Camera Calibration from Out-of-Focus Images

by

Ryan Schmalenberg

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2024

© Ryan Schmalenberg 2024

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

One publication has resulted from the work presented in this thesis.

1. [67]: Ryan Schmalenberg and John Zelek. Out-of-Focus Camera Calibration. Journal of Computational Vision and Imaging Systems, 9(1):50-53, April 2024.

Abstract

For many 3D computer vision applications, accurate camera calibration is a necessary pre-requisite task. Generally, the objective is to find a camera's intrinsic parameters such as focal lengths, or extrinsic parameters such as the camera's pose in 3D space, or both. Camera calibration using structured calibration targets relies on special patterns which contain features that are used to localize control points with sub-pixel accuracy. The most frequently used patterns are checkerboards and circle grids, and in well constrained environments, these patterns are known to provide accurate feature correspondences for accurate camera calibration results.

One challenging case for camera calibration is in the instance of calibrating a long focal length camera. In this case, the focal plane can be too far away in distance, and the only practical solution is to capture images of the calibration pattern out-of-focus while it is closer to the camera. Due to the radial distribution of out-of-focus blur, and biases created by a lack of distance preservation, as well as changes in spatial blur with perspective, checkerboard patterns have been proven to lose accuracy when they are captured in out-of-focus images, and with increased blur, can fail to provide feature correspondences all together.

To address this, phase-shift circular gradient (PCG) patterns had been proposed as a method to encode control point positions into phase distributions, rather than through pixel intensities. Our work aims to validate previous authors claims of out-of-focus blur invariance and accuracy when using PCG patterns. Using PCG, small circle, and concentric circle grid patterns, we made comparisons using their respective retrieved pixel value focal lengths, and in-focus vs. out-of-focus percentage differences.

Initial comparisons showed that PCGs were largely invariant to blur. However, their accuracy was marginally worse than comparable small circles when real-world noise was introduced. In this real case, a 7-DOF robot arm was used for repeatable calibration target positioning. The recorded set of poses was also used to mirror conditions in a further synthetic experiment. From this work, PCGs initially showed mixed results, but when extended beyond real-world conditions, PCGs were the only pattern that worked under the most severe levels of out-of-focus blur. This validated their improved detectability under extreme blur, and theoretical effectiveness for use with long focal length cameras.

From these results, this study acknowledges the trade-offs in calibration pattern selection for respective use cases. It also highlights the importance of ellipse fitting techniques, as well as acknowledging the role of other learned methods. Finally, this study outlines the benefits that were observed when using robotic target positioning, and our synthetic validation pipeline for experimentation with calibration patterns under various conditions.

Acknowledgements

I would like to first thank my supervisor Professor John Zelek who made this work possible by setting up this research project with our industry partner, and for providing valuable knowledge, insights, support, guidance, and mentorship throughout. Through his hard work and dedication, Professor Zelek continues to create excellent opportunities for students such as myself. I would also like to thank our anonymous industry partner for a great deal of support, providing much of their time, knowledge, expertise, and advice. They helped to push the boundaries of my knowledge while I learned a great deal about camera calibration.

I would also like to acknowledge my lab-mate Kris (Chun Cheng Feng) who was involved in a different aspect of this research earlier on. I would like to thank Kris for his voluntary work towards helping me tediously collect calibration images, as well as being a wonderful lab-mate to work with and a source of encouragement. I would also like to thank Zobeir Raisi who supported our project as a post doc earlier on. Zobeir was dedicated to his work and helped me with many of the ins and outs of LaTeX as well as lots of initial proofreading of my literature review which later provided content for this thesis.

I would also like to thank the rest of my fellow lab-mates for making my time at the VIP lab intellectually stimulating and enjoyable. Of special mention, I would like to thank Adam Srebrnjak Yang, for his help with object segmentation, Matthew Bradley for his technical pointers and advice on helping me finish my thesis, and Kevin Hu for his continual encouragement, technical help, and genuine commendation of my work. I also appreciate Yuxiang Huang's curiosity and questions about this camera calibration work that helped me to think about it more clearly.

A special thanks goes out to my coding buddy Dennis Nguyen-Do who I know from McMaster University. Over the years, Dennis has constantly pushed me to continue growing my knowledge, and through regular meetings we have exchanged a lot of advice about anything Python related, as well as all kinds of other interesting tools and updates on newly released research.

I also have to thank Professor Brandon Dehart, and Alexander Werner for their excellent teaching of the Autonomous Mobile Robots course, and support in the RoboHub. Also, a huge thanks to Professor Dehart for being my first point of contact, and a model ambassador to the university through his unhesitating willingness to help and answer my many questions almost a year before I applied to the program.

Also, thank you to my Professors Jonathan Kofman, and Pascal Poupart for their exceptional courses, and their course projects that helped to support my research. Also thank you to Zahra Gharaee for her enlightening course on computer vision.

Thank you to my reading committee members Professors David Clausi and Oleg Michailovich, for their time and help with refining my work. I realize that camera calibration is not a frequently reviewed topic, so thank you for taking on the review of my work on this subject.

Most of all, I would like to thank all of my family who provided an enormous amount of support while I completed this work. I would especially like to thank my wife, Aya, who finished her MASc. concurrently while I started mine, while also being a super mom and working full-time to support our family. To my mom and my aunt Mina for their continual support, and especially to my mom for her habit of lifelong learning and for continually encouraging me to further my education. To my aunt Lolita and uncle Blair for making the distance between home and Waterloo feel much shorter, and for a huge amount of support from my wife's family, (Lolo & Lola), and her brothers for all of the times that we visited and booked the "study room" where the majority of the most deeply focused work was completed towards this thesis. I would also like to acknowledge some inspirational family members, uncles (Courosh, Badi, and Timo), who's academic accomplishments also inspired me to strive for further education. Also, thank you to uncle Courosh for advice on machine learning, and other academic related subjects.

Finally, I would like to thank my father, who although is no longer physically with us, has inspired me through his exemplary lifelong pursuit of knowledge, his technical background, his career, his academic pursuits, and his strong moral character, to continually search for knowledge, make efforts to help others, and help to advance society wherever possible. Dad, we miss you dearly, but I know you have been with me in spirit through this whole journey, and will continue to do so.

Land and Territorial Acknowledgement

As we explore the world through the advancement of science and technology, we must anchor our knowledge in ground truths. Likewise, to continue strengthening our foundations, we must acknowledge and learn about our past, the history of this nation, and the land that we use.

Our work has taken place on the traditional territory of the Neutral, Anishinaabeg, and Haudenosaunee peoples. The University of Waterloo is situated on the Haldimand Tract, 950,000 acres promised to the Six Nations that includes six miles on each side of the Grand River. Today, approximately 48,000 acres ($\sim 5\%$) of this land remains as the current Six Nations reserve.

As we have made use of this traditional territory, let us be thankful for the scientific and technological knowledge that it has helped us to acquire. Let us also strive to balance this newfound knowledge with learning about the land, renewed respect for its diverse peoples, reflection on the connections between nature and science, and understanding about how to make meaningful contributions going forward.

As the University provides resources through the Office of Indigenous Relations, let us commit, with an open mind, to draw on these resources, and to learn about the Anishinaabeg and Haudenosaunee confederacy nations, their cultures, the land, the proclamation and treaties that had been established and broken, what needs to be done, and how we can open new dialogs and share collective knowledge going forward.

Dedication

This thesis is dedicated to my wife, our daughter, and my parents.

Table of Contents

Author's Declaration ii

Statement of Contributions iii

Abstract iv

Acknowledgements v

Land and Territorial Acknowledgement vii

Dedication viii

List of Figures xiii

List of Tables xvi

Table of Notation xvii

1 Introduction 1

1.1 Background 1

1.2 Problem 3

1.3 Aim, Objective, & Contributions 3

2 Theory **5**

2.1 Pinhole Model Camera Calibration 5

2.2 Additional Camera Models 9

2.3 Theory of Defocus Blur 10

2.4 Center Point Correction 12

3 Related Work **15**

3.1 Checkerboard and Circle Methods 17

3.2 Blur Characteristic and Depth from Defocus Methods 21

3.3 Active Display and Phase-Shifting Methods 21

3.3.1 Square Patterns 22

3.3.2 Circular Patterns 23

3.4 Deep Learning Methods 32

3.4.1 Camera Calibration Using Deep Learning and Checkerboards 34

3.4.2 Camera Calibration Using Deep Learning from Blurred Images . . . 35

3.4.3 Auto / Self, and Other Deep Learning Based Calibration Methods . 41

3.5 Other Methods 45

3.5.1 Fiducial Markers and Other Patterns 46

3.5.2 Calibration Projection Method and Toolbox 48

3.5.3 Deblurring Images Prior to Calibration 49

3.6 Related Work Summary 49

4 Methods **51**

4.1 Standard Camera Calibration Method 51

4.2 Control Point Detection Methods 52

4.2.1 PCG Patterns 52

4.2.2 Control Point Localization and Calibration Pipeline 58

4.2.3 Circular Patterns 60

4.3 Camera Calibration from Real Images . . . 62
4.3.1 Initial Trials and Foundational Knowledge . . . 62
4.3.2 Robot Guided Real-World Experiment Method . . . 62
4.4 Camera Calibration from Synthetic Images . . . 66
4.4.1 Initial Synthetic Experiment Method . . . 66
4.4.2 Calibration Validation and Focal Length Ground Truth Recovery Method . . . 69
4.4.3 Advantages of Synthetic Camera Calibration . . . 70
4.4.4 Method for Synthetic Experiment Aligned with Robot Guided Real-World Experiment . . . 70
4.5 Methods Summary . . . 73

5 Results and Discussion **74**
5.1 Comparison of Focal Lengths . . . 74
5.2 Initial Synthetic Experiment Results . . . 75
5.3 Robot Guided Real-World Experiment Results . . . 77
5.4 Results for Final Synthetic Experiment Aligned with Robot Guided Real-World Experiment . . . 80
5.5 Results Summary . . . 87

6 Conclusions and Recommendations **88**

Table of Copyright Content Reuse Permissions and Licenses **91**
License No. 5873930308736 . . . 92
License No. 5873100004458 . . . 94
License No. 5873921430784 . . . 96
License No. 5875531099349 . . . 98
License No. 5875531364412 . . . 100
License No. 5873920499397 . . . 102

License No. 5875540019944 104
Reuse Permission Email 106
Optica Publishing Group Permission Email 108

References **110**

APPENDICES **120**

A APPENDIX A (Blender Focal Length Recovery and Lighting) **121**

A.1 Example Calculation of Focal Lengths from Blender Perspective Camera Model 121
A.2 Synthetic Approximation of Lighting Conditions 124

List of Figures

2.1 Pinhole camera projection 7
2.2 Point projection vs. observed points 8
2.3 Additional camera models 9
2.4 Thin lens in-focus and out-of-focus 10
2.5 Concentric circle center point distances 12

3.1 Taxonomy of camera calibration methods 16
3.2 Testing available code for blurred checkerboard detection 18
3.3 Localization error of checkerboard method proposed by Duda et al. 19
3.4 Localization error of method from Choi et al. 20
3.5 Experimental setup of Ha et al. 23
3.6 Wang et al. proposed PCG method 24
3.7 MRE comparison of PCG synthetic experiment 26
3.8 Wang et al. color PCGs 29
3.9 Chen et al. LBP PWD varied period circles 29
3.10 Chen et al. LBP PWD encoding method 30
3.11 Chen et al. experimental setup 31
3.12 Zou et al. neural network guided robot 33
3.13 MATE patterns 34
3.14 CCDN architecture 35
3.15 MS-TENet process 36

3.16 Camera calibration for robot in tunnel boring machine 37
3.17 Phase-Net architecture 38
3.18 Huo et al. method 38
3.19 Huo et al. calibration patterns 39
3.20 Huo et al. results 40
3.21 Huo et al. experimental setup 41
3.22 MTL-CPL architecture 43
3.23 CTRL-C architecture 44
3.24 Fundamental matrix MLP architecture 45
3.25 Mono-spectrum AR fiducial marker 46

4.1 Implemented PCG calibration vs. traditional checkerboard method 53
4.2 Implemented control point recovery method 54
4.3 Double Wrapped Phase Discontinuity 56
4.4 Center Point Correction 57
4.5 General camera calibration process implemented 58
4.6 Small and large circle patterns 61
4.7 Concentric circle patterns 61
4.8 Initial lab experiment 63
4.9 Computar C23-5028-5M 50mm camera lens 64
4.10 Our lab setup 65
4.11 Pose rendering and wrapped phase image visualization 67
4.12 Configuration of Blender scene rendering for PCG images 67
4.13 Spatially dependant perspective blur visualization 68
4.14 Synthetic ground truth recovery and validation 69
4.15 Pose aligned real and simulated images 71
4.16 Franka Emika Panda robot in Gazebo 72

5.1 Focal lengths of initial PCG synthetic validation experiment 76
5.2 Percent Differences Between Real-World In and Out-of-Focus Focal Lengths 79
5.3 Final Experiment Synthetic Images 81
5.4 Final Synthetic Experiment Focal Lengths Plot 83
5.5 Final Synthetic Experiment Percent Differences Plot 84

A.1 Blender Focal Lengths 123
A.2 Lab Lighting 124
A.3 Blender Scene Lighting 125

List of Tables

1 Table of Notation xvii

3.1 Wang et al. table of results 25
3.2 Cai et al. table of results 27
3.3 Chen et al. table of results 31
3.4 Huo et al. table of results 42

4.1 Phase-Shift Circular Grating (PCG) Method Steps 55
4.2 Calibration cases 60

5.1 Results for initial synthetic image experiment. 77
5.2 Results for robot guided real-world experiment. 78
5.3 Results for final synthetic experiment 80
5.4 Comparison between real-world experiment and final synthetic experiment 86

Table 1: Notation

Symbol	Definition
λ	Scale factor
$\hat{x}_{img}$	Re-projected image points
x_{img}	Image points
x_w	World points
R_w	Rotation matrix
t_w	Translation vector
K	Intrinsic matrix
T	Extrinsic matrix
P	Projection matrix
H	Homography matrix
a	object distance
b	image distance
i_{blur}	blurred image
i	sharp image
k	blur kernel
n	noise
r	blur radius parameter
C	normalization constant
σ	standard deviation
f	Focal length
f_x	Focal length x
f_y	Focal length y
c_x	Camera center x
c_y	Camera center y
u	Image pixel coordinate in x direction
v	Image pixel coordinate in y direction
ϕ	Eigenvectors
ψ	Eigenvalues
ξ	Eigenvector representing direction of 3D perspective transformation

Note: Notation based on Forsyth and Ponce [33] and Labussière [52]

Chapter 1

Introduction

1.1 Background

A common challenge in various camera-based computer vision applications is to accurately relate the 2D points in a captured image to the 3D points in a natural scene. This relationship requires accurate knowledge of the camera's internal (intrinsic) parameters. It also requires knowledge of the camera's extrinsic parameters, which define the camera's pose, or translation and orientation in 3D space. Fundamentally, camera calibration is a procedure that is performed to determine these parameters and typically relies on a calibration target with patterns of known shape and size. Although several methods [75, 87] have been well established, camera calibration continues to be an active area of research as the requirements of dependant computer vision applications grow.

Generally, a camera requires initial calibration after the physical components such as the lens, image sensor, and electronics have been manufactured and assembled. However, they can also require re-calibration in the field. One reason is that re-calibration might be an inherent requirement for a particular computer vision pipeline. Applications such as structure from motion (SfM), and simultaneous localization and mapping (SLAM), require frequent translations between 2D and 3D, and continuous re-calibration can be required to update camera parameters when a camera is re-focused on a new object. Cameras can also require re-calibration because of other factors such as wear and tear, vibrations that shift components out of alignment, or other environmental factors such as temperature that cause a camera's parameters to gradually change over time [47].

Depending on the application, camera calibration methods can follow different approaches. Vision systems in mobile robots [22], or autonomous vehicles [47], require fre-

quent camera parameter estimation updates, and cannot rely on structured calibration targets owing to the dynamic nature of their working environment. These applications often rely on self or auto-calibration methods that use natural scene features which are found in streams of images where features like planes and vanishing points can be identified and used to recover camera parameters [15, 37]. Other attempts have also been made to recover camera parameters directly from images as a data-driven end-to-end process using deep learning (DL) techniques [43, 57, 82, 90]. Although several auto-calibration methods currently exist, they approach calibration in different ways using different numbers of images, stereo vs. monocular camera rigs, and different training sets that are derived from real or synthetic images.

In well constrained environments, it is also possible to perform camera calibration using carefully designed calibration targets. Compared to auto-calibration, calibration using targets can be a better controlled approach, which is preferable for initial calibration where higher accuracy is needed. Camera calibration targets can consist of 3D objects [39, 81], flat planes with 2D patterns [17, 18, 24, 26, 31, 46, 63, 77, 78], and 1D patterns [88]. Owing to simplicity and low cost, 2D calibration patterns have been the most popular choice of calibration target, and have been studied the most [17, 18, 24, 46, 77, 78]. 2D Calibration targets typically use checkerboard or circle grid patterns, but many other patterns have also been proposed [30, 39]. It is also common to distinguish between passive and active 2D calibration targets. Passive targets rely on a single image which can be printed and mounted on a rigid surface, and active targets rely on a digital screen which is needed to display multiple consecutive images of a calibration pattern. Another clear distinction is that passive targets reflect light, and active targets emit light.

For both active and passive methods [13], when 2D targets are used, multiple images need to be captured of the target in different poses. Once the images are acquired, feature points or control points are detected from the 2D image of the pattern. These control points provide input to the subsequent calibration process steps along with prior established metric information about the calibration pattern, such as feature size and distance between features. Using these inputs, initial estimations of 2D to 3D correspondence are made to determine the homography between target planes with an approximated camera model containing intrinsic parameters. Further refinement is performed using associated points from all other images by re-projecting them with the initially estimated camera model. Using the reprojected points, the refinement step is cast as an optimization problem where camera parameters are varied to minimize re-projection error. Although many methods [11, 56, 58, 75, 88] exist, a typical example would be to minimize error using linear least-squares minimization, followed by non-linear refinement using one of many potential methods such as Gauss-Newton [70]. Other approaches could be to find the fundamental matrix by

solving a system of equations and matching feature pairs which can be performed using random sample consensus (RANSAC) [64]. In any case, the accuracy of calibration is affected by the accuracy of control point localization and the degree of optimization process convergence, which can be significantly influenced by the choice of the loss function and error approximation threshold.

1.2 Problem

High calibration accuracy is essential in 3D computer vision applications where metric information is considered. In 3D reconstruction and non-contact optical measurement applications, inaccurate calibration will lead to incorrect 2D to 3D correspondences. It can cause vision systems to fail or operate outside of acceptable tolerances. Although several accurate methods [29, 87] have been proposed to date, accurate determination of camera parameters can be hindered under certain conditions. This is particularly true when calibrating long focal length cameras in tightly space-constrained environments. In these situations, it may not be possible to place the calibration target within the camera's minimum focusing distance, as a result, blurred or out-of-focus images will be captured. Consequently, with out-of-focus images, traditional calibration methods can return inaccurate parameters or fail altogether [46].

1.3 Aim, Objective, & Contributions

To address the problem of not being able to calibrate a long focal length camera within its minimum focusing distance, the aim of this work is to better understand the conditions that lead to more accurate feature detection for camera calibration, and to find the most accurate method of performing camera calibration from out-of-focus images. To achieve this objective, an extensive literature review outlines related work and methods, and experimental work is conducted to validate a recommended approach.

Using synthetic and real-world data, we compare phase-shift circular gradient (PCG) patterns and circle grids to examine their relative calibration accuracy under out-of-focus blurred conditions. Since the task of accurate control point localization from blurred images is an initially ill posed problem, our first contribution is to demonstrate a technique for positioning the calibration target consistently to remove geometric variability, and to provide comparable results between focus levels. These are considerations that have been

frequently overlooked in past work, and when left unaddressed, can contribute unwanted geometric error, and generate results that are difficult to compare.

To solve this problem, we use a robot arm for repeated calibration target positioning in our real-world experiments. We also ensured that different focus level images were captured without modifying the camera's intrinsic parameters, with the goal of minimizing the differences between estimated parameters that are recovered after calibration. Further to this contribution, we utilize the robot pose data to replicate and extend real-world experimentation in a synthetic rendering environment, while simulating the real-world camera setup and scene lighting conditions. This allows us to compare real vs. sim differences as well as to obtain further results that would not be possible to observe in the real-world setup, due to field of view boundary limitations vs. the degree of out-of-focus blur. To our knowledge, this is the first work that has attempted to observe results from real-world coupled synthetic calibration images, with this degree of replication accuracy and detail. We also calculate calibration accuracy based on the distances between focal lengths for in, and out-of-focus cases, which we believe is a better standard for comparison overall.

Chapter 2

Theory

2.1 Pinhole Model Camera Calibration

In this section, we outline the theory of camera calibration as it relates to direct linear transform (DLT) and the pinhole camera model. We also describe the most common method that is regarded as a standard method [87]. From here, we establish a basis for discussing the nuances and general approaches to solving the problem of performing calibration from blurred images. For ease of understanding, we chose notation that best relates to the broadest collection of previous works [33, 42, 52, 87].

First, we describe the typical relationship for the projection between 2D and 3D, where x_{img} is a set of image point coordinates, or pixel coordinates of a point in the captured image, K is the intrinsic matrix, T is the extrinsic matrix, and x_w are the world point coordinates or coordinates of a point in the real scene:

$$x_{img} = KTx_w. \tag{2.1}$$

To visually represent the relationship of points, as shown in Figure 2.1, we can see where $x_w = (x, y, z)$ are the 3D coordinates of a point in the scene, and $x_{img} = (u, v)$ are the corresponding 2D pixel coordinates in an image.

From (2.1), the goal of camera calibration is to determine the camera's intrinsic parameters which are contained in matrix K (2.2) where (f_x, f_y) are scale factors [42]. In the instance of a digital camera, these scale factors can be interpreted as the focal lengths in the x and y directions in units of pixels. The parameter s represents the pixel skew angle,

and (c_x, c_y) are the camera center. Most often, the image sensor pixels are square, making $s = 0$ and $f_x = f_y$ [52]:

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.2}$$

Expanding this relationship and building from the notation of Hartley and Zisserman [42], Forsyth and Ponce [33], and primarily Labussière [52], a pinhole camera (2.1), in homogeneous coordinates can be represented as (2.3):

$$x_{img} = K_{pinhole} P T x_w \Leftrightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \propto \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_w & t_w \\ 0_{1x3} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{2.3}$$

where P is the projection matrix and T is the extrinsic matrix that defines the camera pose in world coordinates. Within T, R_w is the rotation matrix, and t_w is the translation vector. When P and T are multiplied, this can also be known as the pose matrix [64].

Using this projection model, K can be computed after finding an initial estimate of H where (2.4):

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}, \tag{2.4}$$

and λ is a scale factor. Note that the matrix consisting of r_1, r_2, t is a 3×3 matrix because z is set to equal 0.

In DLT, we assume that the coordinates of x_{img} and x_w are known. Using a minimum of 6 sets of these corresponding point coordinates, we can perform singular value decomposition (SVD) to find H. For camera calibration, Zhang's method [87] is often used, and is widely regarded as a standard calibration method owing largely to its implementation within OpenCV [5] and MATLAB [1, 70]. In this approach, only 4 corresponding sets of points are needed to find an initial estimate of H, but unlike in DLT, the z coordinates of x_w, are not known. To provide a starting point, metric information about the calibration target pattern is substituted into x_w and z is set to equal 0. The most typical example is

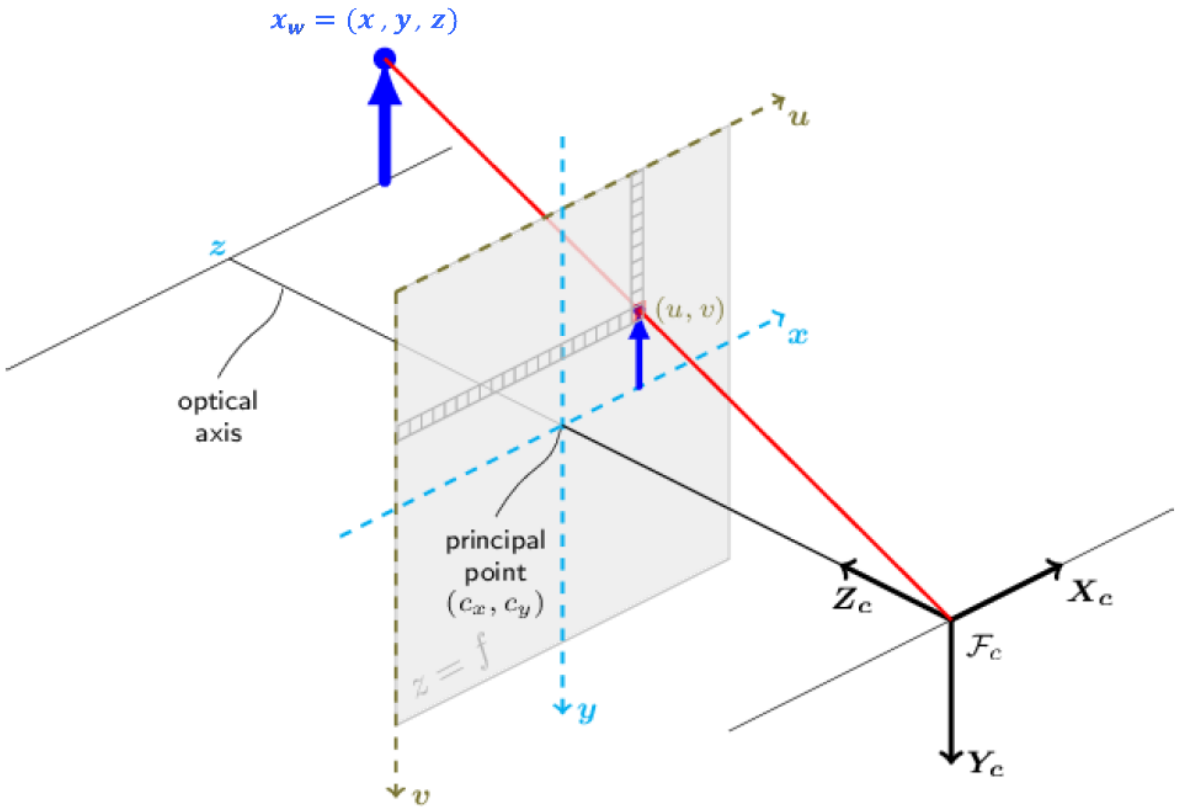


Figure 2.1. Pinhole camera based point projection from 3D points $x_w = (x, y, z)$ to 2D points $x_{img} = (u, v)$ adapted from "OpenCV: Camera Calibration and 3D Reconstruction" [5]. Image licensed under the OpenCV Apache License, Version 2.0 (the "License").

using a checkerboard as the calibration pattern where the size and number of checkers are known, and a minimum of 4 points can be provided relative to the checkerboard coordinate system itself which provides starting values to x_w.

