10)$$

5. Upsample the column or downsample the template. Template  $T$  is, due to its concatenation, not representative of the shape of the actual column. In this research upsampling of the column is used. Linear interpolation of column  $C$  results in upsampled version  $U$  with:

$$U = \{C(0), \frac{C(1) + C(0)}{2}, C(1), \dots, \frac{C(N) + C(N-1)}{2}, C(N)\} \quad (3.11)$$



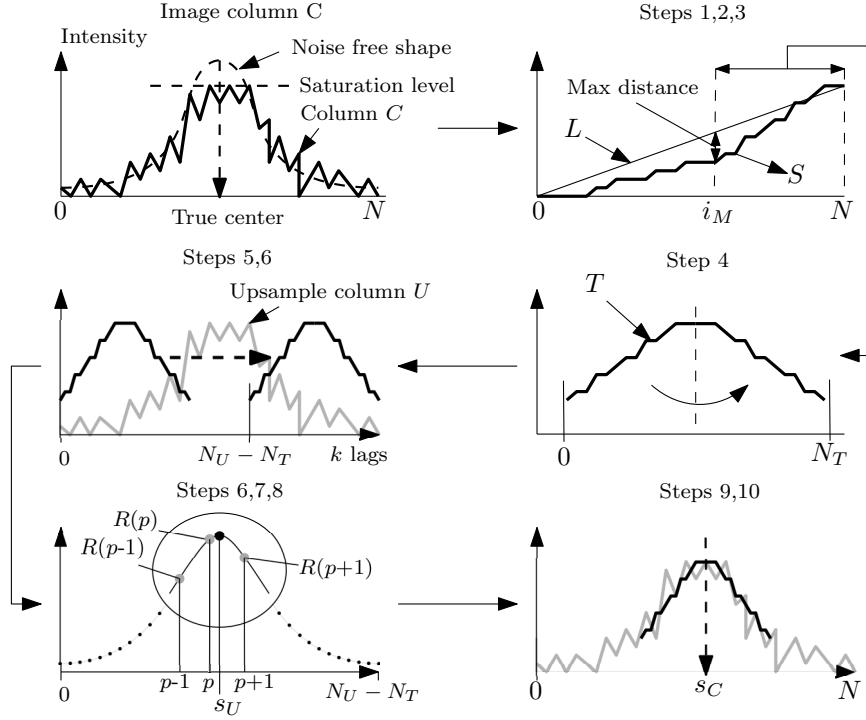


Figure 3.3: Centerline extraction: The column-wise adaptive centerline extraction algorithm (CACE). From left to right and top to bottom: (Noisy) image column  $C$  is thresholded due to pixel saturation. Create sorted column  $S$  and segment laser line from background using slope  $L$  where max distance index  $i_M$  determines the segmentation point (Steps 1,2,3). Create template  $T$  from mirror (Step 4). Create upsampled column  $U$  from  $C$  and generate cross-correlation function (Steps 5,6). Find subsample peak  $s_U$  (Steps 6,7,8). Downsample to get center  $s_C$  (Steps 9,10).

6. Generate the cross-correlation function  $R$  for all possible positions of the template on top of the column  $C$  where  $N_T = 2(N - i_M) - 1$  is the number of data points in the template  $T$  and  $N_U = 2N - 1$  is the number in upsampled column  $U$ .

$$R(i) = \sum_{k=0}^{N_T} T(k)U(i+k) \quad \forall \quad i \in \{0, \dots, N_U - N_T\} \quad (3.12)$$

7. Find the peak location  $p$  in  $R$  by:

$$p = \operatorname{argmax}(R) \quad (3.13)$$

8. Apply quadratic interpolation to achieve subpixel accuracy. Using 3-points  $R(m-1)$ ,  $R(m)$ ,  $R(m+1)$  leads to peak location  $s$  with:

$$s = p + \frac{R(p-1) - R(p+1)}{2(R(p-1) - 2R(p) + R(p+1))} \quad (3.14)$$

9. Add known center of template to find peak location  $s_U$  in column  $U$ .

$$s_U = s + N - i_M \quad (3.15)$$

10. Downsample back to find peak location  $s_C$  in column  $C$ .

$$s_C = \frac{s_U + 1}{2} \quad (3.16)$$

For template construction and matching no fixed parameters are used. The algorithm does not use filtering techniques either. The central believe is that the created template  $T$  is a reasonable approximation of the noise free column. For the construction of  $T$ , array  $L$  is used to separate the background from intensities belonging to the laser line which serves as an alternative to differencing which is more sensitive to noise.

### 3.3 Calibration point extraction

Camera calibration is the process of developing a (back-)projection model such that whenever a centerline is measured in an image it can be backprojected to the real-world. The calibration method, which is used to estimate the parameters of the projection model, is discussed in Chapter 4 and needs feature correspondences for calibration. A feature correspondence is defined as a feature where its coordinates are known in the real-world, or the laser plane and in the image acquired from the camera. Point correspondence is used in this research, although line correspondence is possible too as explained in [10], where ideally an exact relation is known between a point  $\mathbf{x}_l$  in the laser plane ( $O_L, X_L, Y_L$ ) and the corresponding (distorted) point  $\mathbf{x}_{di}$  in the image plane ( $O_I, X_I, Y_I$ ).

The artifact pattern used for calibration in this research is specifically designed to simplify feature point detection. The designed pattern is a triangular wave in coordinate system ( $O_A, X_A, Y_A$ ) as shown in Figure 3.4 where the intersection or corners of the pattern are used as feature points for calibration. In reality the produced artifact will have a surface roughness and other sources of errors relative to the ideal pattern. Since for this research the artifact is printed by the AM system itself, it will be prone to additional errors due to over- and under-filling as explained in Chapter 1.

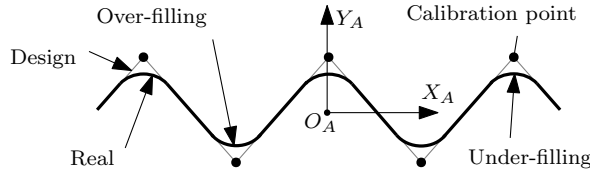


Figure 3.4: Calibration pattern: Real calibration pattern on surface of printed artifact will deviate from ideal pattern due to over- and under-filling as a result of fabrication through additive manufacturing. Calibration points are located at the intersection of the ideal line segments in the pattern. The pattern in coordinate system ( $O_A, X_A, Y_A$ ) exists at every cross section of the artifact.

The projection of a point  $\mathbf{x}_a$  in a artifact cross section plane ( $O_A, X_A, Y_A$ ) to a distortion-free point  $\mathbf{x}_i^n$  in image plane ( $O_I, X_I, Y_I$ ) for image  $n$  is described by combining Equations 3.5 and 2.17 such that:

$$\mathbf{x}_i^n = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} sfr_{00} & sfr_{01} & 0 \\ sfr_{10} & sfr_{11} & 0 \\ r_{20} & r_{21} & t_z \end{pmatrix}}_{\mathbf{M}_h} \underbrace{\begin{pmatrix} 1 & 0 & x_0 + c\Delta z^n \\ 0 & 1 & y_0 + ab\Delta z^n \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_t^n} \underbrace{\begin{pmatrix} a & 0 & 0 \\ bc & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_r} \mathcal{H}[\mathbf{x}_a] \right] \quad (3.17)$$

$$= \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}}_{\mathbf{H}} \mathcal{H}[\mathbf{x}_a] \right] \quad (3.18)$$

where  $\mathbf{H} = \mathbf{M}_h \mathbf{M}_t^n \mathbf{M}_r$  is a perspective projection matrix or *homography*. This transformation is invariant to lines meaning that lines in the artifact pattern plane appear as lines in the image plane as described in [10]. The triangular artifact pattern is constructed from linear line segments connected at the location of the calibration points. The observation of this pattern in the image plane will therefore also be an interconnection of linear line segments. Lens distortion bends lines and therefore the assumed invariance to lines does not hold. The distance between calibration points is however assumed to be relative small. Therefore linear approximation is thought to be reasonable.

After centerline extraction in the image plane using the algorithm as described in Section 3.2.2 the corresponding position of the calibration points in the image plane are found using piecewise linear regression using the method explained in [28]. The piecewise fit is first done using all data in the centerline. Due to over- and under-filling this fit is however biased since the round-off edges influence the calibration point positions. To overcome this bias, after the first fit a second fit is applied. This fit uses the point locations as determined by the first fit as initial conditions and only uses the center half of the centerline points between the calibration point positions from the first fit. The reasoning behind this is that the center half between the calibration points is not biased due to over- or under-filling. This assumption is however dependent on the severity of filling errors. Figure 3.5 is a visualization of the previously explained piecewise fitting process.

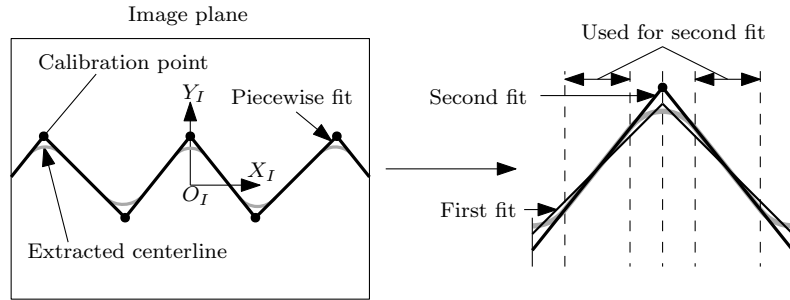


Figure 3.5: Calibration point extraction: Piecewise fitting of line segments to extracted centerline. First fit is used to find approximate location of calibration points. Second fit is conducted using only the center half of data points between each approximated location such that influence of over- and under-filling is reduced.



