Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Chair of Computer Science VII
Robotics and Telematics

Bachelor's thesis

Calibration of a Fisheye Lens for Coloring of 3D Laser Data

Jochen Barf

October 2016

Examinor: Prof. Dr. Andreas Nüchter
Supervisors: MSc. Dorit Borrmann & Prof. Dr. Andreas Nüchter

Abstract

3D laser scanning is a powerful tool to capture a realistic image of a scenery. The measurement method, which is briefly explained in the introduction, does not provide any color information. The color is therefore extracted from fotos of the scenery. In this thesis the hypothesis, if a camera with a fisheye lens can be used to reliably color the 3D point cloud is evaluated. In that process, the mathematical concept of the calibration and mapping is developed and implemented, using the software libraries OpenCV and ThreeDTK. The usage of the implementation is then described in detail for further utilisation. An experiment to test the mathematical approach and the implementation was conducted and analysed. The examination of the results allows to draw conclusions regarding the quality of the calibration process. It was also possible to determine the optimal number of photos needed to cover the whole point cloud.

Contents

1 Introduction

1.1 3D Laser Scanning

1.1.1 Concept

Range measurements of active laser scanners are based on the concept of "time-of-flight". The scanner emits laser light towards an object, the light is reflected at its surface and reradiated towards the scanner. The scanner receives the signal and measures the time of travel. With that time, the range to the object can be calculated. In addition to the range also the reflectance of the object's surface can be determined from characteristics of the reradiated signal. To take measurements of more than one spot the laser beam must be directed in different directions. This can be done by rotating the scanner or by deflecting the beam with a system of mirrors. That implies that the measurements of multiple spots do not take place at the same time but one point at a time. This has no effect on the resulting data if the scanner does not change its position and observes a static environment. In this thesis the environment is always assumed to be static and the scanner to be stationary. The output of the 3D laser scanner is a three-dimensional point cloud. This 3D data, however, does not give any color information of the color of the surface. Therefore, the color must be acquired with another sensor and the data has to be merged.

1.1.2 Applications

3D laser scanning has a wide range of applications in a huge variety of disciplines. According to [Ebr14] the entertainment industry makes excessive use of 3D laser imaging. There it is used to construct a scene from real data and later modify it on the computer. But there are also more serious fields of application. For example, the system can be utilised to document crime scenes or accidents. Also hazardous environments can be safely explored while still providing realistic data as if one was on site. The concept can be useful in the industry to map buildings, bridges and industrial plants. The data can help to determine the state of construction and to analyse the quality of constructed structures. Especially the mining industry can benefit from 3D laser scanners. Not only is it possible to create a 3D map of the tunnels but one can also analyse the tunnel while it is build and compare it to a CAD model. One of the advantages of laser scanning is that there is a large amount of data recorded in a relatively small period of time. So the measurement team can clear the construction site faster and analyse the data later on the computer. With repeated measurements of the same tunnel over a large period of time it is possible to detect structural changes and rock deformations (cf. [vdMA12]).
Science also benefits from 3D laser scanning. In [MHH07] for example a geological experiment with a laser scanner to measure erosion and deposition volumes at a glacier is described. In the field of archeology, laser scanning makes the documentation of historical sites faster, more

accurate, and easier than ever before. The collected data can be used for restoration works, for educational purposes or simply for conservation.

1.2 Setup

This work deals with one particular setup. It is a combination of a laser scanner and a camera which is mounted on top of the scanner. The full setup can be seen in figure 1.1. The 3D laser scanner is the Riegl VZ400. With its multi-facet rotating mirror it reaches a maximum of 100° vertical field of view (from $-40°$ to $+60°$). By rotating the whole scanner it reaches a horizontal field of view of 360°. The scanner operates in the near infrared spectrum and has a minimum range of 1.5m (cf. [Rie14]). The camera is the Canon 1000D which is a standard reflex

Figure 1.1: Photo of the setup with laser scanner, camera and lens

camera. Its CMOS sensor has 10 megapixels on an area of 22.2×14.8mm (cf. [Can08]). The advantage of this camera is that there exists a communication interface via USB that allows it to be controlled with a computer. The camera is equipped with the Walimex Pro II fisheye lens with manual focus and aperture. It has a focal length of 8 mm and a maximum field of view of 180°. The closest focusing distance is 0.3m (cf. [wal16]).

1.3 Motivation

3D point clouds alone already give a very good idea of an object or an environment. But as explained in section 1.1.1 there is no color information available. The data gets much more intuitive and simpler to understand if every dot also has the correct color. There is currently a software toolchain in ThreeDTK available to successfully color the scan data with image data from the camera, but it was developed for a lens with a far smaller field of view and thus smaller distortions. ThreeDTK is a software library in `C++` with many algorithms to process 3D point clouds. The improvement that a lens with a wider field of view brings is that less photos are required to cover the whole 360° view. It thus reduces the acquisition time, the amount of data and the computational time in the post-processing. But the distortions must be handled differently. The computer vision library OpenCV introduced a new fisheye model for camera calibration in its version 3.0. In this thesis it is to be determined whether the coloring of 3D scan data with photos from a fisheye lens is possible by implementing the procedure using the fisheye model within OpenCV and evaluating the results.

2 Theory

2.1 Calibration

The laser scanner and the camera each have their own coordinate system an they practically never coincide. The way to map a coordinate from one reference frame to the other is described in this section. The coordinate system of the 3D point cloud has its origin in the position of the scanner and is here referred to as world coordinate system. Coordinates in the world frame are labeled with the subscript w and coordinates in the camera reference frame are labeled with the subscript c. Furthermore, there is the reference frame of the image, that is only two-dimensional and its coordinates are labeled with the subscript i.

2.1.1 Extrinsic

The camera's perception of the world underlies a transformation due to its position and pose with respect to the world origin. This transformation is given by a rotation R and a translation T. Coordinates of the world reference frame $\begin{bmatrix} x_w & y_w & z_w \end{bmatrix}^T$ are transformed into coordinates in the camera reference frame $\begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^T$ by

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{2.1}$$

where R is a 3×3 rotation matrix and T a 3×1 translation vector. Both are parameters that are to be measured for the specific system and are called the extrinsic parameters. Due to the rotation of the scanner, the rotation R differs for every photo. The transformation is described in section 2.2.

