



The University of Southern Denmark
Faculty of Engineering
Maersk Mc-Kinney Moller Institute

Flexible Camera Calibration

Project in Advanced Robotics

Authors:

Jakob Yde-Madsen	Oliver Holck Vea
Malte Nygaard Jensen	Simon Lyck Bjært Sørensen

Supervisor:

Jakob Wilm

May 2020

The University of Southern Denmark
Campusvej 55, 5230 Odense
Faculty of Engineering

Authors:

Name: Jakob Yde-Madsen
Mail: jayde16@student.sdu.dk

Name: Oliver Holck Vea
Mail: olvea16@student.sdu.dk

Name: Malte Nygaard Jensen
Mail: malje16@student.sdu.dk

Name: Simon Lyck Bjært Sørensen
Mail: simso16@student.sdu.dk

Supervisor

Name: Jakob Wilm

Date of Submission:

The May 28, 2020

Pages: 35

Abstract

This paper researches the possibilities of using a generic model to calibrate a camera instead of the parametric one used in most applications today. The generic model is constructed using a b-spline and optimized using bundle adjustment. This results in a model that seems to have lower systematic error and in some cases also a lower RMS reprojection error than the parametric model. The generic model has been tested on multiple cameras and achieves decent results on all of them, however, it does take significantly longer to calibrate than the parametric model.

Acknowledgements

We would like to thank Jakob Wilm Assistant Professor, Ph.D., Biomedical Engineering at The University of Southern Denmark for his guidance during the project.

Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
2 Previous Work	2
3 Data	3
4 Parametric Model	4
4.1 Results	5
5 Generic Central Model	11
5.1 B-Spline	11
5.2 Bundle Adjustment	15
5.3 Results	19
6 Discussion	25
7 Conclusion	26
8 Future Work	27
8.1 Initialization of the Forward Projection	27
8.2 Multi-scale Bundle Adjustment	27
8.3 Test for Overfitting	27
8.4 General Optimization	28
8.5 Improving Feature Detection	28
9 Bibliography	29
1 Appendix	30
1.1 Division of Labor	30
1.2 GitHub	30
1.3 Normality for the Scorpion camera	31
1.4 Charuco Board	33
List of Figures	34
List of Tables	35

1 Introduction

Computer vision systems rely on camera models to create accurate representations of their environment. Classic parametric camera models are based on physical interpretations of cameras, where a mathematical model is constructed for the source of incoming light. Complex lens distortion is a challenge for the parametric models since the few degrees of freedom limit the detail in the model. The resulting loss of accuracy is generally small compared to the vision algorithms that are applied after, but the error persists regardless of the chosen method and can also introduce systematic error. In computer vision and statistics, it is common to use methods that rely on the residuals of the camera model being normally distributed, which is far from guaranteed when using a classic parametric model.

This paper describes the application of a different method for generic camera modeling based on a generic model described in the article: "Why Having 10,000 Parameters in Your Camera Model is Better Than Twelve" [1].

The goal of the project is to implement the methods for calibrating cameras using both the generic and the parametric model and comparing their performance on several parameters:

- Structural error in the residuals.
- Overall error in the residuals.
- Normality of the residuals.

2 Previous Work

Generic models for camera calibration have already been created in different forms. Article [2] does it theoretically with a closed-form solution, whereas article [1] and [3] does this based on a b-spline. This project attempts to re-create the results of the latter two and compare the performance with parametric models.

3 Data

To get a comprehensive idea of the performance of the methods, seven different data sets were collected. The data sets consist of both chessboard and charuco board calibration images. The number of images, the boards, the cameras, and subdivisions of data is shown in table 1. The data set made with a Point Grey Scorpion camera is from DTU [4]. It was made as part of a different project [5]. The rest of the data is made as a part of this project. To limit calibration time and memory usage, the data sets were split into six folds.

In the data sets we produced, a charuco board was used as the known 3D structure. The reason for using the charuco board instead of the chessboard is that the charuco board allows for greater detection versatility as only two markers is needed for a calibration corner to be detected where a chessboard requires the detection of all of the corners.

The used charuco board can be found in the appendix. The board was generated using the board generator from [6]. The charuco board has 19 squares in the height and 28 squares in the width. Each square is 10 mm wide and the aruco markers are 7.5 mm wide. The dictionary for the charuco board is a DICT_4X4. To ensure that the paper with the board is planar, the charuco board paper is glued to a glass plate with thin spray glue. The pixel size can be used to compare the reprojection errors presented in this report with other cameras by multiplying the reprojection errors with the pixel size to get the error in μm .

TABLE 1: Table of the data used in this project. The camera module SLSI_SAK2L4SX is the rear module on the Samsung Galaxy S10 Plus. For the folds, the number in the parentheses indicate the number of images removed due to large errors

Camera	Lens	Pixel size	Focal length	Board	Resolution	# images	Subdivisions					
							Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6
Point Grey Scorpion	-	-	12	Chessboard	1600 x 1200	94	-	-	-	-	-	-
SLSI_SAK2L4SX	Telephoto	1.0 μm	-	Charuco	4032 x 3024	216	37 (0)	37 (0)	36 (0)	36 (0)	34 (2)	36 (0)
	UltraWide	1.4 μm	-	Charuco	4608 x 3456	389	65 (1)	66 (0)	65 (0)	65 (0)	64 (1)	64 (1)
	WideAngle	1.0 μm	-	Charuco	4032 x 3024	223	38 (0)	37 (0)	37 (0)	37 (0)	37 (0)	37(0)
NikonD3100	Nikon DX AF-S NIKKOR 18-55mm 1:3.5-6.6G	4.94 μm	18	Charuco	4608 x 3072	45	-	-	-	-	-	-
			22	Charuco		585	97 (1)	98 (0)	98 (0)	97 (1)	98 (0)	97 (0)
			55	Charuco		313	53 (2)	48 (7)	55 (0)	53 (3)	53 (2)	52 (3)
			300	Charuco		523	91 (5)	88 (8)	84 (12)	90 (6)	83 (13)	87 (9)

4 Parametric Model

A parametric model is calibrated for all data sets using the OpenCV library [7]. The parametric model is both used for initializing the generic model and also for comparison. In many cases, the parametric model is the industry standard, and therefore it provides a good performance goal for the generic model.

The calibration uses the charuco board as a known 3D object. First step in the pipeline is to find corners in the board images and identify them using the charuco markers. This is done with OpenCV aruco functions called: *DetectorParameters_create()*, *detectMarkers()*, *refineDetectedMarkers()*, *interpolateCornersCharuco()* and *cornerSubPix()*. Given a list of greyscale images containing a charuco board, this sequence of functions will return corners of the boards in the images. For marker detection the Otsu standard deviation was increased from the default 5 to 12 after experiencing errors. For corner refinement a *winSize* of (10, 10) is used and the refinement is continued until the change is less than 0.001 or 30 iterations are reached.

The found and refined corners can then be fed into the function *calibrateCameraCharucoExtended*, which performs the calibration and returns the calibration error, camera matrix, extrinsic parameters, distortion parameters etc. As flags for the calibration we set *cv2.CALIB_RATIONAL_MODEL*, *cv2.CALIB_THIN_PRISM_MODEL* and *cv2.CALIB_TILTED_MODEL* defining the models used for distortion.

Two models with different amounts of distortion coefficients are used, one with five considering only lower-order radial distortion and tangential distortion. The other with 14 parameters considering radial, tangential, and thin-prism distortion. The models are chosen, such that we have both the simplest and the most advanced parametric models from OpenCV to compare with the generic model.

A Root Mean Square (RMS) of the remaining reprojection error for the camera model can be calculated in the image plane, which gives an error in pixel units. Some of the images used for the calibration returned unusually high error and were sorted out as outliers using a threshold.

4.1 Results

The parametric model was used to calibrate all of the data subdivisions described in 3. An overview of the resulting RMS reprojection error is shown in table 2. In table 3 the std.dev for the intrinsic and extrinsic parameters is shown.

TABLE 2: Table of the RMS reprojection error from calibrating the parametric model on each of the folds. The mean \pm standard deviation is shown for each camera

Model	Camera						
	NikonD3100				Samsung Galaxy S10 Plus		
	18 mm	22 mm	55 mm	300 mm	Telephoto	UltraWide	WideAngle
Parametric (5)	0.7055	2.5160 ± 2.5449	0.9571 ± 0.0240	3.2616 ± 1.3953	37.2395 ± 11.4746	1.2587 ± 0.0130	26.0533 ± 7.1821
Parametric (14)	0.5650	0.9949 ± 0.0815	0.9049 ± 0.0274	2.8849 ± 1.3458	2.1222 ± 0.2000	4.6745 ± 2.8601	2.5608 ± 0.1717

TABLE 3: Table of the means of the standard deviations from calibrating the parametric model on each of the folds. The mean \pm standard deviation for the standard deviation of intrinsic-s/extrinsics is shown for each of the cameras

Model	Camera			
	NikonD3100			
	18 mm	22 mm	55 mm	300 mm
Parametric (5)	$0.1793 \pm 0.3528 / 0.0002 \pm 0.0002$	$0.4346 \pm 1.4125 / 0.0004 \pm 0.0006$	$1.7834 \pm 3.5201 / 0.0007 \pm 0.0005$	$1.1188 \pm 3.9801 / 0.0015 \pm 0.0047$
Parametric (14)	$0.4045 \pm 0.5195 / 0.0002 \pm 0.0001$	$0.6401 \pm 0.9496 / 0.0002 \pm 0.0002$	$1.8241 \pm 3.2278 / 0.0006 \pm 0.0006$	$2.8919 \pm 6.3452 / 0.0030 \pm 0.0055$
Model	Samsung Galaxy S10 Plus			
	Telephoto	UltraWide	WideAngle	
	$8.7358 \pm 16.5793 / 0.0073 \pm 0.0229$	$0.0998 \pm 0.1919 / 0.0003 \pm 0.0004$	$2.3727 \pm 4.7019 / 0.0029 \pm 0.0028$	
Parametric (5)	$1.3639 \pm 1.9315 / 0.0008 \pm 0.0015$	$1.2886 \pm 3.6737 / 0.0025 \pm 0.0035$	$2.2748 \pm 3.6077 / 0.0005 \pm 0.0004$	

From table 2 it is seen that the parametric model with 14 distortion parameters performs better on all of the NikonD3100 and Samsung data sets, except for 'UltraWide'. In table 3 it is seen that the model with the lowest standard deviation of the parameters is more divided between the 14 and 5 distortion parameter models.