## 4 Camera calibration

Camera calibration is the process of estimating the parameter vectors  $\mathbf{h}$ ,  $\mathbf{k}$  and  $\mathbf{p}$  in the inverse projection model as described in Equation 2.21 such that when images are acquired from the camera, the estimated model can be used to project image coordinates to real-world coordinates in the laser plane. This chapter explains the developed calibration procedure which estimates the parameters based upon known point correspondences in the artifact and the acquired images of the camera.

The transformation of a point  $\mathbf{x}_a$  in the artifact pattern plane ( $O_A, X_A, Y_A$ ) to a distorted point  $\mathbf{x}_{di}^n$  in the image plane ( $O_I, X_I, Y_I$ ) can be described by combining Equations 2.19 and 3.5 into:

$$\mathbf{D}^{-1}(\mathbf{k}, \mathbf{p}, \mathbf{x}_{di}^n) = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} h_1 & h_4 & 0 \\ h_2 & h_5 & 0 \\ h_3 & h_6 & h_9 \end{pmatrix}}_{\mathbf{M}_h} \underbrace{\begin{pmatrix} 1 & 0 & x_0 + c\Delta z^n \\ 0 & 1 & y_0 + ab\Delta z^n \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_t^n} \underbrace{\begin{pmatrix} a & 0 & 0 \\ bc & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_r} \mathcal{H}[\mathbf{x}_a] \right] \quad (4.1)$$

Assuming that  $N$  images are acquired by the camera while moving at a known constant velocity  $V$ , each image  $n$  is time stamped with a time  $T^n$ , the coordinates of  $K$  calibration points ( $\mathbf{x}_a^1, \mathbf{x}_a^2, \dots, \mathbf{x}_a^K$ ) in the artifact pattern plane ( $O_A, X_A, Y_A$ ) are known and  $K$  corresponding calibration points are acquired in each of the  $N$  images ( $\mathbf{x}_{di}^{11}, \mathbf{x}_{di}^{12}, \dots, \mathbf{x}_{di}^{NK}$ ) using the line and point extraction algorithms as discussed in Chapter 3.

Initially in the calibration procedure, lens distortion is neglected and it is assumed that there is no artifact rotation ( $\mathbf{M}_r = \mathbf{I}$ ) such that:

$$\mathbf{x}_i^{nk} = \mathbf{x}_{di}^{nk}, \quad \mathbf{x}_p^k = \mathbf{x}_a^k \quad \forall (n, k) \in S_1 \quad (4.2)$$

where  $S_1 = \{(n, k) \in \mathbb{Z}^+ \times \mathbb{Z}^+ | n \leq N \wedge k \leq K\}$ ,  $\mathbf{x}_i^{nk}$  is an undistorted calibration point in the image plane ( $O_I, X_I, Y_I$ ) and  $\mathbf{x}_p^k$  is the corresponding calibration point in an artifact plane ( $O_P, X_P, Y_P$ ) parallel to the laser plane ( $O_L, X_L, Y_L$ ). Equation 4.1 can then be reduced to the following relationship for every calibration point correspondence as defined in Equation 4.2 as:

$$\mathbf{x}_i^{nk} = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} h_1 & h_4 & 0 \\ h_2 & h_5 & 0 \\ h_3 & h_6 & h_9 \end{pmatrix}}_{\mathbf{M}_h} \underbrace{\begin{pmatrix} 1 & 0 & x_0 + c\Delta z^n \\ 0 & 1 & y_0 + ab\Delta z^n \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_t^n} \mathcal{H}[\mathbf{x}_p^k] \right] \quad \forall (n, k) \in S_1 \quad (4.3)$$

### 4.1 Multi-image direct linear transform

Matrices  $\mathbf{M}_h$  and  $\mathbf{M}_t^n$  in Equation 4.3 can be combined and simplified into a matrix  $\mathbf{H}^n = \mathbf{M}_h \mathbf{M}_t^n$  which is dependent on image  $n$  as follows:

$$\mathbf{x}_i^{nk} = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} h_1 & h_4 & h_7^n \\ h_2 & h_5 & h_8^n \\ h_3 & h_6 & h_9^n \end{pmatrix}}_{\mathbf{H}^n} \mathcal{H}[\mathbf{x}_p^k] \right] \quad \forall (n, k) \in S_1 \quad (4.4)$$

where

$$h_7^n = h_1(x_0 + c\Delta z^n) + h_4(y_0 + ab\Delta z^n) \quad (4.5)$$

$$h_8^n = h_2(x_0 + c\Delta z^n) + h_5(y_0 + ab\Delta z^n) \quad (4.6)$$

$$h_9^n = h_3(x_0 + c\Delta z^n) + h_6(y_0 + ab\Delta z^n) + h_9 \quad (4.7)$$

Estimation of  $\mathbf{H}^n$  for every image  $n$  in Equation 4.4, is performed using the *Direct linear transform* as described in [10]. Although it is possible to determine the parameters for each image separately, as described in [12] and [29], it is suboptimal in the sense that shared parameters in each image are not exploited in their estimation. Observation of matrix  $\mathbf{H}^n$  shows that parameters  $h_{1-6}$  are independent on  $n$  and therefore the same for every image. Parameters  $h_{7-9}^n$  are however dependent meaning that each image needs its own set of parameters. Rewriting Equation 4.4 leads to the following two non-linear equations for each calibration point correspondence  $\mathbf{x}_p^k = (x_p^k \ y_p^k)^T \rightarrow \mathbf{x}_i^{nk} = (x_i^{nk} \ y_i^{nk})^T$ :

$$x_i^{nk} = \frac{h_1 x_p^k + h_4 y_p^k + h_7^n}{h_3 x_p^k + h_6 y_p^k + h_9^n}, \quad y_i^{nk} = \frac{h_2 x_p^k + h_5 y_p^k + h_8^n}{h_3 x_p^k + h_6 y_p^k + h_9^n} \quad \forall (n, k) \in S_1 \quad (4.8)$$

Although non-linear, the equations are linear in parameters which can be represented in matrix form as:

$$\mathbf{D}^{nk} \mathbf{h}_c + \mathbf{E}^{nk} \mathbf{h}^n = \mathbf{0}_2 \quad \forall \quad (n, k) \in S_1 \quad (4.9)$$

where

$$\mathbf{D}^{nk} = \begin{pmatrix} -x_p^k & 0 & x_i^{nk} x_p^k & -y_p^k & 0 & x_i^{nk} y_p^k \\ 0 & -x_p^k & y_i^{nk} x_p^k & 0 & -y_p^k & y_i^{nk} y_p^k \end{pmatrix}, \quad \mathbf{E}^{nk} = \begin{pmatrix} -1 & 0 & x_i^{nk} \\ 0 & -1 & y_i^{nk} \end{pmatrix}$$

$$\mathbf{0}_2 = (0 \ 0)^T, \quad \mathbf{h}_c = (h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6)^T, \quad \mathbf{h}^n = (h_7^n \ h_8^n \ h_9^n)^T$$

Stacking of the two equations for each point correspondence leads to the following set of equations for each image:

$$\mathbf{B}^n \mathbf{h}_c + \mathbf{C}^n \mathbf{h}^n = \mathbf{0}^n \quad \forall \quad n \in S_2 \quad (4.10)$$

where  $S_2 = \{n \in \mathbb{Z}^+ | n \leq N\}$  and:

$$\mathbf{B}^n = \begin{pmatrix} \mathbf{D}^{n1} \\ \vdots \\ \mathbf{D}^{nK} \end{pmatrix}, \quad \mathbf{C}^n = \begin{pmatrix} \mathbf{E}^{n1} \\ \vdots \\ \mathbf{E}^{nK} \end{pmatrix}, \quad \mathbf{0}^n = \begin{pmatrix} \mathbf{0}_2 \\ \vdots \\ \mathbf{0}_2 \end{pmatrix}$$

with shapes  $\mathbf{B}^n \rightarrow 2K \times 6$ ,  $\mathbf{C}^n \rightarrow 2K \times 3$  and  $\mathbf{0}^n \rightarrow 2N \times 1$ . At last all point correspondence blocks for each image are combined leading to the following system of homogenous linear equations:

$$\mathbf{A} \mathbf{h} = \mathbf{0} \quad (4.11)$$

where the size of matrix  $\mathbf{A}$  is  $2NK \times 3N + 6$  and the number of variables in  $\mathbf{h}$  is  $3N + 6$  with:

$$\mathbf{A} = \left( \begin{array}{c|ccc} \mathbf{B}^1 & \mathbf{C}^1 & & \\ \vdots & & \ddots & \\ \mathbf{B}^N & & & \mathbf{C}^N \end{array} \right), \quad \mathbf{h} = \begin{pmatrix} \mathbf{h}_c \\ \mathbf{h}^1 \\ \vdots \\ \mathbf{h}^N \end{pmatrix}, \quad \mathbf{0} = \begin{pmatrix} \mathbf{0}^1 \\ \vdots \\ \mathbf{0}^N \end{pmatrix} \quad (4.12)$$

The system of homogenous equations 4.11 is solved using the procedures as described in [25] where first the calibration points are normalized. Then the system is solved using the *singular-value-decomposition*(SVD) and at last the obtained result is de-normalized. Normalization is used to lower the condition number of matrix  $\mathbf{A}$  such that the SVD produces more accurate results.