2.1.2 Intrinsic

Of course this relation does not give any information on the correspondence between pixels in the image taken by the camera and world coordinates. To assign a pixel to each world coordinate, the optical system must be taken into account. The pinhole camera model as seen in figure 2.1 is the usual way to describe an optical system. That model describes the angle θ between the principal axis and the incoming ray as

$$\theta = \arctan(r) \tag{2.2}$$

$$r^2 = \left(\frac{x_c}{z_c}\right)^2 + \left(\frac{y_c}{z_c}\right)^2. \tag{2.3}$$

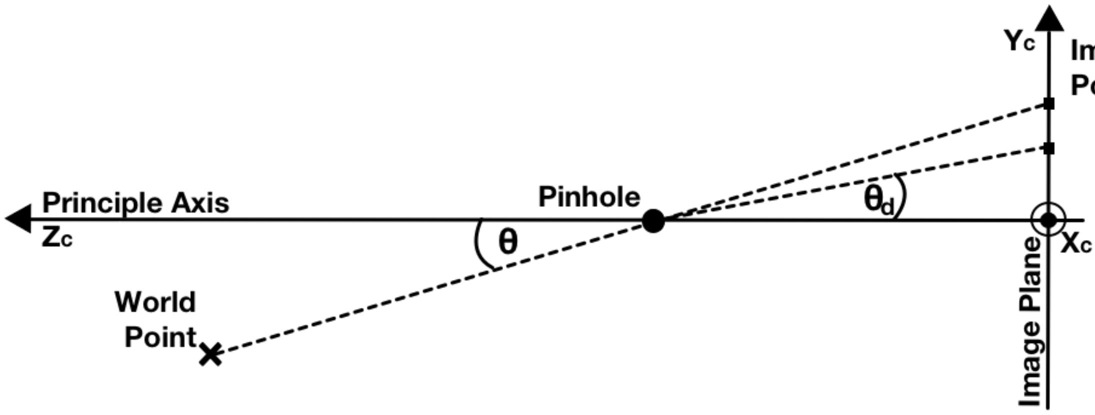


Figure 2.1: Sketch of the Pinhole Camera Model

The distortion of the pinhole camera model introduced in [Zha09] is not applicable for optical systems with very high distortions such as fisheye lenses. Therefore, the distortion model introduced in [KB] is used which defines the distortion of θ as

$$\theta_d = \theta(1 + k_1\theta^2 + k_2\theta^4 + k_3\theta^6 + k_4\theta^8) \tag{2.4}$$

where $D = (k_1, k_2, k_3, k_4)$ are the distortion parameters. In [KB] the distorted pixel coordinates $[\, x'_c \; y'_c \,]^T$ are then further described as

$$\begin{bmatrix} x'_c \\ y'_c \end{bmatrix} = \frac{\theta_d}{r} \begin{bmatrix} x_c \\ y_c \end{bmatrix}. \tag{2.5}$$

According to the pinhole camera model the distorted coordinates are then further transformed with respect to the focal length $f = (f_x, f_y)$ and the camera center $c = (c_x, c_y)$. That transformation is represented by

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_c \\ y'_c \\ 1 \end{bmatrix}. \tag{2.6}$$

The formulas in this and the last section define a relationship between a world coordinate $[\, x_w \; y_w \; z_w \,]^T$ and the corresponding pixel coordinate $[\, x_i \; y_i \,]^T$ in the image frame. The matrix in equation 2.6 containing f and c is called the camera matrix K.

2.1.3 Determination of Parameters

In this model there are 14 degrees of freedom, three for translation T, three for rotation R, four for distortion D, two for focal length f and another two for the camera center c. The tuple (T, R) is called the extrinsic parameters and the tuple (K, D) the intrinsic parameters where $K(f, c)$ is the camera matrix from equation 2.6. A point pair is a tuple of two points where one point is a 3D point in the world coordinate system and the other point is a 2D point in the image frame and both are located at the same feature:

$$pp = \left(\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}, \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right) \tag{2.7}$$

$M = \{m_1, m_2, ..., m_{|M|}\}$ is a set of sets of point pairs where each set of point pairs $m_i = \{pp_1, pp_2, ..., pp_{|m_i|}\}$. The calibration parameters can be determined from such a set of sets of point pairs using the Levenberg-Marquardt algorithm. It is possible to determine one optimal tuple of intrinsic parameters for the whole set M plus one tuple of extrinsic parameters for every element of M. The algorithm therefore iteratively minimises the sum of all squared reprojection errors given by

$$\sum_{i=1}^{|M|} \sum_{j=1}^{|m_i|} ||pp_{ij}^p - proj((K, D), (R, T)_i, pp_{ij}^w)||^2 \tag{2.8}$$

where pp_{ij}^p is the 2D pixel coordinate and pp_{ij}^w the world coordinate of the point pair j in set i. $proj((K, D), (R, T)_i, pp_{ij}^w)$ denotes the projection of the world point pp_{ij}^w into the image according to the formulas in sections 2.1.1 and 2.1.2. This procedure delivers one (K, D) tuple and $|M|$ times the $(R, T)_i$ tuple but for the mapping there is only one rotation and one translation required. In theory every $(R, T)_i$ tuple should be identical but real measurements have errors and in this case even huge outliers. Therefore, a median filter, specified in [BAEN12] as emed, that calculates the median of each component in T and R individually is a suitable way to determine one (R, T) tuple.

2.2 Mapping

Section 2.1 states a relationship between a world point and an image point which is true in theory but in reality image points are not infinitely small. Pixels on a image sensor have a discrete size and all light that falls onto the area of that pixel is accumulated into one value. So there is actually not one ray that falls exactly on one image point as illustrated in figure 2.1, but a rectangular cone full of rays that fall onto a rectangular pixel. Figure 2.2 illustrates that cone as the grey area. All world points that are located inside that cone are mapped onto one and the same pixel. That also implies that world points that are exactly behind each other from the cameras point of view, are also mapped onto the same pixel. However, the equations derived in section 2.1 can be used without any limitations. The difference is, that the calculated image point is discretised and hence one pixel possesses a correspondence to an infinite number of possible world points. That makes it impossible to find exactly one world point for one pixel in the image.