To visualize the distortion present in the cameras a series of distortion plots were made, see figure 1 and 2, where the distortion is shown in magnitude and angle.

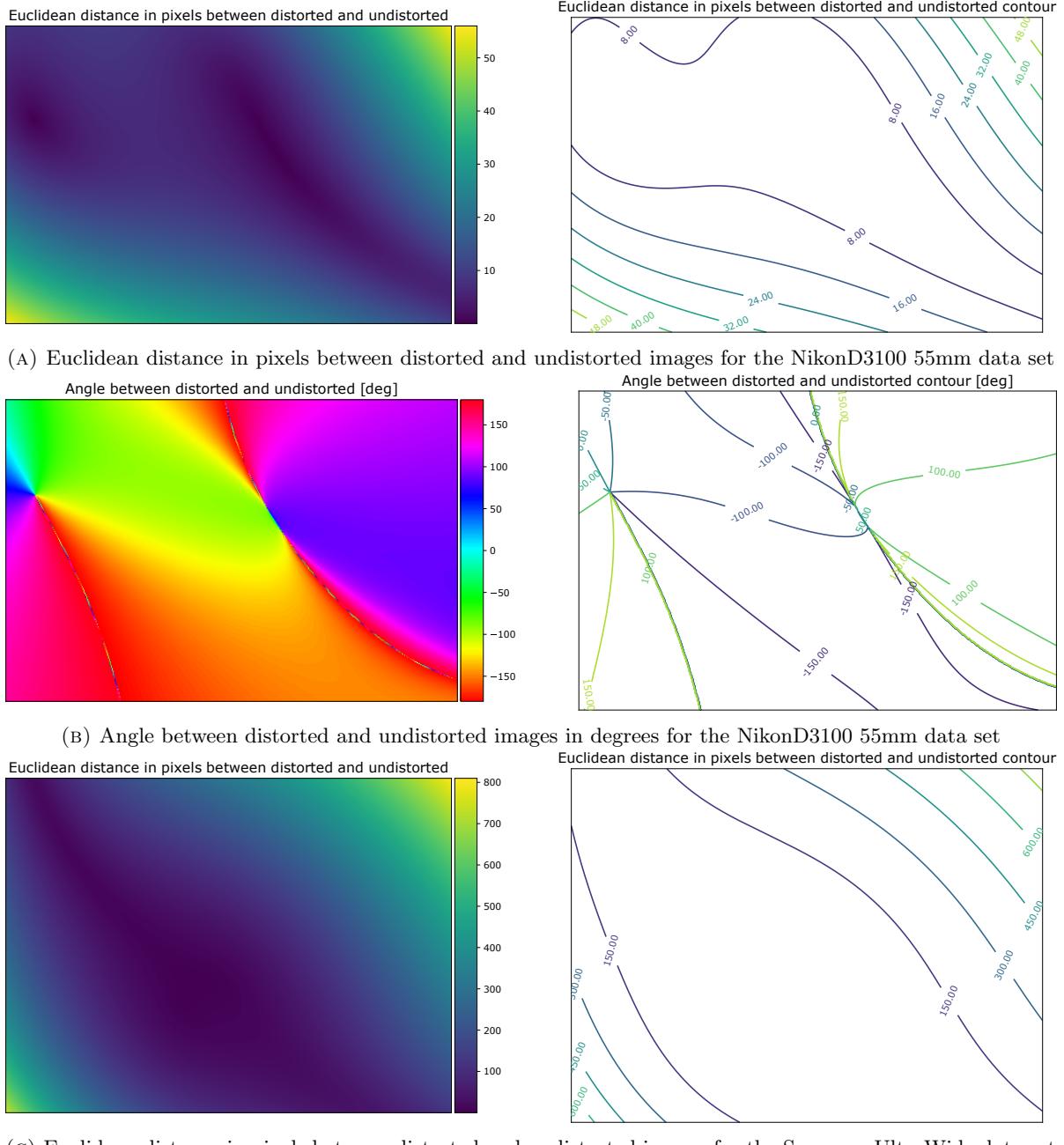
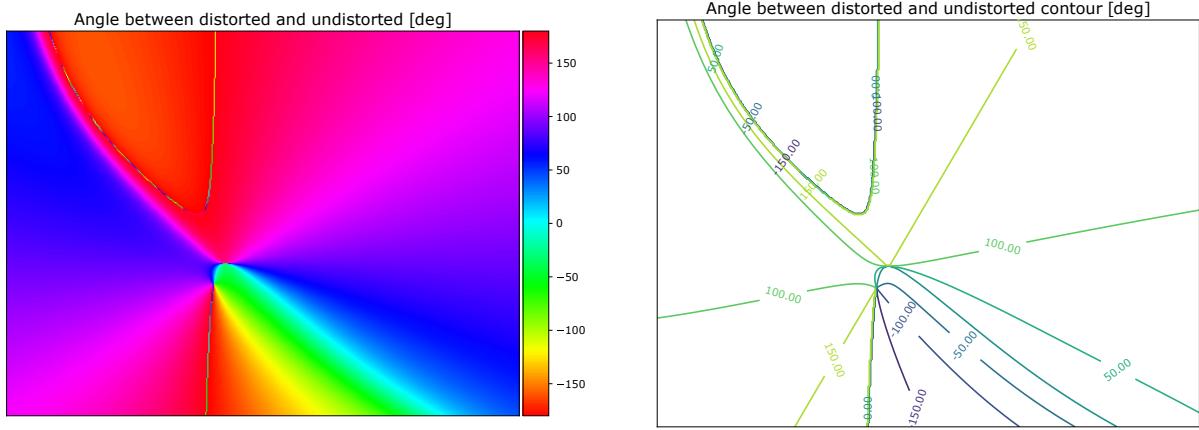


FIGURE 1: Two examples of undistortion using the parametric (14) model. Both magnitude and angle is shown. $\text{angle} = 0$ is equal to displacement to the right in the image and the degrees are measured positive in the counterclockwise direction.



(A) Angle between distorted and undistorted images in degrees for the Samsung UltraWide data set

FIGURE 2: Two examples of undistortion using the parametric (14) model. Both magnitude and angle is shown. $\text{angle} = 0$ is equal to displacement to the right in the image and the degrees are measured positive in the counterclockwise direction.

The distortion contours show that the images contain significant distortion. The largest distortion generally occurs along the perimeter of the image, which means it is likely the effect of radial distortion and a lack of features. In general, there is a high amount of distortion in the cameras. The camera with the largest distortion in the data set is the 'UltraWide' Samsung camera, which is also expected as it is a fisheye like lens with 123° field of view (FOV) [8].

The distortion contours show how the pixels are moved when they are undistorted by the model. To show the error that remains after correcting for distortion a series of Voronoi diagrams are used, see figure 3.

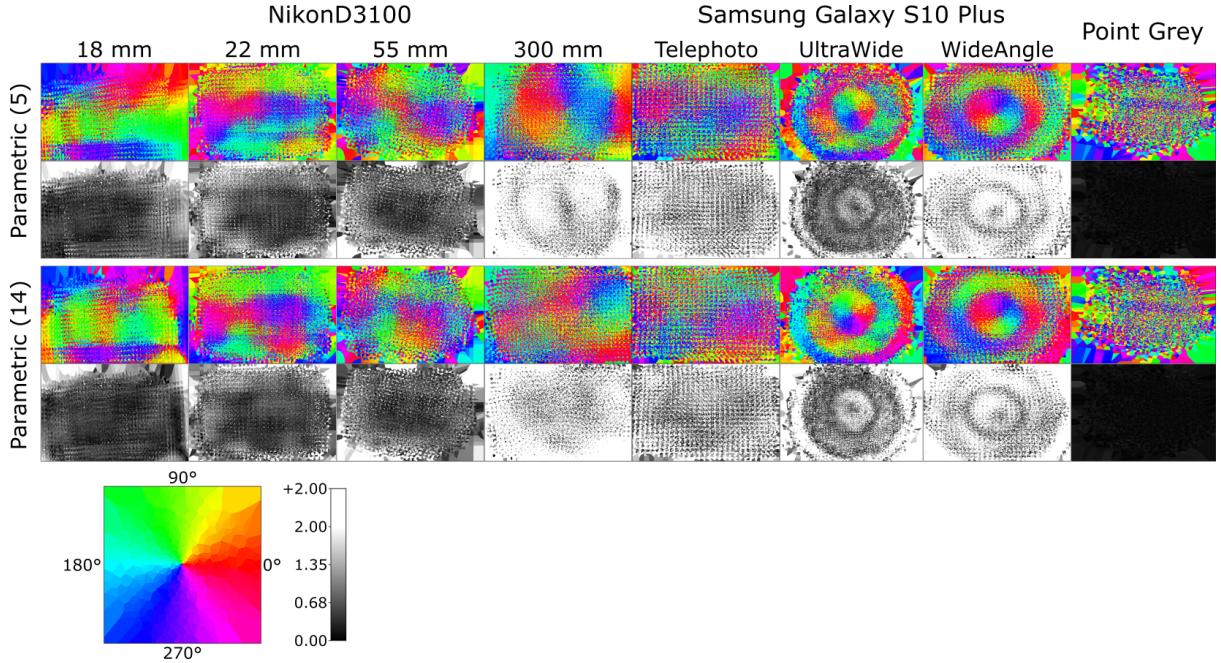


FIGURE 3: Voronoi diagram of the residual error after using parametric camera calibration on one fold of each data group presented in 1. The color diagrams show, which direction the error is pointing in the image as illustrated at the bottom of the figure. The greyscale images show the magnitude of the residuals

You can see on the overall color of the magnitude plots that the models find it harder to model some cameras than others. The patterns on the magnitude diagrams show that there are specific regions in the images, where errors are larger or smaller than in others. The 'UltraWide' and 'WideAngle' dataset have a circular pattern, which might be an artefact of radial distortion. These patterns become even clearer when you look at the angle diagrams. Any continuous region of similar color shows that the residuals are pointing in the same direction. These systematic residuals are what can bias the performance of the model. Figure 4 shows model checks for normality in the residuals in the form of a boxplot, scatter matrix, and QQ-plot.