Under the assumption that there are enough images and calibration points in each image, the system  $\mathbf{A}\mathbf{h} = \mathbf{0}$  is *over-determined* meaning that the number of equations is larger than the number of variables or  $2NK > 3N + 6$ . The SVD is then minimizing  $\|\mathbf{A}\mathbf{h}\|^2$  in the least squares sense which is a algebraic error and not directly related to a geometrical error. Ideally, the geometrical error is minimized which requires non-linear optimization. Therefore a final nonlinear optimization step is conducted in Section 4.5 where the obtained values for the model parameters from linear optimization serve as initial conditions.

## 4.2 Artifact slope and angle estimation

The estimated parameters in  $\mathbf{h}$  can now be used to estimate the artifact slope angle  $\beta$  and rotation angle  $\alpha$  as well as the last unknown parameter  $h_9$  in  $\mathbf{M}_h$  using equations 4.8. These equations can be written in matrix form as:

$$\mathbf{L}_1 h_9 + \mathbf{L}_2 \mathbf{b}_2 + \mathbf{L}_3^n \mathbf{b}_3 = \mathbf{h}^n \quad \forall n \in S_2 \quad (4.13)$$

where

$$\begin{aligned} \mathbf{L}_1 &= (0 \ 0 \ 1)^T, \quad \mathbf{L}_2 = \begin{pmatrix} h_1 & h_4 \\ h_2 & h_5 \\ h_3 & h_6 \end{pmatrix}, \quad \mathbf{L}_3^n = \mathbf{L}_2 \Delta z^n \\ \mathbf{b}_2 &= (x_0 \ y_0)^T, \quad \mathbf{b}_3 = (x_1 \ y_1)^T, \quad x_1 = c, \quad y_1 = ab \end{aligned}$$

Combining all terms into a single matrix  $\mathbf{L}$  results in the following set of ordinary linear equations:

$$\mathbf{L}\mathbf{b} = \mathbf{k} \quad (4.14)$$

where

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_1 & \mathbf{L}_2 & \mathbf{L}_3^1 \\ \vdots & \vdots & \vdots \\ \mathbf{L}_1 & \mathbf{L}_2 & \mathbf{L}_3^N \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} h_9 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{pmatrix}, \quad \mathbf{k} = \begin{pmatrix} \mathbf{h}^1 \\ \vdots \\ \mathbf{h}^N \end{pmatrix}$$

which can be solved using the *singular-value-decomposition* as explained in [30]. The estimated parameters  $x_1$  and  $y_1$  in vector  $\mathbf{b}_3$  can, with the use of the definition as stated in Equation 3.4, be written as:

$$\begin{aligned} x_1 &= c = \tan(\alpha) \\ y_1 &= ab = \cos(\alpha)^{-1} \tan(\beta) \end{aligned}$$

An estimate for artifact rotation angle  $\alpha$  can now be determined by:

$$\alpha = \tan^{-1}(x_1) \quad (4.15)$$

which can be used to estimate the artifact slope angle  $\beta$  as:

$$\beta = \tan^{-1}(y_1 \cos(\alpha)) \quad (4.16)$$

### 4.3 Iterative linear parameter estimation

The estimated parameters  $(h_1, h_2, h_3, h_4, h_5, h_6)$  in Section 4.1 and  $(h_9, \alpha, \beta)$  in Section 4.2 are determined based upon the assumption made in Equation 4.2 that  $\mathbf{x}_p^k = \mathbf{x}_a^k$  which is inaccurate. If matrix  $\mathbf{M}_r$  is not ignored the relation in Equation 4.4 would look like:

$$\begin{aligned} \mathbf{x}_i^{nk} &= \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} h_1 & h_4 & 0 \\ h_2 & h_5 & 0 \\ h_3 & h_6 & h_9 \end{pmatrix}}_{\mathbf{M}_h} \underbrace{\begin{pmatrix} 1 & 0 & x_0 + c\Delta z^n \\ 0 & 1 & y_0 + ab\Delta z^n \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_t^n} \underbrace{\begin{pmatrix} a & 0 & 0 \\ bc & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_r} \mathcal{H}[\mathbf{x}_a^k] \right] \\ &= \mathcal{H}^{-1} \left[ \begin{pmatrix} h_1 a + h_4 bc & h_4 & h_7^n \\ h_2 a + h_5 bc & h_5 & h_8^n \\ h_3 a + h_6 bc & h_6 & h_9^n \end{pmatrix} \mathcal{H}[\mathbf{x}_a^k] \right] \\ &= \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} \hat{h}_1 & h_4 & h_7^n \\ \hat{h}_2 & h_5 & h_8^n \\ \hat{h}_3 & h_6 & h_9^n \end{pmatrix}}_{\hat{\mathbf{H}}^n} \mathcal{H}[\mathbf{x}_a^k] \right] \quad \forall \quad (n, k) \in S_1 \end{aligned}$$

Matrix  $\hat{\mathbf{H}}^n$  is unequal to  $\mathbf{H}^n$  in Equation 4.4 due to the following inequalities:

$$\hat{h}_1 = h_1 a + h_4 bc \neq h_1 \quad (4.17)$$

$$\hat{h}_2 = h_2 a + h_5 bc \neq h_2 \quad (4.18)$$

$$\hat{h}_3 = h_3 a + h_6 bc \neq h_3 \quad (4.19)$$

which leads to inaccurate estimation of parameters in Section 4.1 and 4.2. The estimated artifact slope  $\beta$  and rotation angle  $\alpha$  can however be used to provide for a better estimate of the calibration points  $\mathbf{x}_p^k$  with:

$$\mathbf{x}_p^k = \mathcal{H} \left[ \underbrace{\begin{pmatrix} \cos(\alpha)^{-1} & 0 & 0 \\ \tan(\beta) \tan(\alpha) & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}_r} \mathcal{H}[\mathbf{x}_a^k] \right] \quad \forall k \in S_3 \quad (4.20)$$

where the angles are used to form an estimate of matrix  $\mathbf{M}_r$  and  $S_3 = \{k \in \mathbb{Z}^+ | k \leq K\}$ . The estimation steps taken as discussed in Section 4.1 and Section 4.2 can now be repeated using the updated calibration points  $\mathbf{x}_p^k$  such that in an iterative way the parameters are estimated. It is expected that the difference between the estimated and used slope and rotation angles converge such that the expression in Equation 4.4 holds. When converged the used artifact rotation and slope angles  $\alpha, \beta$  and the estimated parameters  $h_1, h_2, h_3, h_4, h_5, h_6$  and  $h_9$  of the last iteration are used to determine lens distortion in the next section and serve as initial conditions for a final non-linear optimization step in Section 4.5.

### 4.4 Lens distortion estimation

The parameters estimated in the previous section can now be used to estimate the parameters  $\mathbf{k} = (k_1 \ k_2)^T$  and  $\mathbf{p} = (p_1 \ p_2)^T$  in the inverse lens distortion model as defined in Equation 2.18 using a linear estimation method equivalent to the method used in [12]. The undistorted coordinates of the calibration points  $\mathbf{x}_i^{nk}$  in the image



plane  $(O_I, X_I, Y_I)$  are calculated using the known calibration points  $\mathbf{x}_a^k$  and the estimated parameters of the previous section using Equation 4.17 such that:

$$\mathbf{x}_i^{nk} = \mathcal{H}[\mathbf{M}_h \cdot \mathbf{M}_t^n \cdot \mathbf{M}_r \mathcal{H}[\mathbf{x}_a^k]] \quad \forall (n, k) \in S_1 \quad (4.21)$$

Known distorted calibration points  $\mathbf{x}_{di}^{nk} = (x_{di}^{nk} \ y_{di}^{nk})^T$  in the image plane and estimated undistorted calibration points  $\mathbf{x}_i^{nk} = (x_i^{nk} \ y_i^{nk})^T$  can now be used to estimate the parameters in the inverse distortion model  $\mathbf{x}_i^{nk} = \mathbf{D}^{-1}(\mathbf{k}, \mathbf{p}, \mathbf{x}_{di}^{nk})$ . Using the expressions in Equation 2.13 the following relations are obtained for the inverse distortion model:

$$\begin{aligned} x_i^{nk} - x_{di}^{nk} &= (x_{di}^{nk} r_{di}^{nk^2})k_1 + (x_{di}^{nk} r_{di}^{nk^4})k_2 + (2x_{di}^{nk} y_{di}^{nk})p_1 + (r_{di}^{nk^2} + 2x_{di}^{nk^2})p_2 \\ y_i^{nk} - y_{di}^{nk} &= (y_{di}^{nk} r_{di}^{nk^2})k_1 + (y_{di}^{nk} r_{di}^{nk^4})k_2 + (r_{di}^{nk^2} + 2y_{di}^{nk^2})p_1 + (2x_{di}^{nk} y_{di}^{nk})p_2 \end{aligned}$$

where decentering distortion is left out and only two parameters each are used to compensate for radial and tangential distortion as discussed in Section 2.5. These equations are linear in the parameters  $k_1, k_2, p_1, p_2$  and can be written in matrix form for every point correspondence as:

$$\mathbf{F}^{nk} \mathbf{k} + \mathbf{G}^{nk} \mathbf{p} = \mathbf{d}^{nk} \quad \forall (n, k) \in S_1 \quad (4.22)$$

where

$$\begin{aligned} \mathbf{F}^{nk} &= \begin{pmatrix} x_{di}^{nk} r_{di}^{nk^2} & x_{di}^{nk} r_{di}^{nk^4} \\ y_{di}^{nk} r_{di}^{nk^2} & y_{di}^{nk} r_{di}^{nk^4} \end{pmatrix}, \quad \mathbf{G}^{nk} = \begin{pmatrix} 2x_{di}^{nk} y_{di}^{nk} & r_{di}^{nk^2} + 2x_{di}^{nk^2} \\ r_{di}^{nk^2} + 2y_{di}^{nk^2} & 2x_{di}^{nk} y_{di}^{nk} \end{pmatrix} \\ \mathbf{d}^{nk} &= \mathbf{x}_i^{nk} - \mathbf{x}_{di}^{nk}, \quad \mathbf{k} = (k_1 \ k_2)^T, \quad \mathbf{p} = (p_1 \ p_2)^T \end{aligned}$$