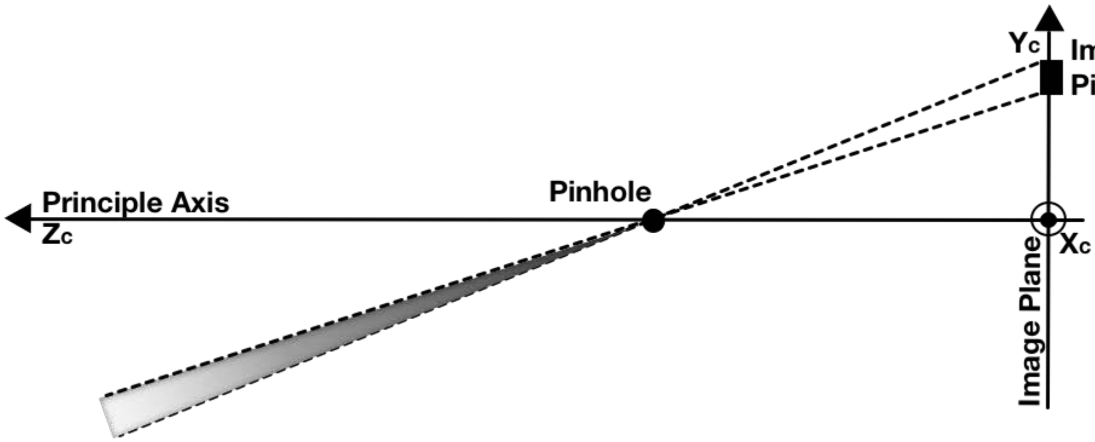


Figure 2.2: Pinhole camera model for discretised pixel sizes

2.2.1 Multiple Poses

The rotation R defined by equation 2.1 refers to one pose. In order to cover the whole 360° view, one must take multiple photos at different angles. Let the number of poses that are needed be k. For a counter-clock-wise rotation the rotation angle for a pose $i \in [0, k-1]$ is

$$\beta_i' = 2\pi \left(1 - \frac{i}{k}\right) \tag{2.9}$$

Here it is assumed that the first photo is taken at the exact same pose as the calibration was made, which is generally not true. To correct this the difference between the angle γ, at which the calibration was made and the angle δ, at which the first photo was made is to be added:

$$\beta_i = \beta_i' + \gamma - \delta \tag{2.10}$$

For the assumption that the camera is rotated around the z-axis of the world coordinate system the rotation matrix for the camera at the pose i is

$$R_i = R_c \begin{bmatrix} \cos\beta_i & -\sin\beta_i & 0 \\ \sin\beta_i & \cos\beta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

where R_c is the rotation that was found by the calibration in section 2.1.3.

2.2.2 Projection Ambiguity

Due to the symmetry of the pinhole camera model it is possible that world points that lie behind the camera are mapped onto the image plane. Figure 2.3 illustrates a ray through the image point and the pinhole that intercepts two world points that are on different sides of the image plane. In order to distinguish between those world points that lie in front of the camera and

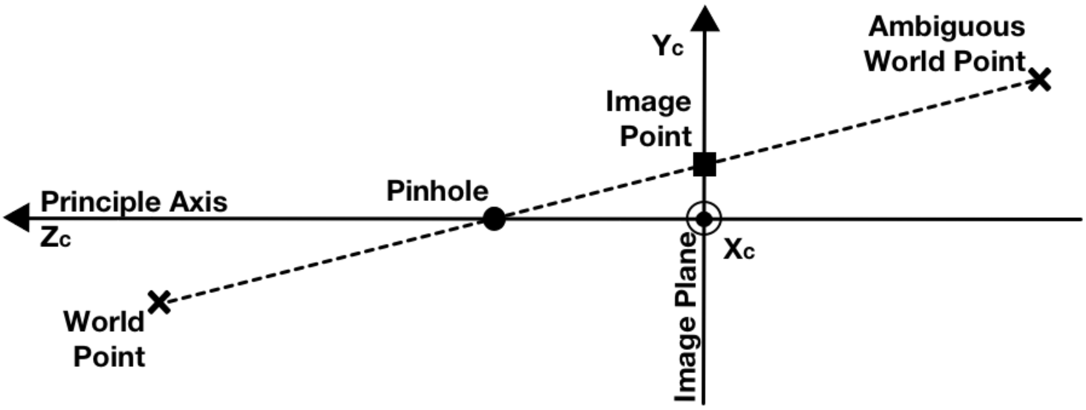


Figure 2.3: Ambiguity of the Pinhole Camera Model

those that lie behind it one must transfer the wold point in question into the camera coordinate system. This can be done with equation 2.1. A world point $[\, x_w \; y_w \; z_w \,]^T$ is then located behind the camera if and only if the component in direction of the principle axis of its representation in camera the frame is negative, that means for $z_c < 0$.

3 Implementation

The implementation is divided in three parts, the ClickTool, the CalibrationTool and the ColorMapper. All three of them were implemented in C++. The CalibrationTool and the ColorMapper are not part of the ThreeDTK library whereas the ClickTool is. That is because the fisheye namespace of the OpenCV library was introduced in version 3.0.0 but ThreeDTK does not support this version yet. The ClickTool is used to find point pairs from calibration scans and photos. The CalibrationTool then can calculate the calibration parameters from those point pairs. The ColorMapper was implemented to color laser scan data with photos from the same scene by using the calibration parameters determined by the CalibrationTool. Each tool is explained in detail in the following sections.

3.1 ClickTool

The ClickTool includes functions from the libraries OpenCV 2.4.9 and Boost 1.55.0 and is itself part of the ThreeDTK library within the FBR toolbox. It was developed and tested on Ubuntu 15.04 with the compiler g++ 4.9.2.