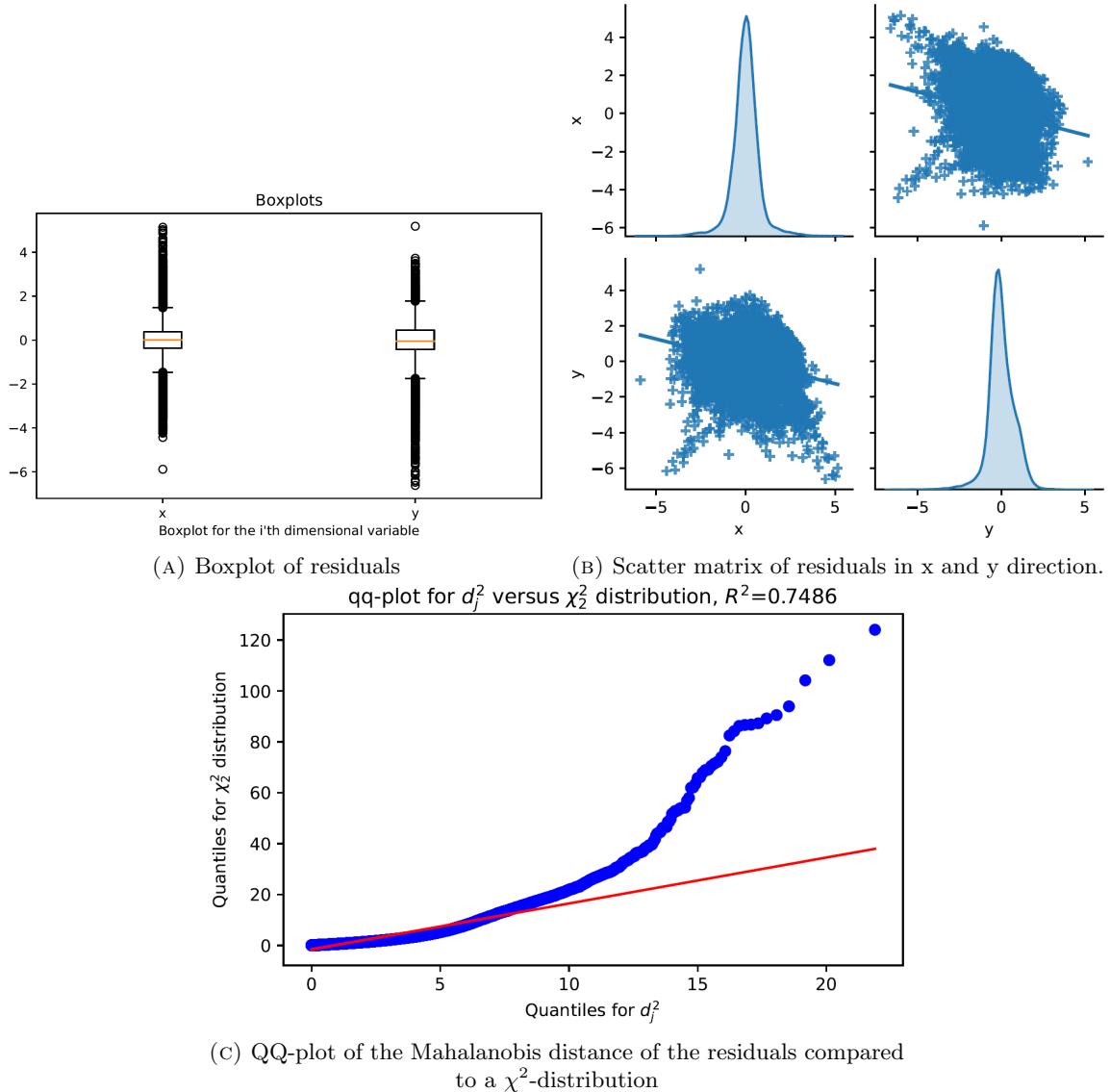


FIGURE 4: Model check of residuals of normality for the parametric model on one of the 22 mm NikonD3100 camera folds

Ideally, the residuals would be normally distributed, but based on the QQ-plot in figure 4c it does not seem to be the case. The curvature of the QQ-plot is a clear indication of skew.

In table 4 the 2D residuals in image dimensions x and y are tested for correlation.

TABLE 4: Table of the correlation test for the parametric model. cc is the correlation coefficient and p is the p-value. p-values with the value 0.0 is zero down to at least four decimals

Model	Camera													
	NikonD3100								Samsung Galaxy S10 Plus					
	18 mm		22 mm		55 mm		300 mm		Telephoto		UltraWide		WideAngle	
	cc	p	cc	p	cc	p	cc	p	cc	p	cc	p	cc	p
Parametric (14)	-0.0875	0.0	-0.2105	0.0	-0.0395	0.0	-0.0576	0.0	-0.0680	0.0	-0.0449	0.0	0.0290	0.0002

From table 4 it is seen that the residuals of the parametric model are not correlated.

5 Generic Central Model

To improve on the general performance of the parametric model described in chapter 4, a generic model as the one described in articles [1] and [3] is implemented. This generic model is a model with an analytical definition of the backward projection. This backward projection is modeled by a b-spline surface, mapping between input coordinates in pixels and outgoing rays.

5.1 B-Spline

A b-spline is a method of generating a smooth surface based on a set of points called control points, a . The values of the control points 'pull' the curve depending on the position and order of the points. To calculate the weighing of the points for any given input, basis functions are used.

The basis functions can be seen in figures 5b and 5a and map between the location in which the spline is sampled ($[0, 1]$) and the weight of the corresponding control point. In a unidimensional input space, there is one basis function for each control point deciding the weight of that point. In this project, a two-dimensional input (pixel coordinate) is used, meaning that there are two sets of basis functions, one containing a function for each column and one containing a function for each row. The basis functions are defined by the order of the spline, k , and the knot vectors, t .

The knot vector is a vector containing values that relate the control points of the spline to the input. In this project, two knot vectors are used as the input space is two-dimensional. This means there is a horizontal and a vertical knot vector. The only requirement for a knot vector is that the values inside must be ordered in ascending order, the values themselves determine the distribution of the control points across the spline.

There are several methods for creating a knot vector. Firstly, in this project the knot vector is uniform. This means that the separation of the control points is uniform in the input-space, ie. the control points are evenly distributed on the image plane. This does not necessarily mean that the control points are spread with the same distance in the u- and v-direction, however. The method for knot generation used for this project is called

'open uniform'. It is made by 'bunching up' k additional knots at the ends of the knot vector, where k is the order of the spline. This allows the spline to vary its value and derivative at the edges of the spline as opposed to an ordinary 'uniform' method. This can be seen in figure 5.

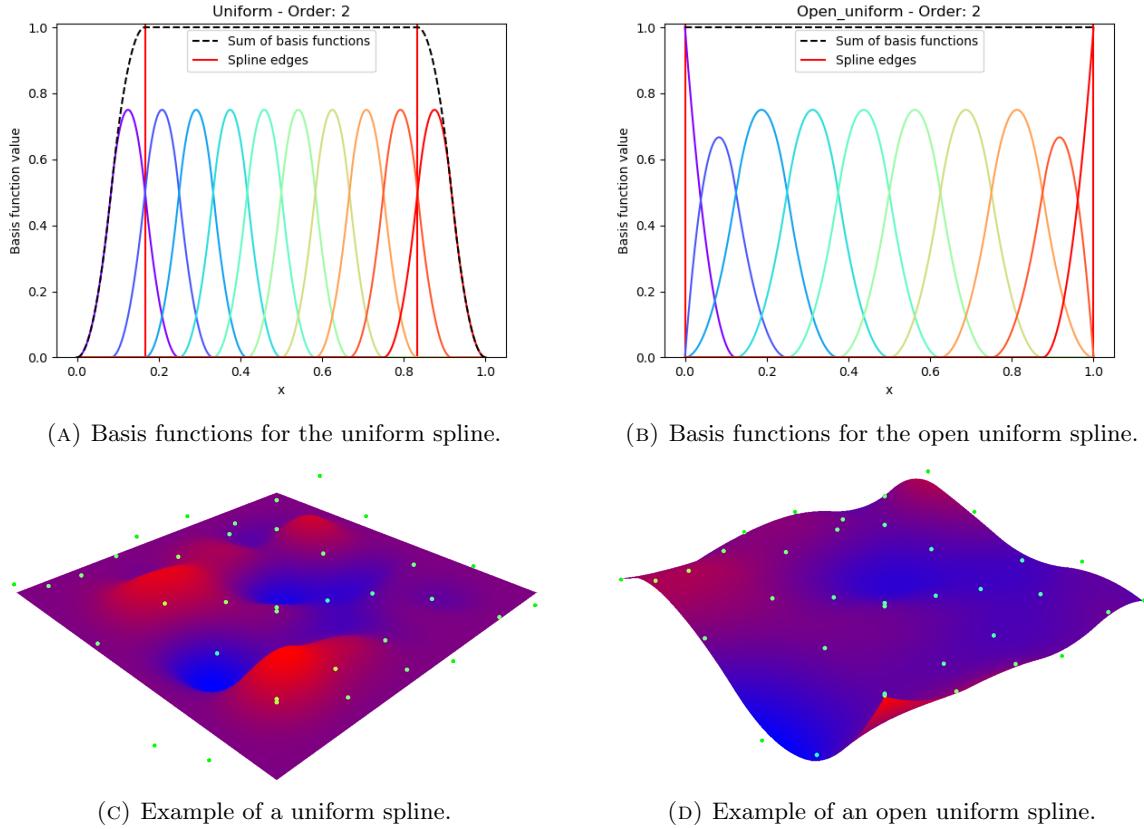


FIGURE 5: Examples of methods of knot vector generation on identical control points.

As shown in figure 5b, the basis functions sum up to one for any input value (defined between 0 and 1) in the open uniform method. As the basis functions weigh the control points, this means that the uniform method has a constraint along the edge of the surface as opposed to the open uniform method. This also means that the absolute height of the control points matter in the uniform spline whereas only the relative height of each control point matters in the open uniform spline. Because of this, to avoid issues with the edge constraint and to allow each control point an approximately equal effect on the shape of the spline, the open uniform method of knot generation is used.

An important parameter of the spline is the order. The order, k , decides the smoothness of the spline, generally C^{k-1} [3]. The order changes the amount of overlap in the basis

functions, ie., how many control points are used to calculate a sample of the spline. This increases the computational complexity of the spline quadratically, as the input space is two-dimensional.