Stacking all point correspondences leads to the following system of ordinary linear equations:

$$\mathbf{Q} \mathbf{v} = \mathbf{d} \quad (4.23)$$

where

$$\mathbf{Q} = \begin{pmatrix} \mathbf{F}^{11} & \mathbf{G}^{11} \\ \vdots & \vdots \\ \mathbf{F}^{NK} & \mathbf{G}^{NK} \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} \mathbf{k} \\ \mathbf{p} \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} \mathbf{d}^{11} \\ \vdots \\ \mathbf{d}^{NK} \end{pmatrix}$$

which is solved using the *singular-value-decomposition* as described in [30] such that the parameters in  $\mathbf{k}$  and  $\mathbf{p}$  are estimated.

## 4.5 Nonlinear refinement

After linear estimation of the parameters  $\mathbf{h}$  in Section 4.1,  $\alpha, \beta$  in Section 4.2 and  $\mathbf{k}, \mathbf{p}$  in Section 4.4 a final non-linear optimization step is used to minimize the geometrical projection error in the laser plane using all parameters where the estimated parameters through linear methods are used as initial conditions.

The (back-)projection of the distorted calibration points  $\mathbf{x}_{di}^{nk}$  onto the image plane  $(O_I, X_I, Y_I)$  is represented by points  $\overleftarrow{\mathbf{x}}_l^{nk}$  and can, as defined in Equation 2.21, be written as:

$$\overleftarrow{\mathbf{x}}_l^{nk} = ILP(\mathbf{h}, \mathbf{k}, \mathbf{p}, \mathbf{x}_{di}^{nk}) \quad \forall (n, k) \in S_1 \quad (4.24)$$

The (forward-) projection of the calibration points  $\mathbf{x}_a^k$  in the artifact pattern plane  $(O_A, X_A, Y_A)$  onto the laser plane is represented by points  $\overrightarrow{\mathbf{x}}_l^{nk}$  and can, as defined in Equation 3.6, be written as:

$$\overrightarrow{\mathbf{x}}_l^{nk} = ALP(\alpha, \beta, V, T_1, T^n, \mathbf{x}_a^k) \quad \forall (n, k) \in S_1 \quad (4.25)$$

The goal is to *minimize* the projection error  $E$  which is defined as:

$$E(\mathbf{h}, \mathbf{k}, \mathbf{p}, \alpha, \beta) = \sum_{n=1}^N \sum_{p=1}^P \|\vec{\mathbf{x}}_l^{np} - \vec{\mathbf{x}}_l^{np}\|^2 \quad (4.26)$$

This is a non-linear least-squares minimization problem which is solved using the *Levenberg-Marquardt algorithm* as described in [25]. The algorithm essentially estimates the orientation of the artifact and the projection of points in the image plane to the laser plane simultaneously. The obtained parameters  $\alpha$  and  $\beta$  only provide knowledge about the orientation of artifact while the obtained parameters vectors  $\mathbf{h}$ ,  $\mathbf{k}$  and  $\mathbf{p}$  can be used to project image plane coordinates to laser plane coordinates as follows:

For each column  $C$  in each acquired image  $I$ :

1. Obtain distorted image plane point  $\mathbf{x}_{di}$  using the column-wise adaptive centerline extraction (CACE) algorithm as described in Section 3.2.2

$$\mathbf{x}_{di} = \text{CACE}(C) \quad (4.27)$$

2. Undistort the distorted point using the inverse  $\mathbf{D}^{-1}$  of distortion model  $\mathbf{D}$  as described in Equation 2.18 using the obtained parameters  $\mathbf{k}$  and  $\mathbf{p}$

$$\mathbf{x}_i = \mathbf{D}^{-1}(\mathbf{k}, \mathbf{p}, \mathbf{x}_{di}) \quad (4.28)$$

3. Project the undistorted image points to the laser plane using matrix  $\mathbf{M}_h$  which can be constructed from parameter vector  $\mathbf{h}$  such that coordinates in the laser plane are obtained by:

$$\mathbf{x}_l = \mathcal{H}^{-1} \left[ \underbrace{\begin{pmatrix} h_1 & h_4 & 0 \\ h_2 & h_5 & 0 \\ h_3 & h_6 & h_9 \end{pmatrix}^{-1}}_{\mathbf{M}_h^{-1}} \mathcal{H}[\mathbf{x}_i] \right] \quad (4.29)$$

## 5 Results and Discussion

This chapter presents calibration results for simulated as well as experimentally obtained data. The first section describes how calibration and measurement accuracy is determined. The second section describes the simulation results and the last section describes the experimental results.

### 5.1 Accuracy and precision evaluation

This section describes how calibration and measurement error in simulation as well as the experiment is evaluated. It describes how the training error obtained in the calibration process can be used as an upper bound on the actual or real error in measurement operation.

#### 5.1.1 Disturbed projection

In reality the artifact will have a surface roughness and is therefore not equal to the artifact pattern that is used for calibration. Detection of the calibration points with the centerline and point extraction algorithms, as discussed in Section 3, is prone to error as well. These two sources of noise are modelled as being normally distributed and independent of each other. The projection of an ideal point  $\mathbf{x}_{ideal}$  in an artifact cross section, which is used for calibration, to a point  $\mathbf{x}_{di_{real}}$  in the image plane is now redefined for the presence of noise. The ideal observed points  $\mathbf{x}_{li_{ideal}}$  in the laser plane are, as described by Equation 3.5, obtained through:

$$\mathbf{x}_{li_{ideal}}^{nk} = \mathcal{H}^{-1}[\mathbf{M}_t^n \cdot \mathbf{M}_r \cdot \mathcal{H}[\mathbf{x}_{a_{ideal}}^{nk}]] \quad \forall (n, k) \in S_1 \quad (5.1)$$

Due to surface roughness of the artifact the real observed points  $\mathbf{x}_{l_{real}}$  are defined as:

$$\mathbf{x}_{l_{real}}^{nk} = \mathbf{x}_{li_{ideal}}^{nk} + \underbrace{\begin{pmatrix} \mathcal{N}(0, \sigma_{a_x}) \\ \mathcal{N}(0, \sigma_{a_y}) \end{pmatrix}}_{\mathbf{N}_a} \quad \forall (n, k) \in S_1 \quad (5.2)$$

where noise vector  $\mathbf{N}_a$  with standard deviations of  $\sigma_{a_x}$  and  $\sigma_{a_y}$  represent the surface roughness of the artifact in two dimensions. The ideal observed points  $\mathbf{x}_{di_{ideal}}$  in the images acquired by the camera are, with the use of the result obtained in Equation 2.15, defined as:

$$\mathbf{x}_{di_{ideal}}^{nk} = \mathcal{H}^{-1}[\mathbf{M}_i \cdot \mathcal{H}[\mathbf{D}(\mathcal{H}^{-1}[\mathbf{M}_f \cdot \mathbf{M}_e \cdot \mathcal{H}[\mathbf{x}_{l_{real}}^{nk}]])] \quad \forall (n, k) \in S_1 \quad (5.3)$$

Due to observation error, introduced by the centerline and point extraction algorithm, the real observed points  $\mathbf{x}_{di_{real}}$  are modelled as:

$$\mathbf{x}_{di_{real}}^{nk} = \mathbf{x}_{di_{ideal}}^{nk} + \underbrace{\begin{pmatrix} \mathcal{N}(0, \sigma_{d_x}) \\ \mathcal{N}(0, \sigma_{d_y}) \end{pmatrix}}_{\mathbf{N}_d} \quad \forall (n, k) \in S_1 \quad (5.4)$$

where noise vector  $\mathbf{N}_d$  with standard deviations  $\sigma_{d_x}$  and  $\sigma_{d_y}$  represent the observation error in both dimensions.