Finding the initial value of H is then repeated for each image. Once the initial estimations of H are computed based on the corresponding sets of points from multiple sets of images, K can then be separated from H by exploiting certain constraints knowing that K remains a fixed component across all images, while R_w and t_w vary. Once K is found, the

extrinsic parameters can be solved from (2.4) and (2.5):

$$\begin{cases} r_1 = \lambda K^{-1} h_1 \\ r_2 = \lambda K^{-1} h_2 \\ r_3 = \lambda K^{-1} h_3 \end{cases} \quad \text{where: } \lambda = \frac{1}{||K^{-1}h_1||} = \frac{1}{||K^{-1}h_2||}. \tag{2.5}$$

Using the recovered intrinsic parameters K, and extrinsic parameters r and t, as per (2.4) H is known and is used to make an initial projection from x_w to points $\hat{x}_{img}$ on the image plane. The objective is to then minimize the sum of the error between the observed x_{img}'s and the projected $\hat{x}_{img}$'s based on H (Figure 2.2). In solving this minimization, we can model a more complex nonlinear function that accounts for radial and other types of distortion. Conceptually, this nonlinear minimization task is categorized as a maximum likelihood estimation (MLE) problem and can be solved using a Levenberg-Marquardt algorithm [87] or any other preferred iterative MLE method.

Finally, the minimization problem can be generally stated as per equation (2.6):

$$\min_H \sum_i \sum_{img} dist(x_{i,img}, \hat{x}_{i,img}). \tag{2.6}$$

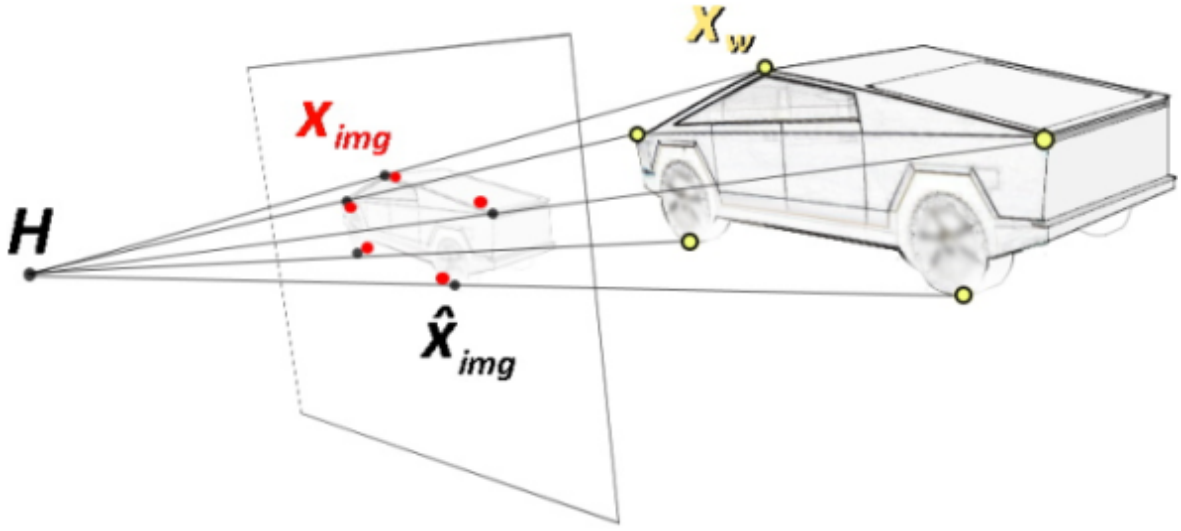


Figure 2.2. Initial projection $\hat{X}_{img}$ vs. observed X_{img} points, projected to x_w world points.

2.2 Additional Camera Models

Since the pinhole camera model only works for focused images, further models have been created to account for the secondary light rays that refract through the lens. Labussière [52] provides some examples of these models, which are shown in Figure 2.3.

As noted by Labussière [52], the thin lens equation, also known as the Gaussian lens equation, establishes the relationship between object distance a, image distance b, and focal length f and is defined by:

$$\frac{1}{f} = \frac{1}{a} + \frac{1}{b}. \tag{2.7}$$

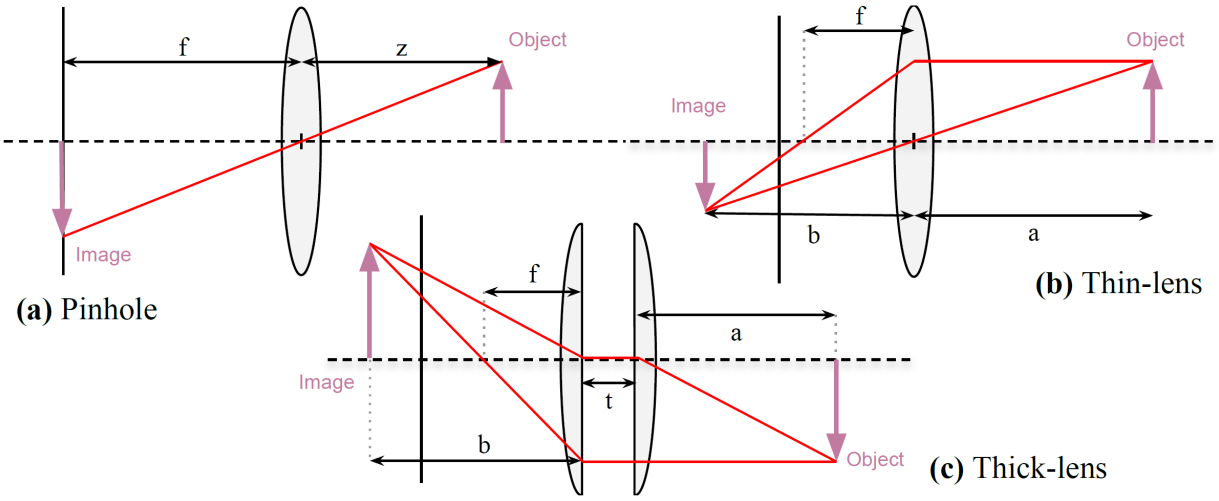


Figure 2.3. Additional camera models, reproduced with permission from Labussière [52].

Based on this, Labussière [52] establishes the projection equation (2.8) for the thin-lens model:

$$x_{img} = K_{thin-lens} T x_w \Leftrightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \propto \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} R_w & t_w \\ 0_{1x3} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{2.8}$$

and Labussière [52] also provides an equation (2.9) for projection using the thick lens

model:

$$x_{img} = K_{thin-lens} T x_w \Leftrightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \propto \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{f} & 1 \end{bmatrix}}^{K_{thick-lens} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 - \frac{t}{f} & 0 \\ 0 & 0 & -\frac{1}{f} & 1 \end{bmatrix}} \begin{bmatrix} R_w & t_w \\ 0_{1x3} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (2.9)$$

2.3 Theory of Defocus Blur

According to Joshi [48], *defocus blur* is defined as: *"a loss of sharpness that occurs due to integrating light over an aperture with a nonzero area where the light source is off of the image focal plane"*. From Wang [77], using the thin lens model, we can see that the projection of a focused point, Figure 2.4 (a) converges at the focal (or sensor) plane. However, the defocused image, 2.4 (b) converges past the focal plane. As a result, both Labussière and Tang [52, 72] describe that the resulting projection of a point will create an image of a disk on the sensor plane.

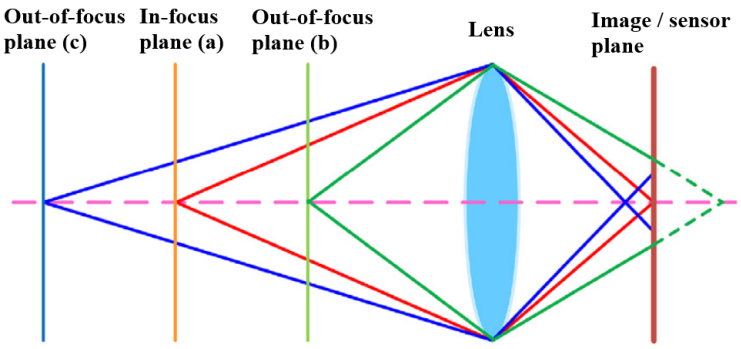


Figure 2.4. From Wang [77], thin lens model (a) in-focus, and (b) projecting defocus blur. Adapted with permission from Elsevier.

From Joshi [48], the amount of blur depends on the aperture opening size, image sensor pixel size, and object and focal depth. Under thick and thin lens model assumptions [72],

defocus blur can be modeled using a point spread function (PSF) [48]. As the name implies, a PSF describes explicitly how a point, such as the one in Figure 2.4 is spread across the image. Conversely, without blur, every scene point would directly correspond with a single matching image point versus being spread across multiple points in the image [48].

Mathematically, a simplified linear model for blur can be stated as [48]:

$$i_{\text{blur}} = i * k + n, \tag{2.10}$$

where i_{blur} is the blurred image, and i is the sharp image, which is a convolution of a blur kernel k with the addition of n noise. Typically, the addition of n is modeled by the input of additive, zero mean, white Gaussian noise, where the standard deviation controls the amount of blur:

$$n \sim \mathcal{N}(0, \sigma^2), \tag{2.11}$$

and $\mathcal{N}$ denotes the normal distribution. In practice, the simplified linear model in (2.10) does not typically hold to be true. Owing to the dependency on depth, a more accurate blur model is needed to account for variations of depth at every point in the scene and across every point in the image. This might not be the case for a specially devised scene, but most natural scenes exhibit differences in depth at different points that will spatially vary the amount of blur that is captured [48].

Additionally, under the weak assumption that a PSF must be of known parametric form [48], two common parametric defocus blur models are (2.12) and (2.13):

$$k(u, v) = \begin{cases} 0 & \sqrt{u^2 + v^2} > r \\ \frac{1}{\pi r^2} & \sqrt{u^2 + v^2} \leq r. \end{cases} \tag{2.12}$$

The "pillbox" or circular disk kernel [20, 48], equation (2.12) and the 2D Gaussian kernel, which is circularly symmetric [12, 48] equation (2.13) as follows:

$$k(u, v) = C\left(-\frac{u^2 + v^2}{2\sigma^2}\right), \tag{2.13}$$

where r in equation (2.12) and σ in equation (2.13) are the blur parameter, C is a normalization constant, and (u, v) are the image coordinates in the (x, y) directions.

2.4 Center Point Correction

As mentioned in various literature [17, 25, 40, 41, 83, 84], since the projection of a circle in perspective becomes an ellipse, it is incorrect to assume that the center of the circle is also the center of the ellipse. From Figure 2.5, we can see that perspective angle relative to the observer affects the distance between ellipse centers and the true center of the circle. As the radius increases, so does the error, and conversely, the smaller the circle and respectively fitted ellipse, the smaller the error.

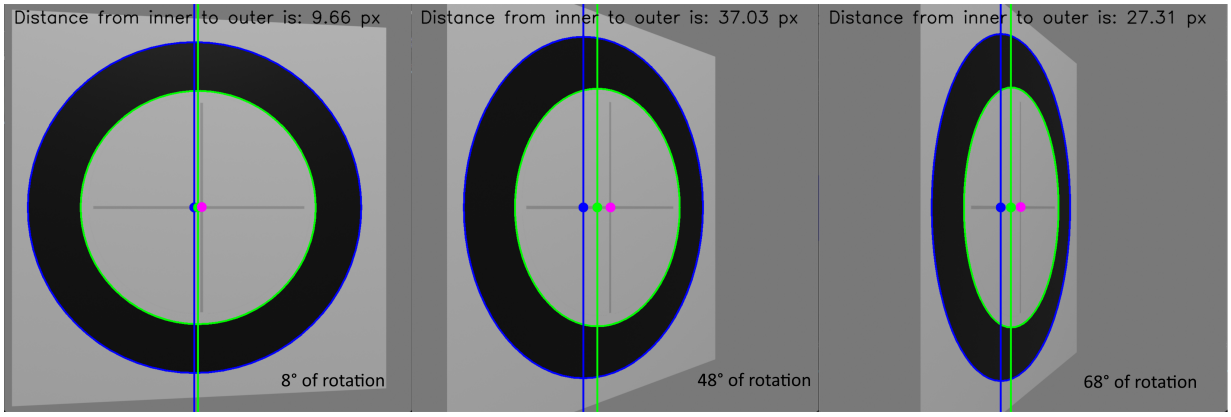


Figure 2.5. We rendered concentric circles and fitted ellipses to depict the horizontal spread of center points from the inner (green) and outer (blue) ellipses under varying degrees of rotation. The computed correct center point is plotted in magenta over a grey cross which is part of the original pattern and validates the computed corrected center point.

Since it is not possible to detect and fit an infinitely small ellipse upon an infinitely small circle, we referred to Cai *et al.* [17], and Zhang *et al.* [84], who outline a center point correction method from concentric circles.

The general equation of an ellipse can be stated as (2.14) [9]:

$$Q(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \tag{2.14}$$

where (x, y) are a set of points on the cartesian plane. The quadratic equation can also be written in the form shown in (2.15) where the vector $\mathbf{x}$ is defined as (2.16) [9]:

$$\mathbf{x}^T A_Q \mathbf{x} = 0, \tag{2.15}$$

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{2.16}$$

In matrix form, this can be stated as (2.17), where matrix A_Q is a square matrix and $A_Q \in \mathbb{R}^{n \times n}$:

$$A_Q = \begin{bmatrix} A & \frac{1}{2}B & \frac{1}{2}D \\ \frac{1}{2}B & C & \frac{1}{2}E \\ \frac{1}{2}D & \frac{1}{2}E & F \end{bmatrix}. \tag{2.17}$$

Using two fitted ellipses (one inner A (green in Figure 2.5), and one outer B (blue in Figure 2.5)), the matrix pair, (A, B) can be decomposed into eigenvectors and eigenvalues by solving the generalized eigenvalue problem. Kressner [51] describes that the eigenvalues of (A, B) are the roots (α, β) of the polynomial (2.18):

$$\left|\beta A - \alpha B\right|. \tag{2.18}$$

From Ghojogh *et al.* [36], (2.19) describes the generalized eigenvalue problem in matrix form, where ϕ are the eigenvectors, and the diagonal elements of ψ are the eigenvalues. Note that the order of A and B matter, and (ϕ, ψ) is known as the eigenpair of (A, B):

$$A\phi = B\phi\psi. \tag{2.19}$$

Practically, (ϕ, ψ) contains the 3-dimensional direction of perspective transformation between the original circles in the scene and their respective projection to ellipses that are captured in the image. To find (ϕ, ψ), it is possible to use MATLAB's polyeig() function [6, 17, 84] or Python's scipy.linalg.eig library function [7].

The MATLAB function polyeig [6] solves the polynomial variant of the generalized eigenvalue problem, and uses QZ factorization to compute intermediate results for finding generalized eigenvalues as well as to ensure that the results are well determined. The polynomial eigenvalue problem is stated to be a variant of the generalized (also known as standard or ordinary) eigenvalue problem where polynomials can be solved instead of linear terms. Python's scipy.linalg.eig function solves only the generalized eigenvalue problem to find eigenvalues as well as the right or left eigenvectors from a general matrix [7]. The linalg.eig function relies on the Fortran 77 subroutine library known as LAPACK [2], whenever arrays are in column major order.

In the case of a pair of ellipses (A, B), three eigenvalues and respective eigenvectors can be found. From the three eigenvalues, the minimum distance between eigenvalues can be computed to find the closest eigenvalue pair. The repeating pair of eigenvalues can be removed along with the respective eigenvectors from ϕ to leave only one remaining eigenvector ξ which represents the 3-dimensional direction of perspective transformation (2.20):

$$\xi = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{2.20}$$

Finally, under weak perspective projection [89], from ξ we can recover the corrected center point from using (2.21) where Z_c is the depth from the centroid:

$$\begin{aligned} x &= \tfrac{X}{Z_c} \\ y &= \tfrac{Y}{Z_c}. \end{aligned} \tag{2.21}$$

Chapter 3

Related Work

To solve the problem of performing camera calibration from out-of-focus images, it is important to consider, which approaches could be taken towards improving calibration from blur. Specifically, there are three potential directions that can be mentioned. In procedural order, these can be: (1) attempt to initially de-blur out-of-focus images before performing feature detection and subsequent calibration, (2) improve methods for detection of features given uncorrected images that still contain defocus blur, and (3) Improve the calibration method itself so it can more robustly accept loose feature correspondences, or, work with a different calibration framework altogether. In this section, we explore the taxonomy of recent camera calibration research as it relates to performing calibration from out-of-focus images, and primarily outline work that has been done to advance approach #2, finding methods to improve feature detection, given that the majority of related past work has been focused in this area.

Currently, most calibration methods [13, 17, 18, 23, 26, 30, 38, 77, 78] are based on 2D planar calibration patterns. In related work such as the work of Schmalz *et al.* [68], these 2D calibration patterns are classified under two main categories: passive and active calibration targets. ElSheikh *et al.* [32] use infrared checkerboard patterns, and distinguish that their passive targets rely on emissivity, while active patterns rely on a combination of emissivity as well as a heated pattern that emits infrared light. Therefore, one common distinguishment between active and passive targets is that passive targets reflect light and active targets emit light, however, this distinction is often also qualified by the ability to display multiple images (for active targets). This convention does not hold for E-Ink displays because they rely on reflecting an external light source and do not typically emit their own light, however they are capable of displaying multiple consecutive images. Therefore we can see that some contradictions in terminology are present.

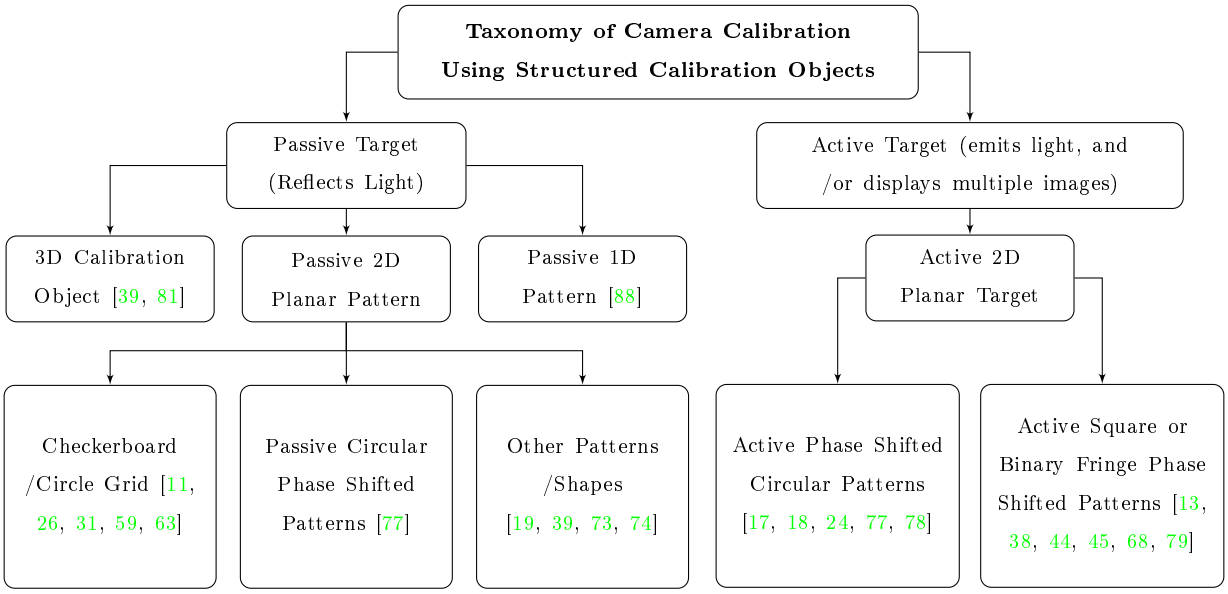


Figure 3.1. General taxonomy of camera calibration methods using structured calibration objects.

With these contradictions in mind, we will qualify the term "active target" by its ability to either emit light, or display multiple images, or both.

Passive patterns. Include any static pattern such as a checkerboard, circle grid, asymmetric circle grid, or other shapes that are typically printed on a flat surface and are used to acquire calibration images in various poses.

Active patterns. Can use digital displays such as TVs, computer monitors, or mobile device display screens. These digital screens are often used to display consecutive patterns that are combined to locate control points, which get passed into a subsequent calibration operation.

For blurred images, most of the recent research has been aimed at improving feature detection by making patterns with features that are easier to detect or by improving the feature detection method itself. It is also worth noting that many of the following works rely on the same method by trying to pass detected feature control points to the calibration method proposed by Zhang [87]. This method forms the basis for the calibration function that is written into OpenCV [5], as well as MATLAB [70], and since it was used in the

majority of reviewed work, we refer to it as the standard calibration method or Zhang's method.

In the following section we review the previous work. These works are organized into the following sections:

- 3.1 Checkerboard and Circle Methods [26, 31, 63],
- 3.2 Blur Characteristic and Depth from Defocus Methods [11, 60, 86],
- 3.3 Active Display and Phase Shifting Methods [13, 17, 24, 38, 77, 78],
- 3.4 Deep Learning Methods [16, 23, 30, 46, 53, 64],
- 3.5 Other Methods [19, 39, 54, 56, 73, 74].

Although some work might be presented in one of these five categories, it may span more than one. For instance, some work has been done that uses a phase-shifting method for calibration and includes another fundamentally different approach, such as deep learning in part of the implementation.

3.1 Checkerboard and Circle Methods

In efforts to increase the robustness of checkerboard-based calibration methods, several researchers have attempted to improve checkerboard corner detection by focusing on improving robustness to blur.

Based on the original OpenCV checkerboard detector, which was based on the work of Vezhnevets [76], Rufli *et al.* [63] focused on improving detection for checkerboard corners for fisheye, catadioptric, and omnidirectional cameras. Using an improved novel heuristic for detecting checkerboard corners, Rufli *et al.* [63] created an automatic checkerboard extractor which could be implemented into several MATLAB toolboxes[1]. From the original OpenCV method, checkerboard checkers are segmented using adaptive thresholding, and subsequent erosion passes separate the checkers from each other until a contour finder can generate quadrangle polygons. Using a quadrangle linking heuristic, corners of the quadrangles are linked, and the arithmetic mean of the links is used to represent the detection of the original corner locations.

[1] Rufli *et al.* [63] also provided freely available C++ code and binaries https://sites.google.com/site/scarabotix/ocamcalib-omnidirectional-camera-calibration-toolbox-for-matlab

Since the original method only returns checkerboard corners if a checkerboard with a complete pattern of checkers is detected, one important improvement that was made is that the new method could return checkerboard corners even if a full pattern was not detected. This improvement was possible because the standard calibration method that was used would not need the entire set of corners to perform calibration. Other improvements were found by alternating between "cross" and "square" erosion kernels to help preserve the aspect ratio of smaller checkers, which would help them shrink uniformly. Further, a modified quadrangle corner linking heuristic was used that only matches corners if they lie within a certain geometric area. This was found to be necessary for dealing with matching the correct corners in highly distorted images. Additional improvements were made to correct some inaccuracies with the distance metric used for corner linking to make it relative to the erosion state, which would improve linking in lower resolution images.

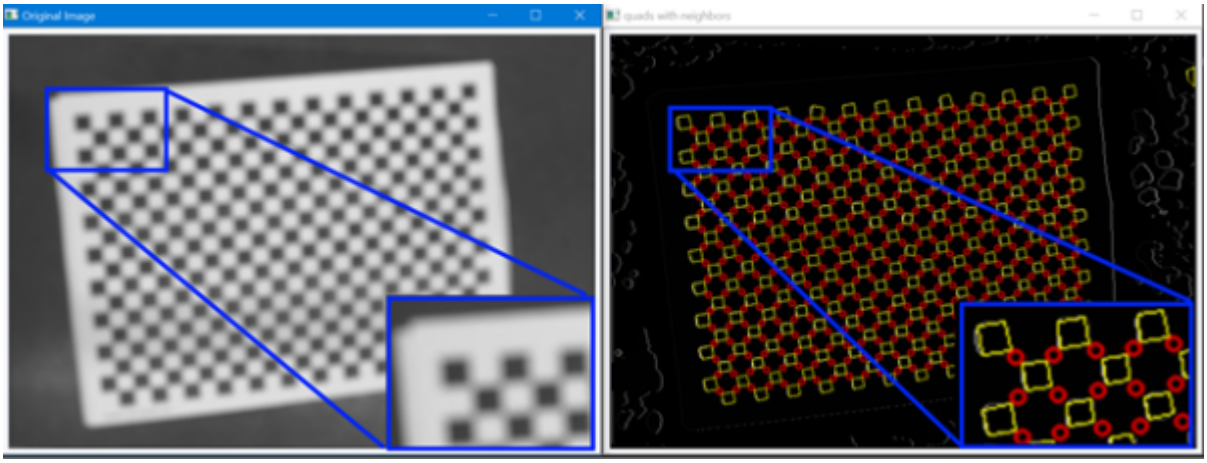


Figure 3.2. Our testing of available code from Rufli et al. and Scaramuzza et al. [63, 65, 66] used to observe checkerboard corner detection on blurred image, successfully found all corners after successive erosion passes.

Further to these improvements, Rufli, *et al.* also addressed blur originating from anisotropy, which is found in omnidirectional cameras where the entire surface of the mirror cannot be in-focus. The anisotropy in this case was described to occur where the image is blurred radially towards the edges or the center of the image and would be dependent on the camera's focal length [63]. To deal with this, they extended the original approach by combining detected checkers from successive erosion passes. This was necessary, because depending on the level of blur in different parts of the image, not all quadrangles would separate during the same pass of the erosion run, and some might even disappear before others would begin to separate. Therefore, it was necessary to combine all partial sets of

successfully detected quadrangles from different erosion passes. However, this improvement was noted as being the smallest contributor to increasing the number of corner detections but was the method that is of key interest for dealing with images that contain uneven blur. Overall, using the improved heuristics, this implementation worked well for low resolution, severely distorted, and blurred images. In the case of images from a hyperbolic setup, the mean corner inaccuracy was reduced from 1.48 to 0.62px over the OpenCV checker detection method.