A technical issue with the open uniform knot generation is caused by the mathematical expression of the spline. Equation 1 is used for calculating a sample for the normalized input image coordinates (u, v) .

$$S(u, v) = \sum_i \sum_j a_{i,j} B_{i,k}(u) B_{j,k}(v) \quad (1)$$

The control points contained in a are weighted by the corresponding horizontal and vertical basis values and summed. The expression for the basis function for the i -th control point, $B_{i,k}$, can be seen in equations 2 and 3.

$$B_{i,0}(x) = \begin{cases} 1 & \text{if } t_i \leq x < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$B_{i,k}(x) = B_{i,k-1} \frac{x - t_i}{t_{i+k} - t_i} + B_{i+1,k-1} \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} \quad (3)$$

These equations directly relate the knot vector, t , to the input coordinate as this is used to calculate at which location the spline is being sampled. The problem is if the denominators of the fractions are zero, which they are at the edge of the spline, resulting in the spline being undefined at the edge. A solution to this is the introduction of a term that makes the knot values in the ends of the vector deviate slightly from one another. e.g. the order 2 knot vector $[0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1]$ could be changed to $[-0.002, -0.001, 0, 0.2, 0.4, 0.6, 0.8, 1, 1.001, 1.002]$, negating the problem. In the implementation of the generic model, this term is called *end_divergence* and defaults to $1e-10$.

5.1.1 Using a B-Spline as a Camera Model

A camera model is a mathematical model used to describe the projection between real-world coordinates and pixel coordinates in the image plane. One of the most simple camera models is the pinhole camera model. In this model, light rays from the world

are gathered in a single point and then projected onto the image plane. This means that each light ray passes through the same point, the pinhole, with a unique direction before hitting the image plane in a unique position. The mapping between the direction and the position is described by the camera matrix, K , by the location of the pinhole, and the focal length of the camera. Calculating the pixel positions of a ray is called forward projection while the inverse is called backward projection.

In the generic model, the b-spline defines the backward projection, ie. it takes pixel coordinates and calculates the direction of the corresponding ray. An illustration of samples from the model can be seen in figure 6.

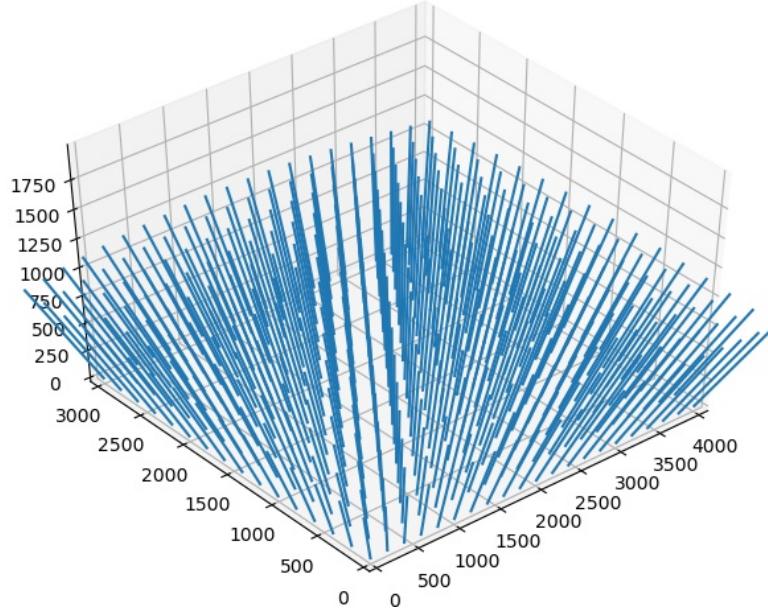


FIGURE 6: A grid of samples from the generic model. The x-y plane is the image plane while the lines show the corresponding direction.

This means that the backward projection is analytically defined, but since a b-spline is not easily invertible, using the model for forward projection is non-trivial. Since the models tend to be smooth, the forward projection is calculated as an optimization problem searching to minimize the difference between any given ray and a sample from the spline at any point. The point that minimizes the difference will be the result of the forward projection. This means that the forward projection will be some factor slower to compute than the backward projection, as the forward projection includes the backward projection in the computation.

A b-spline has multiple useful properties when used as a camera model. The first, most obvious, property is the adjustable amount of control points in the spline. The amount changes the fidelity of the spline, allowing it to model variations with higher frequencies. If the model tends to overfit or does not converge, it can be easily simplified by reducing the fidelity.

Another property is the C^{k-1} continuity that allows the surface to stay continuous after taking multiple derivatives. This is useful when performing the bundle adjustment, as the Hessian is used to guide the parameter search, ensuring a continuous and smooth search space for the bundle adjustment.

5.2 Bundle Adjustment

To calibrate the generic model, bundle adjustment is performed. The objective of the bundle adjustment is to tune the parameters to reduce the residuals. The parameters, in the case of the generic model, are the extrinsics of the checkerboards and the control points of the b-spline.

5.2.1 3D Residuals

As forward projection is much more computationally expensive than backward projection, calculating the residuals in 2D space is not feasible. Instead, the residuals are calculated in 3D. This is done by estimating two rays and finding the difference. The first ray is the direction of any given object point from the camera center while the other ray is backward projected by sampling the b-spline at the pixel coordinates of the corresponding feature. A flowchart illustrating the process of calculating the 3D residuals can be seen in figure 7.

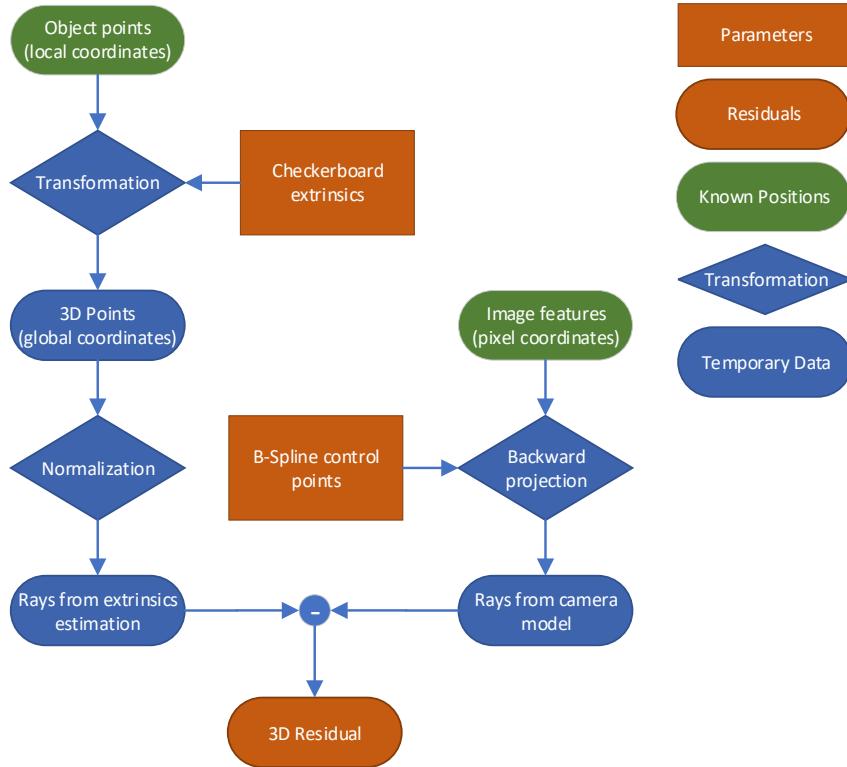


FIGURE 7: Calculation of the 3D residual.

For each checkerboard feature detected, a 3D residual vector is calculated. This results in a total of $3 \sum_{i=1}^n m_i$ residual values for the bundle adjustment to optimize, where n is the number of images and m_i is the number of features detected in the image. These residual values are reduced to one RMS value which determines the progress of the calibration process.

5.2.2 Model Initialization

Bundle adjustment generally requires a good initial value for the parameters. For the extrinsics of the checkerboards, the extrinsics from the parametric (14) calibration described in chapter 4 are used. To initialize the control points of the generic model, a grid of rays is calculated using the intrinsic parameters and distortion coefficients found in chapter 4. The model is then fitted to these rays in a small-scale, more constrained bundle adjustment. This step is referred to in the code as the pre-fit. A simplified (uni-dimensional output) example of the pre-fit can be seen in figure 8.

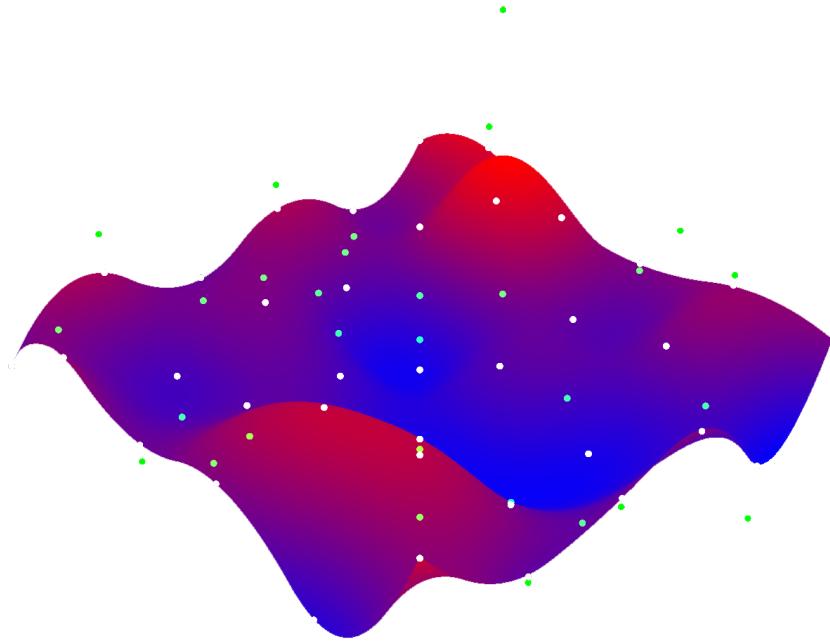


FIGURE 8: The spline is fitted to specific values (white) resulting in the control points shown (green). A video can be found on [9]

5.2.3 Sparsity Matrix

To improve the speed of the bundle adjustment, a sparsity matrix is constructed. The matrix relates specific parameters to specific residuals, which aids the least squares optimization in the search by describing the sparsity of the problem. For the mapping between residuals and checkerboard extrinsics, this is simple. Each image has one checkerboard with 6 parameters, 3 describing the 3D position, and 3 describing the orientation. This results in the two diagonals seen on the right side of figure 9. The sparsity of the control points is slightly more difficult to compute. Firstly, the subset of control points used to calculate any value is a square with the side length $k + 1$. To find which control points are used to calculate any residual, this square is found for the pixel coordinates of the residual. This results in the more complex pattern seen in the left part of figure 9. For building an intuition for the sparsity of the control points, see figure 5b.