3.1.1 Purpose

To find the extrinsic and intrinsic calibration one musst know point correspondences between pixels and world points. With enough correspondences one can calculate the extrinsic and intrinsic parameters according to section 2.1.3. In [BHH+15] a method to automatically detect point correspondences is proposed. Due to the high aperture angle of the fisheye lens, patterns appear very small in the image. To make them appear greater one would have to get them closer to the camera but that would not help either, because that would bring the pattern out of focus since the focus must be set to infinity in order to focus on the objects that are later photographed. A second possibility to allow automatic detection of correspondences would be to increase the size of the pattern which is not applicable for long journeys. The solution that was found in this thesis is to put multiple patterns into the picture. This, however, rules out the possibility to detect them automatically, which arose the demand for this tool. One can set the correspondences manually by clicking on the features appearing in both images. Therefore, the 3D laser scan point cloud is transformed into a 2D panorama image. With this method it is not necessary to use any kind of pattern but it is more convenient and simpler to precisely identify and mark correspondences.

3.1.2 Usage

In order to enable the compilation of the ClickTool, the FBR toolbox of the ThreeDTK library is to be enabled. After compilation the binary file can be found in the `Slam6d/bin` directory, where it can be executed with the command

```
$ ./click_tool [options] /path/to/directory/
```

The tool was derived from the `scan_to_panorama` tool, so the main options are identical. There are, however, some additional options which are listed in table 3.1. It is assumed that the photos, the scan file and the pose file are all in the specified directory. It is further assumed that the file name format of the photos is photo[*poseNumber*]_[*photoNumber*] where [*poseNumber*] is the zero-padded, three digit index of the pose and [*photoNumber*] is the index of the HDR photo. A typical command to start the tool would be

Option	Value	Description
-a	integer	sets the number of poses in one scan at which photos were taken
-c	integer	sets the number of photos that were taken at one pose
-h	string	sets the format of the pictures: [PNG | JPEG | JPEG2000 | TIFF]
-d	float	sets the lower boundary for range normalisation
-D	float	sets the upper boundary for range normalisation
-r	none	enables the rotation of the photo

Table 3.1: Additional Start Options of the ClickTool

```
$ ./click_tool -s0 -e0 -a5 -c3 -hJPEG -tRIEGL -frxp -r -D10 MyData/
```

It would open the scan file `MyData/scan000.rxp`, the pose file `MyData/scan000.pose` and the photo files `MyData/photo00[i]_[j].jpg` where $i \in [0, 4]$ and $j \in [0, 2]$. Once the tool is started it shows three windows, the photo, the panorama image of the scan and the control panel. With the trackbars on the control panel the size of the windows can be modified. The user interface is operated with mouse and the keyboard in combination. An overview of all available controls is given in table 3.2. The HDR photo that is displayed may be changed with the left and right arrow keys. There are two possibilities how the scan is displayed, either as range image (see fig. 3.1a) or as reflectance image (see fig. 3.1c). The range image illustrates the range by showing farther objects brighter than closer objects. The reflectance image does not give any information about the range but the intensity of the reflection of the surface. In most cases the reflectance image is of more use than the range image because it makes paint and other differences in the surface visible where the range image just shows a homogenous area. If the range image does not show enough contrast, the normalisation boundaries must be adjusted on startup (-d and -D options, see table 3.1). With the +/- keys it can be zoomed into each image where the center of zoom is the position of the mouse. Once zoomed one can shift the field of view inside the image by using the w/a/s/d keys. To mark a pixel as feature point a left click on the desired pixel is required. Once a feature point is marked in one window the corresponding feature point must be marked in the other window before another feature can be marked. The point pairs are

Key	Description
$\leftarrow$ $\rightarrow$	changes the displayed HDR image
t	toggles scan image between range image and reflectance image
$\uparrow$ $\downarrow$	changes the current PPR file
$+$ $-$	zooms to the position of the mouse
left click	adds a feature point at the Position of the mouse
right click	removes the feature point at the position of the mouse
w	navigation upwards
s	navigation downwards
a	navigation to the left
d	navigation to the right
f	toggles visibility of all feature points of all PPR files in the directory
ESC	terminates program

Table 3.2: Controls of the ClickTool

saved in files with the extension `.ppr`. The CalibrationTool calculates the extrinsic parameters for each PPR file, so it is advantageous to group close feature points into one PPR file. The PPR file can be changed with the up and down arrow keys. With the f key all PPR files in the directory are loaded and displayed. This is enabled on startup and can be disabled with the f key, aswell. The escape key opens the next scan or terminates the program.

3.2 CalibrationTool

The CalibrationTool utilises the libraries OpenCV 3.1.0 and Boost 1.59.0. It was developed and tested on Mac OS 10.11.6 using the Apple LLVM 8.0 compiler with optimisation level O3.

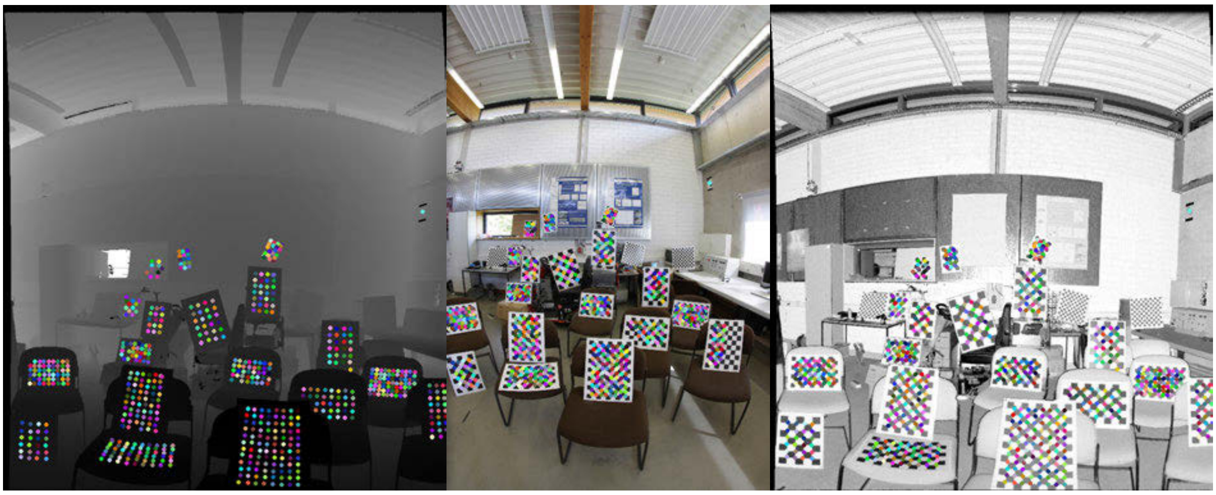
3.2.1 Purpose

With the ClickTool, explained in section 3.1, point pairs of features seen in the scan and the photo were found and saved. Those pairs are now used to find the intrinsic and extrinsic parameters of the setup. This is done with the method described in section 2.1.3. The output is a full set of extrinsic and intrinsic parameters that can be used by the ColorMapper to find the correct color for each world point.