As a new method for increasing checkerboard calibration robustness, Duda *et al.* [31] exploited the symmetry of checkerboard corners using a method that resembles a localized Radon transform [27] with large box filters to better detect checkerboard corners. Using this method, improved sub-pixel accuracy could be achieved under a variety of conditions such as motion and Gaussian blur. Their work was tested on synthetic data that was perturbed by sub-pixel shifting, rotation, and perspective, as well as Gaussian blur and noise. The proposed method was robust to low contrast, image noise, and rotation, and also maintained high subpixel accuracy. However, it resulted in greater error under increased defocus blur when tested against the standard OpenCV "findChessboardCorners" function (Figure 3.3). It should be noted that the opposite result of Figure 3.3 was later observed by Choi *et al.* [26]. It should also be mentioned that Duda *et al.*'s method was also implemented as an available function within OpenCV as "findChessboardCornersSB" [5].

Figure 3.3. has been removed due to copyright restrictions. See Figure 5 (a) in [31]. Localization error of method proposed by Duda et al. [31] vs. Open CV checkerboard method.

Choi *et al.* [26] proposed a calibration method that captured images of a grid of circles that were projected using a digital screen. The screen would consecutively display progressively smaller concentric circles to be captured. This method is noteworthy because it falls under the active calibration method category but does not employ phase shifting. Using the consecutive images of the circle grid, the circles were captured in a memory buffer and then stacked to form 3D cone profiles. Using these profiles, from a top-down view, x & y slices were taken to create 2D profiles of the cones. Polynomial fitting was then performed on the 2D profiles to estimate the vertices, which would localize the control points for calibration. In comparison to phase-shifting methods, this method was chosen to improve calibration under severely blurred conditions and also to reduce inaccuracies that could be attributed to variations in lighting and distortion caused by perspective. Through a series of three experiments, the active circle grid method was compared against two reference checkerboard calibration methods. One checkerboard method was the standard OpenCV

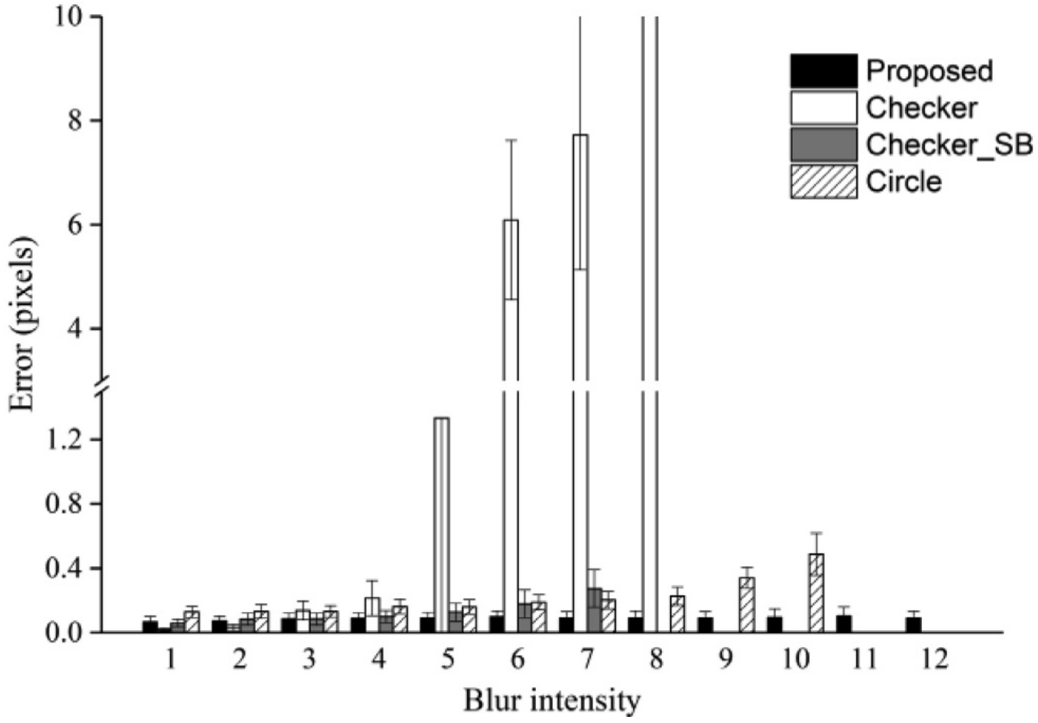


Figure 3.4. Localization error of method proposed by Choi et al. [26] vs. Open CV Circle and Checker as well as Checker_SB, which is the method proposed by Duda et al. Reproduced with permission from Elsevier.

implementation, while the other method, referred to as "Checker_SB" [26] as previously mentioned, was based on the method presented by Duda *et al.* [31]. One of the three experiments was designed to test accuracy against 12 different levels of artificially generated Gaussian blur. From this experiment, it was possible to compare the proposed consecutive circle method to the standard OpenCV-based checkerboard and circle implementations and Checker_SB. It was found that the checker methods [5, 31] performed better up to a certain level of blur. Beyond a blur intensity of 2, Checker_SB greatly outperformed the OpenCV checker method and performed well up until a mid blur level. However, all checker methods would fail past a certain point (Figure 3.4). Although the proposed consecutive circle method was less accurate than the checkerboard methods under low blur conditions, it performed consistently well even when blur was increased. Although the localization error for the proposed method was stated to have remained stable at around 0.88px, from the provided bar chart (Figure 3.4), it appears that the authors may have instead intended

to state that it was 0.088px on average across all blur conditions.

3.2 Blur Characteristic and Depth from Defocus Methods

Another approach is to estimate or model the blur characteristic itself and, in doing so, find ways to extract edges or tangible points that can be used to locate control features. Baba *et al.* [11] proposed a geometric approach for approximating the width of blur, from which to find the edges of a feature (such as a checkerboard checker or circle) in order to locate centers or intersection points. This method accounted for lens thickness and added a blur width parameter to be minimized upon the basis of Tsai's calibration method [75]. Another perspective can be seen in the work of Mannan, and Langer [60] where the objective is to perform better Depth from Defocus (DFD) detection. Typically, this problem entails capturing two defocused images with two different camera settings to infer defocus depth. In this example, calibration of point spread functions (PSFs) is performed to model how a point light sources appear under different camera settings for accurate blur kernel approximation. Similarly, prior work from Subbarao *et al.* [71] discusses two methods of PSF recovery, one using a simple approximation with only the blur parameter σ as an input, and the other using a more complex approach that takes a detailed mathematical model into consideration with several additional input parameters. Although neither of these DFD methods are directly related to recovering camera calibration parameters, Zhang *et al.* [86] proposed an algorithm that takes two arbitrarily defocused images taken with different camera settings to directly recover calibration parameters. This proposed method estimates the difference in blur between the two defocused images and also takes a geometric approach to calibration by considering thin lens geometry.

3.3 Active Display and Phase-Shifting Methods

In this section, we explore recent work in chronological order starting from 2015 and onward to review approaches that use active display calibration methods [13, 17, 24, 38, 46, 77, 78] that are robust to out-of-focus blur. Active calibration targets use a digital screen to display consecutive images that are combined to recover control points more accurately. Some active methods [13, 38] display consecutive shapes or patterns to generate intersections of lines or directly locate points, while others work with phase shifting methods where special

patterns and phase recovery methods are used to accurately estimate control points under blurred conditions [17, 24, 46, 77, 78]. Generally, the phase shifting methods that have been encountered have used some variation of square or circular patterns. It should also be noted that some phase shifting methods can be decoupled from following an active calibration approach. In this case, a single image can be printed and used to construct a planar calibration pattern. This case is also briefly discussed in this section.

3.3.1 Square Patterns

Following the assumption that defocus blur could be modeled as a convolution of the focused image and a Gaussian point spread function (PSF), Ha *et al.* [38] proposed a method using defocused unidirectional binary pattern images and the PSF to robustly detect feature points under defocused conditions. In a series of physical experiments, an iPhone screen was used to consecutively display square binary patterns in front of a stereo camera setup. Both cameras were focused at infinity, and the iPhone display was placed within a very close distance to capture the severely defocused images. With this image capture setup, image pre-processing and initial feature detection are first performed, followed by solving a 1D Gaussian deconvolution problem with subsequent sub-pixel refinement. The 1D Gaussian deconvolution problem involves modeling the blur gradient from a set of complementary binary patterns and overlaying their gradient profiles to find the intersection, which represents a line that will define control points. Refraction correction is also performed considering Snell's law to account for the glass thickness of the iPhone display. The results were also obtained by testing different approaches using real data and synthetic data.

The proposed approach was tested against three other known checkerboard approaches, and one method known as "zero-crossing" [38]. The proposed method demonstrated lower error than the other tested approaches and was noted as being 0.12 px for single-camera calibration and 0.34 for stereo, without refraction correction. Another scatter plot was also presented that showed the accuracy of each feature refinement method plotted against increasing standard deviation of Gaussian blur in pixel units. Although specific numbers were not provided, in synthetic experiments, the proposed method can be seen as having less than 0.025 across all blur levels of the mean distance error in pixels from a ground truth location of a single known feature.

As in Ha *et al.*'s work [38], Bell *et al.* [13] also adopted a calibration approach using a digital display and square binary patterns. Their choice of square binary patterns, as opposed to sinusoidal patterns, was motivated by using the highest possible contrast patterns

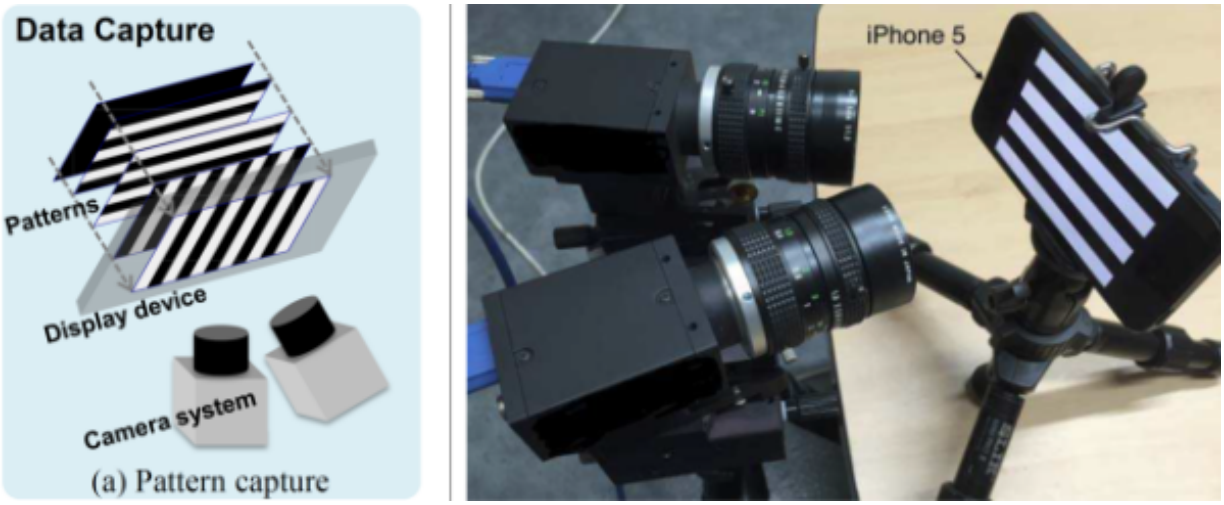


Figure 3.5. From the work of Ha et al. [38] (left) illustration of capture setup and unidirectional binary images to be successively displayed (right) setup with iPhone screen with stereo camera setup. © [2015] IEEE.

under defocused conditions, and being invariant to the nonlinear gamma of an LCD screen. This proposed method also uses a slightly different approach as it incorporates grey-coded vertical and horizontal phase maps to encode the feature or control point locations. Using a phase-shifting method to recover a wrapped phase, they also distinguished between spatial and temporal phase unwrapping, where temporal phase unwrapping avoids referring to relative phase information from neighbouring pixels and allows for the unwrapping of absolute phase values. This distinction led to their choice of the temporal phase shifted method instead of a spatial phase-shifted pattern. To recover intrinsic parameters from located control points, this research also used the standard camera calibration method built into OpenCV. The maximum difference between focal lengths of severely blurred and focused images was stated to be 0.2%. We would also like to note that a similar previous method from Schmalz *et al.* [68] was also found, and although this work does not specifically address out-of-focus blur, it does also address the topic of ray offsetting due to glass plate refraction. This work also provides publicly available code [3].

3.3.2 Circular Patterns

Another method that is well researched is to use phase-shift circular grating (PCG) patterns. PCGs are another type of pattern that are stated to work well because they are

mainly invariant to severe blur allowing for accurate control point localization under defocus conditions. The methodology behind PCGs is to capture images of radial gradient patterns that are based on specified shifted phase values. From these shifted phase patterns, a wrapped phase pattern can be calculated, which contains a pronounced edge, essentially the point or circle in the case of PCGs where maximum intensity meets minimum intensity. A circle or ellipse fitting technique can then be used to fit a circle or ellipse around the pronounced edge. Ellipse fitting techniques are preferred when the calibration pattern plane is on an angle and exhibits perspective distortion. Once the circle or ellipse has been fitted, it then becomes possible to locate the center of the pattern. The following presented work is based on variations of this approach.

Wang *et al.* [78] proposed an active calibration method using a laptop screen displaying three consecutive grids of PCG pattern arrays. Grids of circular PCG patterns are organized using specific rules for spacing distance, which can aid programmatic implementation.

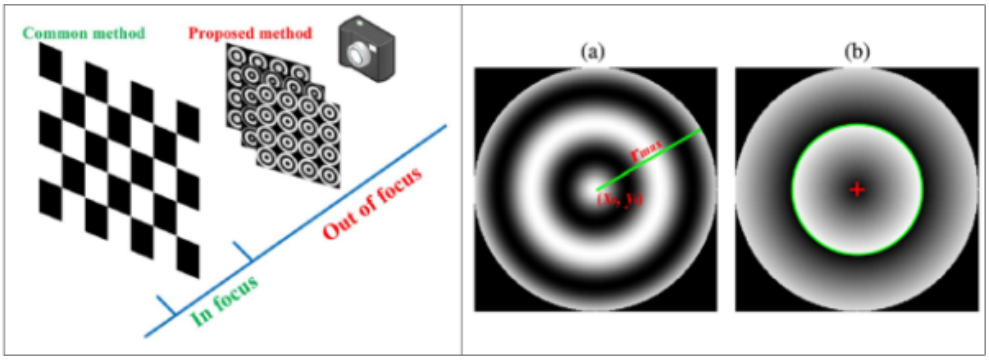


Figure 3.6. Proposed method from Wang et al. [78] (left). An image of PCG pattern, (right "a") and wrapped phase with detected circle and center (right "b"). Adapted with permission from [78] ©Optical Society of America.

Using a phase shift of $2\pi/3$, the following set of equations is used to describe the pixel intensities for each of the three phase shifted patterns. Equation (3.1) from Wang *et al.* [78] describes the brightness intensities for each pixel of each of the three phase shifted patterns:

$$\begin{cases} I_1(x,y) = A + B\ cos[\phi(x,y) - 2\pi/3], \\ I_2(x,y) = A + B\ cos[\phi(x,y)], \\ I_3(x,y) = A + B\ cos[\phi(x,y) + 2\pi/3], \end{cases} \tag{3.1}$$

where, A and B are constants, and $\phi(x, y)$ is the unwrapped phase described by equation (3.2). In the equation for the unwrapped phase (3.2), T is the PCG period, and (x, y) are the Euclidean distance between any point on the PCG pattern and its center:

$$\phi(x, y) = \frac{2\pi r(x, y)}{T}. \tag{3.2}$$

Once the images of the grids of phase shifted patterns are captured, the following equation (3.3) from Wang *et al.* [78] is used to create the wrapped phase where $J_i(u, v)(i = 1, 2, 3)$ and i is each of the three phase patterns:

$$\psi(u, v) = arctan\left(\sqrt{3}\frac{J_1 - J_3}{2J_2 - J_1 - J_3}\right), \tag{3.3}$$

Through a series of steps that involve erode and dilate functions, Sobel edge detection, and least squares ellipse fitting, the wrapped phase of the PCG is used to calculate center points and perform subsequent sub-pixel optimization.

Group	Pattern	f_u	f_v	u_0	v_0	MRE
First	PA	1543.5	1543.4	972.1	619.8	**0.0788**
	CB	1543.7	1543.6	971.2	618.6	0.0845
Second	PA	2633.4	2633.2	979.1	597.1	**0.0779**
	CB	2484.8	2483.4	978.4	581.7	1.1080
Third	PA	4543.8	4544.3	993.7	554.3	**0.0755**
	CB	5753.9	5758.2	981.1	572.5	7.2483

Table 3.1: Real-world results from Wang *et al.* [78] where groups 1 to 3 are different focus levels, f_u, f_v are focal lengths in u and v directions, u_0 and v_0 are camera centers. PA and CB are PCG arrays and checkerboards respectively. Reprinted with permission from [78] ©Optical Society of America.

Real experiments (Table 3.1), as well as experiments using synthetic data (Figure 3.7), are conducted. All programmatic work was conducted in MATLAB using the Image Processing Toolbox and Camera Calibration toolbox [1, 4] . For real world experiments, indoor lighting was turned off. The synthetic data experiments used images with varying standard deviations of Gaussian blur to create several levels of artificial blur, and were tested against a checkerboard for comparison. Accuracy was captured in mean reprojection error (MRE). The results of synthetic experiments showed that the PCG array method returned an MRE

of 0.08px under focused conditions, which was slightly better than the checkerboard, and stayed at 0.08px as Gaussian blur was introduced. Conversely, the checkerboard error increased significantly as blur increased (Figure 3.7). Also, it was noted that the chosen period needs to be suitable for circle detection, and 2π phase-detection leads to the highest accuracy. Finally, several sizes of PCG array were tested. Overall, it was determined that a larger array of 8×8 PCGs was best for returning the lowest MRE.

It should also be noted that in (Table 3.1), the focal lengths for each group are different. From these results, we see that the objective was to demonstrate minimization of MRE. The stable MRE is promising in both Table 3.1 and Figure 3.7. However, the differing focal lengths are not overly meaningful since one group cannot be compared as a ground truth to the others, this reflects the experimental setup, and therefore these results require further investigation to determine if repeatable focal lengths (reflecting overall calibration accuracy) can be returned under out-of-focus conditions.

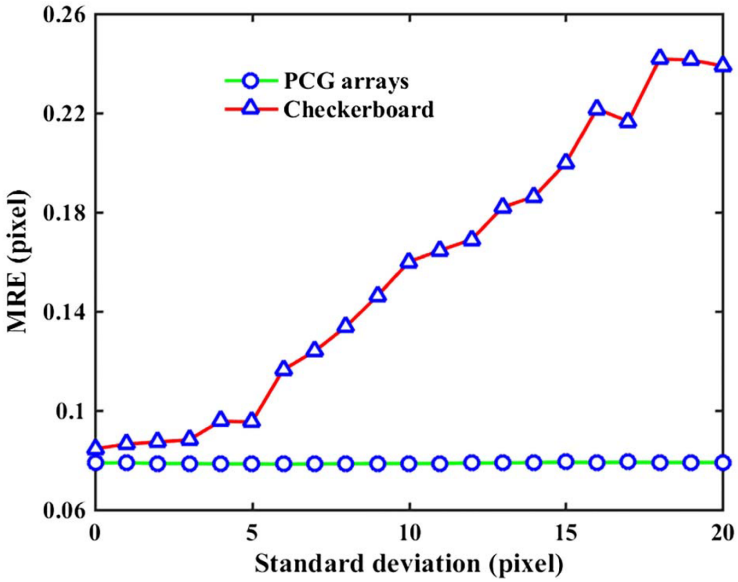


Figure 3.7. Mean reprojection error from synthetic experiment using PCG method proposed by Wang et al. [78]. Proposed method holds at 0.08px MRE under increased blur. Reprinted with permission from [78] ©Optical Society of America.

Similar to the previous work of Wang *et al.*, using the same PCG method, Cai *et al.* [17] presented an improved method for circle center localization using spatial projection relationships of concentric circles. They also used canny edge detection instead of Sobel and presented a sorting algorithm for individually characterizing each PCG target in the

array. Also, they conducted real and synthetic experiments and in the synthetic case, used images of the PCG arrays from 6 different orientations. Three different types of tests were conducted to study the influence of PCG array size, PCG period, and robustness to noise. Results were captured as a relative error (%), absolute error (px), and root mean square reprojection error (RMSE). For real experiments, a mechanical lift and turntable were used to orient a camera at different angles between the camera's optical axis and a computer monitor. Tests were conducted using in-focus and varying degrees of blurred images. In blurred cases, arrays of concentric circle patterns were tested against PCG patterns both at a number of different angles, and the RMSE and focal lengths were reported (Table 3.2). Under perspective and blur, the PCG arrays were found to outperform the concentric circle arrays, and RMSE varied from 0.045 to 0.057 under different blur conditions. In contrast, the concentric circle array started with an RMSE of 0.054 under low blur and increased significantly to 0.179.

	Pattern	f_u	f_v	u_0	v_0	k_1	k_2	RMSE
Trial 1	PCG arrays	2748.012	2747.892	982.443	616.473	−0.010	0.099	**0.045**
	Concentric circle array	2745.681	2745.370	982.433	615.732	−0.012	0.103	0.054
Trial 2	PCG arrays	2732.675	2732.785	980.751	615.591	−0.012	0.105	**0.048**
	Concentric circle array	2721.353	2720.358	972.573	609.420	−0.041	0.182	0.136
Trial 3	PCG arrays	2674.015	2674.918	974.139	614.406	−0.047	0.174	**0.057**
	Concentric circle array	2680.222	2678.179	974.413	616.083	−0.060	0.243	0.179

Table 3.2: Real image calibration results from Cai *et al.* [17], where Trial 1 is using focused images, and Trials 2 and 3 have increasing levels of out-of-focus blur. Reproduced under the terms and conditions of the Creative Commons Attribution (CC-BY) license https://creativecommons.org/licenses/by/4.0/

Extending their original work [78], Wang *et al.* used only a single captured image of a color PCG array from each pose to perform out-of-focus calibration [77]. This work is motivated to reduce the time required for calibration from multiple images and to reduce potential variations due to changes in lighting or other disturbances such as vibration over time. To accomplish this, three color channels of the PCG arrays are first calculated. Unlike in Wang *et al.*'s earlier work [78], the PCG arrays are not used to make separate images that will be captured, but instead, each of the three arrays are used for the separate RGB color channels that are merged into a single-color image array containing all three phases. An added advantage of having only one merged color array is that there is no requirement for a digital display like with other active methods. Therefore, the array could optionally be printed to construct a passive planar calibration target.

Using the color array, the wrapped phase can be recovered along with its 2π discontinuity. Since the circle of the 2π discontinuity can only be produced with pixel-level accuracy, sub-pixel optimization using linear least-squares fitting is performed to locate the circle more accurately. Subsequently, an ellipse is fitted based on the refined circle, and the refined center point is located based on the ellipse.

For validation, two experiments were conducted. In the first experiment, the color PCG array was tested against a symmetric circle grid pattern using the standard circle calibration method from OpenCV. Although not necessary, the arrays of each pattern were still displayed on a computer monitor, and 20 poses of each were captured for three different levels of blur. For the first set, the images were nearly in focus and the color PCG array with an RMSE of 0.065 slightly underperformed against the circle method, which returned an RMSE of 0.062.

Under blurred conditions, the color PCG method significantly outperformed the circle grid, and under the highest level of defocus, returned an RMSE of 0.081 compared to 0.147 from the circle grid method. For the second experiment, since it had been stated that the color PCG arrays could achieve better performance at longer distances, three sets of images were captured at three different distances by moving the monitor and keeping the camera pose and intrinsics fixed. This also produced three levels of defocus and was found to return a relatively consistent $0.06 + / - 0.005$ RMSE value.

Cai *et al.* [18] proposed a calibration method, which relied on similar PCGs as in their previous work [17], and Wang *et al.*'s work [78], and was deemed to work effectively under blur. This proposed method relies on both detection of PCG target centers as well as vanishing line estimation. The effectiveness of this method was confirmed through experiments where the proposed method was compared to Wang *et al*'s previous results [78], and the results were found to be similar. It is unclear however if either slightly improved or worsened results were achieved between the proposed and reference method. Although new results were published, it is also unclear if there was any difference from the method proposed in Cai *et al.*'s initial work [17].

In relation to previous PCG based methods, Chen *et al.* [24] proposed a novel approach using phase-shift wedge gratings (PWG) and rotation invariant local binary patterns (LBP) as another calibration pattern method. Like previous methods, this method is also robust to blur, and works even if part of the image falls outside of the field of view (FOV) and is cut off. It was also suggested that this proposed method could be used in both monocular and multi-camera applications. Using a similar set of phase shift pattern equations, circular PWG patterns are generated that have gradients that vary with angle of rotation around the center. From these patterns a wrapped phase is calculated allowing for lines to be

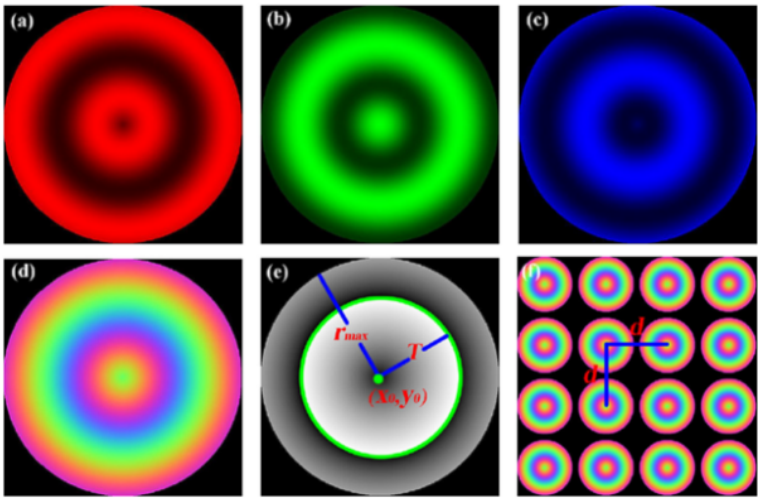


Figure 3.8. From Wang et al. [77] images (a to c) are the RGB PCGs and (d) is the PCG consisting of the combined RGB channels. Image (e) is showing the wrapped phase with fitted ellipse, and (f) is depicting the full array of PCGs with separation distances. Reproduced with permission from Elsevier.