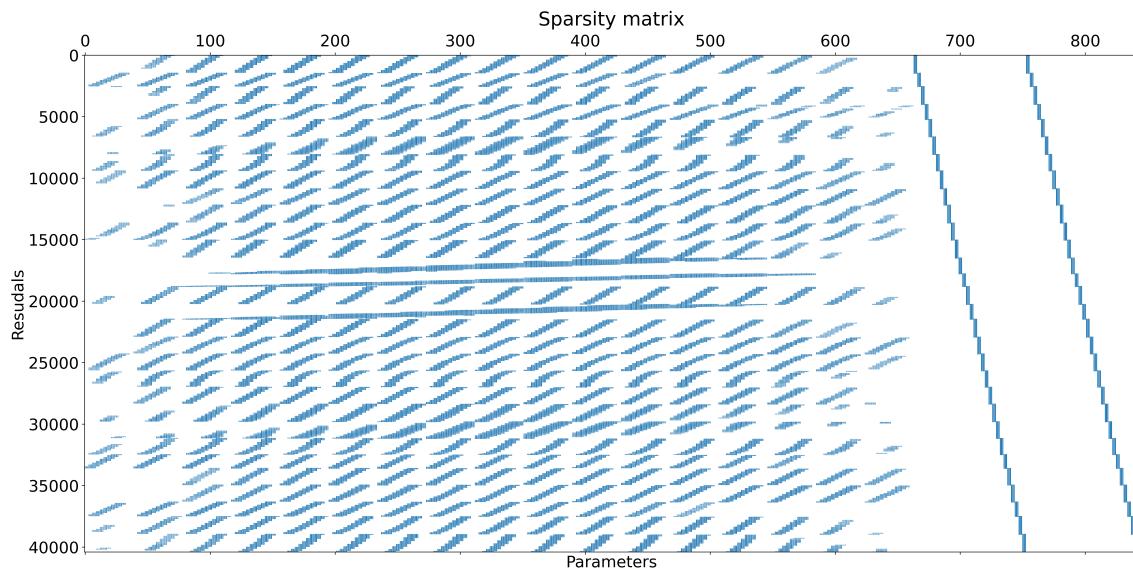


FIGURE 9: Example of the structure of the sparsity matrix.

5.2.4 2D Residuals

To be able to compare the results of the generic model to the parametric model described in chapter 4, the residuals have to be comparable. As 2D pixel-based residuals are the most commonly used residuals, these are used for the comparison. This is done by calculating the 2D residuals of the model after it has been calibrated using the bundle adjustment. A flowchart illustrating the calculation of the 2D residuals can be seen in figure 10.

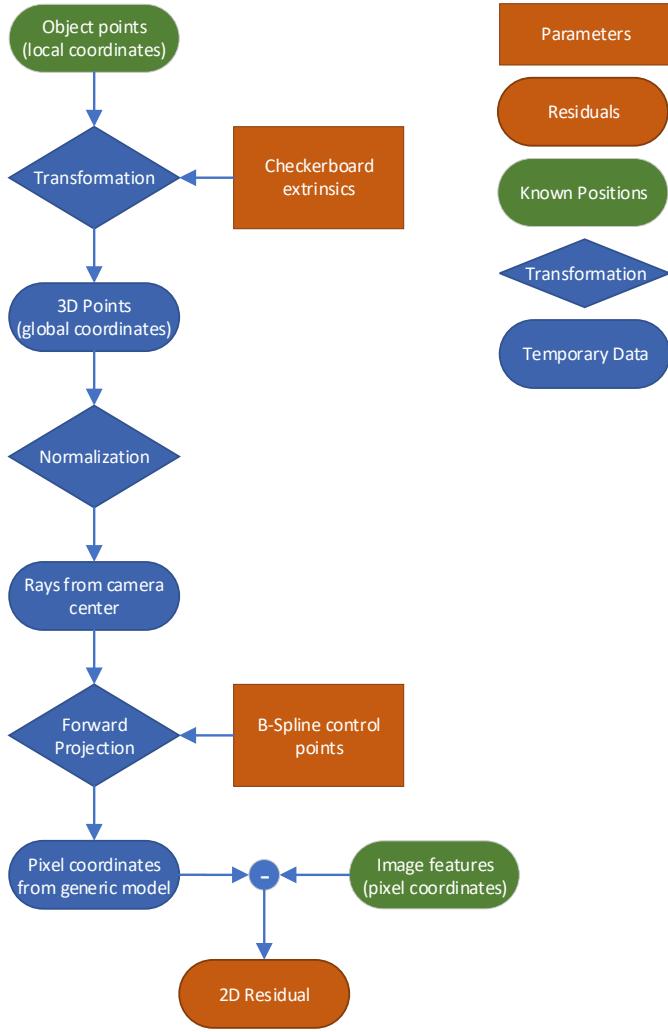


FIGURE 10: Calculation of the 2D residual.

5.3 Results

The generic model requires several parameters to define how the backward projection is calculated. Two of the most important parameters are the order and the step size. The step size is the distance between each control point in the image plane. To find the values of these parameters, a small parameter search was conducted on a subset of the images from the Samsung camera in 'WideAngle' mode. Orders 2, 3, and 4 were tested with different step sizes. The results from the test are shown in figure 11.

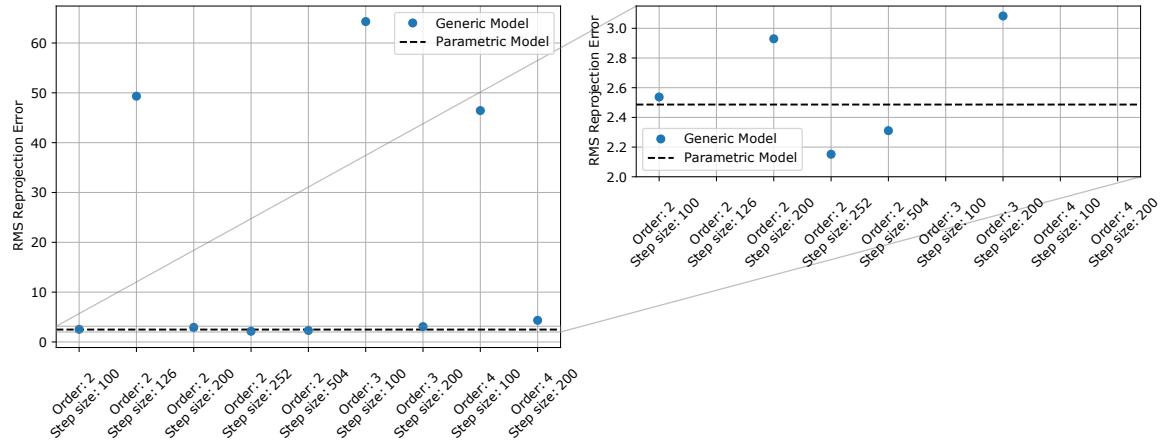


FIGURE 11: Reprojection error for calibration with different b-spline settings. Calibration performed on images from the Samsung 'WideAngle' data set, see table 1.

From figure 11 it is seen that an order of 2 with a step size of 252 achieved the best performance. The run time for the parametric model as well as all the generic models are shown in figure 12.

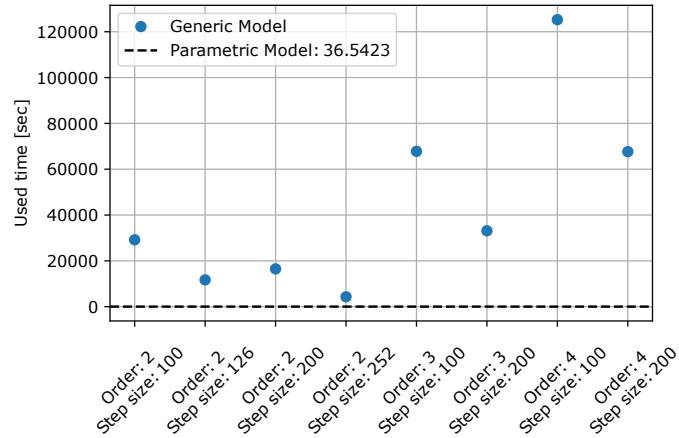


FIGURE 12: Run time for the test displayed in figure 11

The parametric model is significantly faster than any of the generic models. The run time of the generic models increase with higher order and lower step size.

To test how the generic model performs with these parameters compared to the parametric model, both were used to calibrate the different cameras. For this calibration, all the data subdivisions described in section 3 were used.

In table 5 the RMS reprojection errors for the NikonD3100 camera and the cameras on the Samsung phone are shown and the improvement is shown in figure 13.

TABLE 5: Table of the RMS reprojection error for the different cameras. The Generic Model uses a order of 2 and 252 in step size

Model	Camera					
	NikonD3100			Samsung Galaxy S10 Plus		
	22 mm	55 mm	300 mm	Telephoto	UltraWide	WideAngle
Parametric	0.9949 ± 0.0835	0.9049 ± 0.0274	2.8849 ± 1.3458	2.1222 ± 0.2000	4.6745 ± 2.8601	2.5608 ± 0.1717
Generic	1.2699 ± 0.1720	0.9608 ± 0.0507	2.9292 ± 1.4327	2.0472 ± 0.1938	3.8802 ± 1.8355	2.3219 ± 0.1449

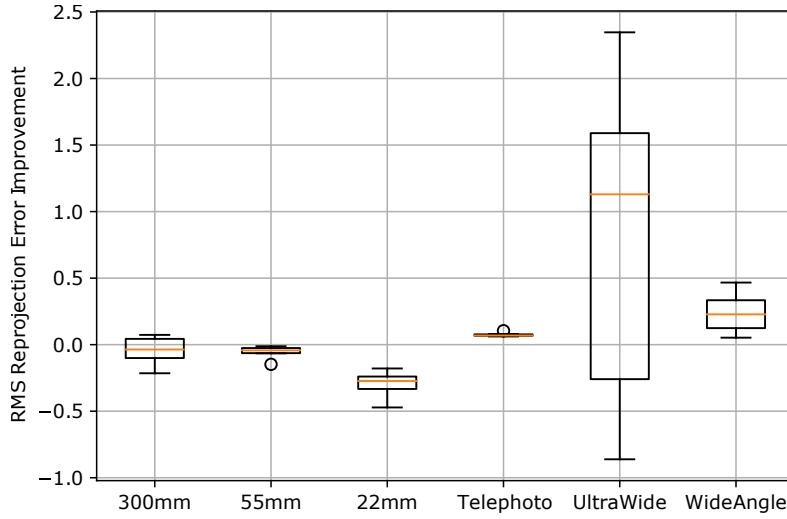


FIGURE 13: RMS reprojection errors for the NikonD3100 camera and the cameras on the Samsung phone.