3.2.2 Usage

The CalibrationTool is configured with a configuration file in the XML format. It is given to the program on startup as argument:

```
$ ./CalibrationTool path/to/configurationFile.xml
```



(a) range image **(b)** photo **(c)** reflectance image

Figure 3.1: Screenshots of the ClickTool

If the specified file does not exist, an example file is saved at that location. In the file one can change the settings by editing the content of the xml files. For every option there is a comment that explains it in detail. There are two modes that the CalibrationTool can be used in, the intrinsic-only calibration and the full calibration. The intrinsic-only calibration is based on photos of one or more chessboard patterns and only determines the intrinsic parameters. In this mode the chessboard patterns can be detected automatically and the point pairs from the ClickTool are not needed. Therefore, the path to the photos of the patterns must be given. For the tool to be able to detect the patterns the width, the height in terms of numbers of inner corners and the maximum number of occurrences of each pattern must be specified in the configuration file. There is the option to save and later load the detected patterns to and from a specified file path. It is also possible to undistort photos in a specified directory with a given camera matrix and distortion parameters. Furthermore, the file paths to the locations where the camera matrix and the distortion parameters are to be saved and the number of threads that are to be used for the detection must be specified. If the intrinsic-only mode is disabled, the CalibrationTool determines the intrinsic as well as extrinsic parameters from point pairs. Therefore, the superior directory containing all PPR files created by the ClickTool must be specified. In difference to the intrinsic-only calibration it is also required to provide the width and height of the photos as well as the two file paths were the rotation and the translation is to be saved. For both modes it is possible to give an intrinsic guess in the form of a camera matrix and a distortion from specified paths. Typically one would determine the intrinsic parameters with the intrinsic-only calibration and use it as intrinsic guess for the full calibration.

3.3 ColorMapper

The ColorMapper was developed and tested with the same configuration as the CalibrationTool.

3.3.1 Purpose

The output of the CalibrationTool, explained in the last section, is the camera matrix, the distortion parameters, the rotation and the translation of the setup. The purpose of this tool is to use this information to conduct the final step and find the correct color of each world point in the photos. The output is a 3D point cloud with color information for every point. Points that could not be mapped onto any photo get the color black.

3.3.2 Usage

Very similar to the CalibrationTool, the ColorMapper is configured with an XML file. It is also launched in the same way:

```
$ ./ColorMapper path/to/configurationFile.xml
```

If there is no configuration file specified in the argument, the ColorMapper creates a sample file at the specified location. In that configuration file one must specify the location of the calibration parameter files, the location of the scan file, the directory of the photos and the location of the output. The scan file must contain the 3D point cloud in ThreeDTK's `uos` format. Such a file can be created from any other format with the `export_points` tool. The output file will be in the `uos_rgb` format which can be viewed with the tool `show`. Both, `export_points` and `show` are tools delivered with the ThreeDTK library. In addition, it must be specified how many equiangular poses per scan were used to take photos as well as how many photos were taken at one pose for HDR photos. Furthermore, it is required to specify at what angle the photos for the calibration were taken - all calibration photos must be taken at the same angle. It is not to be confused with the angle at which the first photo was taken, which must also be given. Then there is the possibility to enable the HDR photo composition. If this option is disabled the index of the HDR photo that is to be used must be specified. There is an option that specifies the behaviour if a world point was mapped onto multiple photos. If the averaging option is enabled, the mean of all color information for one world point is used, otherwise the color is always replaced by newer information. Finally, there is one option to specify how many threads are to be used and one option to rotate the image counter-clock-wise before mapping.

4 Evaluation

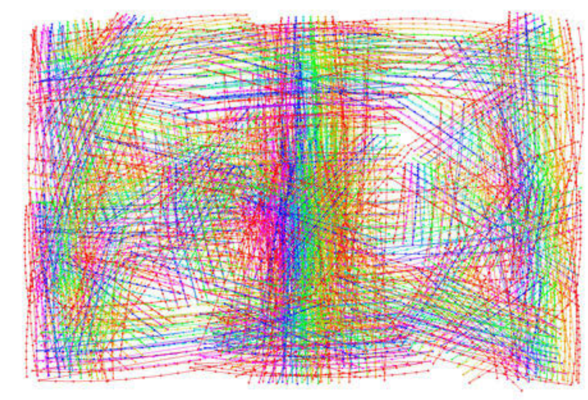
4.1 Intrinsic Calibration

As explained in section 3.2.2, the CalibrationTool can be used to determine the intrinsic parameters from photos of chessboard patterns. To analyse the performance of this method a test with 111 photos similar to the one in figure 4.1b of a chessboard pattern was conducted. The chessboard had a width of 8 and a height of 17 inner corners and the photos had a resolution of 3888×3592 pixels. In the photo the high distortions can easily be seen, especially at the door frame on the left hand side. To sample the distortion sufficiently well, the whole field of view should be covered with a dense grid of feature points. To accomplish that, the camera was rotated for every photo so that the pattern covers another part of the image. The image in figure 4.1a illustrates all detected patterns of all photos drawn into one single image. It is a good indication that the whole field of view was sampled with reasonable density. Using the CalibrationTool with the detected feature points from all photos, the intrinsic parameters of the optical system were determined and are listed in table 4.1. With those parameters the photo from figure 4.1b was undistorted and the resulting image is displayed in figure 4.1c. The almost identical values for the vertical and horizontal focal length suggest that the focal length is the same in both dimensions. This is expected since the lens should be build symmetrically. More interesting is that the center point is shifted by more than 28 pixels to the left from the actual image center. That has the visible affect on the undistorted image that its corners on the right hand side are less spread than on the left hand side. The intrinsic calibration can then be con-