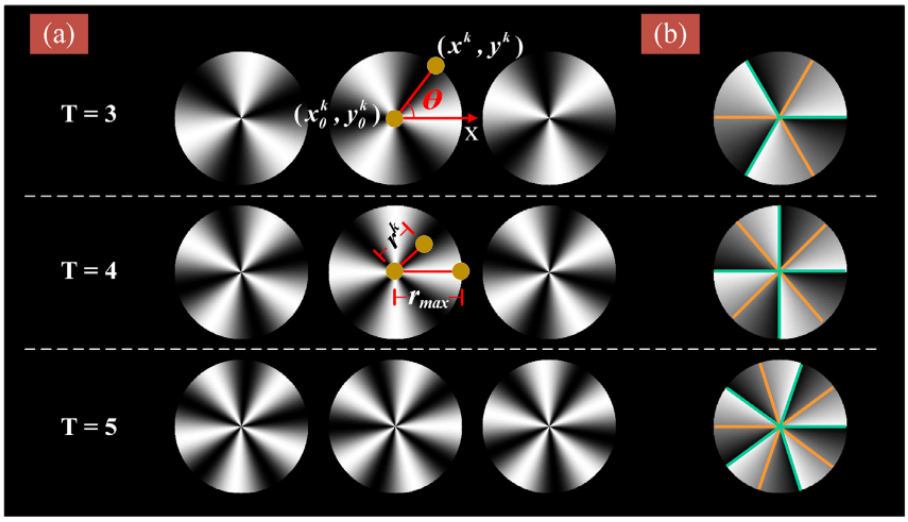


Figure 3.9. From Chen et al. [24] (a) PWG circles with different T period values and (b) their wrapped phase images with fitted lines. Reproduced with permission from Elsevier.

detected for the number of phase discontinuities that exist in the pattern (Figure 3.9). The effectiveness of three different line fitting techniques was investigated. After testing

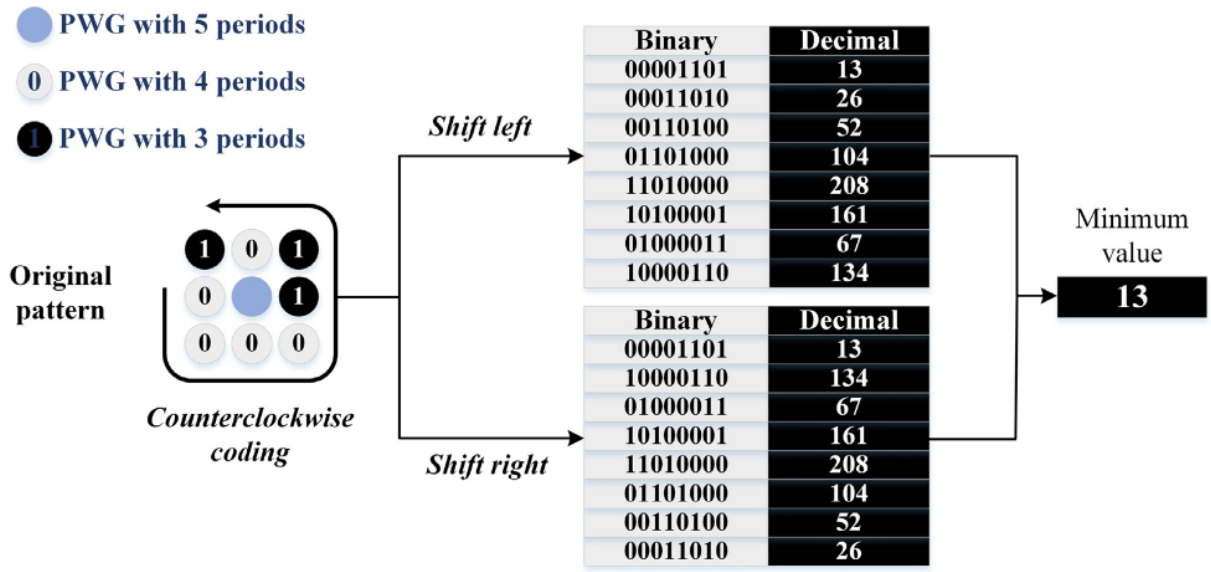


Figure 3.10. From Chen et al. [24] local binary patterns (LBP) encoding scheme based on Figure 3.9. Reproduced with permission from Elsevier.

RANSAC, least squares fitting, and Hough Transform under different conditions, it was determined that least squares fitting returned the lowest extraction error and was selected as the best line fitting method. Using this method, the lines are first detected with pixel level accuracy and then undergo further sub-pixel optimization. Using the average value of the intersections of the optimized lines, a control point is then estimated.

To encode relative location into the patterns, 3×3 arrays of PWGs are created with (T) period values of 3, 4 or 5. In each 3×3 grid, the $T = 5$ pattern is always placed in the center, and coded arrangements of $T = 3$, and $T = 4$ patterns are placed around the center pattern where $T = 4$ patterns are equal to binary 0, and $T = 3$ patterns are equal to binary 1 (Figure 3.10). Using these coded arrangements from the 9 PWG (3×3) patterns, a clockwise or counterclockwise search can be performed to compute the lowest decimal value using the 8 surrounding PWG pattern, or 8-bit binary equivalent bits. This search is performed 8 times from each possible starting position of the surrounding PWG patterns. The lowest value of the search then represents a keyword for the (3×3) array. The location of the 3×3 array allows for images to be taken of a much larger grid of PWGs, which was stated to be useful for stereo camera calibration with non-overlapping FOVs.

In an initial experiment, a 9×12 PWG grid was displayed on an iPad screen, and a camera was placed at close proximity to the screen so that only part of the screen was in the

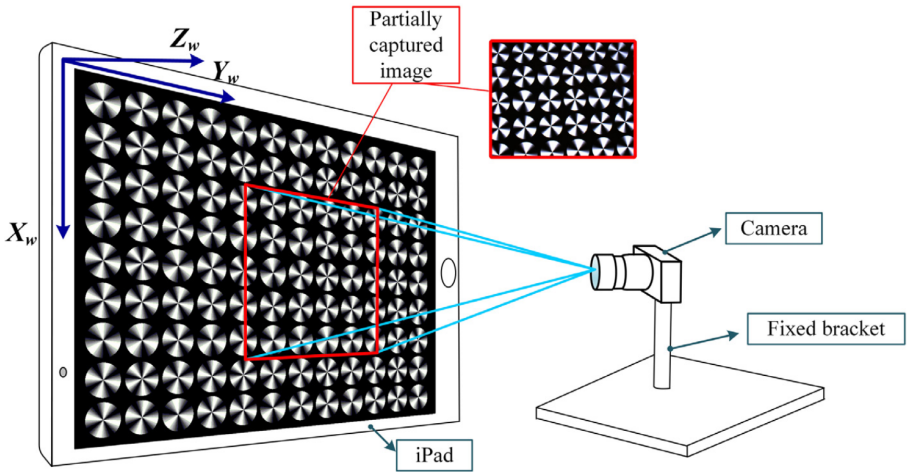


Figure 3.11. From the work of Chen et al. [24], this illustration shows the experimental setup using an iPad and image capturing of the LBP encoded PWG grid where partial images were captured at different rotations about Z_w & X_w. Reproduced with permission from Elsevier.

Distance(cm)	f_u	f_u	u_0	v_0	d_1	d_2
13	1723.6	1722.8	637.6	530.8	−0.128	0.291
20	1725.9	1726.6	639.4	530.1	−0.115	0.221
25	1717.3	1719.0	639.0	532.6	−0.110	0.201

Table 3.3: Results of camera calibration from real images from Chen *et al.* [24], where 20 cm is in-focus and 13cm and 20cm are out-of-focus. *Reproduced with permission from Elsevier.*

FOV. Using a number of rotations that were applied to the iPad screen, images of the PWG grid were captured to perform calibration. Around the optical axis, the screen was rotated at large angles of 0°, 90°, 180° and 270°. Under these conditions, it was found that if only a single keyword coded 3×3 array was visible in the larger array, no matter the orientation, it was still possible to accurately detect a set of control points. This characteristic was found to be helpful and added flexibility under varied image capturing distances where only part of the pattern is visible. To test the effectiveness of their method on out-of-focus images, in their second experiment, the camera was fixed in place and the camera's internal parameters were not changed. It is important to note that this is one of the few experiments that follows a similar method to ours where camera parameters were explicitly fixed between capturing images at different focus distances. In this second experiment, the

iPad was moved to three different distances 13cm, 20cm and 25cm where 20cm was for capturing the in-focus set of images. Results of this experiment can be seen in Table 3.3. Further experiments were performed with comparisons to circle and checkerboard patterns, and it was reported that the proposed method performed well under blurred conditions returning an error in the order of 0.04px as a standard deviation of projection error. The setup for these experiments is shown in Figure 3.11.

3.4 Deep Learning Methods

According to Liao *et al.* [55], camera calibration research dates back as far as 60 years. Not surprisingly, an extensive amount of research has been produced that analyzes different facets of this domain within the field of computer vision. Depending on the application, some broader facets of research include improving conventional camera calibration methods or models, finding ways to deal with specific kinds of distortion and noise, multi-view and stereo camera calibration problems, and registration with other sensors such as lidar. These areas of camera calibration research have been approached using analytical solutions as well as learned methods based on machine learning (ML), and deep learning (DL) algorithms.

Camera calibration can also be classified into two general categories. The first category is calibration using targets with patterns, and the second category, is self, or auto-calibration. Calibration using targets is the most common and well-established camera calibration method. However, some drawbacks of non-learning target-based calibration methods such as Zhang's method include difficult setup, and laborious manual intervention. Also, low accuracy or calibration failure can occur when dealing with out-of-focus and motion blur if poor feature or control point correspondences are provided as inputs.

With major advances in DL [69], a growing number of researchers have opted to experiment with DL as a complimentary way to enhance Zhang's method, or even bypass it all together by creating an end-to-end learned approach. Further motivating these efforts are potential areas of improvement such as reduced operator error, improved generalizability, and potential reductions in the number of required calibration images to be captured.

Target based methods are also limited in use, and cannot be used in applications where camera parameters are dynamically changing [47]. To solve this problem another branch of camera calibration techniques known as online, self, or auto-calibration methods are also being actively researched. Self or auto-calibration is a process where calibration is performed using natural scene features in images such as planes or vanishing points [15, 37], or by inferring relationships through an end-to-end data driven process such as through DL [43, 57, 82].

With these considerations in mind, the following section, aims to explore learned methods that are either directly related to calibration from out-of-focus images, or follow different approaches such as methods that relate to auto-calibration. Our hope is to provide details on contrasting learned methods to inspire consideration for how they could be adapted to perform out-of-focus calibration tasks. In this light, the following example based on earlier literature illustrates how calibration can be performed using a learned method while entirely bypassing the often-preconceived necessity to find intrinsic and extrinsic parameters.

Zou *et al.* [90] performed calibration for vision-based robot localization without requiring extrinsic or intrinsic parameters such as K (2.2). Using a mobile robot, markers are positioned on the floor with known 2D floor coordinates in robot space. As the robot moves, images are collected from the robot's camera. The camera is angled downwards towards the floor, and the respective 2D image coordinates of floor markers are identified. Using these corresponding sets of points: (2D floor marker positions in robot space, respective 2D floor marker positions in image space), both sets of points are provided to a simple multilayer perceptron (MLP) for training. The trained MLP is then used to transform new image space points to points in robot space. The new robot space coordinates can then be used with an extended Kalman filter (EKF) to track the robot's position (Figure 3.12).

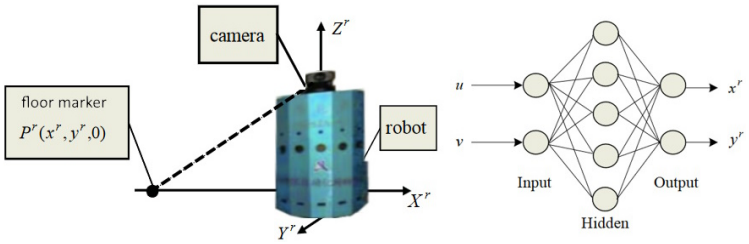


Figure 3.12. From Zou et al. [90], vision guided robot configuration using a simple neural network. Reproduced with permission from Springer Nature.

This method simplifies the process and entirely bypasses Zhang's method [87], requiring no calibration targets or time-consuming setup. Although this method cannot be evaluated against most other modern methods, it provides a simple example of how calibration can be learned in a fundamentally different way than through conventional analytical based methods.

3.4.1 Camera Calibration Using Deep Learning and Checkerboards

To improve checkerboard corner detection, Donné *et al.* [30] implemented MATE (machine learning for adaptive template extraction). A large example set of chessboard images was used to train a three-layer convolutional neural network (CNN) to recognize checkerboard corners. The training set included images that were perturbed with artificial noise and severe lens distortion.

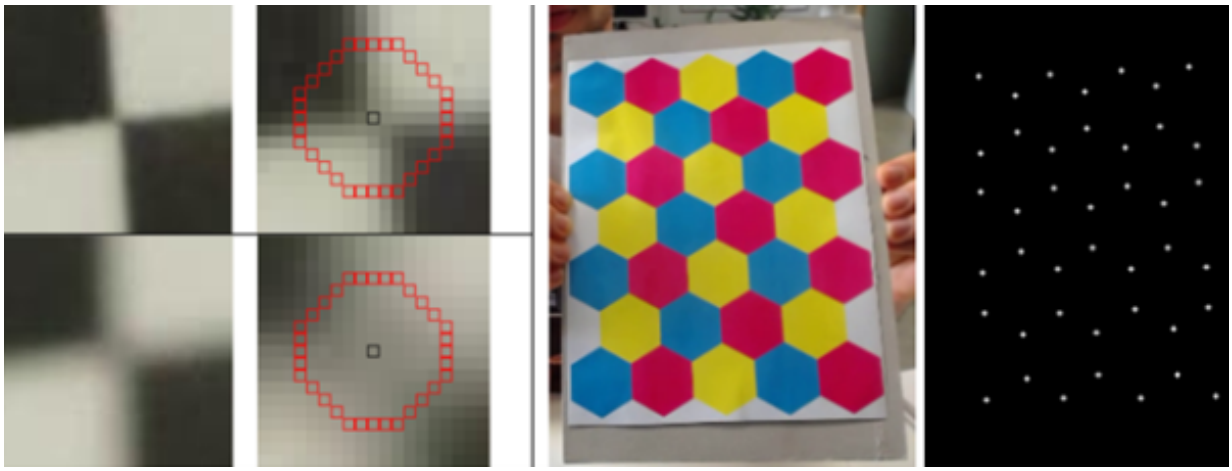


Figure 3.13. Work of Donné et al. [30] (left) showing checkerboard corner blur and (right) example of control point detection using CNN trained on hexboard pattern. Reproduced under the terms and conditions of the Creative Commons Attribution (CC-BY) license https://creativecommons.org/licenses/by/4.0/.

Inference of checkerboard corners was noted to have performed well against other known methods [14, 61]. Across four data sets, MATE returned a range of accuracies from 0.835 to 1.160 px. MATE was also adapted to other calibration pattern shapes. This was demonstrated separately with the network trained to recognize control points on a CMYK hexboard calibration pattern (Figure 3.13). MATE was not intentionally trained on heavily blurred images, however the authors noted that the radius of the convolution filter kernel in the first layer of the network could be increased to deal with more significant defocus blur.

Chen *et al.* [23] proposed checkerboard corner detection network (CCDN) (Figure 3.14) which was shown to improve robustness over MATE. This improvement was reported when comparing inference on two data sets (uEye and GoPro) that were also used by MATE. For CCDN, on the two data sets, the range of accuracies were found to be between 0.576 to

0.812px. To improve CCDN, a deeper network with more filters, improved learning rate, and loss function was chosen. For CCDN, cross entropy was selected to match the binary ground truth for center point detections. CCDN was also shown to reduce the number of overlapping and false detections than MATE, due to improvements of thresholds that were set for clustering as a post processing method. Also, Chen *et al.* [23] continued the discussion from Donné *et al.* [30] to highlight the trade-off of the first layer kernel size which is used to extract useful features from the input image. It was mentioned that the spatial support radius of the filters should be large enough to suppress the effect of blur and noise, but a larger radius could reduce recall of the real corners while a smaller radius could miscategorise background pixels as checkerboard corners. To solve this problem, Chen *et al.* [23] chose a filter radius that balanced this trade-off.

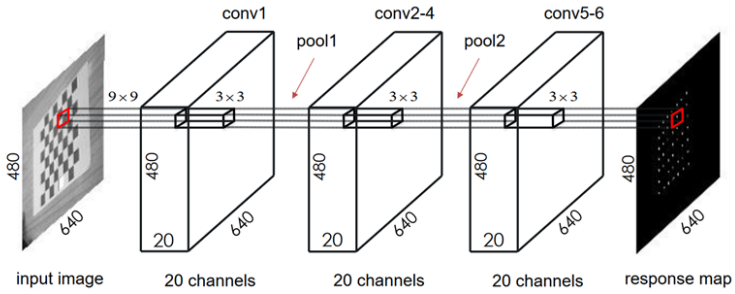


Figure 3.14. From Chen et al. [23], the fully convolutional network architecture of CCDN with six convolutional layers, ReLU activations, and max-pooling after layers 1 and 4. CCDN outputs a single-channel response map equal to the input size. Reproduced with permission from Springer Nature.

3.4.2 Camera Calibration Using Deep Learning from Blurred Images

To improve camera calibration from blurred or noisy images, Zhang *et al.* [85] proposed a multi-scale target enhancement network (MS-TENet) to recover sharp target images using blurred and potentially noisy images as an input. Using a supervised learning approach, MS-TENet was trained using 68 sharp checkerboard images in a variety of poses with varied angles and translations. The captured sharp images were synthetically blurred and re-scaled to generate a larger training set of 8160 images. Using a modified Unet architecture with fewer feature channels as well as added multi-scale feature extraction

and multi-scale residual blocks (MSRBs), MS-TENet was able to re-construct the sharp checkerboard images from defocused ones.

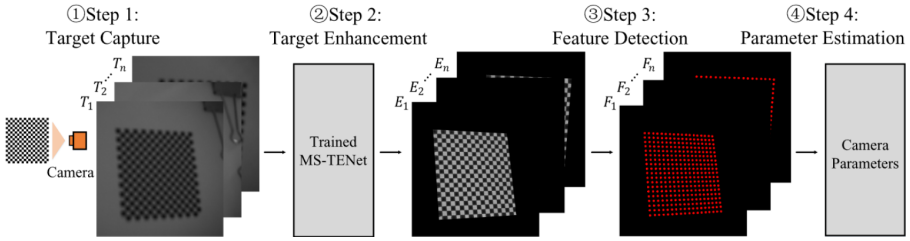


Figure 3.15. From Zhang et al. [85], calibration process using MS-TENet. Reprinted with permission from [85] ©Optical Society of America.

Camera calibration using MS-TENet was conducted with the following steps: (1) Given an uncalibrated camera, a checkerboard target was rotated and translated to create a variety of poses for images to be captured from. These images were also all captured with the camera being setup so that the camera was capturing images out-of-focus. (2), The trained MS-TENet used the out-of-focus images as an input and generated re-constructed focused images as an output. (3) Using Geiger *et al.'s* feature detection method [35], and the re-constructed sharp images, control points (which in this case are the intersections of checkerboard checkers) were detected. (4) Camera calibration was performed using the widely used method from Zhang [87] (Figure 3.15).

To evaluate the performance of MS-TENet, Zhang *et al.* [85] performed experiments using active and passive targets (AT & PT) which consisted of a 1440×900 resolution digital display that displayed an image of a checkerboard for the AT, and a printed checkerboard pattern attached to a carboard backing for the PT. Several tests were performed comparing the base Unet to MS-TENet for re-construction of sharp images and subsequent camera calibration using both AT and PT targets under three camera defocus levels where "Defocusing Degrees = $\sigma 2 < \sigma 1 < \sigma 3$". For the lowest level, calibration was performed without image re-construction / enhancement using only the feature detection and calibration steps which was labeled as the (None) method. As expected, the (None) method was not successful using the more severely blurred images. From levels $\sigma 1$ and $\sigma 3$, however, MS-TENet succeeded to deblur the images and obtain calibration results.

In the context of a tunnel boring machine (TBM) where a vision-based robot would be required to operate in a space constrained environment (Figure 3.16), Huo *et al.* [46] presented an alternative camera calibration method using DL and phase-shift circular gradient (PCG) patterns to perform calibration from blurred images.

Like in the past work of Cai *et al.* [17, 18], the proposed method employs the same PCG patterns with ellipse eccentricity correction. With Wang *et al.* [78], and Cai *et al.* [17, 18], three images of a grid of greyscale PCG patterns are displayed consecutively using a digital display screen. The three consecutive images (I_1, I_2, I_3) are iterated for every variation of screen or camera pose. From the captured images, a sharp wrapped phase image is computed using the three-step phase shifting arctan function (equation (3.3)) from the three PCG pattern images. The sharp image is used for ellipse fitting and center point (or control point) detection which is then passed to Zhang's method for calibration.

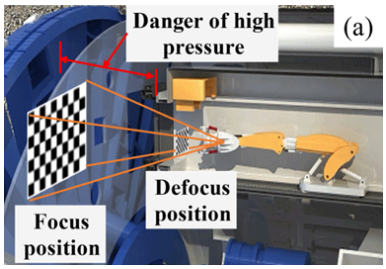


Figure 3.16. From Huo et al. [46], camera calibration in the context of a TBM. Reproduced with permission from Elsevier.

Unlike these past methods, the proposed method of Huo *et al.* [46] only requires one of the PCG pattern images (I_1 or I_2 or I_3) to recover the sharp wrapped phase image using a phase-recovery network or (Phase-Net) 3.17. Phase-Net is trained on real, and data-augmented sets of images of grids of PCGs. These sets are of the three images (I_1, I_2, I_3) of PCGs which are shifted $(0, 2\pi/3, 4\pi/3)$ respectively. The training labels are images of the respective wrapped phase calculated by equation (3.3). Phase-Net was built on a pre-trained VGG-16-BN backbone and was modified to perform saliency detection which essentially provides circular masking of the PCG circular targets under inference (Figure 3.17).

Phase-Net training was performed on a GPU accelerated workstation. The inference was performed on mid-grade consumer-level hardware in sub 1 second times. Figure 3.18 provides an illustration of Phase-Net vs. a traditional checkerboard-based calibration method.

Physical experiments were performed to compare Phase-Net with a traditional checkerboard calibration method using Harris corner detection, as well as a standard PCG method. The experiments were conducted using a tablet display screen so all four patterns could be projected successively to capture images of each pattern for every pose. The tablet was fixed in place, and a robot arm mounted to a multi-axis linear motion table, was used for camera positioning (Figure 3.21). Figure 3.19 shows the target patterns that were used.

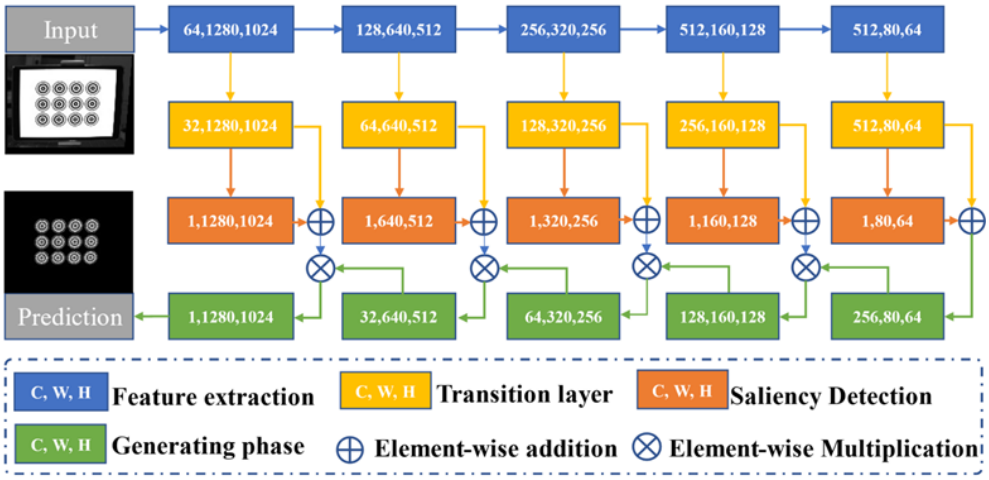


Figure 3.17. From Huo et al. [46], Phase-Net architecture. Reproduced with permission from Elsevier.

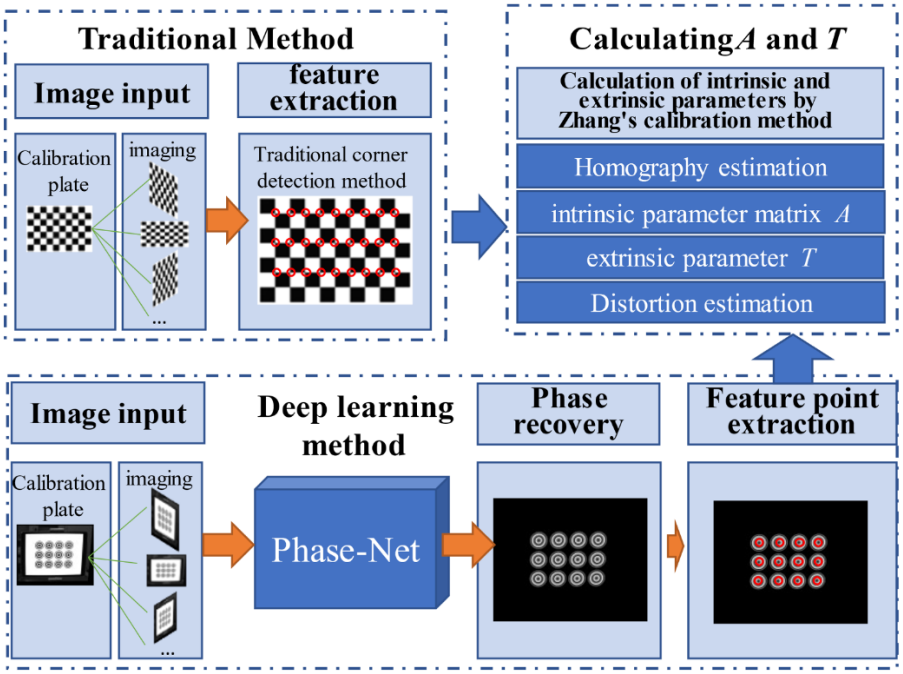


Figure 3.18. Method proposed by Huo et al. [46] vs. traditional checkerboard calibration method. Reproduced with permission from Elsevier.