### 5.1.2 Error definition

After calibration, the obtained model is supposed to be used for measurement operation as discussed in Section 4.5. The estimated coordinates in the laser plane are, with the use of the estimated model, obtained by:

$$\mathbf{x}_{i_{est}}^{nk} = \mathbf{D}_{est}^{-1}(\mathbf{x}_{di_{real}}^{nk}) \quad \forall(n, k) \in S_1 \quad (5.5)$$

$$\mathbf{x}_{l_{est}}^{nk} = \mathcal{H}^{-1}[\mathbf{M}_{h_{est}}^{-1} \cdot \mathcal{H}[\mathbf{x}_{i_{est}}^{nk}]] \quad \forall(n, k) \in S_1 \quad (5.6)$$

where (without mentioning the parameters)  $\mathbf{D}_{est}^{-1}$  and  $\mathbf{M}_{h_{est}}$  describe the estimated model. Training error  $\mathbf{e}_{train}^{nk}$  is defined as the difference between the estimated coordinates and the ideal (known) coordinates. This error is essentially minimized by the non-linear optimization step in the calibration method as described in Section 4.5 and is defined as:

$$\mathbf{e}_{train}^{nk} = \mathbf{x}_{l_{ideal}}^{nk} - \mathbf{x}_{l_{est}}^{nk} \quad \forall(n, k) \in S_1 \quad (5.7)$$

Real error  $\mathbf{e}_{real}^{nk}$  is defined as the difference between the estimated coordinates and the real (unknown in experiment, known in simulation) coordinates and is defined as:

$$\mathbf{e}_{real}^{nk} = \mathbf{x}_{l_{real}}^{nk} - \mathbf{x}_{l_{est}}^{nk} \quad \forall(n, k) \in S_1 \quad (5.8)$$

Assuming that lens distortion is negligible, so  $\mathbf{x}_i = \mathbf{D}(\mathbf{x}_{di}) \approx \mathbf{x}_{di}$ , and that the estimated projection matrix has approximately zero error,  $\mathbf{M}_{h_{est}} \approx \mathbf{M}_{h_{real}}$ , the distribution of the training as well as the real error will approximately follow a zero mean normal distribution. These errors serve as a bound on the "best" achievable errors.

*Accuracy* is defined as the mean error which, in this case, is zero for both the training ( $\mathbf{e}_{train}^{best}$ ) and real ( $\mathbf{e}_{real}^{best}$ ) error. Therefore if modelled perfectly the accuracy will be 100%. *Precision* is the variability of errors defined by the standard deviations of the distributions of the training and real errors. Perfect modelling does not mean the errors will have a zero standard deviation since the training and real errors are:

$$\mathbf{e}_{train}^{best} \approx \underbrace{\begin{pmatrix} \mathcal{N}(0, \sigma_{train_x}^{best}) \\ \mathcal{N}(0, \sigma_{train_y}^{best}) \end{pmatrix}}_{\mathbf{N}_{train}^{best}}, \quad \mathbf{e}_{real}^{best} \approx \underbrace{\begin{pmatrix} \mathcal{N}(0, \sigma_{real_x}^{best}) \\ \mathcal{N}(0, \sigma_{real_y}^{best}) \end{pmatrix}}_{\mathbf{N}_{real}^{best}} \quad (5.9)$$

which is derived from the previous results. The errors are normal distributed as well since all commutative and associative properties for normal variables are also valid for random variables. Standard deviations  $\sigma^{best}$  represent the error in their corresponding direction of the laser plane and form a lower bound on the achievable precision if modelled perfectly.

In reality the estimated model will deviate from the real underlying model. Therefore the previous result does not hold. It is assumed that lens distortion is also negligible in reality. The estimated projection matrix however is prone to error, such that  $\mathbf{M}_{h_{est}} \neq \mathbf{M}_{h_{real}}$ . Therefore bias and added variability is introduced into the training and real errors. It is expected that the estimated models lead to normally distributed training ( $\mathbf{e}_{train}^{est}$ ) and real ( $\mathbf{e}_{real}^{est}$ ) errors, with means  $\mu^{est}$  representing the accuracy. in the following form:

$$\mathbf{e}_{train}^{est} \approx \underbrace{\begin{pmatrix} \mathcal{N}(\mu_{train_x}^{est}, \sigma_{train_x}^{est}) \\ \mathcal{N}(\mu_{train_y}^{est}, \sigma_{train_y}^{est}) \end{pmatrix}}_{\mathbf{N}_{train}^{est}}, \quad \mathbf{e}_{real}^{est} \approx \underbrace{\begin{pmatrix} \mathcal{N}(\mu_{real_x}^{est}, \sigma_{real_x}^{est}) \\ \mathcal{N}(\mu_{real_y}^{est}, \sigma_{real_y}^{est}) \end{pmatrix}}_{\mathbf{N}_{real}^{est}} \quad (5.10)$$

### 5.1.3 Accuracy estimation

The relation between training error and real error is investigated for four different cases where it is assumed that the estimated model is exact or  $\mathbf{M}_{hest} = \mathbf{M}_{hreal}$ . The cases are enumerated as follows:

1.  $\mathbf{N}_a = \mathbf{0}, \mathbf{N}_d = \mathbf{0}$ : No noise present in the system such that the training error and real error are zero. Therefore  $\mathbf{x}_{di_{real}} = \mathbf{x}_{di_{ideal}}$  and  $\mathbf{x}_{l_{real}} = \mathbf{x}_{l_{ideal}}$ . This then results in:

$$\mathbf{x}_{l_{ideal}} = \mathbf{x}_{l_{est}} \quad (5.11)$$

$$\mathbf{x}_{l_{real}} = \mathbf{x}_{l_{est}} \quad (5.12)$$

which means that:

$$\sigma_{real}^{best} = \sigma_{train}^{best} = \mathbf{0} \quad (5.13)$$

2.  $\mathbf{N}_a \neq \mathbf{0}, \mathbf{N}_d = \mathbf{0}$ : The centerline and point extraction algorithms detect with zero error but the artifact has a surface roughness such that  $\mathbf{x}_{di_{real}} = \mathbf{x}_{di_{ideal}}$  and  $\mathbf{x}_{l_{real}} \neq \mathbf{x}_{l_{ideal}}$ . This then results in:

$$\mathbf{x}_{l_{ideal}} \neq \mathbf{x}_{l_{est}} \rightarrow \mathbf{e}_{train} = \mathbf{N}_{train} = \mathbf{N}_a \quad (5.14)$$

$$\mathbf{x}_{l_{real}} = \mathbf{x}_{l_{est}} \rightarrow \mathbf{e}_{real} = \mathbf{0} \quad (5.15)$$

which means that:

$$\sigma_{train}^{best} > \sigma_{real}^{best} = \mathbf{0} \quad (5.16)$$

3.  $\mathbf{N}_a = \mathbf{0}, \mathbf{N}_d \neq \mathbf{0}$ : The artifact is perfectly smooth but there is a detection error. The pattern used for calibration is an exact representation of the reality. Therefore  $\mathbf{x}_{di_{real}} \neq \mathbf{x}_{di_{ideal}}$  and  $\mathbf{x}_{l_{real}} = \mathbf{x}_{l_{ideal}}$  which results in:

$$\mathbf{x}_{l_{ideal}} \neq \mathbf{x}_{l_{est}} \rightarrow \mathbf{e}_{train}^{best} = \mathbf{N}_{train}^{best} = \mathbf{N}_{real}^{best} \quad (5.17)$$

$$\mathbf{x}_{l_{real}} \neq \mathbf{x}_{l_{est}} \rightarrow \mathbf{e}_{real}^{best} = \mathbf{N}_{train}^{best} = \mathbf{N}_{real}^{best} \quad (5.18)$$

which means that:

$$\sigma_{train}^{best} = \sigma_{real}^{best} \neq \mathbf{0} \quad (5.19)$$

4.  $\mathbf{N}_a \neq \mathbf{0}, \mathbf{N}_d \neq \mathbf{0}$ : In this case both the sources of disturbance are present. Therefore it can be derived that:

$$\mathbf{x}_{l_{ideal}} \neq \mathbf{x}_{l_{est}} \rightarrow \mathbf{e}_{train}^{best} = \mathbf{N}_{train}^{best} \quad (5.20)$$

$$\mathbf{x}_{l_{real}} \neq \mathbf{x}_{l_{est}} \rightarrow \mathbf{e}_{real}^{best} = \mathbf{N}_{real}^{best} \quad (5.21)$$

$\mathbf{N}_{train}^{best}$  is a combination of  $\mathbf{N}_a$  and  $\mathbf{N}_d$  while  $\mathbf{N}_{real}^{best}$  is only affected by  $\mathbf{N}_d$  since the real coordinates in the laser plane are known. Therefore the training error is always bigger than the real error.

$$\sigma_{train}^{best} > \sigma_{real}^{best} \neq \mathbf{0} \quad (5.22)$$

From the previous cases it can be concluded that the training error precision is always bigger than or equal to the real error precision if the estimated model is exact and no lens distortion is present. This result means that for the standard deviations in  $\mathbf{N}_{train}^{best}$  and  $\mathbf{N}_{real}^{best}$  the following should hold:

$$\sigma_{train_x}^{best} \geq \sigma_{real_x}^{best} \quad \wedge \quad \sigma_{train_y}^{best} \geq \sigma_{real_y}^{best} \quad (5.23)$$

This result can be used to determine measurement accuracy. By confirming that the training error is normally distributed through normality tests a conservative confidence interval for measurement (real) error can be formed based upon the training error. Conservative in the sense that the real error is only equal to the training error if there is no noise present (case 1) or only noise due to detection error (case 3) which is highly unlikely since the artifact will always have a surface roughness and there will always be a detection error.

## 5.2 Simulation

This section describes the results obtained through simulation. First a vision system model is defined which is based upon the actual experimental setup as described in Section 5.3. Then the measurement accuracy is evaluated in the case that the estimated model is exact which proves (through simulation) that the training error is always bigger than or equal to the real error as proposed by the previous section. The third subsection describes the influence of the amount of data used for calibration on the estimated model accuracy. It also simulates the effect of the spread of calibration points across the field of view of the camera due to the use of an artifact with and without a slope and rotated angular orientation. The last subsection evaluates the measurement accuracy of the estimated model by the calibration algorithm. It also shows that the training error is bigger than or equal to the real error.