Parameter		Value
f_x	$=$	$1.48611 \cdot 10^3$
f_y	$=$	$1.48667 \cdot 10^3$
c_x	$=$	$1.97251 \cdot 10^3$
c_y	$=$	$1.29349 \cdot 10^3$
k_1	$=$	$2.88819 \cdot 10^{-2}$
k_2	$=$	$7.37330 \cdot 10^{-3}$
k_3	$=$	$5.25318 \cdot 10^{-5}$
k_4	$=$	$5.14530 \cdot 10^{-4}$

Table 4.1: Intrinsic Calibration Parameters

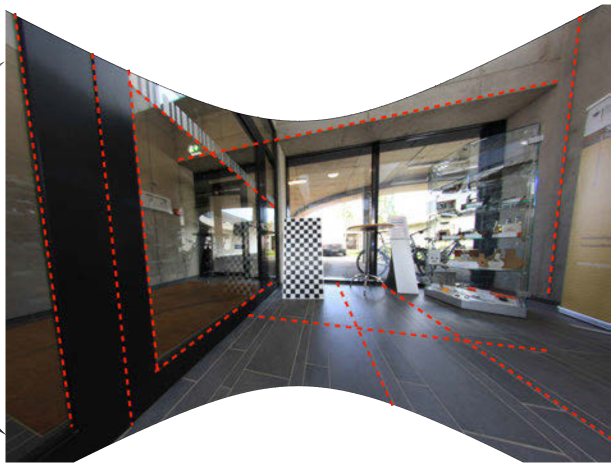
sidered good if straight lines in reality are straight lines in the undistorted image. Figure 4.1d shows a scaled version of the image in figure 4.1c where straight dashed lines were painted into the image on positions where straight lines are expected. It can be seen, that all edges follow



(a) Cover Image

(b) Distorted Image

(c) Undistorted Image



(d) Cropped Undistorted Image with Lines

Figure 4.1

the dashed line perfectly, even the highly distorted door frame on the left hand side. This is a good indication, that the intrinsic calibration is accurate.

4.2 Extrinsic Calibration

The procedure to perform the extrinsic calibration is more complicated than that for the intrinsic calibration. First, one must install the setup explained in section 1.2 and put it in a suitable environment. For the calibration it is not necessary to put chessboard patterns in the field of view but it is more convenient to find a high number of feature points in both images if there are patterns in them. For the extrinsic calibration actually already one photo together with the corresponding part of the laser scan is sufficient. Hence, not a full 360° view but only the part of the scan which is also covered by the photo is needed. Of course it has to be known at which angular position of the scan the photo was taken. In the experiment evaluated here the photo was taken at an angular position of 40°. To achieve a more precise calibration either more patterns can be placed in the scenery or photos and scans can be taken from different positions within the scenery. In this case it is important to ensure that the photo is taken at the same angular position of the scan for each set of data. A photo of the calibration setup for this experiment can be seen in figure 4.2. For this experiment, 6 positions with 20 patterns were recorded.

The next step is to find point pairs. This was done with the ClickTool that was explained in

Figure 4.2: Extrinsic Calibration Setup

section 3.1. All point pairs of one pattern were stored in one PPR file. Thanks to the manual method it is possible to not only create PPR files with point pairs from only one pattern but also with point pairs from patterns that are distributed over the whole field of view. Doing so simulates a large pattern that covers a major area of the image. In [vdL16] it was determined that patterns with a greater area provide better calibration, so it is advisable to include simulated large patterns. From those 6 positions, 57 PPR files with a total of 17.861 point pairs were created. But here it must be noted, that due to a bug in OpenCV 3.1 only 17.720 point pairs could be used for calibration. Figure 3.1 shows all point pairs of one position in the photo and the scan.

These point pairs were then used to find the extrinsic parameters by feeding them into the CalibrationTool. Besides the point pairs, also the intrinsic parameters found in section 4.1 were fed into the calculation as intrinsic guess. The resulting set of parameters is shown in table 4.2. In contrast to the intrinsic parameter set from table 4.1 also the rotation and the translation is included. For the rotation it must be noted that the values are not Euler angles but Rodrigues rotation parameters. The parameters of the camera matrix are only slightly changed whereas the distortion parameters differ in sign and up to 3 orders of magnitude.This might be an effect of the additional information given from the real 3D scan points. Before, in the intrinsic only calibration, the world points were simulated with the third dimension set to zero. The additional information in that dimension might change the optimum of the parameters.

To analyse the quality of the calibration, two scans were performed. The first location was

Parameter		Value
f_x	=	$1.48259 \cdot 10^3$
f_y	=	$1.47988 \cdot 10^3$
c_x	=	$1.96821 \cdot 10^3$
c_y	=	$1.29702 \cdot 10^3$
k_1	=	$3.83024 \cdot 10^{-2}$
k_2	=	$-2.55709 \cdot 10^{-2}$
k_3	=	$3.29389 \cdot 10^{-2}$
k_4	=	$-9.78449 \cdot 10^{-3}$
r_1	=	1.87732
r_2	=	0.606002
r_3	=	1.81290
t_1	=	-20.0241
t_2	=	-1.63506
t_3	=	-8.23834

Table 4.2: All determined calibration parameters

in the robotics hall of the chair of robotics and telematics, were the calibration took place and the second one was outside, in front of the Computer Science building on the Campus of the University of Würzburg. These two scans provided a wide range of distances. With the help of the ColorMapper tool, explained in 3.3, and the calibration parameters from table 4.2, the scan points were colored. A visualisation as a panorama picture, created with ThreeDTK's `scan_to_panorama` tool, can be seen in figure 4.3. The upper picture shows the colored scan from the robotics hall and the lower picture the one taken at the campus. In the following this data is further analysed.

First of all, the scan points seem to be reasonably well colored but the panorama image does not really give the impression of a 3D image. This is due to the fact that the data is viewed from the same position as the camera-scanner-setup was located. It becomes more visible, that there is 3D information if one changes the point of view and the perspective of the virtual camera inside the point cloud. Figure 4.4 shows each data set as seen from a virtual camera that is not located in the world origin. Thanks to the limited vertical field of view, the very spot on which the scanner was positioned is not visible to the scanner and thus appears as a white spot inside a well colored area. Figure 4.4a perfectly shows that phenomenon in the left bottom corner. The other white spots that look like snow are actually no points at all. No scan points were measured at those locations because they were located behind another object and thus not are visible to the scanner. To also cover those spots one would have to move the scanner to a different position, take another scan and merge those two scans. In [BHH+15] an experiment with a mobile robot that autonomously scanned the environment and created a colored 3D model of the environment is described.