Under focused conditions, both Phase-Net and the PCG method were shown to be inferior to the checkerboard method. However, under blurred conditions, both phase-based approaches continued to return consistent results, while the checkerboard method MRE measured in pixels increased significantly (Figure 3.20).

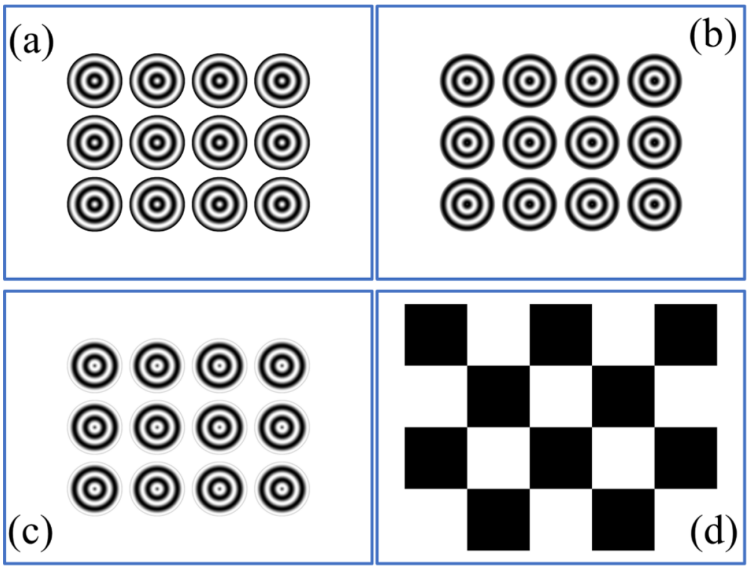


Figure 3.19. Calibration target patterns used by Huo et al. [46] where ***a****,* ***b****, and* ***c*** *are used for PCG calibration, only* ***a*** *is used for the proposed method, and* ***d*** *is used for the traditional Harris corner based method. Reproduced with permission from Elsevier.*

It is important to note, that this experiment was conducted by capturing a set of images from 13 poses in-focus. The out-of-focus cases were produced by Gaussian blurring the real-world images with successively increasing blur kernel sizes. This may not produce the same blur conditions as in real life since blur is affected by changes in perspective. The comparison of camera parameters from the two phase-based methods are shown in (Table 3.4).

It was also observed that the PCG method performed slightly better than Phase-Net in out-of-focus testing. The maximum relative error between Phase-Net and the PCG method was reported to be 1.54%, while the observed MRE values in pixels provided on the chart (Figure 3.20) for the PCG method were roughly 0.04 and for Phase-Net on-average 0.07. Also, for Phase-Net, a maximum MRE of 0.11 pixels was reported for its performance under varied blur conditions. Experiments on the proposed Phase-Net method were also performed to evaluate the influence of changes in lighting and spatial resolution. In this

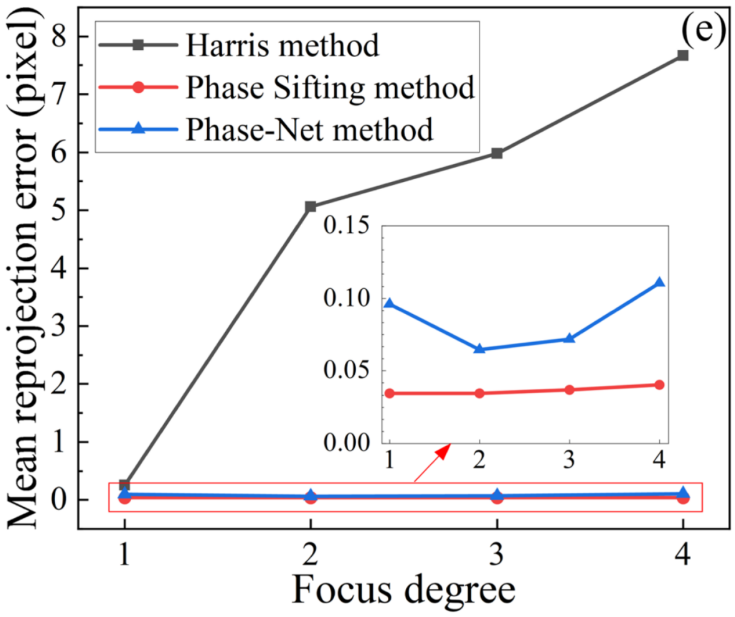


Figure 3.20. Results from Gaussian blurred real-world captured images from Huo et al. [46] where both phase-based methods outperform the checkerboard-based Harris corner method when images are Gaussian blurred. Reproduced with permission from Elsevier.

case, the target distance was varied to three distances, and the camera was re-focused at each distance. These three cases were repeated under different lighting conditions with the camera aperture adjusted accordingly. The results showed that this method was highly insensitive to changes in lighting because the PCG calibration patterns were always being displayed using a backlit display screen. The maximum variation in MRE due to differences in lighting was recorded to be 0.08 pixels.

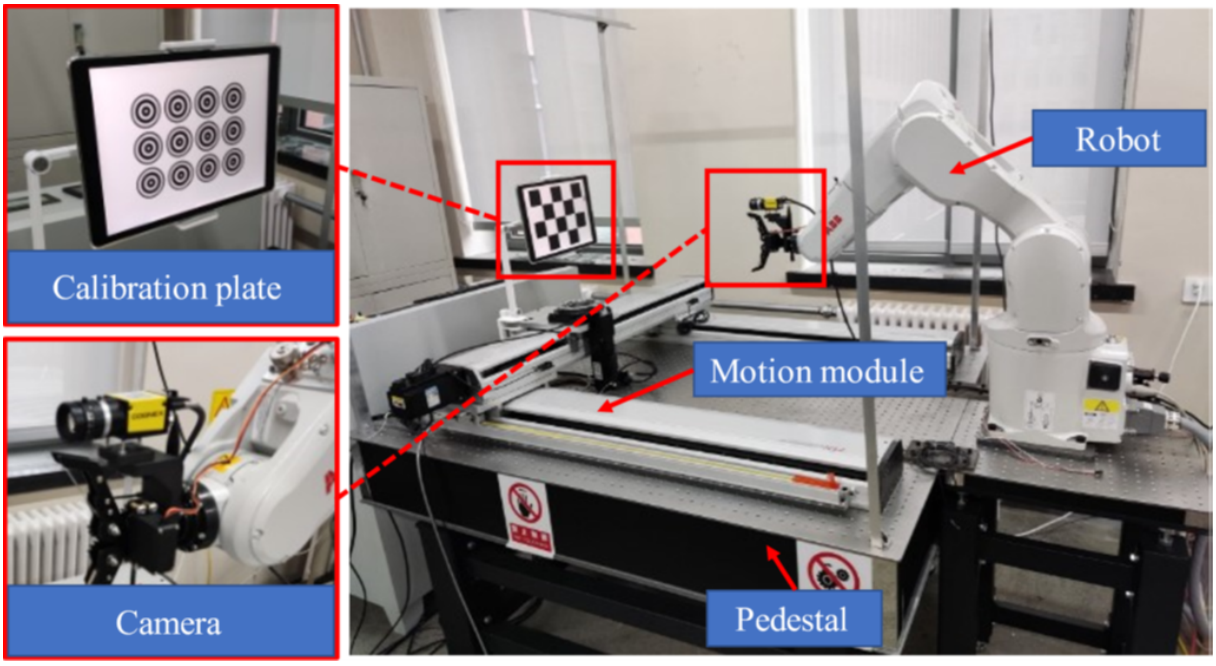


Figure 3.21. From work of Huo et al. [46] defocus calibration test bed. Reproduced with permission from Elsevier.

3.4.3 Auto / Self, and Other Deep Learning Based Calibration Methods

Instead of learning a fully end-to-end representation of camera parameters, Butt *et al.* [16] proposed a novel camera projection loss (CPL), and camera model neural network (NN) to estimate proxy variables (relating to multi-view geometry), and reconstruct 3D points. Using the reconstructed 3D points and a calculated reconstruction loss, the camera parameters are then estimated. The advantage of this method is that it can jointly estimate the intrinsic and extrinsic parameters using a multi-task learning (MTL) approach that considers the underlying analytical equations in its learning (Figure 3.22).

Additionally, Butt *et al.* provides a CVGL camera calibration dataset using the CARLA simulation which is specifically designed for their work, where 10 camera parameters (including intrinsic and extrinsic parameters) are being simultaneously estimated, which is a greater number than from previous data sets. This dataset was also integral for training their MTL method. With raw image inputs, an Inception-v3 model, pre-trained on ImageNet, was first used as a first stage feature extractor, this network was trained to return

Focus status number	method	α (pixel)	β(pixel)	u_0(pixel)	v_0 (pixel)
1	Phase Sifting method [7]	2426.65	2426.48	646.52	498.53
	Phase-Net method	2434.00	2433.80	636.58	505.59
	relative error	0.30%	0.30%	1.54%	1.4%
2	Phase Sifting method [7]	2426.65	2426.48	646.52	498.53
	Phase-Net method	2423.27	2423.23	640.02	495.07
	relative error	0.14%	0.13%	1.01%	0.69%
3	Phase Sifting method [7]	2420.81	2420.51	643.51	499.08
	Phase-Net method	2422.90	2423.00	639.60	501.18
	relative error	0.09%	0.10%	0.61%	0.42%
4	Phase Sifting method [7]	2424.53	2424.15	643.12	501.40
	Phase-Net method	2420.55	2419.20	643.80	504.35
	relative error	0.16%	0.20%	0.11%	0.59%

Table 3.4: Results from Gaussian blurred real-world captured images from Huo *et al.* [46] showing the recovered camera parameters and relative errors. *Reproduced with permission from Elsevier.*

13 variables, 10 of which were proxy variables that do not directly relate to visible image properties but are related to each other and correspond to variables of a multi-view geometry model. The other 3 variables were trained to directly approximate 3D feature point coordinates. Using the 13 variables as input, the second stage is to run a Camera Projection Loss network which has an architecture that is designed to predict camera parameters with different magnitudes. To calculate loss, a single metric is used that compares the estimated camera model network parameters to known parameters to calculate the error of the 2D to 3D projections of points. Where $\omega = (f_x, f_y, u_0, v_0, b, d, \theta_p, t_x, t_y, t_z, X, Y, Z)$ are known parameters and: $\hat{\omega} = (f'_x, f'_y, u'_0, v'_0, b', d', \theta'_p, t'_x, t'_y, t'_z, X', Y', Z')$ are the parameters predicted by the NN. Using the parameters, the error function as per equation (3.4) becomes [16]:

$$L(\omega, \hat{\omega}) = \frac{1}{n} \sum_{i=1}^{n} MAE(\omega, \hat{\omega}). \tag{3.4}$$

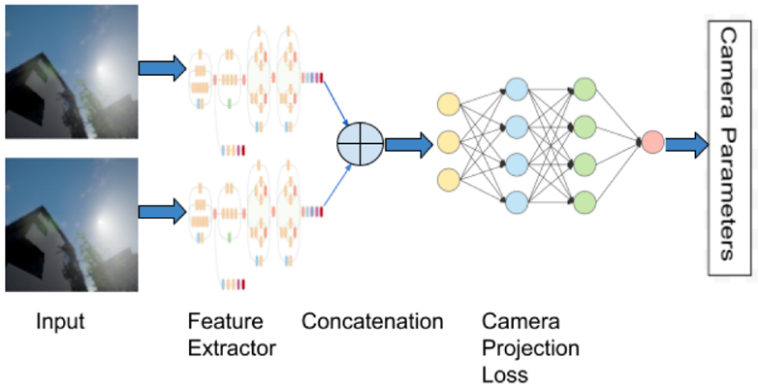


Figure 3.22. From Butt et al. [16] MTL-CPL architecture with Inception-V3 backbone feature extractor and camera projection loss network. © [2022] IEEE.

Furthermore, individual parameters are disentangled from the camera projection loss through individual evaluation to avoid backpropagating errors to other parameters that have less influence on the deviation of a point from its ideal projection. To further normalize the loss function, trained weights are added to the individual parameters to avoid introducing bias from specific error terms. According to Butt *et al.* these biases can be attributed to encountering heterogeneous ranges of parameters.

To evaluate their approach, Butt *et al.* used a real-world Cyclist Detection dataset containing 2914 images. Results were provided as mean absolute errors (MAE) for the 10 parameters on both the synthetic CVGL training dataset (with 23,796 images) and the Cyclist Detection dataset, with comparison to two other state-of-the-art (SOTA) methods. Three versions of the proposed MTL method were tested as an ablation study, one version, MTL-Baseline, demonstrates the inferior performance of end-to-end learning based only on MAE for loss. Two other variants MTL-CPL-U and MTL-CPL-A were tested with uniform and adaptive parameter weighting in the camera projection network respectively. MTL-CPL-U performed better than all other variants as well as the compared SOTA methods, and reported lower MAEs for 8 out of the 10 computed calibration parameters.

Lee *et al.* [53] addresses calibration using single shot natural scene images with the objective of finding zenith, horizon line, and field of view (FOV) parameters. Using a proposed camera calibration transformer with line-classification, (CTRL-C), directly estimates camera parameters from a single image as an end-to-end learned approach. Unlike previous CNN based methods, the authors proposed that CTRL-C is better at capturing long-term dependencies within an image. By using a transformer with an attention mechanism, CTRL-C is said to be more effectively able to learn the global characteristics

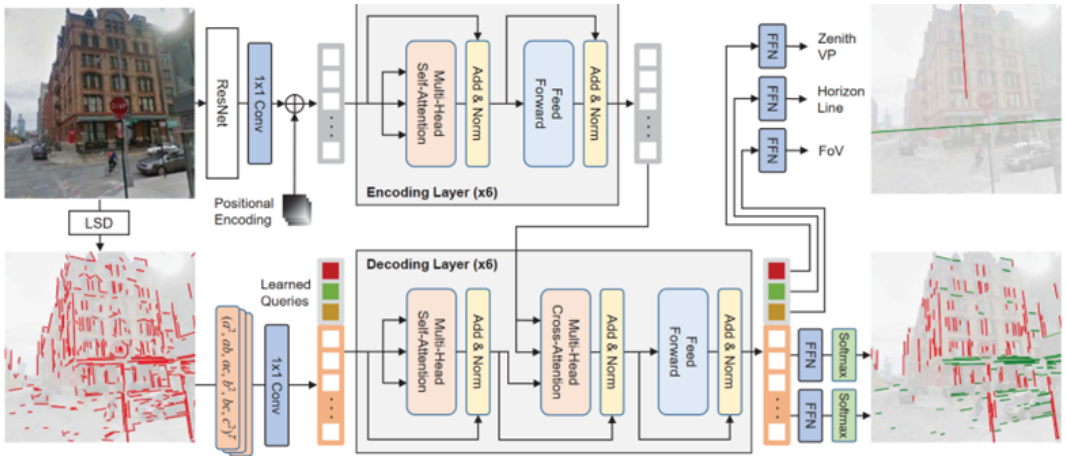


Figure 3.23. From Lee et al. [53] CTRL-C transformer encoder-decoder architecture with ResNet feature input, LSD algorithm line input, positional encoding, multi-headed attention mechanisms, and Feed Forward networks to estimate camera parameters and classify line embeddings into vertical and horizontal convergence lines. © [2021] IEEE.

of an image using multi-modal geometric and semantic cues. Semantic cues are provided using a ResNet backbone feature detection network, and geometric cues are provided by an LSD algorithm for providing lines, all while using a single image as an input. CTRL-C was tested against other SOTA methods and claimed to outperform these methods on the Google Street View (GSV) and SUN360 datasets (Figure 3.23).

Although Lee *et al.* [53] claims to outperform SOTA methods, qualitatively it can be seen that recovery of parameters from a single image is inherently degenerate since it is not always possible to extract 3 perpendicular planes reliably under the Manhattan world assumption. To compensate for this, Samaha *et al.* [64] proposed an end-to-end fully connected multilayer-perceptron (MLP) using fundamental matrices based on the underlying epipolar geometry and principles from Kruppa's equation. Using three sequential camera images, two fundamental matrices are computed as an input to the MLP to directly estimate focal length and camera center intrinsic parameters. For training, Samaha *et al.* [64] used over 80,000 images from their newly created Unity dataset, with ground truth camera parameters as a supervisory signal. The 5-layer MLP can take 2 or more fundamental matrices as a hyperparameter that can be tuned. 10% dropout was applied to the first 4 layers with early stopping, Huber loss, an Adam solver, and a learning rate of 0.0001. The resulting work claimed to outperform other SOTA methods while using many fewer network parameters (17K vs 26M) [64], however, the MAE of 10% across all parameters is

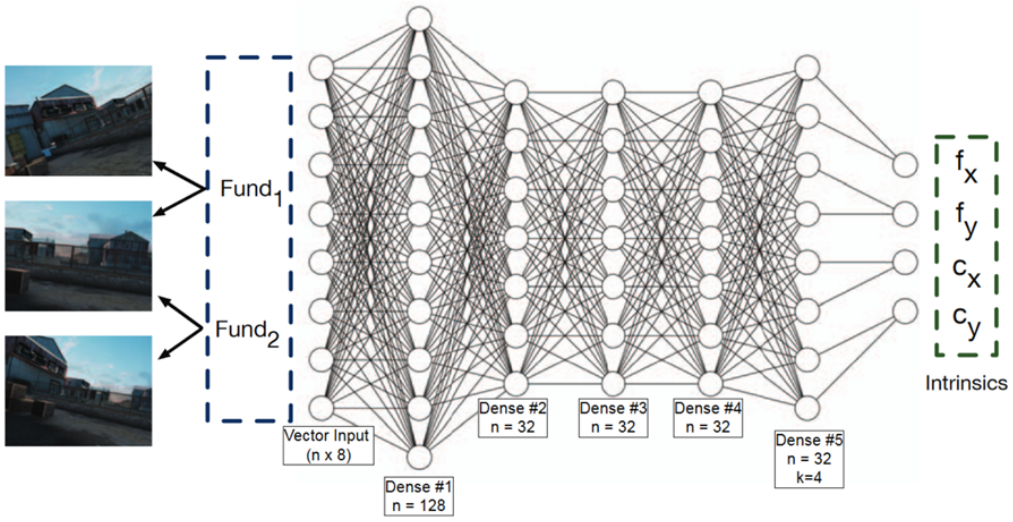


Figure 3.24. From Samaha et al. [64] MLP architecture for Learned Intrinsic Auto-Calibration from fundamental matrices using a minimum of two fundamental matrices as inputs to recover focal length (f_x, f_y) and principal point (c_x, c_y). © [2022] IEEE.

relatively high. In the limitations section Samaha *et al.* [64] discusses the issue with degeneracy in absolute-conic-based algorithms (relating to Kruppa's equation), and that the degeneracy stems from the computation of fundamental matrices that could have multiple solutions due to certain camera motions. References are provided for future work towards addressing these issues.

3.5 Other Methods

In the previous sections of this literature review, we have presented work that relates to camera calibration under blurred conditions, which mostly focuses on finding more accurate ways of detecting control points on calibration targets. Relaxing some constraints from our search, it is possible to present broader coverage of work that indirectly relates to the problem of camera calibration from blurred images. This section discusses the accurate detection of other patterns that may not have been directly discussed in the context of camera calibration, but relate to detection under defocused conditions. Conversely, additional work is presented on another pattern related to camera calibration but has not been tested under blurred conditions. We also briefly discuss other calibration frameworks,

toolboxes, and finally, an altogether different approach that is well researched, which would be to deblur the image prior to performing any feature detection or calibration. It is also worth mentioning that many of these other works include links to publicly available code or data sets, which could be an asset for further assessment and testing.

3.5.1 Fiducial Markers and Other Patterns

Relating to markers for augmented reality (AR), Toyoura *et al.* [73, 74] discusses the issue with using traditional markers such as ARTags, QR codes, and random dot markers that have sharp, high-frequency, binary edges or corners that become attenuated under blurred conditions. Treating this issue as a signal processing problem, they also demonstrate how the frequency spectrum of an image changes and loses higher frequency components once the image becomes blurred. To address this issue, they propose using a grid of low-frequency color gradient circles, which they refer to as a mono-spectrum marker. Specifically, this grid consists of 9 (3×3) different color sinusoidal gradient circles, which are designed to represent specific values for the hue, saturation, and value components of the HSV color space (Figure 3.25). For instance, yellow is reserved for each of the four corners of the mono-spectrum marker. The centers of the four corners are used to provide the homography for the position and orientation of the marker.

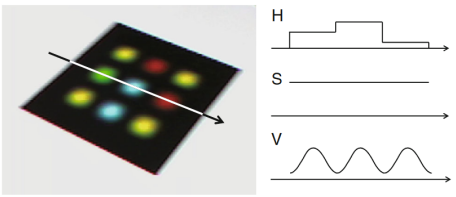


Figure 3.25. From work of Toyoura et al. [73, 74], mono spectrum marker with cross section of HSV profile for determination of orientation. Reproduced with permission from Springer Nature.

Specific marker identification is made possible by alignment of the surrounded colors. In this context, the problem of blur is treated as a low pass filter in the image frequency spectrum. Adjustment of an α value, which controls the circle brightness offset will alter detectability under varying degrees of blur. A series of steps, including removal of zero-frequency components of the image, band-pass filtering of remaining regions, and corner dot registration through color segmentation and alignment, are then performed to detect the markers. Since the proposed method becomes computationally expensive for real-time

detection, the steps were summarized in two separate algorithms, (1. for removal of zero frequency regions, & 2. detection of candidate marker regions) and were implemented using Nvidia's CUDA language and GPU[2] acceleration. It was also noted that multiple markers could be detected in the same image.

Along with the proposed mono-spectrum marker implementation, ARToolKit was also used for marker localization. Since the 3×3 grid was surrounded by a black border, when subjected to blur, localization using the border presents a similar issue to using checkerboards, ARToolKit was said to be able to more accurately localize the markers under conditions where the images were not blurred. To take advantage of this, Toyoura *et al.* implemented a confidence interval that allowed for switching between the use of their mono-spectrum marker corners vs. ARToolKit for accurate marker localization depending on the degree of blur. This consideration seems valuable because it could provide hints towards implementation of hybrid phase / checkerboard-based methods.

Calvet *et al.* [19] proposed a fiducial marker composed of concentric rings to preserve temporal continuity through multiple video frames where severe blur is present. Compared to some prior fiducials such as ARTKPlus, RuneTag, and PRASAD, the proposed method was stated to allow for accurate center point detection under blur. To achieve this, this proposed method relies on a conservation property of gradient magnitudes. Circular gradients are first created using a gradient map and field lines that are based on the concentric rings. In the detection phase, after pixel regions have been selected based on voting criteria, the conservation property enforces matching by ensuring that the ingoing gradient magnitudes passing through the outer ellipse equal the outgoing gradient magnitudes passing through the arc of the innermost ellipse. To allow for identification of individual targets, the proposed method also uses variable concentric ring thicknesses and spacing. The downside of this encoding approach is that there are a limited number of unique ID's that can be generated, but certain re-use schemes can be employed to create further patterns that can allow for the unique encoding of a greater number of markers. In practice, this method was found to return sub-pixel accuracies of 0.4px, to ~ 0.75 px in cases where blur magnitudes were as high as 10 to 15 pixels. Respectively, the detection rate under these conditions was between 57% and 22%. Frame rates were also noted to fall between 4 to 11fps on CPU[3] vs. GPU hardware, respectively. Code is publicly available for this work[4].

Finally, Ha *et al.* [39] proposed a new ecosystem for high accuracy camera calibration using triangular deltille grid patterns. This ecosystem consists of a corner detection algorithm, indexing for partially occluded grids, and DelTags, a coded AR-tag version of a

[2]Nvidia GeForce GTX295 896MB, CUDA 4.0

[3]Intel (i5-4590, 3Ghz), Nvidia GTX 980 Ti, CUDA 7.0

[4]Code is available via re-direct from: http://github.com/lcalvet/CCTag

deltille grid. Although not accounting for real-world blur, this method was proposed as a new higher accuracy replacement for checkerboards. It used a novel localization method by finding an intensity distribution characterized as a monkey saddle surface instead of a plain saddle surface that can be derived from checkerboards. Fundamentally, this method is claimed to be more accurate, because, unlike the intersection of checkerboards with two intersecting edges, the deltille patterns provide intersections with three edges, which provide an additional set of constraints. Calibration testing was performed using a 3D multifaceted calibration object with deltille grid patterns against a similar 3D calibration object using checkerboards. A robot was used to film a video sequence that was used for calibration[5], and the proposed deltille method returned an RMSE of 0.0995 px. A study of detection rates and times on other datasets against other well-known methods [1, 5, 34, 35, 61, 66] was also performed.

3.5.2 Calibration Projection Method and Toolbox

For wide FOV cameras, Lochman *et al.* [56] proposed BableCalib, as a calibration framework that is more robust to outlying detections. Using back projection as an intermediate model followed by regression, the proposed method provides a solver that recovers parameters using many supported forward and backward projection camera models such as Brown-Conrady, Kannala-Brandt, and Unified Camera, to name a few. Although Babel-Calib, was not tested with blurred images, its code is publicly available[6] and may be worth considering as an alternate solver.

Hao *et al.* [41] also proposed a conic tangent geometric method for increased control point localization accuracy based on concentric circles for intrinsic parameter recovery. Compared with Zhang's method, this proposed method was stated to reduce reprojection errors between 17.66% to 21.58% for a stereo camera setup.