From table 5 it is seen that the generic model, in general, was able to improve the RMS reprojection error for all of the cameras on the Samsung phone, while also achieving a lower standard deviation. However, the generic model was not able to improve all the reprojection errors. From figure 13 it is seen that in all the folds for the 'Telephoto' and 'WideAngle' camera on the Samsung phone the generic model was able to improve the reprojection error. For the 'UltraWide' camera the model was generally able to improve the reprojection error, however, some of the folds still had a higher reprojection error with the generic model. For both the 55 mm and 22 mm zoom on the Nikon camera, the generic model had a higher reprojection error than the parametric model. In general, the 300 mm zoom also had a higher reprojection error with the generic model, but some of the folds achieved a lower error than with the parametric model. From figure 13 it is also seen that the camera with the best improvement is the camera that was used to find the parameters for the b-spline.

To test whether another step size could perform better with the other cameras a parameter search using order 2 with 3 different step sizes was made. The order of 2 was chosen, as it was seen in figure 11 that higher orders, in general, achieved higher reprojection errors. The results can be seen in table 6.

The Voronoi diagrams for the parametric and the generic models are shown in figure 14.

TABLE 6: Table of the RMS reprojection errors from calibrating the generic model on one fold with multiple step sizes. The RMS reprojection error for the parametric model calibrated using the Scorpion data set is 0.0926.

Step size	Camera								Scorpion	
	NikonD3100				Samsung Galaxy S10 Plus					
	18 mm	22 mm	55 mm	300 mm	Telephoto	UltraWide	WideAngle			
200 (80*)	-	1.1273	0.9872	4.6522	2.2107	1.9033	3.2341	0.0964*	Scorpion	
250 (100*)		1.0541	4.8034	2.3716	2.2817	1.5686	3.5467	0.0926*		
300 (120*)		1.1597	2.9044	2.3718	2.3202	1.9827	3.4994	0.2365*		

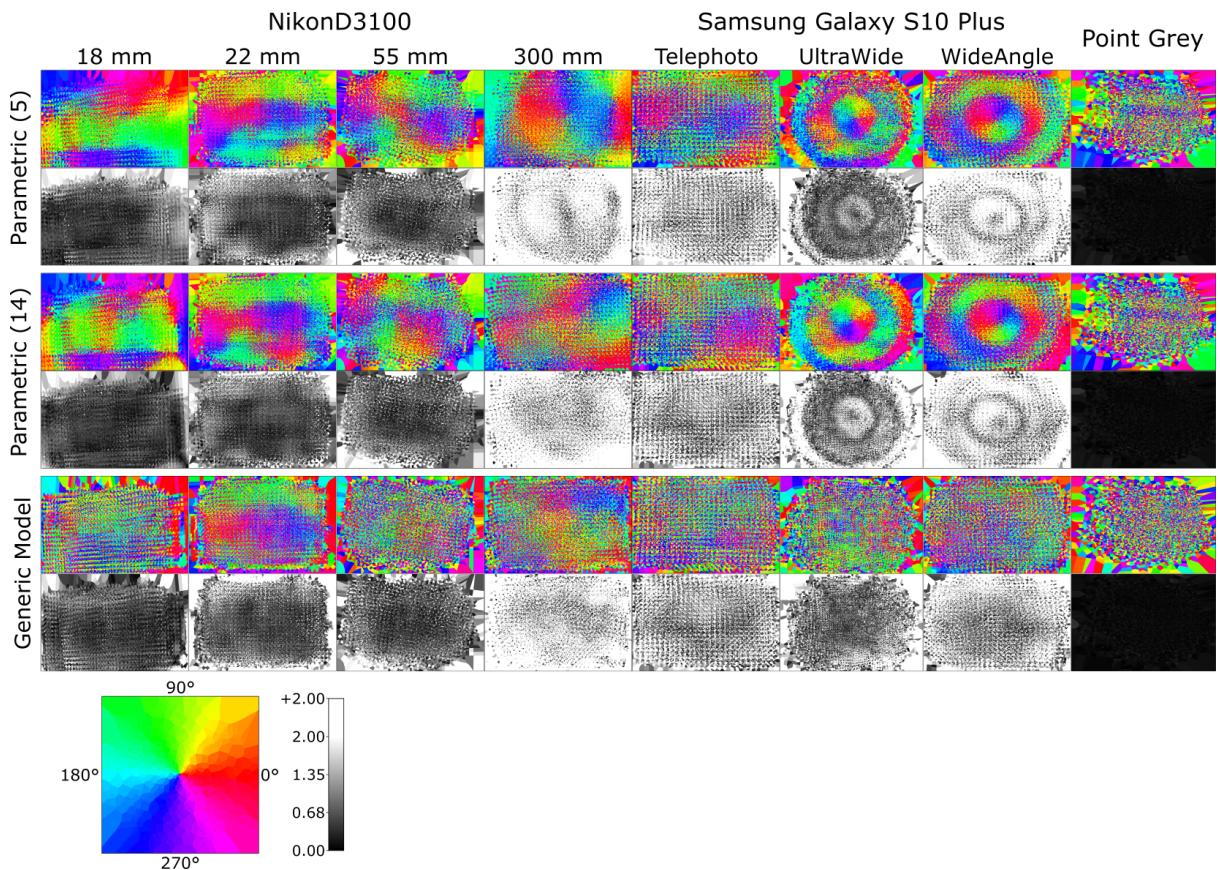


FIGURE 14: Voronoi diagram of the residuals after performing parametric and generic camera calibration on one fold of each data group presented in table 1. The color diagrams show which directions the residuals are pointing in the image as illustrated at the bottom of the figure. The greyscale images show the magnitude of the residuals.

From the Voronoi diagrams, it can be seen that the generic model generally reduces the systematic error as there is a reduction in the number of patterns. The cameras 'UltraWide' and 'WideAngle' had a clear pattern with the parametric model, whereas with the generic model the clear pattern was removed.

A model check of normality of the residuals for the generic model is shown in figure 15.

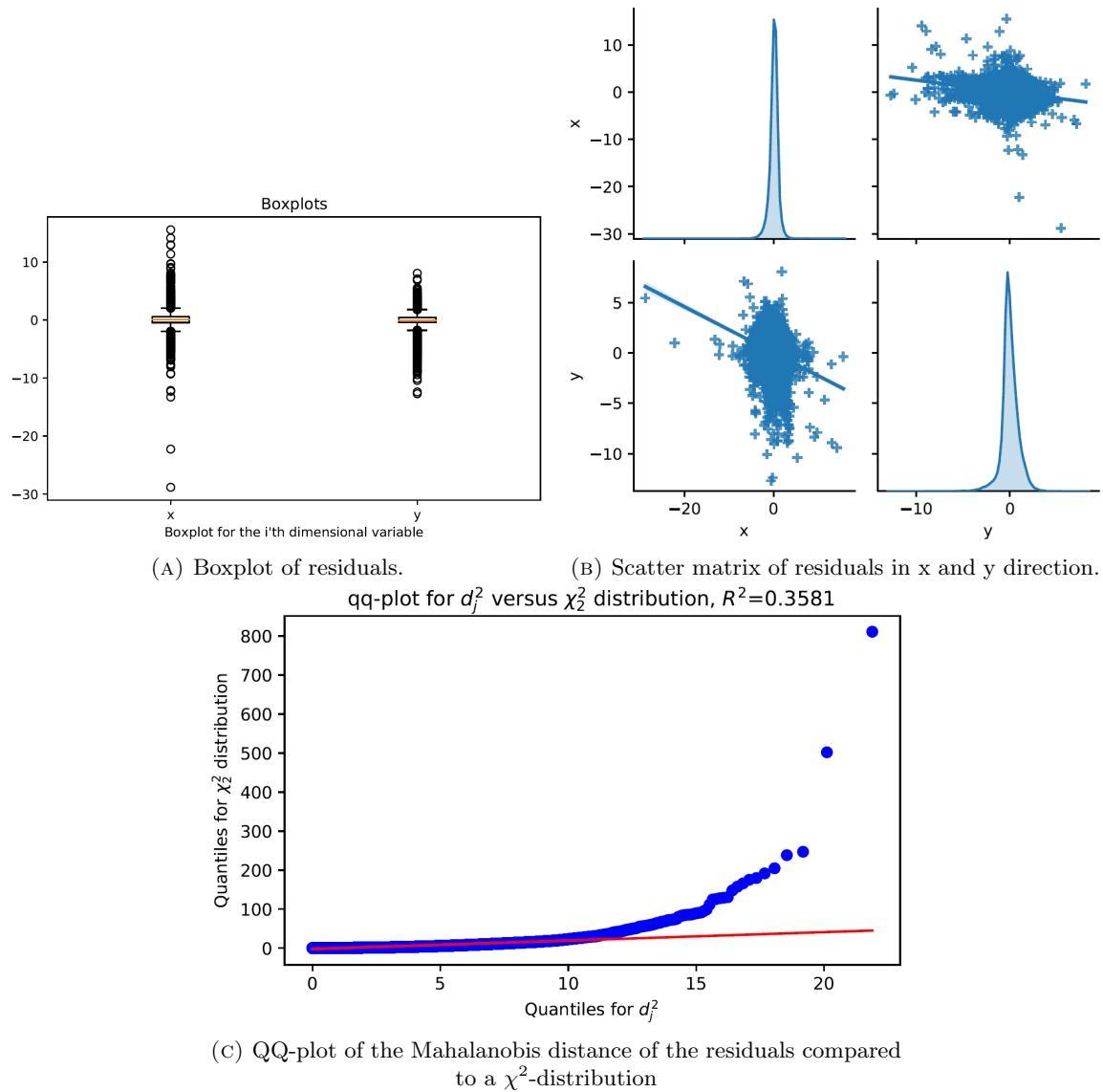


FIGURE 15: Model check of residuals of normality for the generic model on one of the 22 mm NikonD3100 camera folds

The QQ-plot shows that the residuals are not normally distributed, but they do look more Gaussian than for the parametric model.

In table 7 the 2D residuals in x and y are tested for correlation for both the parametric and the generic model.

TABLE 7: Table of the correlation test for the generic model. cc is the correlation coefficient and p is the p-value. p-values with the value 0.0 is zero down to at least four decimals.