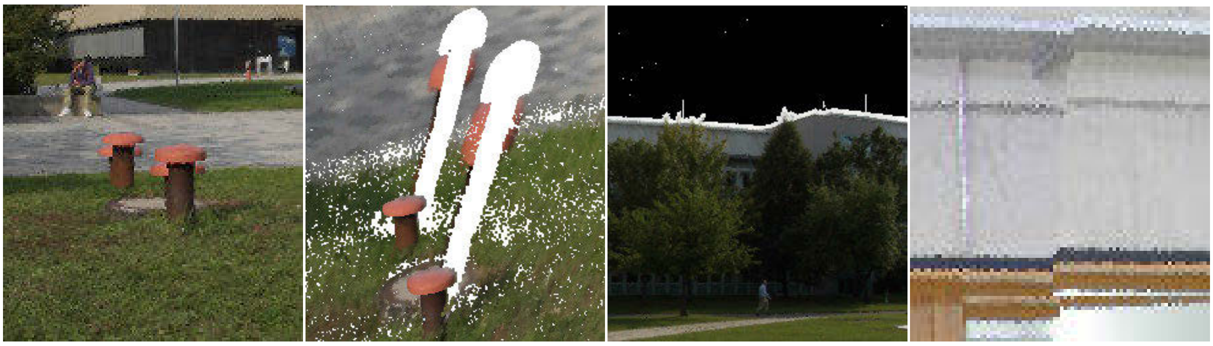
Figure 4.3: Panorama images of the colored 3D point cloud from scans in the robotics hall (top) and from the campus (bottom).

Another effect that is based on perspective can be observed in both scans. The left picture in

(a) (b)

Figure 4.4: View of colored 3D point cloud from a virtual camera

figure 4.5a was cut from the lower panorama pictures in figure 4.3. In the left picture, there seem to be four mushroom-shaped pipes. The picture at the right shows the same objects viewed from a different position. With the second picture it becomes obvious that there are only two objects and that the other two were painted onto the ground. This happens because the scanner and the camera are not located at the exact same spot, instead they are more than 20cm apart. That means both sensors see the objects from different angles, in other words, they have different perspectives. That also means that there might be objects that can be seen by one sensor but not by the other because the object is in another's shadow. In this case the scanner was able to look under the red part of the pipe but the camera just saw the red pipe. The scan points are located behind each other from the camera's point of view and are thus mapped onto the same image point and get the same color.



(a) (b) (c) (d)

Figure 4.5: Errors of the coloring

Of course the calibration is not perfect und thus there are errors in the mapping process. Figure 4.5c is copied from the panorama image of the campus and emphasises a bright line between the roof of the building and the sky (black). This is the result of a calibration inaccuracy that leads to the mapping of scan points of the roof onto the part of the photo already covering the sky. Another effect of imperfect calibration is shown in figure 4.5d. It is a close image of the border between two images. The textures on the wall do not meet perfectly. These errors are regarded as minor imperfections that are due to unavoidable inaccuracies in the calibration process.

4.3 Number of Poses

The advantage of using a fisheye lens is the wider field of view. Thanks to that, there is a reduced number of poses required to cover the whole 360° view. Besides the advantage that the waiting time for one scan is shorter, there is also less time for the environment to change. The time between a photo and the scan is reduced and therefore a more accurate mapping is possible. So it is of interest how many poses are needed to cover the whole 360° view.

From the translation shown in table 4.2 it can be seen that the camera origin is not located on the z-axis of the world coordinate system. Thus, a rotation of the camera around the z-axis does not rotate it on the spot but moves it in the 3D space. Because of that and the fact, that the image is highly distorted, it is not possible to simply calculate the field of view and derive the number of poses from there. Therefore, an experiment was conducted that determined the coverage of a simulated scan. A dense artificial scan point grid was placed at a constant distance to the world origin. The simulated field of view of the scanner was 60° to the top, 40° to the bottom and 360° around the scanner. This generates a point cloud, that has the shape of a sphere that has circular holes at the top and the bottom. The density of the grid was 20 points per degree in both dimensions. With the extrinsic and intrinsic parameters, found in section 4.2, the world points were then mapped onto artificial photos with the procedure that is also used in the ColorMapper (see sec. 3.3). Within that process the coverage of the artificial scan was determined by counting colored and not colored points. This procedure was repeated for multiple numbers of poses and all distances between 20cm and 100cm with a step size of 1cm. Figure 4.6 is a graph illustrating the results of that experiment. It shows the coverage of the

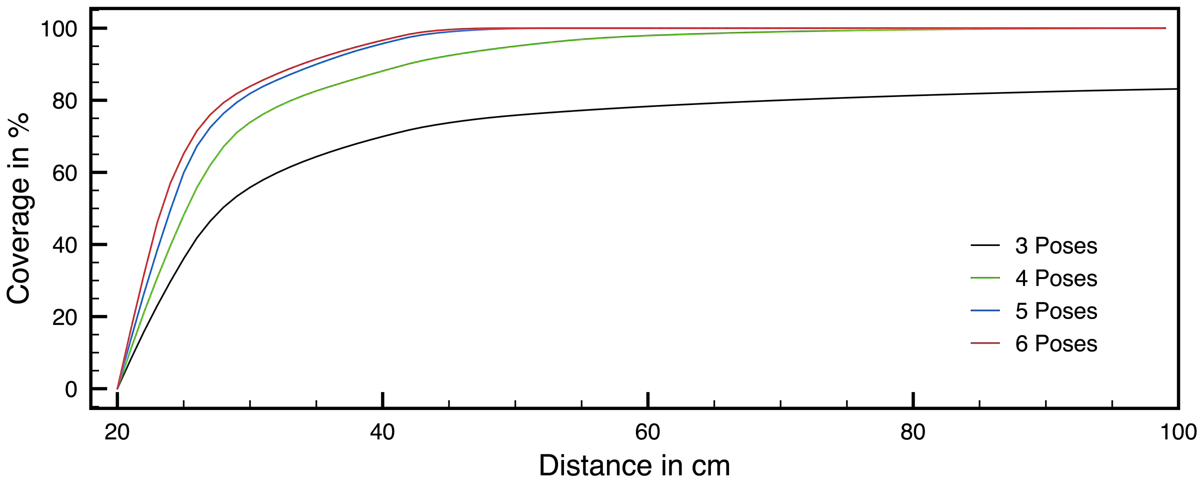


Figure 4.6: Percentage of coverage of the 360° view vs. the distance of the scan points for different numbers of poses