Finally, Rameau *et al.* [62] proposes a calibration toolbox that is designed to work with multiple cameras and markers. The toolbox is designed to require minimal user interaction but is hierarchical in nature to allow for the calibration of complex camera systems. Their work builds on previous calibration toolboxes, and in relating previous work, they have provided a convenient table of previous calibration toolboxes. MC-Calib is also publicly available[7]. We do not recommend using MC-Calib because it suffers from the usual detection problems when dealing with blur. However, its review of other calibration

[5] Code, datasets and extended version of paper are available at: https://github.com/deltille.
[6] Code available at: https://github.com/ylochman/babelcalib.

toolboxes might be a useful starting point for finding other publicly available codes.

3.5.3 Deblurring Images Prior to Calibration

Another approach that could bypass the problem of dealing with calibration from defocused images would be to first deblur the image. In recent work, [54] Lee *et al.* proposes an end-to-end deep learning method that is based on recent filter adaptive networks (FANs). The proposed approach, iterative filter adaptive network (IFAN), is specifically designed to handle defocus blur. For improved image deblurring quality, IFAN considers the spatially varied nature of defocus blur and in doing so, generates deblurring filters for each pixel. For added efficiency, IFAN uses a separable filter architecture and employs a novel iterative adaptive convolution (IAC) layer to iteratively apply separable filters to different features. To supplement training data, the proposed method uses the DPDD dataset from Abuolaim *et al.* [10], and utilizes the dual-pixel (DP) stereo images to increase the accuracy of blur magnitude predictions. Further to this, a new dataset was created using a dual camera setup capturing stereo sets of defocused images[8]. Evaluations were also conducted on a number of other datasets, and both qualitatively and quantitatively, this method was shown to achieve state of the art results. Further to this, there are a number of related works that have code availability and could also be explored to solve this problem [21, 28, 49, 50].

3.6 Related Work Summary

In this chapter we primarily outlined methods for improved control point localization under out-of-focus blur conditions and mainly focused on methods that are based on 2D planar calibration target patterns which rely on Zhang's method.

We reviewed methods that have attempted to use improved checkerboard or circle-based patterns for improved calibration from out-of-focus images. We also reviewed work that was presented on phase shifted methods, and in particular circular phase-based patterns such as PCGs. In the last sections we reviewed other calibration related Deep Learning methods, as well as other methods that are not directly related to out-of-focus blur but might provide some underlying useful techniques and insights.

From the presented checkerboard and circle-based methods, we did not find results that would validate increased accuracy for out-of-focus images. Also, the one non-phase-based

[7] Code available at: https://github.com/rameau-fr/MCCalib
[8] Code and dataset available at: https://github.com/codeslake/IFAN

circle method from Choi *et al.* [26] required 15 images per pose which was less practical than the three images that were required in PCG based methods.

Overall, the reviewed methods were difficult to compare because of a lack of standardization in how the authors chose to present their results. For instance, in many cases, calibration results from different blur levels were presented with different focal lengths. Also, synthetic results were often derived from Gaussian blurring which lacks spatially dependant blur and often produces near perfect results. These two practices were common throughout many of the reviewed papers and make it difficult to clearly understand if the methods are improving accuracy or not.

We also reviewed Deep Learning based methods, and even reviewed one study that incorporates Deep Learning with PCGs to remove the requirement for multiple images to be displayed per-pose. Our take on Deep Learning methods is that they have not been proven to increase accuracy, but can be helpful towards encoding certain parts of calibration or control point localization methods, and can therefor improve their ease of implementation and use.

Finally, we would like to acknowledge that some square phase-based patterns such as Bell *et al.* presented promising results, however, these results were presented with less decimal point accuracy (see Bell *et al.* [13]) and were difficult to validate without fully re-implementing and testing the method.

Further work could be performed to confirm and compare the accuracy of square phase-based methods since this exceeded the scope of our study and our focus was on the branch of work relating to PCGs. This was owing to the PCG method being well-defined in literature, and the somewhat promising results that we chose to investigate further. Although only a limited amount of publicly available code was found overall, it might be worth testing the implementation from Schmalz *et al.* [68][9] due to its similarities to the work from Bell *et al.* [13] and the favorable results that were presented in their work.

[9]Code and dataset available at: http://www5.informatik.uni-erlangen.de/research/software/index.html

Chapter 4

Methods

Our camera calibration work is based on the standard calibration method of Zhang [87] using substituted calibration target patterns to improve control point localization when capturing out-of-focus images. Section 4.1 provides a review of the standard calibration method. Section 4.2 discusses the control point detection pipeline that we implemented based on the work of several authors [17, 24, 46, 77, 78], as well as our overall pipeline. Section 4.3 discusses camera calibration from real-world images, and section 4.4 discusses camera calibration from synthetic images.

4.1 Standard Camera Calibration Method

In this section, we review the camera calibration method proposed by, Zhang [87] which is well regarded as the standard method for camera calibration, and is known to be the method that is employed in OpenCV and MATLAB's camera calibration functions [5, 70].

To help visualize this camera calibration procedure, Figure 4.1 provides a simplified overview of the main steps involved in the calibration process, and depicts the typically used checkerboard calibration target outlined by Zhang [87] vs. PCGs which will be discussed in section 4.2.

For step 1 in Figure 4.1, the checkerboard calibration target is moved within the FOV of the camera and images of multiple poses are captured. Checkerboard intersections are detected and used as control points for step 2. In step 2, camera parameter recovery is performed using Zhang's method which works by decomposing sets of homographies based on the projection equation (2.1), that are collected based on point correspondences between

detected feature control points in the checkerboard images and known coordinates of the checker spacing. Since K and T relate the 2D points to the 3D points, K and T can be separated with the knowledge that K remains the fixed component across all images, and T changes with each pose. After an initial estimate of K is made through algebraic means, Zhang's method uses an iterative maximum-likelihood optimization step to reduce reprojection error and refine the estimation of K and T [87].

4.2 Control Point Detection Methods

Typically, checkerboards are the most used calibration target pattern, but require focused images to avoid poor localization accuracy that result in high MREs [24, 46, 74, 78] and low calibration accuracy. To overcome this issue, we replace checkerboards with PCG patterns (see Figure 4.1), (sub-section 4.2.1), circle grids, and concentric circle grids (sub-section 4.2.3). We also test different cases for control point localization and explain our overall calibration pipeline in (sub-section 4.2.2).

4.2.1 PCG Patterns

As introduced in Chapter 3, Wang *et al.* [78] initially proposed using phase-shift circular grating (PCG) arrays as a pattern that could be displayed on a digital screen and used for camera calibration from out-of-focus images. Mathematically, the steps in Table 4.1 outline this basic method.

Since the goal of our work was to gain more insight into the overall feasibility of PCGs and to determine if it could be the most accurate calibration method when using out-of-focus images, we first re-implemented the work of Wang *et al.* [78], and built upon it with the improvements proposed by Cai *et al.* [17]. Since no publicly available code was found for either of their work [17, 78], new code was written based on Table 4.1 using Python to generate PCG calibration pattern images, handle blob segmentation, recover wrapped phase images, and perform subsequent camera calibration.

In step 1 of Table 4.1, PCG circles I_1 to I_3 were always generated with an odd number diameter of pixels. This was done to ensure that the center pixel would always display a unique pixel intensity. In all subsequent work, we chose PCG circles with a radius of 131px.

After step 2 in Table 4.1 is completed, and sets of three PCG array pattern images are captured from multiple poses, the images can be processed using equation (4.4). To run

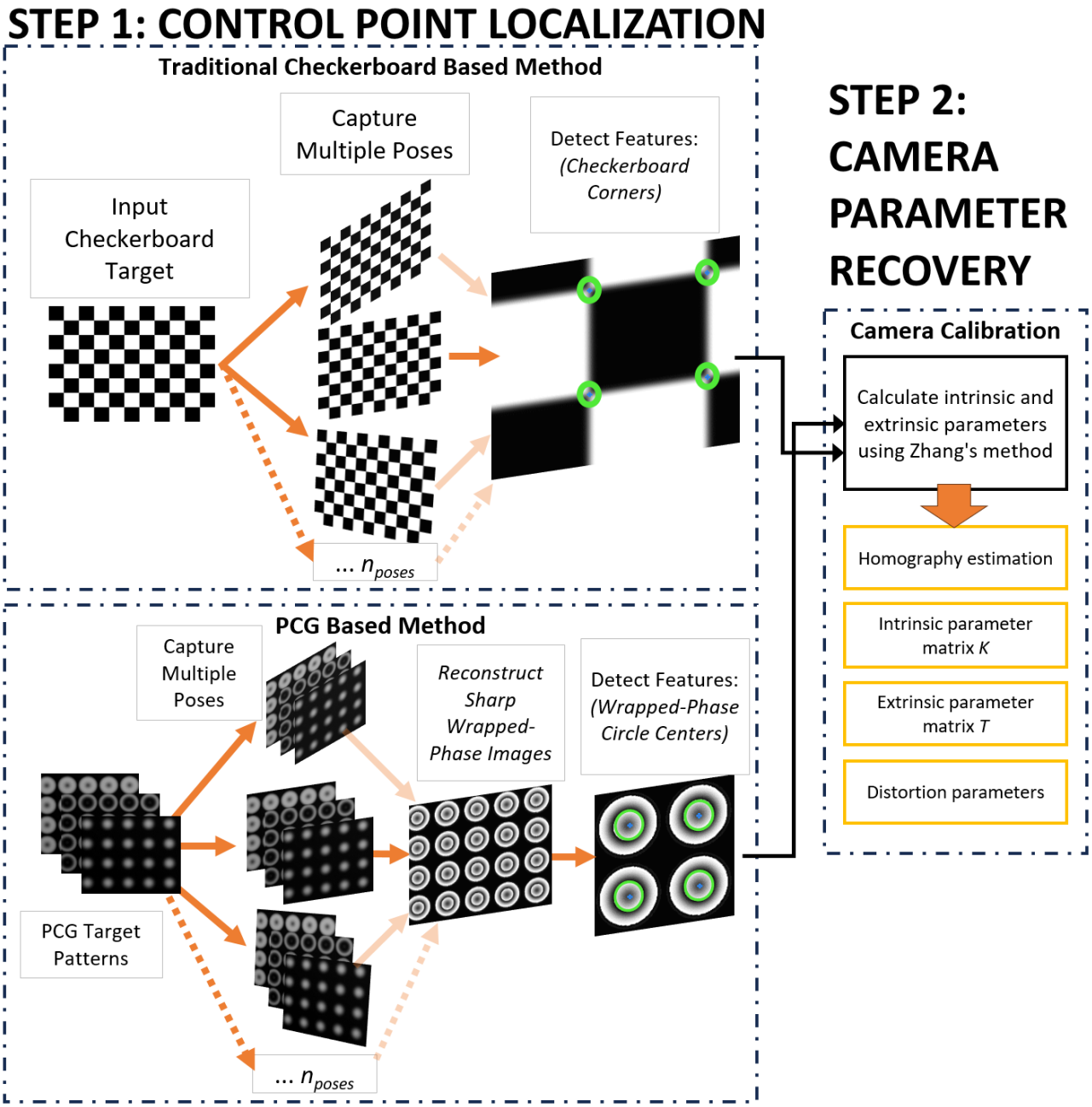


Figure 4.1. From our previous work [67], we illustrate the control point localization methods of Zhang [87] using checkerboards, vs. Wang et al. [78] using PCGs.

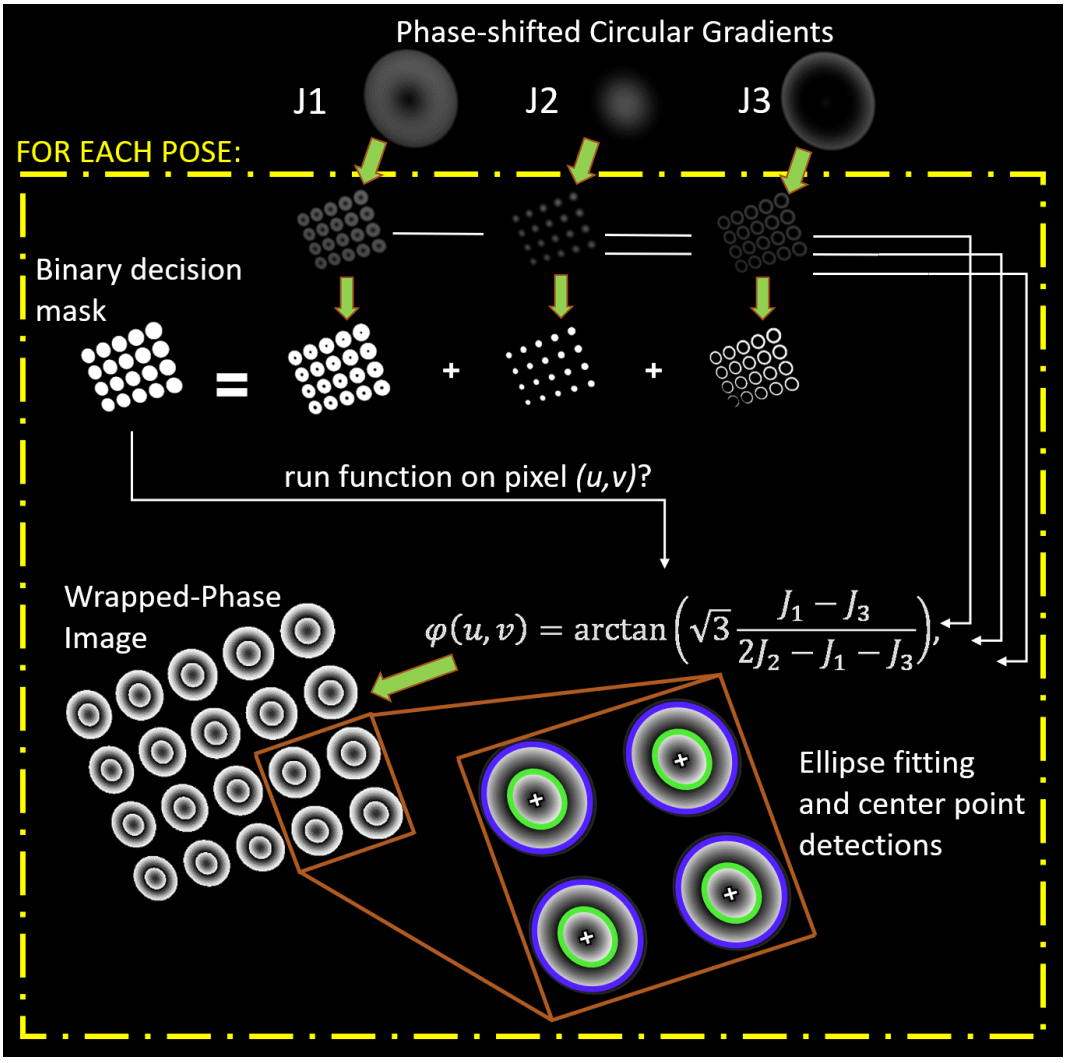


Figure 4.2. From our previous work [67], we illustrate that for each captured or rendered set of three PCG array / grid images, blobs are generated from images I_1 to I_3 for η number of poses. The three blobs per pose are combined to create a combined blob image for each pose. The combined blob images are used as binary masks to determine which pixels to operate the arctan function on, in order to compute sharp wrapped phase images.

equation (4.4), we mask the captured PCG array pattern images separately and combine the masks into blobs that segment the pattern from the background (Figure 4.2). This step is essential to ensure that the arctan function will run without encountering division by

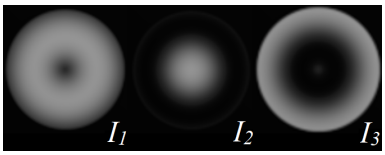
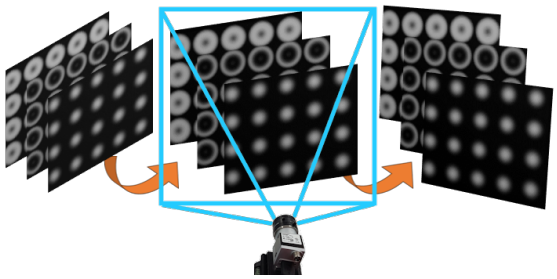
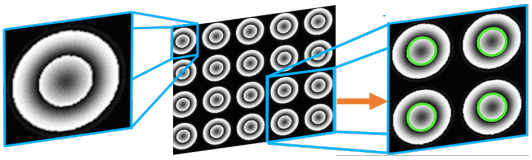
<table>
<tr><th colspan="2">Phase-Shift Circular Grating (PCG) Method Steps [78]</th></tr>
<tr><td>Step1 : Three grayscale circular gradient patterns are created using equation (4.1), where A and B are constants, $\phi(x,y)$ is the unwrapped phase based on (4.2), and I_1 to I_3, are the grayscale pixel intensities at coordinates x, y for each image respectively. Each gradient is shifted by $2\pi/3$ increments. As an input to the unwrapped phase, $r(x,y)$ is the Euclidean distance from the center of the circle based on (4.3), These circular patterns can be referred to as PCGs.</td><td>$$\begin{cases} I_1(x,y) = A + B\ cos[\phi(x,y) - 2\pi/3], \\ I_2(x,y) = A + B\ cos[\phi(x,y)], \\ I_3(x,y) = A + B\ cos[\phi(x,y) + 2\pi/3], \end{cases} \quad (4.1)$$
$$\phi(x,y) = \frac{2\pi r(x,y)}{T}, \quad (4.2)$$
$$r(x,y) = \sqrt{(x-x_0)^2 + (y-y_0)^2}, \quad (4.3)$$
  </td></tr>
<tr><td>Step2 : Each PCG image, (I_1, I_2, and I_3) are tiled into a symmetric grid or array, and displayed using a digital display. Images of the (I_1, I_2, and I_3) arrays are captured consecutively in multiple poses with various rotations and translations within the FOV of the camera and can be captured regardless of the level of focus.</td><td>  </td></tr>
<tr><td>Step3 : For every pose, wrapped phase images are generated using equation (4.4) where $J_1 - J_3$ are respective pixel intensities from $I_1 - I_3$. The phase points lying on the 2π discontinuity (circular transition from light to dark pixels) are used for ellipse fitting and respective center point estimation as inputs for camera calibration.</td><td>$$\psi(u,v) = arctan\left(\sqrt{3}\frac{J_1 - J_3}{2J_2 - J_1 - J_3}\right), \quad (4.4)$$
  </td></tr>
</table>

Table 4.1: Phase-Shift Circular Grating (PCG) Method Steps

zero errors. The computed wrapped phase images are essentially a sharp re-construction of the originally captured sharp or blurred sets of PCG images. These computed images are stored and used in a subsequent camera calibration process.

Following Cai *et al.*'s method [17], we modified the period T of equation (4.2) to create PCGs that will generate two wrapped phase discontinuities. Figure 4.3 shows the modified wrapped phase image with two discontinuities which are the pronounced rings where pixels transition from high intensity white to low intensity black shading.

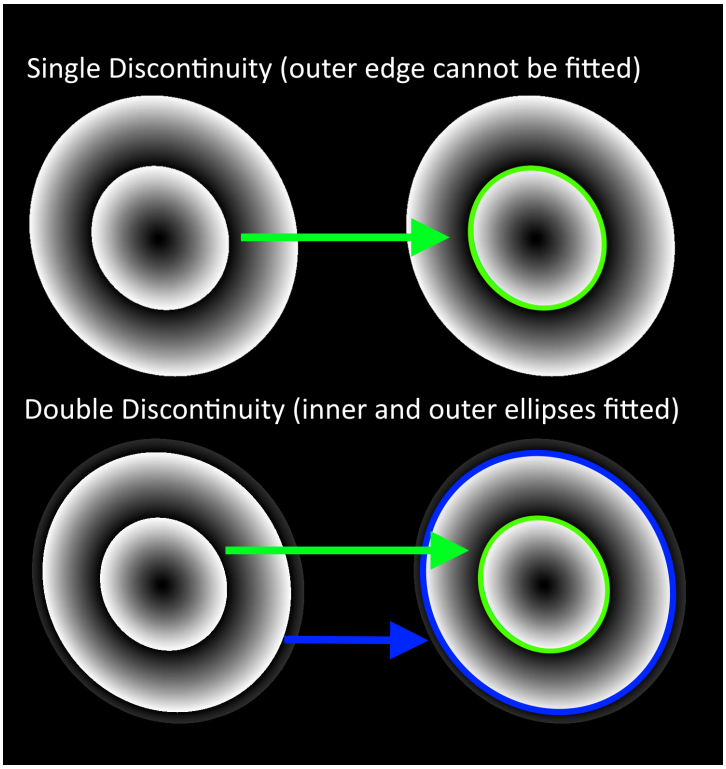


Figure 4.3. shows wrapped phase images of single PCG circles with sharp discontinuities (rings where there are sudden changes of pixel intensities from light to dark) that are used for ellipse fitting. Notice that the below images (similar to Cai et al. [17, 18]), have an outer ring that remains part of the pattern and is not segmented from the background, whereas the above set (similar to Wang et al. [78]) would be segmented at the point of highest intensity, leaving only one discontinuity that can be fitted.

With multiple wrapped phase discontinuities, we applied the theory in section 2.4, to compute corrected center points from the centers of inner and outer ellipses. Figure 4.4

illustrates how the corrected center point typically looks from an enlarged view of the plotted points of a wrapped phase PCG circle grid image generated from synthetic data.

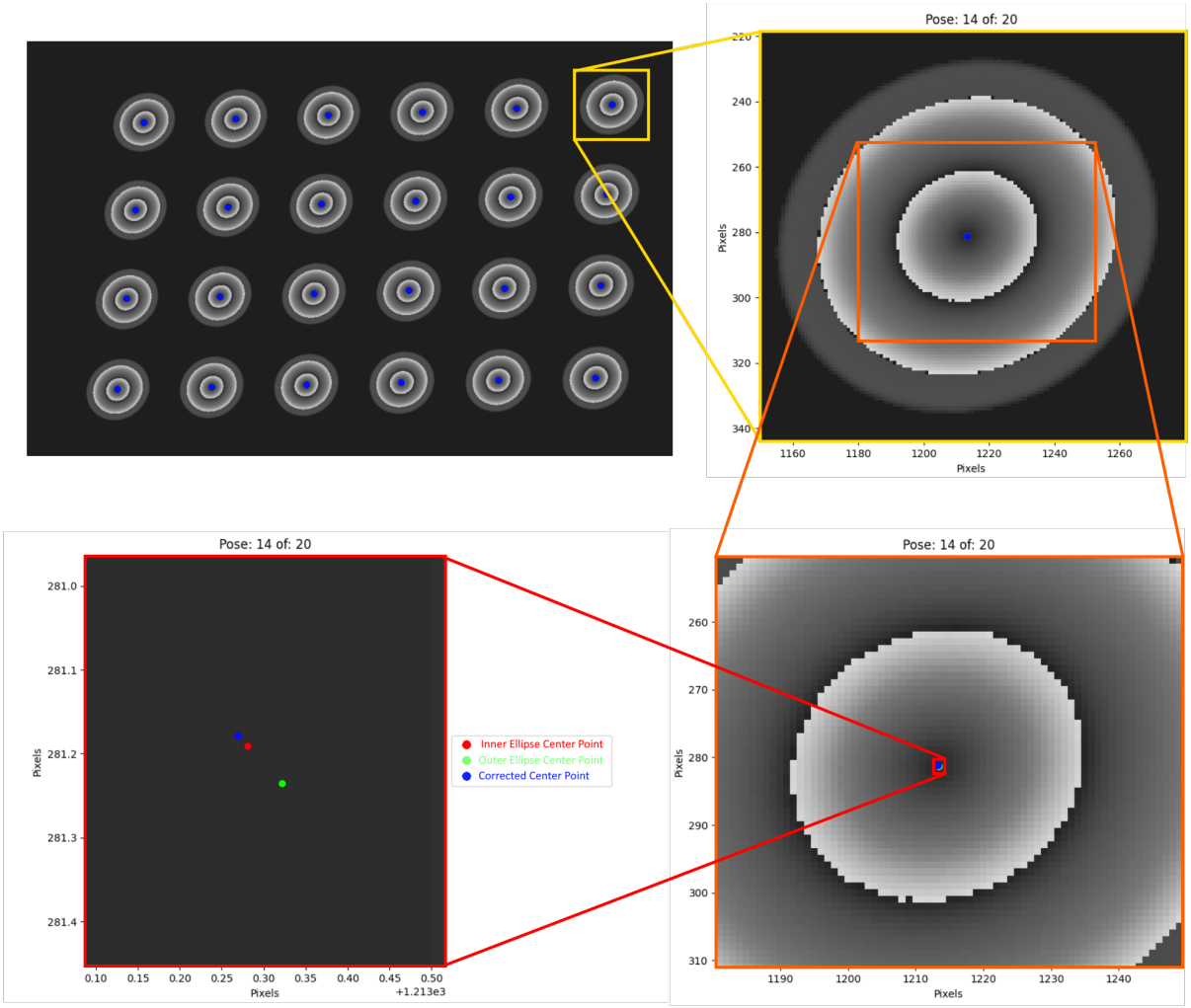


Figure 4.4. shows 3000× magnification of localized center points. Where red is the center of the inner ellipses, green is the center of the outer ellipses, and blue is the corrected center, note that these images were generated from synthetically rendered PCGs for an in-focus case. Also, as demonstrated in Cai et al.'s work [17], the amount of perspective angle of the calibration target has been observed to affect the spread of points respectively.

4.2.2 Control Point Localization and Calibration Pipeline

Similar processes were followed to perform camera calibration using PCG, circle, and concentric circle grid patterns. Figure 4.5 outlines the general process where two alternate methods were followed for extracting control points depending on which pattern is used.