### 5.2.1 Vision system model

Based upon the experimental setup a simulation model is developed. The camera used for the experiment generates images of pixels. Each image has a height  $h_i = 960[\text{pixels}]$  and width  $w_i = 1280[\text{pixels}]$ . The sensor of the camera is defined to have height  $h_s = 3.6[\text{mm}]$  and width  $w_s = 4.8[\text{mm}]$  in reality. Scaling parameter  $s$  in intrinsic matrix  $\mathbf{M}_i$  as described in Equation 2.16 can now be determined by:

$$s = w_i/w_s = h_i/h_s \approx 267 \quad (5.24)$$

The width in reality, of the field of view of the camera, is approximately measured to be  $w_r = 6.7[\text{mm}]$ . The working distance, the distance between the camera sensor and the center of the laser plane, is determined to be  $d = 50[\text{mm}]$ . The focal length  $f$  and translation  $t_z = d + f$  can now be determined using the pinhole camera model as depicted in Figure 5.1. From triangle similarity follows:

$$\frac{w_r/2}{f + d} = \frac{w_s/2}{f} \rightarrow f = \frac{w_s}{w_r - w_s}d \quad (5.25)$$

such that:

$$f \approx 127, \quad t_z \approx 177 \quad (5.26)$$

The camera in the experimental setup is positioned at an approximate angle of  $\theta_x = -45$  degrees relative to the laser plane. No rotation around other axis is present such that  $\theta_y = \theta_z = 0$ . With this knowledge the projection matrix for simulation  $\mathbf{M}_{h_{sim}}$  is fully determined and calculated as:

$$\mathbf{M}_{h_{sim}} \approx \begin{pmatrix} 190.572 & 0 & 0 \\ 0 & 134.755 & 0 \\ 0 & -0.004 & 0 \end{pmatrix} \quad (5.27)$$

Forward lens distortion model( $\mathbf{D}$ ) is unknown and therefore omitted in simulations. Furthermore, it is expected that lens distortion is relatively small and negligible.

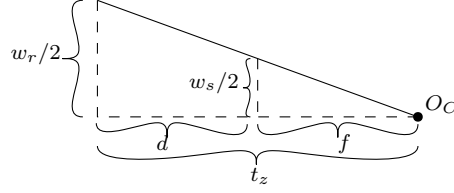


Figure 5.1: Finding focal length: Based upon a given width in reality  $w_r$ , the width of the camera sensor in reality  $w_s$  and the working distance  $d$  the focal length  $f$  and translation  $t_z = d + f$  can be determinant using triangle similarity.

### 5.2.2 Known model precision evaluation

In Sub-section 5.1.3 it is proposed that the variability of the training errors during calibration is always bigger than or equal to the real error or  $\sigma_{train}^{best} \geq \sigma_{real}^{best}$ . To confirm this statement a simulation is performed. Disturbed projection and error estimation as described in Sub-sections 5.1.1 and 5.1.2 is performed using simulation projection matrix  $\mathbf{M}_{h_{sim}}$  without lens distortion. Calibration point correspondences are generated for each pixel location in the image and their correspondence in the laser plane. Artifact surface roughness and detection noise is simulated for a range of standard deviations which are equal in the  $x$  and  $y$  direction. The training and real error in the form of their standard deviations are calculated for each noise combination. The results are shown in Figure 5.2

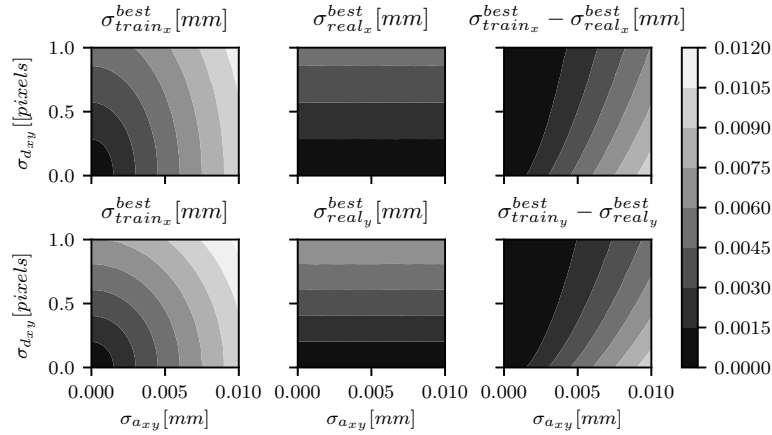


Figure 5.2: Known model errors: The standard deviations  $\sigma_{train}^{best}$  of the training errors are depicted on the left for a range of artifact surface roughness ( $\sigma_{a_{xy}}$ ) and detection noise ( $\sigma_{d_{xy}}$ ) standard deviations. The middle subplot shows the same results for the real error  $\sigma_{real}^{best}$  and on the right the difference is shown. Observation shows that the training error variability is always bigger than or equal to the real error.

Since the estimated model is equal to the real underlying model the training error and the real error in the figure show the best achievable error. The proposed cases as defined in Sub-section 5.1.3 are evaluated as follows.

Observation shows that, Case 1:  $\sigma_{a_{xy}} = \sigma_{d_{xy}} = 0$  (no noise) holds since  $\sigma_{train}^{best} = \sigma_{real}^{best} = 0$ , Case 2:  $\sigma_{a_{xy}} \neq 0 \wedge \sigma_{d_{xy}} = 0$  (no detection noise) also holds since  $\sigma_{train}^{best} = \sigma_{a_{xy}} \wedge \sigma_{real}^{best} = 0$ , Case 3:  $\sigma_{a_{xy}} = 0 \wedge \sigma_{d_{xy}} \neq 0$  (no surface roughness) holds since  $\sigma_{train}^{best} = \sigma_{real}^{best} \neq 0$  and Case 4:  $\sigma_{a_{xy}} \neq 0 \wedge \sigma_{d_{xy}} \neq 0$  (both types of noise) holds since  $\sigma_{train}^{best} > \sigma_{real}^{best} \neq 0$ . Therefore it is concluded that  $\sigma_{train}^{best} \geq \sigma_{real}^{best}$  if an exact model estimate is obtained and no lens distortion is present in the system.

### 5.2.3 Model estimation accuracy versus data-set size and spread

This subsection studies the effect of the training data size on the estimated model accuracy for an artifact with and without a slope. Figure 5.3 shows the mean absolute error in the laser plane for an increasing number of images used for calibration. Five calibration point correspondences are used for each image.

For the artifact with a slope, the spread of calibration points at the maximum number of images simulated(289) covers a large part of the field-of-view of the camera. At the minimum the calibration points are located near the center of the image. For the artifact without a slope all calibration points are located near the center of the image, for a small as well as a large number of images used for calibration, since it does not have slope. Extreme noise values are used for simulation at  $\sigma_{a_x} = \sigma_{a_y} = 0.01$  and  $\sigma_{d_x} = \sigma_{d_y} = 1$ . These standard deviations are equal to the upper right corners of the Sub-plots in Figure 5.2.

Figure 5.3 shows that the artifact with a slope leads to a more accurate estimated model compared to an artifact without a slope if a large number of images is used  $\geq 200$ . This result therefore proves the hypothesis made in Chapter 1 that an artifact with a slope leads to more accurate calibration due to the spread of calibration points across the field-of-view. Therefore, to achieve accurate calibration, at least 200 images should be used for the experiment. The curves in Figure 5.3 are noisy which is caused by sampling errors in simulation. A larger number of trails can be used to get a more realistic result of the estimated accuracy. This is however computationally expensive. The overall trend will however not change, the artifact with a slope achieves more accurate calibration than the artifact without a slope.

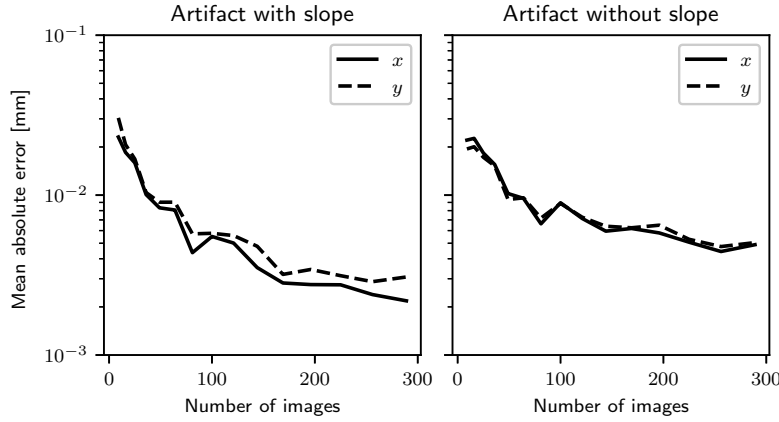


Figure 5.3: Artifact design: A artifact with a slope leads to more accurate calibration compared to an artifact without a slope for a large number of images. The spread of the calibration points across the field-of-view of the camera due to the a sloped artifact is considered the leading cause for higher accuracy.

### 5.2.4 Estimated model accuracy evaluation

Using 200 images of 5 calibration points each, the model accuracy ( $\mu$ ) and precision ( $\sigma$ ) in terms of training and real error is now evaluated for the estimated model at different levels of noise. This is analogous to Sub-section 5.2.2 except that in this case the estimated model is used instead of the known model. Figure 5.4 shows the results for the precisions  $\sigma_{real}^{est}$  and  $\sigma_{train}^{est}$  in the real and training errors respectively.