Model	Camera													
	NikonD3100								Samsung Galaxy S10 Plus					
	18 mm		22 mm		55 mm		300 mm		Telephoto		UltraWide		WideAngle	
	cc	p	cc	p	cc	p	cc	p	cc	p	cc	p	cc	p
Parametric (14)	-0.0875	0.0	-0.2105	0.0	-0.0395	0.0	-0.0576	0.0	-0.0680	0.0	-0.0449	0.0	0.0290	0.0002
Generic	-0.0330	0.0	-0.2411	0.0	-0.1134	0.0	-0.0379	0.0	-0.0568	0.0	-0.0365	0.0	0.0299	0.0001

From table 7 it is seen that neither the parametric nor the generic model's residuals in x and y are correlated.

6 Discussion

In terms of overall accuracy of the generic model does not seem to offer any significant advantage over the parametric models. Based on the RMS reprojection errors found in figure 13 their performance seem roughly similar and the best performing model varies between cameras. If the goal is to minimize this error the choice of model would probably depend on the camera.

The difference between the two models is clearer in terms of the distribution of the residuals as seen in figure 14. Here the generic model seems to have a reduced systematic error across all the different cameras. In short, the generic model seems to offer similar size residuals but distributed better across the image. This comes at the cost of both training time and also complexity in the algorithm. Figure 11 shows clearly that the design of the b-spline affects the performance of the model drastically and any useful calibration would require finding suitable parameters for the b-spline.

As seen in figure 11, the central models with an order of two and a higher step size seem to perform better than the ones with higher order and lower step size. Our hypothesis as to why is that it gets stuck in local minima while performing the forward projection. Since forward projecting with the generic model is not easily solved analytically, it creates an optimization problem that has to be solved numerically and can get stuck in local minima, which can result in the algorithm returning a wrong pixel coordinate.

As seen in figure 12, the generic calibration is much slower than the parametric calibration. However, the time it takes can be difficult to predict, since it might converge faster or slower depending on whether the step size and order is a good match with the residuals. This is also why the model with a step size of 126 is faster than the model with a step size of 200.

The generic model is recommended if calibration time is not of the essence, but removing bias and increasing precision is. Only backward projection is recommended, not forward projection since this cannot be done in real-time and can result in large errors unless an analytical solution can be calculated.

7 Conclusion

When the training converges without seemingly reaching a local minimum, the performance of the generic model seems to be either comparable or a slight improvement to the parametric models based on the difference in RMS errors of the two methods. This can be seen in figure 13. When looking at figure 14, less systematic error can be observed in the residuals of the generic model bundle adjustment, both in the angle and the magnitude. Although neither model seems to have fully normally distributed residuals the residuals from the generic model are much closer to being Gaussian, as seen in figure 15.

8 Future Work

8.1 Initialization of the Forward Projection

As the forward projection optimization can give bad results in the case of local minima, it might be beneficial to do multiple different initializations. At the moment one initialization is used: the center of the camera. An idea could be using the center and then n additional either randomly or uniformly distributed points in the image plane. This would of course require additional search time depending on the number of points. Another idea could be storing forward projected results and then initializing in the closest match in the history. This would require extra memory for each generic model depending on the size of the history. These ideas could be combined to have a semi-intelligent initialization search.

8.2 Multi-scale Bundle Adjustment

To address the instability in training higher fidelity models as described in chapter 6, one could use a multi-scale approach. This means training a model with fewer parameters to ensure convergence and then initializing a model with more parameters with the values from the simpler model. This initialization could be done by interpolating between the control points of the simple model to find the corresponding control points of the more complex model. This process can be repeated to improve the initialization of the complex models, hopefully allowing them to converge.

8.3 Test for Overfitting

To ensure that the calibration models do not overfit, especially since the generic model has many degrees of freedom, but a limited number of images to calibrate on. This could be done by removing some feature points before calibrating the model, then testing the calibrated model on these afterward.

8.4 General Optimization

Optimizing the general performance of the training, backward projection and forward projection of the model would be essential if it had to be used for any real-time application.

8.5 Improving Feature Detection

The feature detection of the charuco images could be improved. Several images had to be discarded because of errors. Improved detection would increase the data set which would, hopefully, increase the general performance of the models.

9 Bibliography

- [1] T. Schöps, V. Larsson, M. Pollefeys, and T. Sattler, “Why Having 10,000 Parameters in Your Camera Model is Better Than Twelve”, 2019. arXiv: [1912.02908](https://arxiv.org/abs/1912.02908).
- [2] S. Ramalingam and P. Sturm, “A Unifying Model for Camera Calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, 2017, ISSN: 01628828. DOI: [10.1109/TPAMI.2016.2592904](https://doi.org/10.1109/TPAMI.2016.2592904).
- [3] J. Beck and C. Stiller, “Generalized B-spline Camera Model”, *2018 IEEE Intelligent Vehicles Symposium (IV)*, no. Iv, 2018.
- [4] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and A. H.l, *Dtu calibration image dataset*, http://roboimagedata.compute.dtu.dk/?page_id=36.
- [5] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and A. H., “Large scale multi-view stereopsis evaluation”, in *CVPR’14*, 2014.
- [6] *Calib*, <https://calib.io/pages/camera-calibration-pattern-generator>, Calib.io I/S.
- [7] G. Bradski, “The OpenCV Library”, *Dr. Dobb’s Journal of Software Tools*, 2000.
- [8] *Samsung galaxy s10 plus camera specs*, <https://www.samsung.com/global/galaxy/galaxy-s10/camera/>, Samsung.
- [9] *B-spline post prefit*, <https://www.youtube.com/watch?v=6fKIYZomn1M>.

1 Appendix

1.1 Division of Labor

	Report	Implementation
1 Introduction	Malte	-
2 Previous Work	Malte	-
3 Data	Simon	Simon
4 Parametric Model	Malte	Simon
4.1 Results	Simon	Simon
5.1 B-Spline	Oliver	Oliver
5.2 Bundle Adjustment	Oliver	Jakob
5.3 Results	Jakob	Simon
6 Discussion	Jakob	-
7 Conclusion	Oliver	-
8 Future Work	Oliver	-
Voronoi diagrams	Simon	Simon

1.2 GitHub

A GitHub repository has been created to host the code files and data. It can be found on <https://github.com/SimonLBSoerensen/Flexible-Camera-Calibration>. The files on the GitHub are identical to the ones in the digital appendix delivered to the supervisor. The repository is separated into separate programs, for which a short explanation can be found in the README file.

1.3 Normality for the Scorpion camera

Figures 16 and 17 show model checks for normality for the data set from the Scorpion camera.

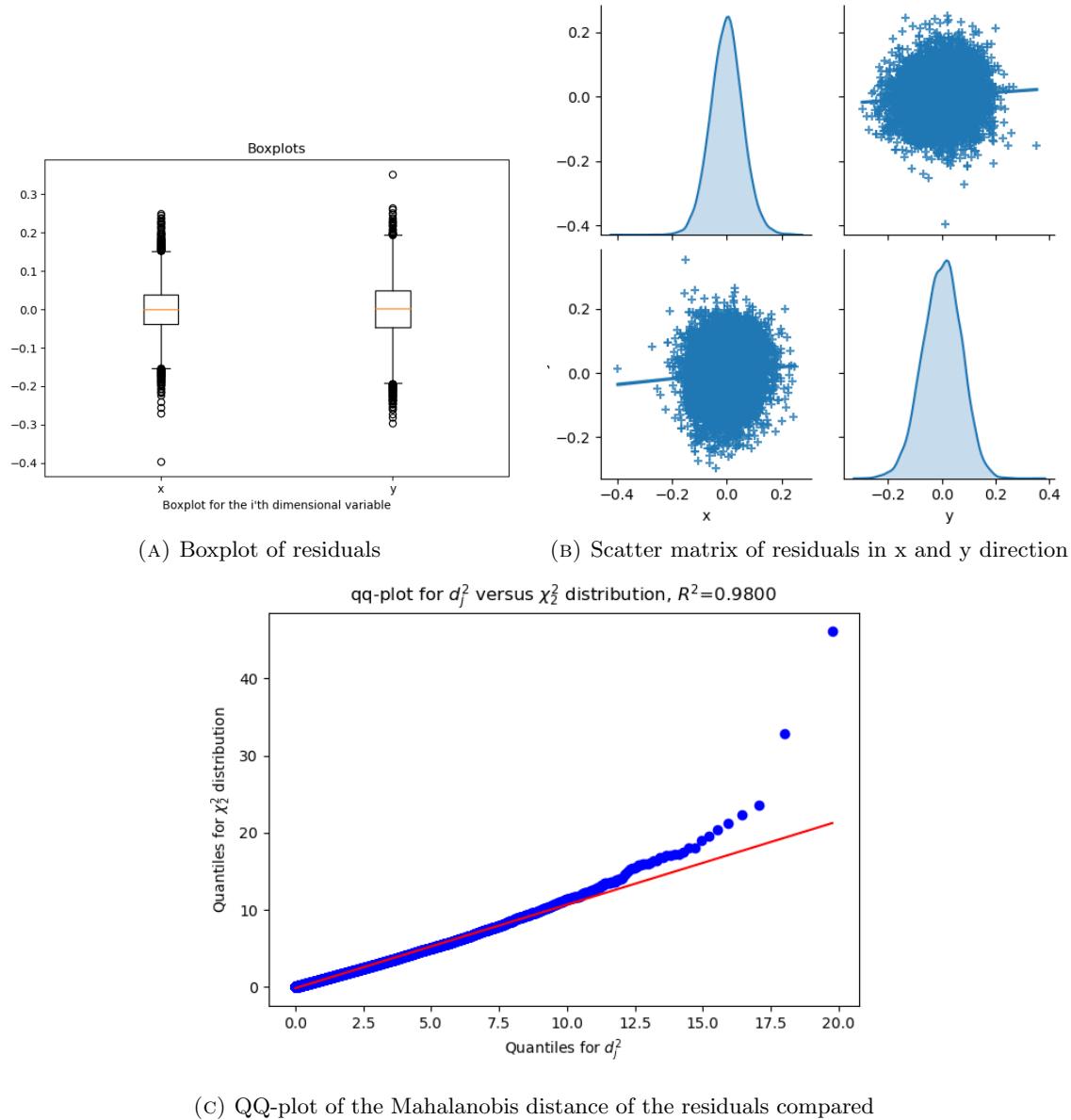


FIGURE 16: Model check of residuals of normality for the parametric model on the Scorpion data set