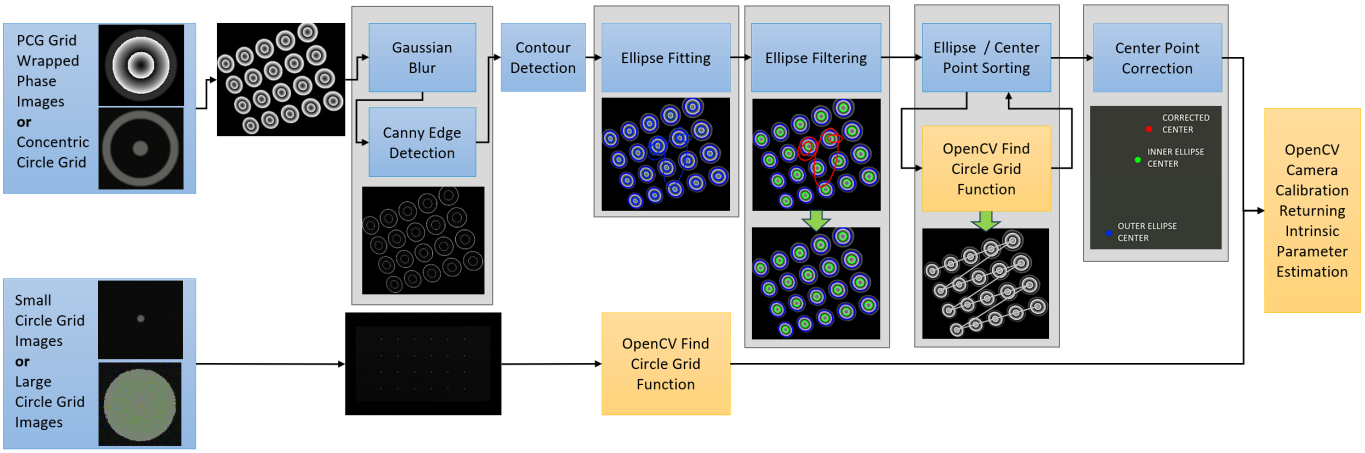


Figure 4.5. General process for center point localization and camera calibration.

The general process for PCGs and concentric circles is to first follow steps that lead to fitting ellipses. This starts by first gaussian blurring the image followed by Canny edge detection, where a blur kernel size, and standard deviation are provided to the gaussian blurring function, and upper and lower hysteresis thresholds are specified for edge detection. Fitting ellipses to concentric circles proved to be more difficult than for PCGs because the re-constructed wrapped phase images of PCGs provide consistent pixel intensities, whereas the concentric circles are sensitive to lighting and resulting contrast.

To improve concentric circle ellipse fitting, additional Canny edge detector parameters were added. These parameters were aperture size, which broaden the range of hysteresis limit functionality, and L2 gradient, which specifies an equation for finding gradient magnitude which increases the accuracy of edge detection. All parameters were hand tuned for respective image data sets, and in the cases of concentric circles, across all images, since lighting conditions could vary across images and different patterns provide different responses for edge detection and subsequent ellipse fitting.

With computed edge thresholds, we then detect contours followed by subsequent ellipse fitting, where a minimum of five contour points are needed to fit an ellipse. Using synthetic data, ellipses can be fit from contours with relative ease and consistency. However, with

the added noise from real-world images, erroneous ellipses often appear linking the wrong contour points, and often need to be filtered. Three methods were devised to filter ellipses, including filtering by major / minor axis ratio, filtering by angle, and pre-filtering by proximity of center points to known coordinates detected using the OpenCV find-circle-grid function. In the latter method, ellipses are thrown out if their x, y center point exceeds specified geometric thresholds of known center points found by the find-circle-grid function, which detects the broader outlining circles of the patterns.

Of these three filtering methods, filtering by major / minor axis ratio proved to be the most effective. Using a user specified ratio cutoff, erroneous ellipses with a ratio of their major to minor axis exceeding the cutoff could be eliminated. This is possible because the majority of erroneous ellipse detections are elongated beyond the amount of elongation caused by calibration pattern perspective shift alone. Although filtering by angle was implemented, it did not end up being used. This was due to the principal angle of ellipses often appearing to be random with no common principal angle being determined for comparison against erroneous ellipses. Pre-filtering by proximity was used in some cases, but only removed a small number of invalid ellipse detections in addition to major / minor axis ratio filtering.

As there are numerous implementation details, Figure 4.5 provides a generalized high-level outline of these processes. All PCG and concentric circle grid patterns use corresponding neighbour matching based on the OpenCV find-circle-grid function to determine the order of points from locations on the grid. In these cases, the OpenCV find-circle-grid function would detect circles on a PCG wrapped phase image. The x, y coordinates of center points from detected ellipses were then compared within specified boundaries (usually within 5 pixels for x, y respectively), if ellipse centers were within the same neighbourhood as the OpenCV points, they would then be assigned to that position on the grid. Since the OpenCV find-circle-grid function relies on an OpenCV based blob detector, in some cases, limitations were found with the detectors ability to find obvious blobs. In these instances, separate masked images were supplemented using images processed using SciKit Image morphology.

Since numerous ellipses are generated per circle, we used various combinations of ellipses to test camera calibration results for different cases. For PCGs and concentric circles, there were two distinct sizes of ellipses, one larger set for the outer circle, and one smaller set for the inner circle. To distinguish between these two, size thresholds were used and could easily segment between the inner and outer ellipse (see green inner ellipses and blue outer ellipses in Figure 4.2.2). It is important to note that for all inner and outer ellipses, often multiple overlapping ellipse detections are found. In all cases, we average the centers of the inner ellipses, and separately averaged the centers of the outer ellipses. From this data,

we were able to experiment with the following cases (Table 4.2):

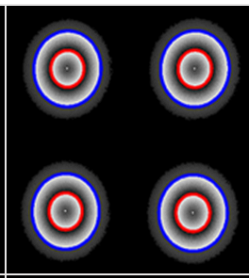
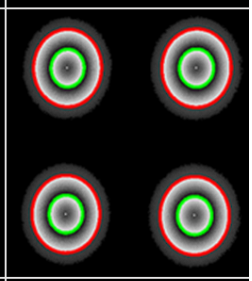
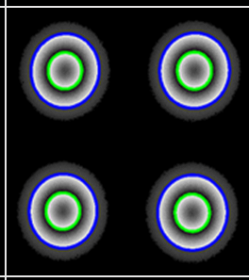
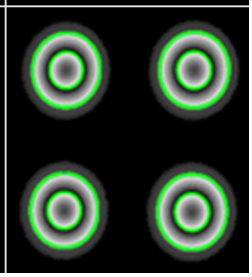
Calibration using inner ellipse centers (center of red ellipse)	
Calibration using outer ellipse centers (center of red ellipse)	
Calibration using a corrected center point using the inner and outer ellipse centers (both green inner and blue outer) as inputs	
Calibration using the average of inner and outer ellipse centers (average centers of all green ellipses)	

Table 4.2: Calibration cases.

4.2.3 Circular Patterns

Two circle grid patterns were generated and used for comparison in experiments. One pattern was using small circles with a radius of 13 px and the other used a grid of large

circles with a radius of 131 px (Figure 4.6). Both patterns were used for comparison with PCGs and used the standard off the shelf OpenCV find-circle-grid, and blob detector functions to detect points for subsequent camera calibration using OpenCV's standard calibration function. Hand tuned parameters were provided for the blob detectors, and a single blob detector profile was used for each set of images. Note that in our last experiment we also gathered data from synthetic sets of concentric circles (Figure 4.7), using a modified branch of our ellipse fitting and center point correction pipeline.

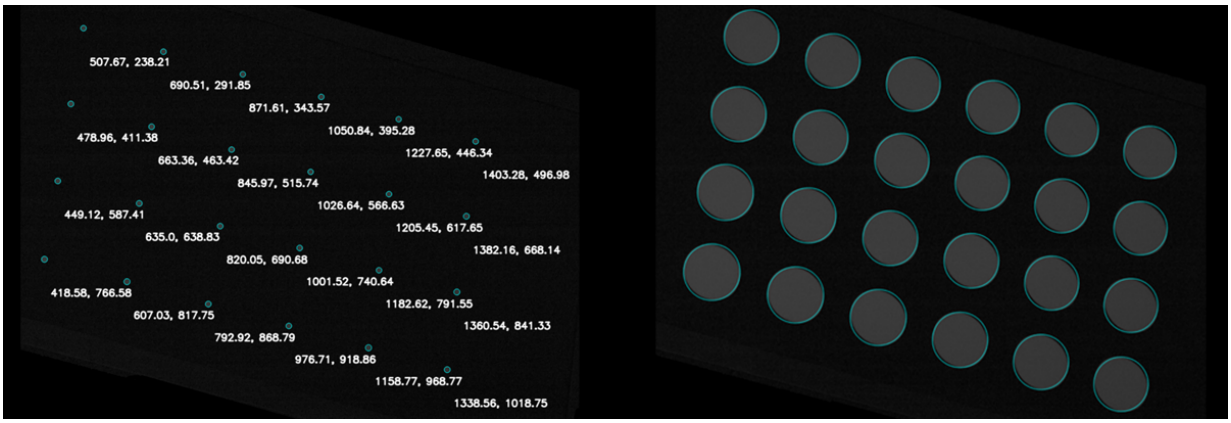


Figure 4.6. Small circle (Left) and large circle (Right) calibration patterns.

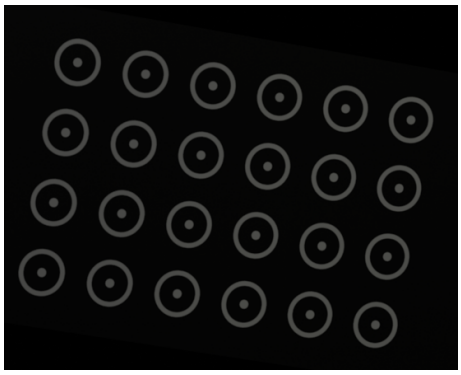


Figure 4.7. Concentric circle pattern, detected using a modified branch of our ellipse fitting pipeline.

4.3 Camera Calibration from Real Images

4.3.1 Initial Trials and Foundational Knowledge

Many lab experiments were conducted to learn about the calibration process and to find a standardized procedure that would allow for valid comparisons between methods. Early experiments were conducted using real world passive checkerboard patterns as well as synthetic checkerboard patterns which confirmed the inability to detect control points beyond certain levels of out-of-focus blur. Further experiments were conducted using LCD display screens, but revealed issues with camera refresh rate synchronization between the LCD screen and the camera, which would cause variations in image brightness depending on exactly when the frame was captured. Although the 1920 x 1080, 144 Hz, 15.6", 141 ppi, anti-glare, IPS display that was later used had less of a synchronization issue due to a relatively fast refresh rate (Figure 4.8), we later decided to try an E-Ink display to eliminate the possibility of synchronization issues all together.

After several experiments, we also realized the subtle difference between finding the intrinsic parameters of the camera's lens element, and the camera system as a whole, which needs to account for the lens's configured settings determining how light is projected onto the camera's image sensor. This underscored the importance of performing calibration with fixed camera intrinsics where it is necessary to leave the camera focus, and aperture (Figure 4.9) unchanged, regardless of capturing in-focus or out-of-focus images.

Essentially, in real world experiments, since there is no way of measuring the focus ring and aperture settings and translating the effect of their set-points on the lens's inherent focal length, to a ground truth focal length, the only way to compare the accuracy of a particular method is to change the distance between the camera and the target, and compare the recovered intrinsic parameters. It is also important to note that other work in literature provided results that did not leave the lens intrinsics fixed, and therefor provided results that were difficult to compare and determine real world accuracy from.

4.3.2 Robot Guided Real-World Experiment Method

All of our real-world experiments were performed using a Basler acA1920-150uc camera with 150 FPS frame rate at 1920x1200 resolution with a 4.8 μm × 4.8 μm sensor (10.9 mm effective diagonal). This was coupled with a Computar C23-5028-5M, 50mm focal length lens (Figure 4.9). Our main experiment was performed in the University of Waterloo's RoboHub lab using a Franka Emika Panda collaborative robot arm, controlled with the

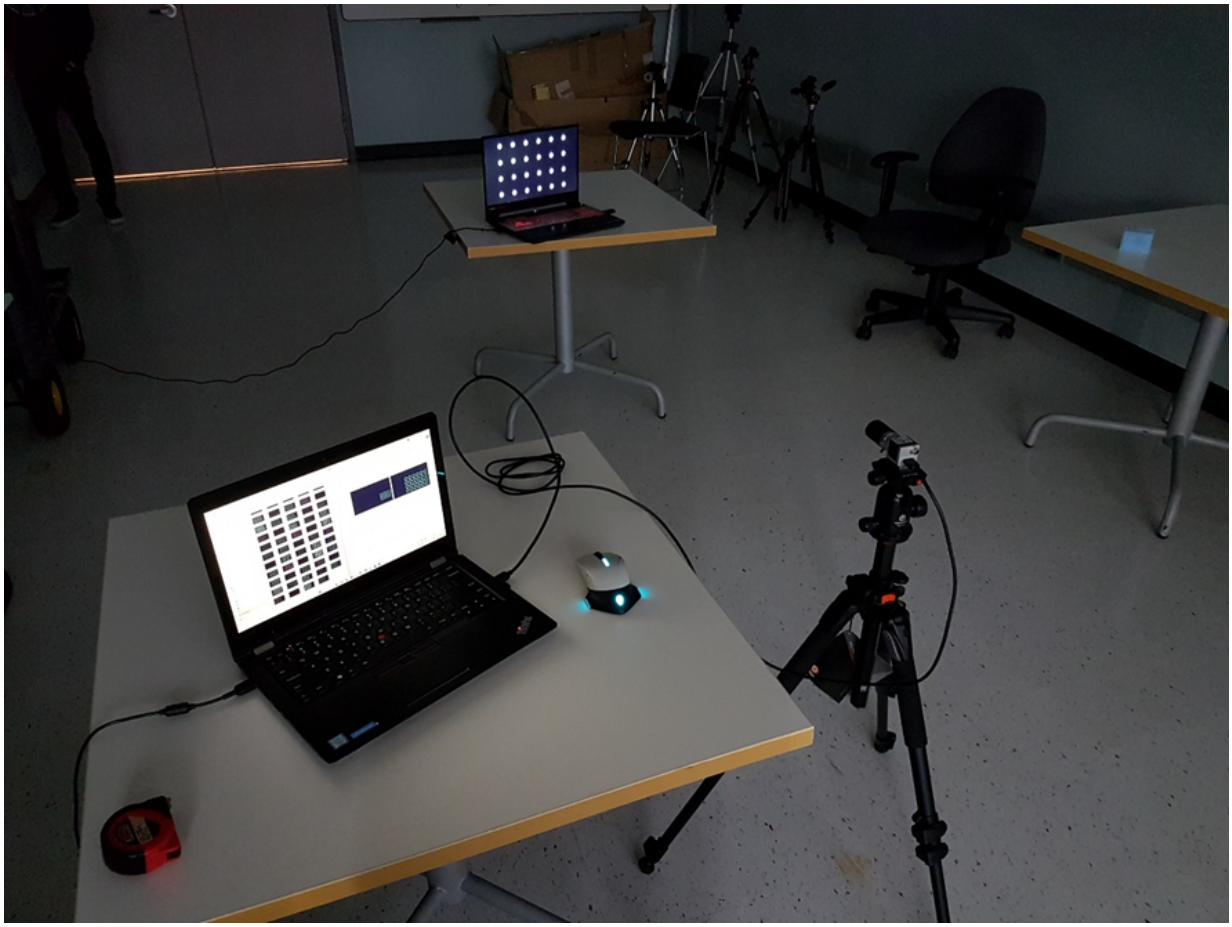


Figure 4.8. Early lab experiments using an IPS LCD laptop display as the calibration target, displaying PCGs and other patterns for comparison in a dark room.

Franka Emika Robot Operating System (ROS) API. A 3200 x 1800 resolution color E-Ink display was attached as the end-effector and camera calibration target. All images were captured at the camera's full resolution of 1920x1200.

With foundational knowledge in place, our established experimental procedure as depicted in 4.10 was as follows:

1. Train the robot using its collaborative functionality to record a sequence of 20 poses with rotations ($\lesssim 45°$) and translations from side to side within the robot's motion envelope with minimal changes in distance from the camera.



Figure 4.9. Computar C23-5028-5M 50mm camera lens with focus and aperture ring adjustment controls.

2. Set up the camera in-focus 4m from the E-Ink display target with extra room to translate the target within the FOV of the camera. The amount of extra room depends on step 4. With our 50mm lens, this worked out to a 4m camera to target distance.

3. Capture the 20 poses of focused images.

4. Leaving the camera intrinsics fixed, move the camera within its macro focus to a maximum distance that allows the pose sequence to translate within the FOV of the camera without cutting off any of the calibration target pattern. In our case a round number distance that worked well was 3m.

5. Capture the same 20 pose sequence from the 3m out-of-focus position.

In our lab setup (Figure 4.10), an E-Ink display was used since it displays a static image with no refresh rate characteristics. The other advantage of this display is that it does not contain a glass plate which could add refraction and require refraction correction using

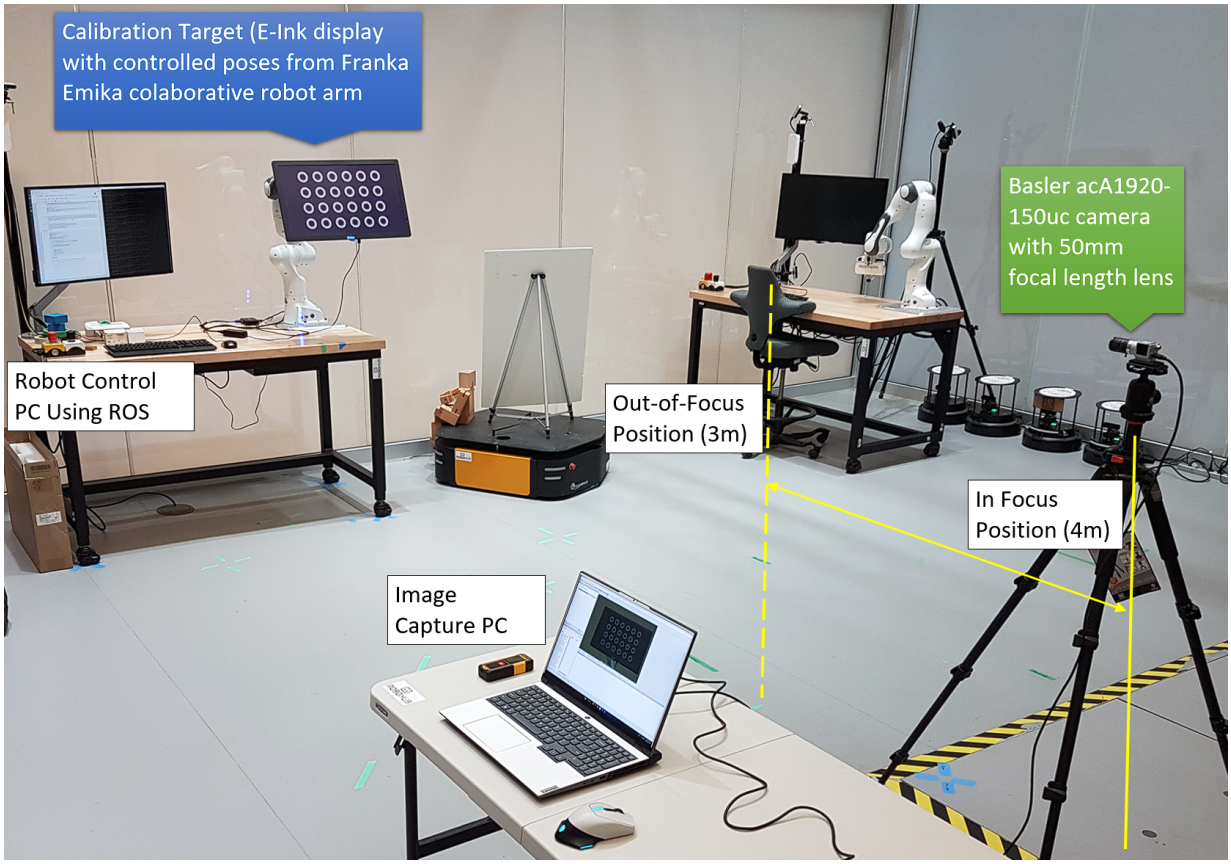


Figure 4.10. As shown in our previous work [67], the above figure depicts the lab setup for our robot guided camera calibration experiment.

Snell's law to eliminate ray offsetting bias [38, 68]. The trade-offs with using this display are that it cannot natively display grayscale pixels. It only achieves a grayscale approximation through dithering, which could potentially introduce an offsetting bias between different patterns. Also, since the E-Ink display does not have a highly mat coating, and is not actively backlit, the environment lighting needs to be carefully set to balance illuminating the E-Ink display but reduce reflections that cause glare. For all 20 poses in the robot's motion sequence, we adjusted individual spotlights and area lights around the ceiling to achieve this balance as much as possible. A minimum 1.5 second delay is built into the motion sequence between poses to allow the robot arm to settle from oscillations before an image is captured.

The fundamental difference between a camera centric and target centric setup is that

in a target centric setup, the camera would need to translate much farther to achieve the same perspective angle towards the target. Unlike with Huo *et al.* [46], where a linear motion table was available in conjunction with a robot arm, since our robotic setup has a more limited range of lateral motion, we chose to use a camera centric setup. In all of our real-world experiments, a 4×6 grid of circles was used. Although higher accuracy can be achieved with more circles, we found that this setup fits the aspect ratio and margins of the screen well, and a sparser control point grid was easier to closely examine. Also, our experiments were mainly aimed at observing the relative accuracy between in and out-of-focus patterns where no other ground truth is present, therefor making an increased number of control points partially redundant.

4.4 Camera Calibration from Synthetic Images

4.4.1 Initial Synthetic Experiment Method

Initial synthetic experiments were conducted to validate the implementation of PCG control point localization and calibration parameter recovery. These initial experiments were also discussed in our previous work [67]. Using the Open-Sourced Blender 3D animation software application, synthetic scene rendering was configured to model real world camera calibration setups. For the initial synthetic experiments, the Blender scenes were configured to render 4×5 PCG grid image sequences from a perspective camera model. The PCG grid pattern was rendered on a flat plane which was animated to rotate and translate through a specified pose sequence. Each image sequence was produced by three copies of a Blender scene containing the three PCG images (I_1 to I_3) for 10 poses each, and a total of 30 images from all three scenes (*see Figure 4.11 above portion*).

The initial 4m distance sequence was rendered in-focus followed by an out-of-focus sequence where the camera focus distance was set to 10m. After 10m, the focus distance was incremented by 5m up to 30m, followed by a final set with a focus distance of 100m where the images were severely out-of-focus. Figure 4.12 illustrates the scene setup and Table 5.1 in section 5.2 outlines results for each of these cases.

The perspective camera was configured to have a 50mm focal length, and an F-Stop of 1 from 4 to 30m followed by an F-Stop of 0.5 for 100m to allow the increased focus distance to render additional blur. In all cases, the synthetic camera model was set to have a horizontal sensor fit with a 36mm width sensor.

It is important to note that in each focus distance set, although the camera focus distance was changed, the actual camera distance in the scene remained the same. This

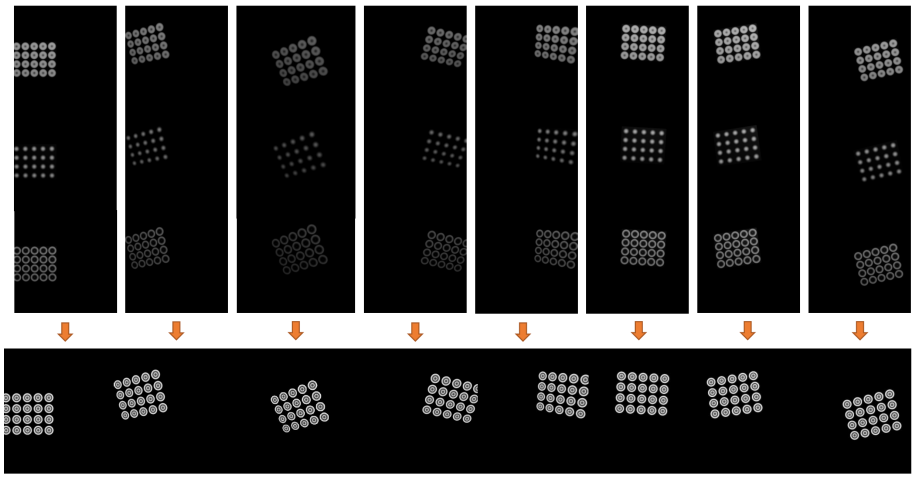


Figure 4.11. Visualization of the first 8 out of 10 rendered poses for (I_1 to I_3) (**Above**) *and respective computed wrapped phase images* (**Below**).