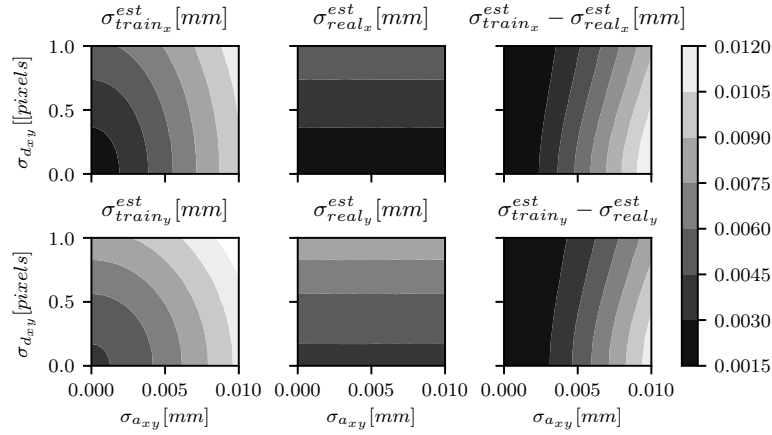


Figure 5.4: Estimated model precision. 200 images were simulated and 5 calibration points are extracted in each image: See Figure 5.2 for explanation. Observation shows that the training error precision is always bigger than or equal to the real error precision

From observation of the figure it can therefore be concluded that the training error is always bigger than or equal to the real error precision, so  $\sigma_{train}^{est} \geq \sigma_{real}^{est}$ , just as in the case of a known model. If the subplots are compared to the subplots of the known model close observations shows that for the estimated model the errors are slightly higher than those for the known model. This makes sense since the errors in the known model are only obtained if the estimated model is exactly equal to the real underlying model.

The estimated model will introduce inaccuracy as proposed by Equation 5.10. The mean errors are therefore calculated as well for the estimated model. The results are shown in Figure 5.5. The inaccuracy is around 0.0003[mm] which is very small compared to the precision as shown in Figure 5.4.

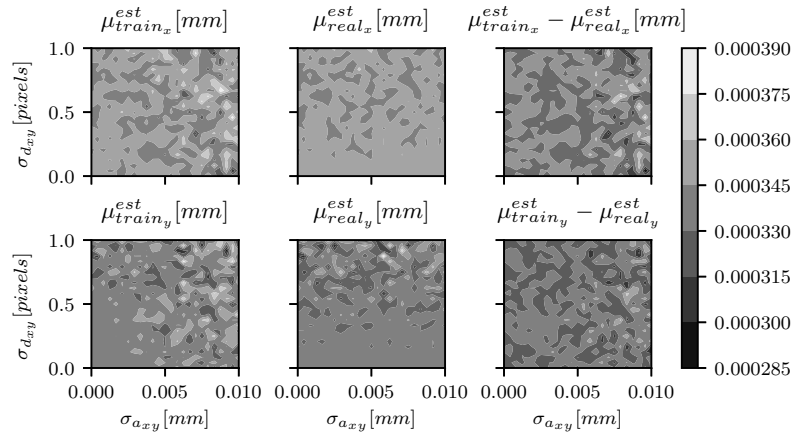


Figure 5.5: Estimated model accuracy. The inaccuracy shows to be around 0.0003[mm] which is very small compared to the precision as shown in Figure 5.4.

### 5.3 Experiment

This section describes the results for a real-world experiment. A prediction is made on the measurement error of the calibrated vision system based upon the distribution of the training error.

#### 5.3.1 Data acquisition

The laser-triangulation based vision system used for the experiment consist out of a camera and a line laser module. The camera used for this experiment is a Dinolite AM7915MZTL which generates images at approximately 30 frames per second. The images are approximately one megapixel in size and the shape of each image is  $960 \times 1280[\text{pixels}]$ . Each image is time-stamped  $(T^1, \dots, T^N)$  and the scan velocity  $V$  is  $5[\text{mm/s}]$ . The line laser module is a Laser-Compnents FP-MV12-450-50D-45-F. The line laser is placed vertical such that the laser plane is vertical. The camera is placed at an angle of approximately  $-45$  degrees relative to the laser plane.

The artifact is produced by a Stratasys J750 additive manufacturing machine. The material used is ABS and a glossy finish is added for additional smoothness of the artifact surface. The artifact pattern is of a triangular shape where the corners are  $0.7[\text{mm}]$  apart from eachother. Each corner has an angle of  $45$  degrees. The artifact has a length of  $50[\text{mm}]$  and the pattern changes  $2[\text{mm}]$  in height between the ends of the artifact. The slope angle is therefore equal to  $\tan^{-1}(2/50) \cdot \pi \cdot 180 \approx 2.29$  degress. 200 images were collected and 5 calibration feature points are extracted from each image. Therefore leading to a total number of a 1000 calibration points correspondes.

#### 5.3.2 Results

The estimated projection model  $\mathbf{M}_{h_{est}}$  is:

$$\mathbf{M}_{h_{est}} \approx \begin{pmatrix} 190.923 & -1.216 & 0 \\ 0.077 & 135.448 & 0 \\ -0.001 & -0.015 & 1 \end{pmatrix} \quad (5.28)$$

which is indeed very close to the simulated model as described in Equation 5.27. The estimated lens distortion parameters for radial and tangential distortion are:

$$k_1 \approx 3.5 \cdot 10^{-6}, \quad k_2 \approx -4.4 \cdot 10^{-11}, \quad p_1 \approx 6.1 \cdot 10^{-4}, \quad p_2 \approx 6.9 \cdot 10^{-6} \quad (5.29)$$

which are very small values indicating that lens distortion is neglegible. Since lens distortion is small the distribution of the training error will be a good indicator for accuracy and precision as discussed in Sub-section 5.1.3. The slope and rotation angle of the artifact are determined to be:

$$\alpha \approx 0.002[\text{degrees}], \quad \beta \approx -2.194[\text{degrees}] \quad (5.30)$$

The negative slope is correct since the artifact was scanned by the vision system in slope decreasing direction. It also shows that the estimated slope angle has an error of approximately  $2.29 - 2.19 = 0.1$  degrees. The small value of the artifact rotation angle makes sense since the artifact was aligned as straight as possible.

The results for the training error  $\mathbf{e}_{train}$  are shown in Figure 5.6. The errors in both the  $x$  and  $y$  direction are tested for normality using the *Shapiro-Wilk test*. The p-values obtained from the test  $x \rightarrow p \approx 0.15$  and  $y \rightarrow p \approx 0.25$  are higher then the alpha-level( $\alpha = 0.05$ ). Therefore it is likely that both samples are drawn from a normal distribution.

The standard deviations of the training errors are  $\sigma_{train_x} \approx 0.004$  for the  $x$  direction and  $\sigma_{train_y} \approx 0.0045$  for the  $y$  direction. The means of the training errors are  $\mu_{train_x} \approx 0.0002$  and  $\mu_{train_y} \approx 0.0001$  for the  $x$  and  $y$  direction respectively.

Since the errors are normal, a 95 percent confidence interval can now be found for both directions resulting in:

$$e_{train_x} = 0.0002 \pm 0.008[mm], \quad e_{train_y} = 0.0001 \pm 0.009[mm] \quad (5.31)$$

These bounds represent a conservative estimate of the real error  $e_{real}$ . Conservative because the real error is more likely to be smaller than the training error as proven in Section 5.2.2.

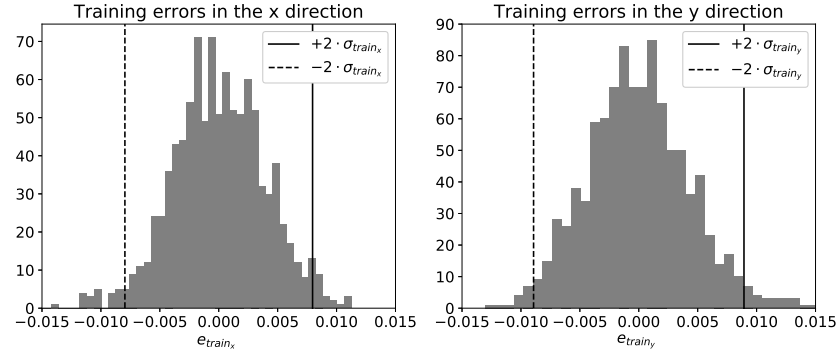


Figure 5.6: Experiment training error showing a predicted measurement accuracy of the calibrated vision system of  $\{-2 \cdot \sigma_{train_x} = -0.008 < e_{train_x} < 0.008 = +2 \cdot \sigma_{train_x}\}$  and  $\{-2 \cdot \sigma_{train_y} = -0.009 < e_{train_y} < 0.009 = +2 \cdot \sigma_{train_y}\}$ .



## 6 Conclusions and future work

This research provides a calibration method for rigid sloped artifacts fabricated through additive manufacturing. Knowledge of the inclination and angular orientation of the artifact is not necessary for calibration. Only the relative coordinates of the cross section pattern in the artifact are needed.

The calibration method first estimates the artifact slope and rotation angle in an iterative way using linear optimization where all calibration points are used simultaneously. After convergence of the angles, lens distortion is approximated using linear methods as well. The resulting projection matrix, distortion parameters and artifact angles are used as initial conditions for a final non-linear refinement step which estimates all model parameters at the same time.