360° view in percent versus the distance of the scan points in cm for different numbers of poses. Of course, a higher number of poses leads to a better coverage, but the difference also disappears with the rising number of poses. The coverage for all numbers of poses at a distance lower or equal 20cm is 0. This can be explained with the position of the camera in the world coordinate system. Table 4.2 shows an absolute value for t_1 of more than 20. Points that are closer than 20cm might lie behind the camera and are ignored as explained in section 2.2.2. The lower the number of poses the more distance is needed to have a full coverage. With only 3 poses the 100% level is never reached. Table 4.3 lists the minimal distances at which a full coverage for a number of poses is reached. As already seen in the graph, there is not much difference for

Poses	Distance
3	∞
4	99 cm
5	52 cm
6	48 cm
7	46 cm

Table 4.3: The minimal distance of an object to the scanner for several numbers of poses for save coloring.

pose numbers above 4, whereas there is a very high difference between 4 and 5 poses. Judging from this data it is advisable to use 5 poses, since it is a reduction of the minimal distance of

almost 50% to 52cm and a higher number would improve the minimal distance insignificantly. But [Rie14] specifies the minimum range of the scanner at 1.5m. That means that if there are objects closer than 1.5m, they are not measured. In this case the optimum number of poses is 4 since one can be sure that there will not be any scan point closer than 1m.
For a closer analysis of the coverage the image in figure 4.7 shall be considered. They are panorama images of the artificial point cloud where the covered areas are colored and the rest is black. Shown are the distances 30cm and 45cm with 3, 4, 5 and 6 poses. It can be seen that the upper part has always low coverage. This is of course also induced by the extended field of view of 20° more at the top than at the bottom. Therefore, it might help to tilt the camera upwards

(a) 3 Poses, 30 cm distance **(b)** 3 Poses, 45 cm distance

(c) 4 Poses, 30 cm distance **(d)** 4 Poses, 45 cm distance

(e) 5 Poses, 30 cm distance **(f)** 5 Poses, 45 cm distance

(g) 6 Poses, 30 cm distance **(h)** 3 Poses, 45 cm distance

Figure 4.7: Panorama pictures of the coverage of simulated scan data at two distances for four different pose numbers. Black areas are not covered, colored areas are covered.

by a few degrees in order to compensate for the asymmetric field of view of the scanner.
A possibility to further reduce the number of poses might be to mount the camera with a rotation of 90° around the principle axis. Due to the format of the image sensor there is a difference of the vertical and horizontal field of view. In this configuration the greater field of view is aligned with the vertical axis, so a rotation by 90° around the principle axis would align the greater field of view with the horizontal axis. With that configuration it might be possible to cover a larger slice of the 360° view at once.

5 Conclusion

The hypothesis under examination in this thesis that the coloring of 3D laser scan data with images from a fisheye lens is possible, can be accepted. In the evaluation it could be shown that the calibration is accurate enough to map each point from the 3D point cloud into a photo and acquire the corresponding color information. Therefore, the theoretical concept of the pinhole camera model with a distortion according to [KB] resulting in 14 degrees of freedom that are determined with the Levenberg-Marquardt algorithm can be approved. It can also be concluded that the implementation of the projection of the 3D point into the image, the optimisation algorithm Levenberg-Marquardt to find the calibration parameters and the detection algorithm for chessboard patterns in OpenCV 3.1 function correctly and terminate in a reasonable amount of time.

Of course, the calibration is not perfect and therefore there are visible errors. For example, one can see the border between two photos in the colored scan due to a small offset of the structures in one photo with respect to the same structure in the other photo. But this is a minor inaccuracy that also occurs in setups without a fisheye lens. The manual procedure to acquire point correspondences is tedious but delivers a reasonable result.

It was determined, that the absolute minimum number of photos that are needed to cover the whole field of view is four. The limiting factor here is the minimum range of the laser scanner. Further efforts should be directed towards the automatisation of the acquisition of the point correspondences and the improvement of the accuracy of the calibration.

Bibliography

[BAEN12] Dorit Borrmann, Hassan Afzal, Jan Elseberg, and Andreas Nüchter. Mutual calibration for 3d thermal mapping. September 2012.

[BHH+15] Dorit Borrmann, Robin Heß, HamidReza Houshiar, Daniel Eck, Klaus Schilling, , and Andreas Nüchter. Robotic mapping of cultural heritage sites, February 2015.

[Can08] Canon. *EOS 1000D User Manual*, July 2008.

[Ebr14] Mostafa A-B Ebrahim. 3d laser scanners: History, applications, and future. *ResearchGate*, October 2014.

[KB] Juho Kannala and Sami S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses.

[MHH07] David J. Milan, George L. Heritage, and David Hetherington. Application of a 3d laser scanner in the assessment of erosion and deposition volumes and channel change in a proglacial river. *Wiley InterScience*, September 2007.

[Rie14] Riegl. *Riegl VZ-400 Datasheet*, September 2014.

[vdL16] Joschka van der Lucht. Robuste kamerakalibrierung für unterwasser 3d-rekonstruktion, April 2016.

[vdMA12] J.W. van der Merwe and D.C. Andersen. Applications and benefits of 3d laser scanning for the mining industry, 2012.

[wal16] Product description walimex fisheye 2 dslr. walimexpro.de/de/produkt/produkt/835-fish-eye-ii-dslr.html, 2016.

[Zha09] Z. Zhang. A flexible new technique for camera calibration. Technical report, Microsoft Research, December 2009.

Proclamation

Hereby I confirm that I wrote this thesis independently and that I have not made use of any other resources or means than those indicated.

Würzburg, October 2016