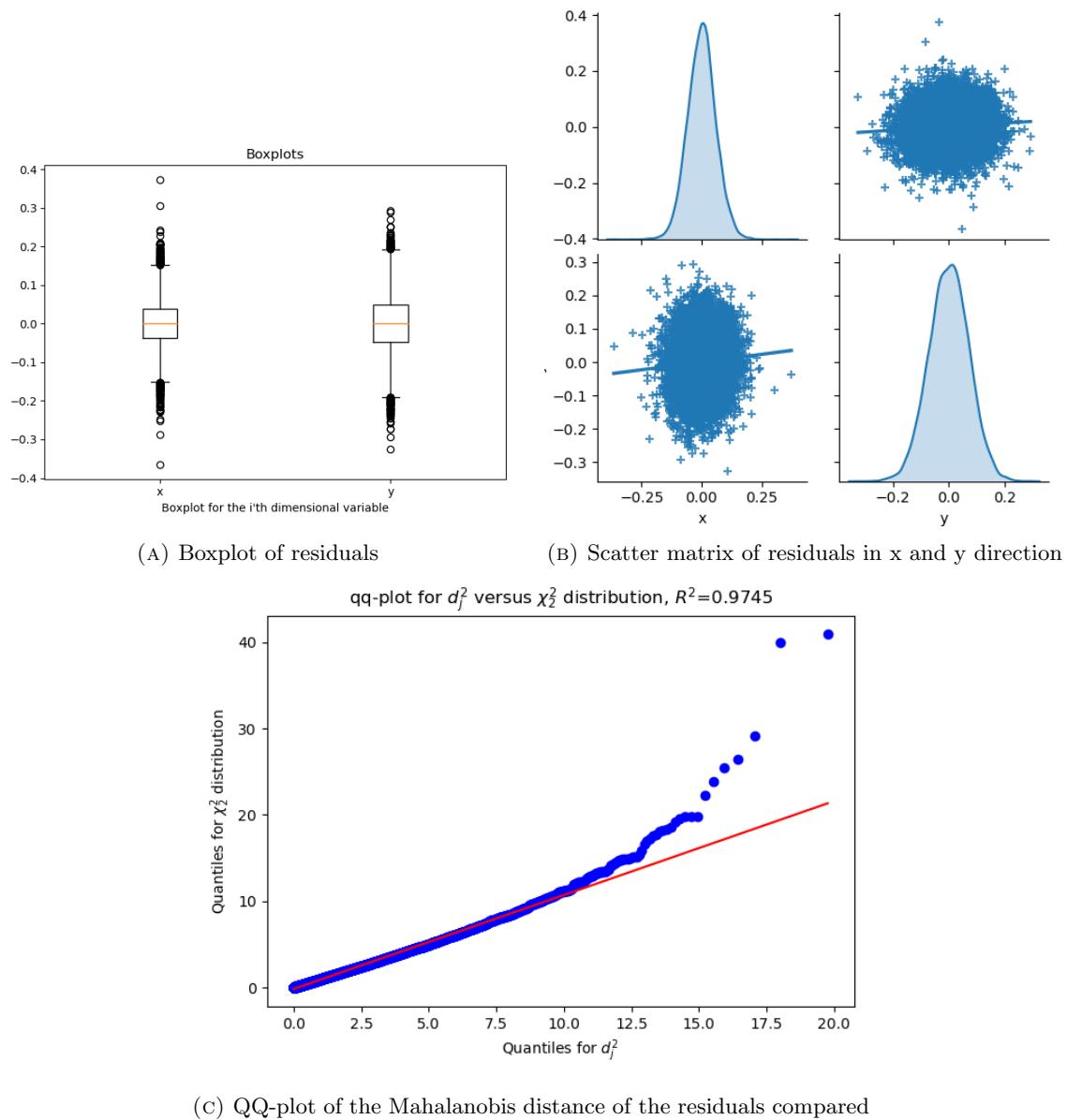
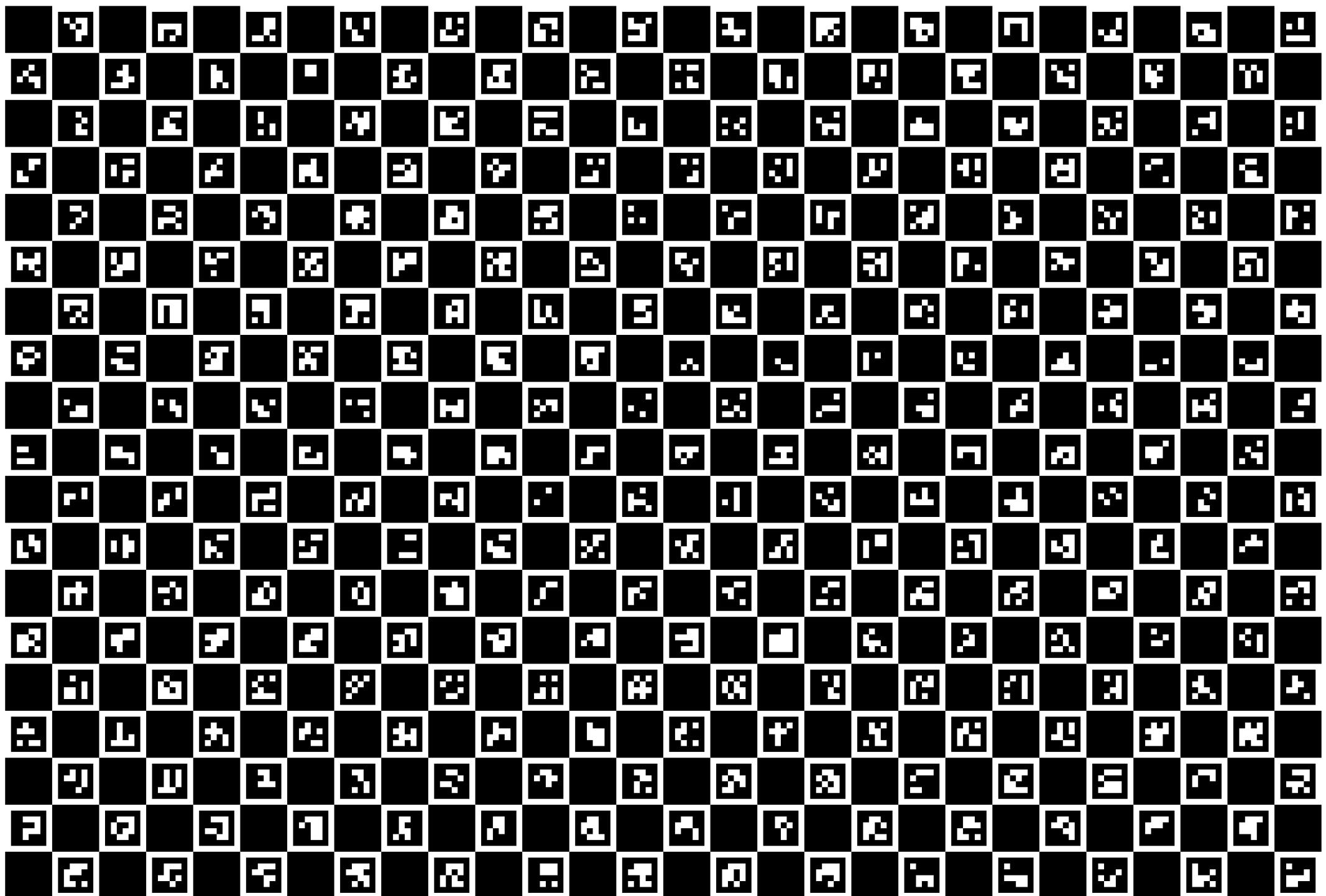


FIGURE 17: Model check of residuals of normality for the generic model on the Scorpion data set



List of Figures

1	Two examples of undistortion using the parametric (14) model. Both magnitude and angle is shown. $angle = 0$ is equal to displacement to the right in the image and the degrees are measured positive in the counterclockwise direction.	6
2	Two examples of undistortion using the parametric (14) model. Both magnitude and angle is shown. $angle = 0$ is equal to displacement to the right in the image and the degrees are measured positive in the counterclockwise direction.	7
3	Voronoi diagram of the residual error after using parametric camera calibration on one fold of each data group presented in 1. The color diagrams show, which direction the error is pointing in the image as illustrated at the bottom of the figure. The greyscale images show the magnitude of the residuals	8
4	Model check of residuals of normality for the parametric model on one of the 22 mm NikonD3100 camera folds	9
5	Examples of methods of knot vector generation on identical control points.	12
6	A grid of samples from the generic model. The x-y plane is the image plane while the lines show the corresponding direction.	14
7	Calculation of the 3D residual.	16
8	The spline is fitted to specific values (white) resulting in the control points shown (green). A video can be found on [9]	17
9	Example of the structure of the sparsity matrix.	18
10	Calculation of the 2D residual.	19
11	Reprojection error for calibration with different b-spline settings. Calibration performed on images from the Samsung 'WideAngle' data set, see table 1.	20
12	Run time for the test displayed in figure 11	20
13	RMS reprojection errors for the NikonD3100 camera and the cameras on the Samsung phone.	21
14	Voronoi diagram of the residuals after performing parametric and generic camera calibration on one fold of each data group presented in table 1. The color diagrams show which directions the residuals are pointing in the image as illustrated at the bottom of the figure. The greyscale images show the magnitude of the residuals.	22
15	Model check of residuals of normality for the generic model on one of the 22 mm NikonD3100 camera folds	23
16	Model check of residuals of normality for the parametric model on the Scorpion data set	31
17	Model check of residuals of normality for the generic model on the Scorpion data set	32

List of Tables

1	Table of the data used in this project. The camera module SLSI_SAK2L4SX is the rear module on the Samsung Galaxy S10 Plus. For the folds, the number in the parentheses indicate the number of images removed due to large errors	3
2	Table of the RMS reprojection error from calibrating the parametric model on each of the folds. The mean \pm standard deviation is shown for each camera	5
3	Table of the means of the standard deviations from calibrating the parametric model on each of the folds. The mean \pm standard deviation for the standard deviation of intrinsics/extrinsics is shown for each of the cameras	5
4	Table of the correlation test for the parametric model. cc is the correlation coefficient and p is the p-value. p-values with the value 0.0 is zero down to at least four decimals	10
5	Table of the RMS reprojection error for the different cameras. The Generic Model uses a order of 2 and 252 in step size	21
6	Table of the RMS reprojection errors from calibrating the generic model on one fold with multiple step sizes. The RMS reprojection error for the parametric model calibrated using the Scorpion data set is 0.0926.	22
7	Table of the correlation test for the generic model. cc is the correlation coefficient and p is the p-value. p-values with the value 0.0 is zero down to at least four decimals.	24