Most calibration methods first estimate the projection matrix using linear methods. The approach in this research is however different in the sense that instead of estimating a projection matrix for each image separately, it is estimated using all calibration points in every image at the same time. The iterative way of estimating the artifact rotation and slope is also a contribution of this research.

Feature point correspondences for calibration are extracted by a newly developed adaptive laser centerline and feature point extraction algorithm. These algorithms do not rely on any fixed parameter. Therefore the algorithm is adaptive to changes in material reflectiveness.

Simulation shows that a sloped artifact leads to more accurate calibration than an artifact without a slope. It is also proven through simulation that the distribution of training errors forms an upper bound on the actual errors in reality. Therefore training errors can be used to quantify measurement accuracy and precision.

An experiment is conducted where a sloped artifact is printed using a Stratasys J750. The calibration result shows that with a 95% confidence the real world measurement errors fall between  $0.0002 \pm 0.008$  and  $0.0001 \pm 0.009$  millimeters in the  $x$  and  $y$  direction respectively for an approximate field-of-view of 7 millimeters.

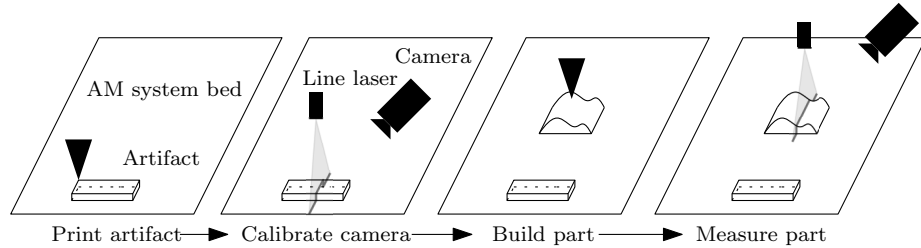


Figure 6.1: Proposed use of the calibration method. The calibration artifact is printed by the AM machine first. Then the vision system is used to scan the artifact and calibrate itself. At last the calibrated vision system is used to scan the parts surface.

The recommended calibration and part surface scanning procedure is depicted in Figure 6.1. Using a CAD model of the artifact, the artifact is printed by the AM machine first. A different angular position of the artifact can be chosen if the camera is set for a larger field-of-view to increase calibration accuracy. Then the artifact is scanned by the vision system and calibrated using the calibration method proposed in this research. Knowledge of the artifact slope and orientation angle are not needed which simplifies the procedure. At last the part surface can be measured using the calibrated vision system.

Future work could involve implementation of the vision system to develop an automated method of steady-state empirical system response modelling as proposed in Chapter 1. Before every build, the vision system is calibrated by the aforementioned procedure first, then for several settings of the process parameters, lines are deposited. At last the geometry of these lines is measured using the calibrated vision system. The data obtained can now be used to estimate a response surface which relates process parameter value with deposition geometry output. This knowledge could then be used to optimize the build process to reduce over- and under-filling at off-nominal system states.

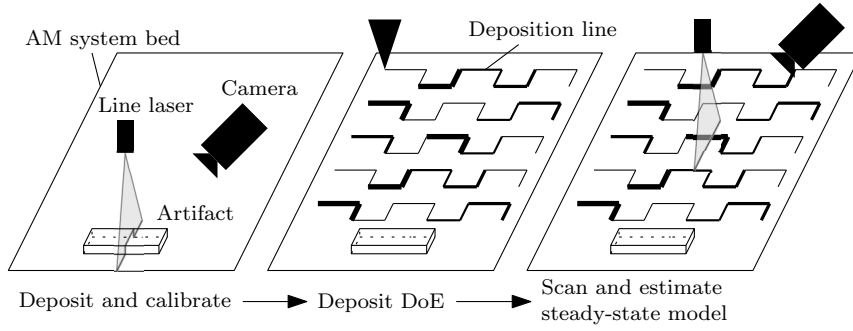


Figure 6.2: Automated empirical response modelling procedure.

## Bibliography

- [1] Julien Gardan. Additive manufacturing technologies: state of the art and trends. *International Journal of Production Research*, 54(10):3118–3132, 2016.
- [2] Terry Wohlers. Wohlers report. *Wohlers Associates Inc*, 2016.
- [3] Ian Gibson, David W Rosen, Brent Stucker, et al. *Additive manufacturing technologies*, volume 17. Springer, 2014.
- [4] JP Patel, CP Patel, and UJ Patel. A review on various approach for process parameter optimization of fused deposition modeling (fdm) process and taguchi approach for optimization. *International Journal of Engineering Research and Applications*, 2(2):361–365, 2012.
- [5] Farzad Liravi, Robin Darleux, and Ehsan Toyserkani. Additive manufacturing of 3d structures with non-newtonian highly viscous fluids: Finite element modeling and experimental validation. *Additive Manufacturing*, 13:113–123, 2017.
- [6] Tingting Huang, Shanggang Wang, and Ketai He. Quality control for fused deposition modeling based additive manufacturing: Current research and future trends. In *2015 First International Conference on Reliability Systems Engineering (ICRSE)*, pages 1–6. IEEE, 2015.
- [7] Paolo Magnoni, Lara Rebaioli, Irene Fassi, Nicola Pedrocchi, and Lorenzo Molinari Tosatti. Robotic am system for plastic materials: tuning and on-line adjustment of process parameters. *Procedia Manufacturing*, 11:346–354, 2017.
- [8] Mihaela Vlasea and Ehsan Toyserkani. Experimental characterization and numerical modeling of a micro-syringe deposition system for dispensing sacrificial photopolymers on particulate ceramic substrates. *Journal of Materials Processing Technology*, 213(11):1970–1977, 2013.
- [9] Matthias Faes, Wim Abbeloos, Frederik Vogeler, Hans Valkenaers, Kurt Coppens, Toon Goedemé, and Eleonora Ferraris. Process monitoring of extrusion based 3d printing via laser scanning. *arXiv preprint arXiv:1612.02219*, 2016.
- [10] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [11] Xi Zhang and Jian Zhang. Summary on calibration method of line-structured light sensor. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1142–1147. IEEE, 2017.
- [12] Robert Andersson. A calibration method for laser-triangulating 3d cameras, 2008.
- [13] Duan Fajie. A new accurate method for the calibration of line structured light sensor. *Chinese Journal of scientific instrument*, 21(1):108–110, 2000.
- [14] Zou Yuanyuan, Zhao Mingyang, and Zhang Lei. Direct calibration method of laser stripe vision sensor based on gauge block. *Chinese J Lasers*, 41(11):1108002, 2014.

- [15] Yuan Li, Qinglin Wang, De Xu, Zhiguo Yan, and Min Tan. Recent developments on welding image processing and features extraction. *Measurement and Control*, 40(5):139–145, 2007.
- [16] Lei Zhang, Mingyang Zhao, and Lihua Zhao. Vision-based profile generation method of twb for a new automatic laser welding line. In *2007 IEEE International Conference on Automation and Logistics*, pages 1658–1663. IEEE, 2007.
- [17] Xiuping Wang, Ruilin Bai, and Ziteng Liu. Weld seam detection and feature extraction based on laser vision. In *Proceedings of the 33rd Chinese Control Conference*, pages 8249–8252. IEEE, 2014.
- [18] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [19] Huu-Cuong Nguyen and Byung-Ryong Lee. Laser-vision-based quality inspection system for small-bead laser welding. *International journal of precision engineering and manufacturing*, 15(3):415–423, 2014.
- [20] Jae Seon Kim, Young Tak Son, Hyung Suck Cho, and Kwang Il Koh. A robust method for vision-based seam tracking in robotic arc welding. In *Proceedings of Tenth International Symposium on Intelligent Control*, pages 363–368. IEEE, 1995.
- [21] De Xu, Zemin Jiang, Linkun Wang, and Min Tan. Features extraction for structured light image of welding seam with arc and splash disturbance. In *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, volume 3, pages 1559–1563. IEEE, 2004.
- [22] Yuan Li, You Fu Li, Qing Lin Wang, De Xu, and Min Tan. Measurement and defect detection of the weld bead based on online vision inspection. *IEEE Transactions on Instrumentation and Measurement*, 59(7):1841–1849, 2009.
- [23] Wei Huang and Radovan Kovacevic. Development of a real-time laser-based machine vision system to monitor and control welding processes. *The International Journal of Advanced Manufacturing Technology*, 63(1-4):235–248, 2012.
- [24] Xiwen Liu, Guorong Wang, and Yonghua Shi. Image processing of welding seam based on single-stripe laser vision system. In *Sixth International Conference on Intelligent Systems Design and Applications*, volume 2, pages 463–470. IEEE, 2006.
- [25] Wilhelm Burger. Zhang’s camera calibration algorithm: in-depth tutorial and implementation. *Tech-nical Report HGB16-05 16th May*, 2016:55, 2016.
- [26] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. ” O’Reilly Media, Inc.”, 2008.
- [27] Janne Heikkila. Geometric camera calibration using circular control points. *IEEE Transactions on pattern analysis and machine intelligence*, 22(10):1066–1077, 2000.
- [28] Charles F Jekel and Gerhard Venter. pwlf: A python library for fitting 1d continuous piecewise linear functions. 2019.
- [29] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.
- [30] Åke Björck. *Numerical methods in matrix computations*, volume 59. Springer, 2015.