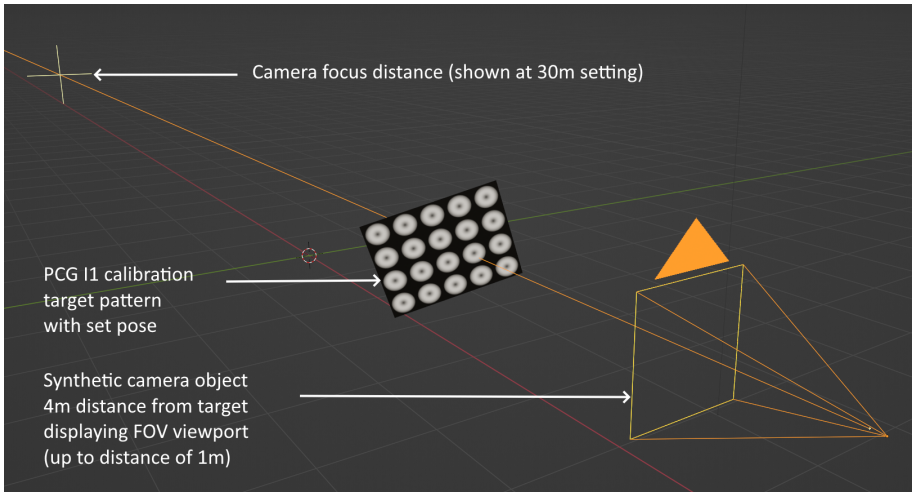


Figure 4.12. Configuration of Blender scene for rendering PCG images.

is a special characteristic that is possible when using Blender for synthetic rendering and allows blur to be decoupled from the spatial resolution of the image. In other words, the calibration target pattern can be the same size in the image, but rendered as if it were a different distance from the camera with a respective change in out-of-focus blur. Unlike in past works [18, 78] where gaussian blurring alone is used, our method of synthetic rendering with Blender is a much closer approximation of real world conditions primarily because

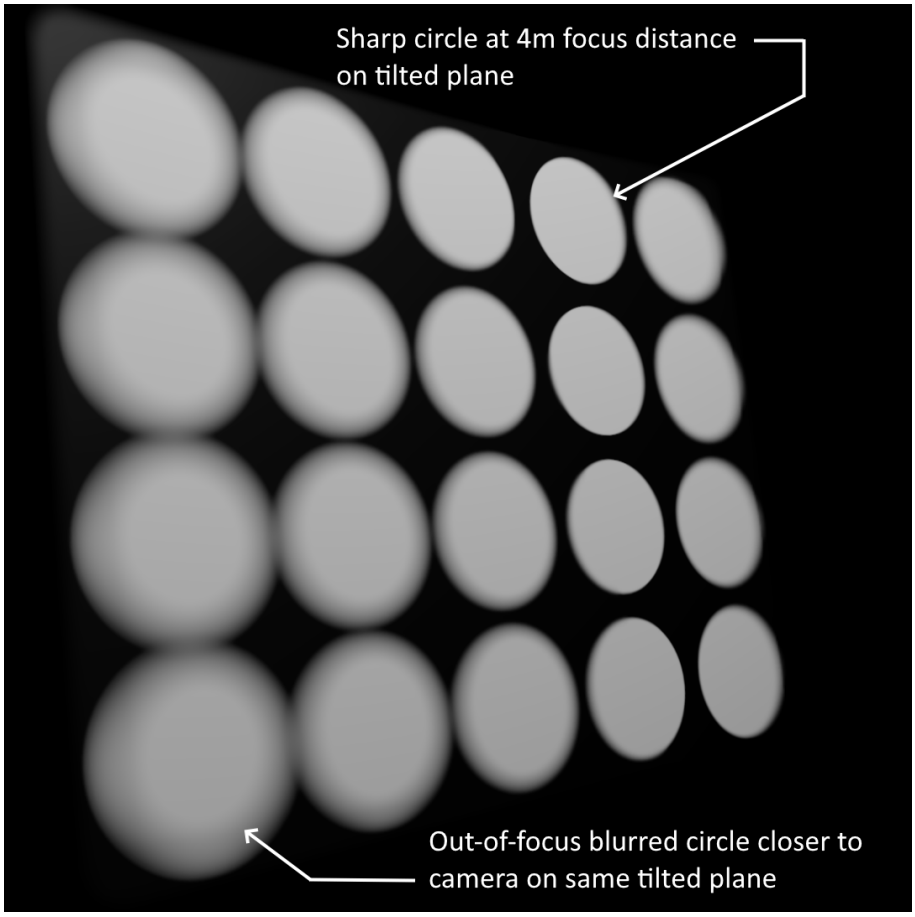


Figure 4.13. For illustration, Blender scene rendered with exaggerated spatially dependent blur of tilted plane rendered with perspective camera model.

spatially dependant blur is being modeled. Figure 4.13 depicts an exaggerated rendering that we created to illustrate this point.

From all of the images that were rendered (7 focus levels $\times$ 3 PCG images (I_1 to I_3),$\times$ 10 poses), equation (4.4) from Table 4.1 was used to generate wrapped phase images (Figure 4.11). Subsequent camera parameter recovery was performed using standard OpenCV find-circle-grid and calibration functions.

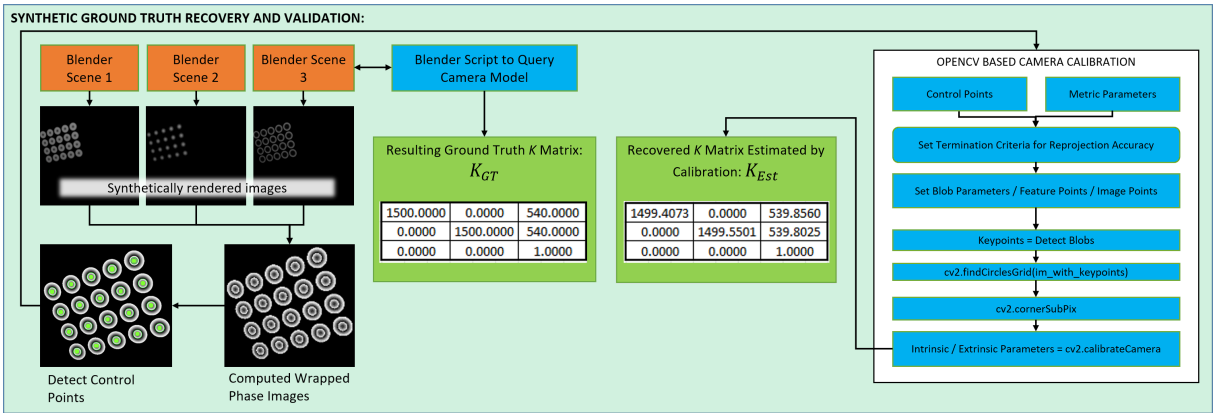


Figure 4.14. Process of recovering ground truth intrinsic parameters (K matrix) from Blender camera model, and comparison with K matrix recovered from calibration.

4.4.2 Calibration Validation and Focal Length Ground Truth Recovery Method

To validate our method and find ground truth focal lengths, Python scripts were devised [8] to query the Blender scenes internal camera model (Figure 4.14). These ground truth calculations are further described in Appendix A.

In all cases of the initial experiments, the focal lengths for the Blender scenes were verified to be exactly 1500px for both the f_x and f_y focal lengths.

4.4.3 Advantages of Synthetic Camera Calibration

From our work using Blender, we can summarize the following advantages that can be attributed to synthetic rendering using these methods:

1. Synthetic rendering is repeatable, target poses, camera positions, lighting, and other variations in background environmental conditions can be eliminated or numerically constrained.
2. Noise can be controlled, either added or removed.
3. The ground truth can be recovered based on the configured camera model.
4. Physical objects like a glass screen plate requiring refraction correction can be added or eliminated.
5. Spatial resolution can be preserved, the focus distance and blur can be varied without changing the image size.
6. Programmability. Synthetic rendering methods can be controlled programmatically to iterate through more experimental conditions faster.

4.4.4 Method for Synthetic Experiment Aligned with Robot Guided Real-World Experiment

After gathering results from our robot guided real-world lab experiment (section 4.3.2 for method, and section 5.3 for results), we devised a synthetic experiment using the recorded poses of the robot to validate our synthetic rendering method against the real-world image dataset. The purpose of this work was also to observe if there are noticeable differences in results with real world factors like dithering and noise removed.

To do this, we recovered the recorded poses from the robot motion sequence and verified them in the ROS based Gazebo simulator. Using the Gazebo simulation, we recorded the quaternion angles of the simulated robot's left end-effector gripper (Figure 4.16), which was found to have the same orientation and similar offset to the real world origin of the E-Ink display.

The quaternion angles were converted to roll-pitch-yaw (RPY) and XYZ distances in meters and inserted into a Blender scene containing a plane with a shader material (similar

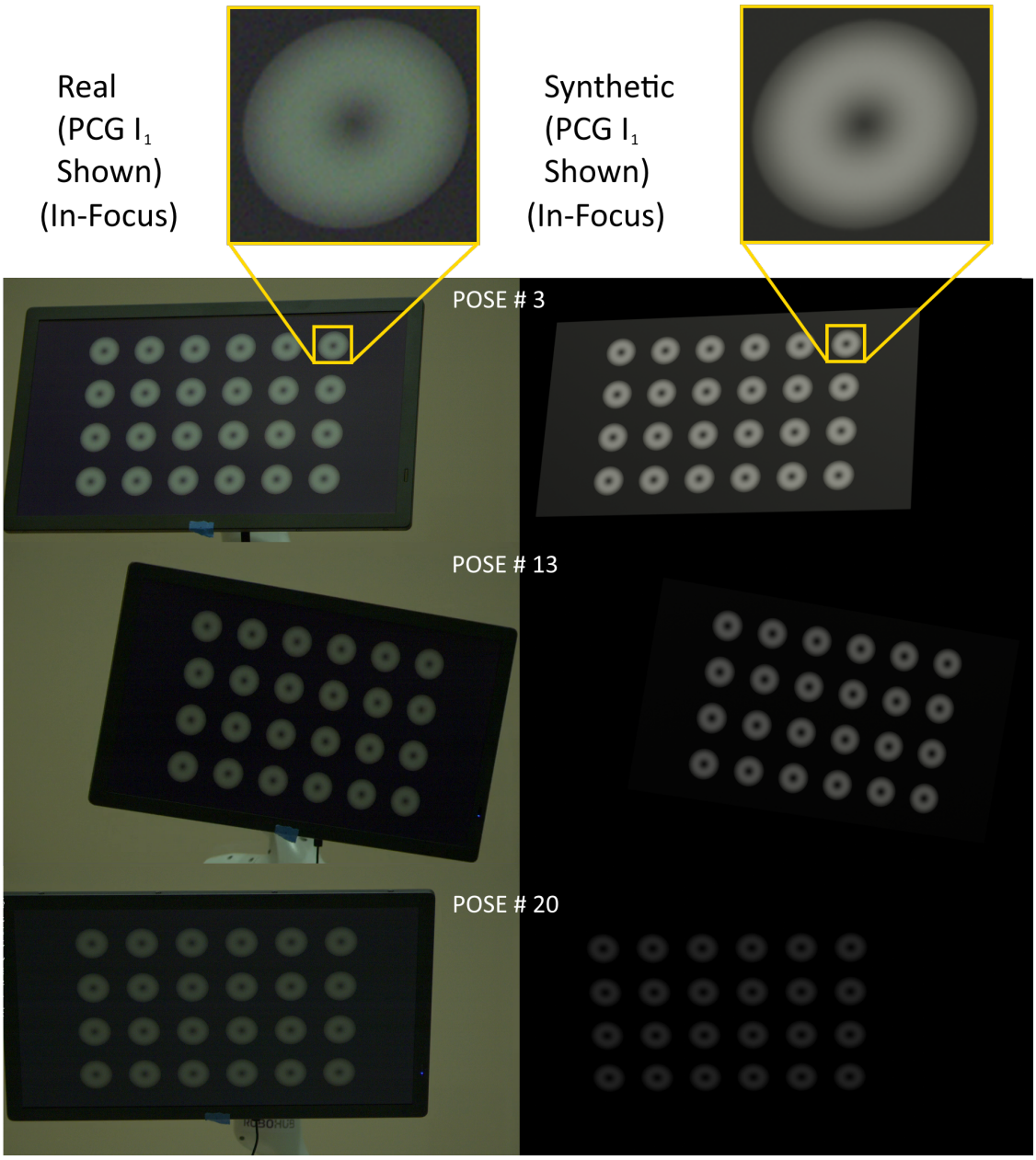


Figure 4.15. Sample of real images captured in different poses (left column), and respective pose-aligned synthetic images (right column). Note that the final 2 out of 20 poses (bottom row shown as sample) had a slight lighting mismatch where the downward facing synthetic images turned out darker due to a lack of full room modeling, where light was reflected from the floor and glass walls in the real world.

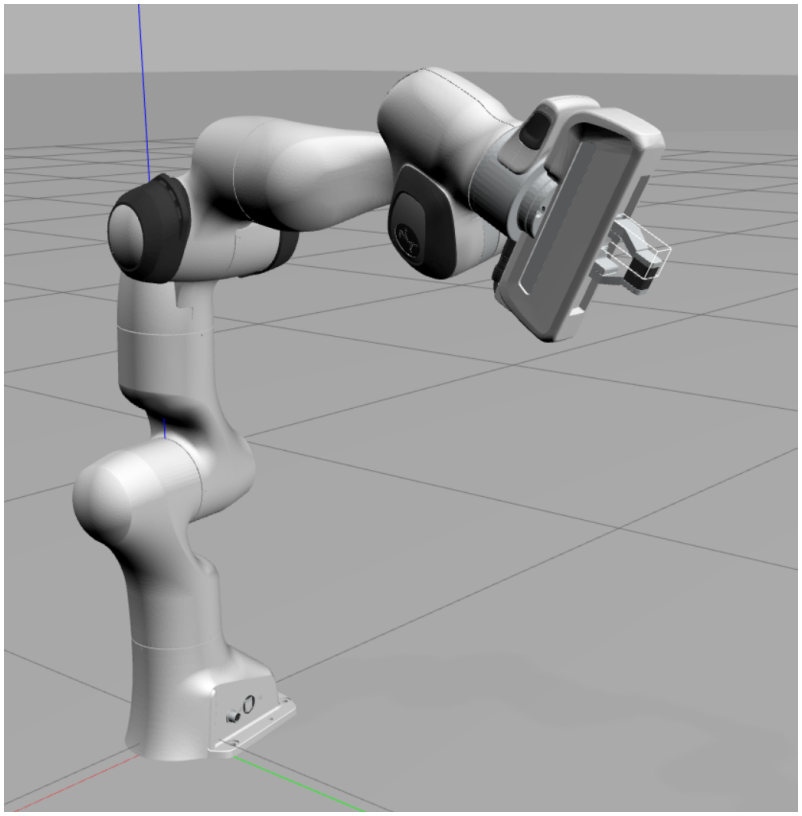


Figure 4.16. Franka Emika Panda robot shown in Gazebo simulation with left gripper finger queried for end-effector pose.

to the procedure in 4.4.1), only with the same 4×6 grid and using the same patterns found in section 4.3.2.

Special consideration was also taken to approximate the positioning and intensity of lights from the lab within the simulated Blender scene. Figure 4.15 illustrates the similarity between images captured in the lab vs. the pose aligned renderings.

4.5 Methods Summary

In chapter 4, we provided a recap of the workings of Zhang's method which we also refer to as the standard calibration method. We also explain the detailed steps of our re-implementation of a PCG method, and outline the detailed steps of pattern image generation, as well as wrapped phase image re-construction and our control point localization pipeline based on ellipse fitting.

We also highlighted the subtleties of control point detection and localization as well as the magnitude of sub pixel localization, control points that can be used for different calibration cases, and detection of control points from small circles as well as concentric circles.

We also discussed the provisions that were made for real-world experiments, including details about our experimental setups, and also explain the details about our synthetic experiment methods, synthetic ground truth focal length recovery method, and a summary of the overall advantages that our synthetic experimentation pipeline was found to provide.

Finally, we discussed how our last experiment was performed which involved using real-world robot guided calibration target poses to re-create similar conditions for a synthetic experiment. This closely approximated real world conditions, but eliminated certain aspects of real-world noise, to provide further insight into the underlying methods.

Chapter 5

Results and Discussion

Our results are shown in the following sections. Section 5.1 discusses how we chose to make relative comparisons between recovered focal lengths. Section 5.2 shows our initial validation of the PCG method based on a 4×5 PCG array in a simplified Blender environment without lighting or extensive scene modifications. Section 5.3 discusses the results of our robot guided real-world experiment, and section 5.4 discusses the results of the final experiment where the real-world experiment robot poses were synchronized with synthetically rendered images.

5.1 Comparison of Focal Lengths

In the following sections, we compare focal lengths estimated from camera calibration to either, known ground truth focal lengths in the case of synthetic data 5.2, or, in-focus to out-of-focus focal lengths in the case of real-world data 5.3, or both, in the case of our final results 5.4. In all cases, we provide the comparisons for both f_x and f_y, and represent the relative differences using the percent difference formula found in equation (5.1) [80]. Note that this differs from other methods of measuring relative change, such as percent error [80], and percent change. Since we cannot assume a ground truth in real cases, for consistency, we use percent difference for comparison across all cases:

$$Percent\ Difference = \frac{|E_1 - E_2|}{\frac{1}{2}(E_1 + E_2)} \cdot 100. \tag{5.1}$$

5.2 Initial Synthetic Experiment Results

Table 5.1 which was discussed in our previous work [67], shows the results of our initial synthetic experimental work where we confirmed that our PCG rendering and calibration pipeline was working as intended. Figure 5.1 shows that the initially recovered in-focus focal lengths at 4m distance are within 1.7 pixels of the ground truth, which is an average of 0.1092% difference from the ground truth. As out-of-focus blur increases, between 15m and 20m focus distances, the estimated focal length crosses the ground truth, and at 30m focus distance is only 0.0023% different from the ground truth. To avoid exhaustive dataset iteration, we tested an extremely blurred 100m focus distance case following 30m, and the resulting recovered focal lengths are just over 2 pixels away from the ground truth. An average percent difference of only 0.1440%.

We believe that these results validate the approach of Wang *et al.* [78], with respect to 3.7 demonstrating strong blur invariance. We also believe that this work shows a more realistic result, since Wang *et al.* only used Gaussian blurring as their synthetic blurring method, and we used a fully featured rendering environment that accounts for spatially dependant blur as shown in section 4.4.1 Figure 4.13. Although MRE and recovered focal lengths are closely related, we also believe that our results shown in focal lengths and percent differences are a better indicator of real-world effectiveness, because focal lengths incorporate the effects of MRE differences as well as the directionality of point spread biases, and are a final output for subsequent computer vision tasks.

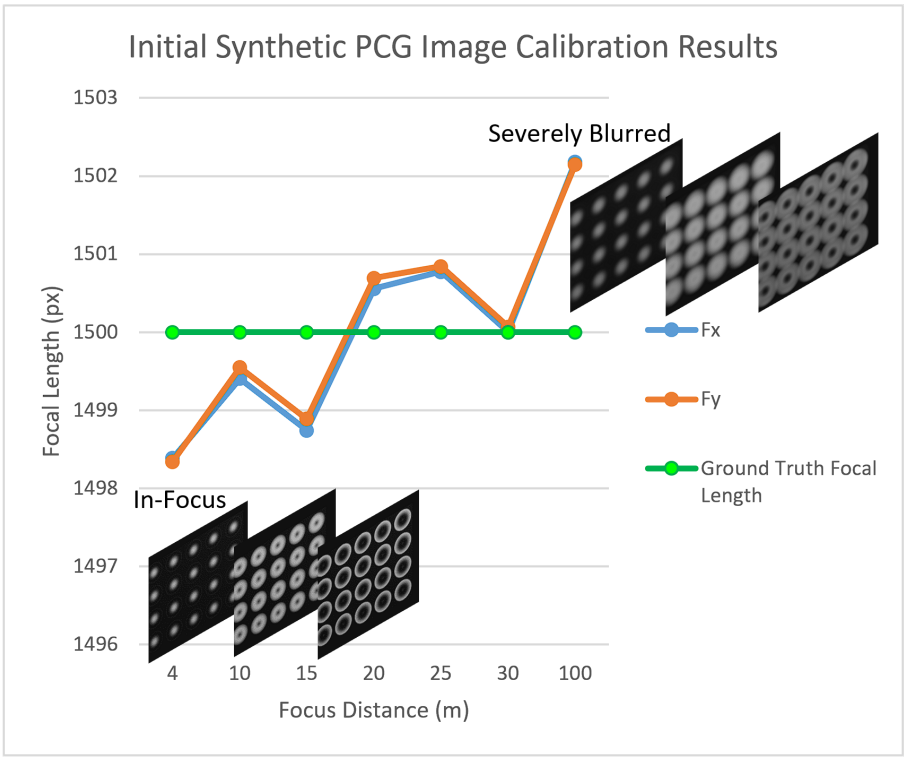


Figure 5.1. shows a plot of the focal lengths recovered from calibration using synthetically rendered PCGs, with increasing focus distance and respective blur. Each focus distance represents a set of 3 PCG images rendered in 10 different poses.

Table 5.1: Results for initial synthetic image experiment.

Initial Synthetic Image Experiment Results								
Focus Distance (m)	F-Stop	Ground Truth Focal Length (f_x & f_y)	**Recovered Focal Length f_x**	**Recovered Focal Length f_y**	Distance from Ground Truth f_x (px)	Distance from Ground Truth f_y (px)	Percent Difference f_x (%)	Percent Difference f_y (%)
4	1	1500	1498.39	1498.34	1.615	1.65819	0.108%	0.111%
10	1	1500	1499.41	1499.55	0.5927	0.4499	0.040%	0.030%
15	1	1500	1498.74	1498.89	1.25676	1.11229	0.084%	0.074%
20	1	1500	1500.56	1500.69	-0.55769	-0.69376	0.037%	0.046%
25	1	1500	1500.78	1500.84	-0.77567	-0.84371	0.052%	0.056%
30	1	1500	1500.00	1500.07	-0.00126	-0.06827	**0.000%**	**0.005%**
100	0.5	1500	1502.18	1502.15	-2.18149	-2.14609	**0.145%**	**0.143%**

5.3 Robot Guided Real-World Experiment Results

Table 5.2, which was also presented in our previous work [67], shows the results of our lab experiment where a robot was used to consistently position an E-Ink display for 20 poses. Three patterns were compared in this experiment which included small circles, large circles, and PCGs. Since an exact ground truth is not known, as a base case, we collected calibration focal length estimates from in-focus images for each pattern, for comparison to the out-of-focus estimates. Our goal is to find the most accurate out-of-focus calibration method. As it is presumed that the case with the smallest difference is the most likely to be estimating the focal lengths accurately, due to the estimates being closely in agreement, we consider the methods with the smallest difference between in, and out-of-focus images to be the best cases.

From PCG wrapped phase images, estimated focal lengths are provided using different combinations of the fitted inner and outer ellipses. In Table 5.2, we can see the four PCG cases where focal lengths were recovered using only inner ellipses, only outer ellipses, an average of the center points from both inner and outer ellipses, and center points that have been corrected using the method discussed in 2.4. Figure 5.2 is based on Table 5.2 and shows a plot of the differences between the in-focus cases, and out-of-focus cases. Here, we can see that small circles have the smallest difference between in, and out-of-focus images, which is the best case. The second-best case is using the corrected center points from PCG

images. Using averages of f_x, & f_y focal lengths, the difference between these two methods is very small, only 0.059%.

Large circles performed worse than small circles, but surprisingly the case using PCG outer ellipses was the worst overall. This result is counter intuitive because the large circles have a radius of 131 pixels, whereas the PCG outer ellipses (based on the PCG wrapped phase discontinuity circles) fall somewhere within this radius and are therefore smaller. Based on this result, we believe that the PCG ellipses are not being fitted as accurately as they are with OpenCV's internal circle grid detection function for the small and large circles. This could be due to fundamental differences with the PCG patterns response to noise, E-Ink display dithering, lighting, and pattern intensity, or it could also suggest that there is room for improvement with the ellipse fitting method that we used for PCGs. Although previous attempts had been made to use OpenCV's circle grid detection on the PCG wrapped phase images, we found that the OpenCV function had limitations stemming from blob detection consistency, and more direct control over parameters was required to improve detection. Therefore, it was necessary to follow the method in section 4.2.2 to obtain results.

Table 5.2: Results for robot guided real-world experiment.

	In-Focus		**Out-of-Focus**					
	f_x	f_y	f_x	f_y	Δ f_x (px)	Δ f_y (px)	Δ f_x (%)	Δ f_y (%)
Small Circles	9903.623	9903.631	9926.619	9926.94	**22.996**	**23.309**	**0.232%**	**0.235%**
Large Circles	9888.723	9887.93	9929.184	9929.735	40.461	41.804	0.408%	0.422%
PCG Inner Ellipses	9886.005	9886.428	9923.071	9923.057	37.066	36.629	0.374%	0.370%
PCG Outer Ellipses	9849.408	9850.551	9908.761	9909.463	59.353	58.912	0.601%	0.596%
PCG Averaging All Ellipses	9867.738	9868.61	9916.957	9917.389	49.219	48.779	0.498%	0.493%
PCG Center Point Correction	9898.482	9898.689	9927.695	9927.448	**29.213**	**28.759**	**0.295%**	**0.290%**

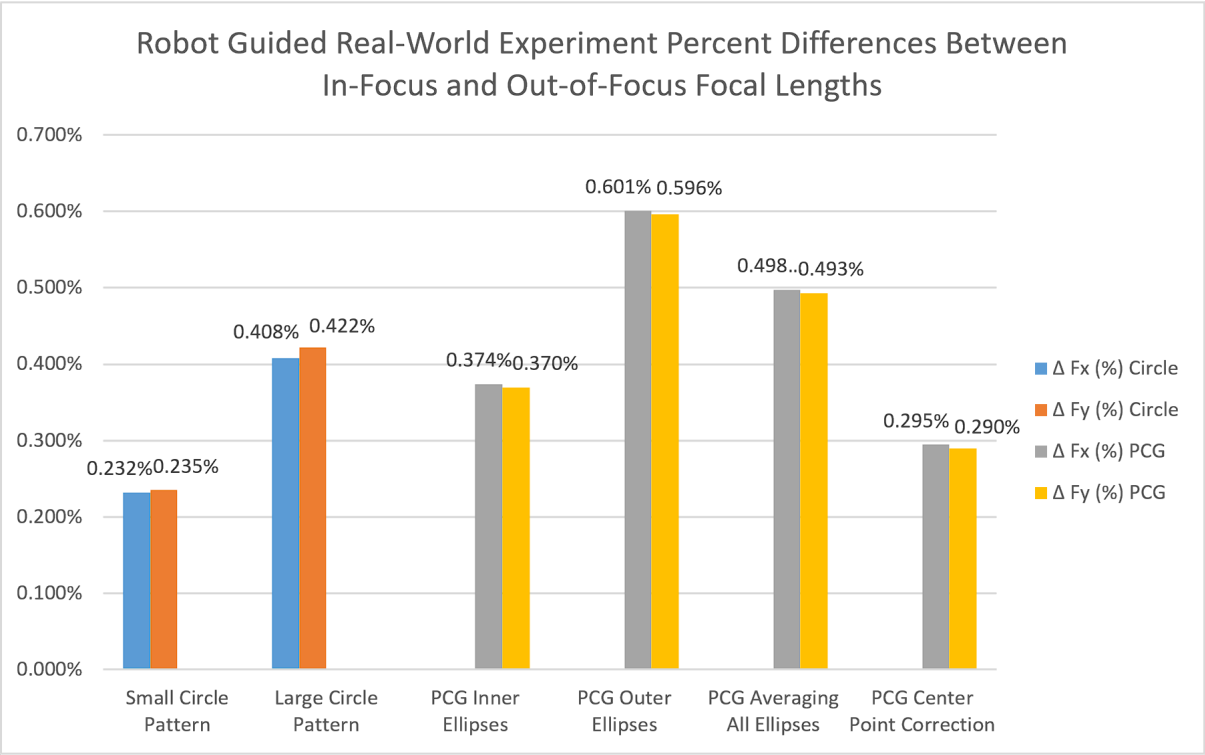


*Figure 5.2. shows a plot of the percent differences between real-world in-focus, and out-of-focus focal lengths on several patterns and control point localization methods. (***smaller percent differences are better).**

5.4 Results for Final Synthetic Experiment Aligned with Robot Guided Real-World Experiment

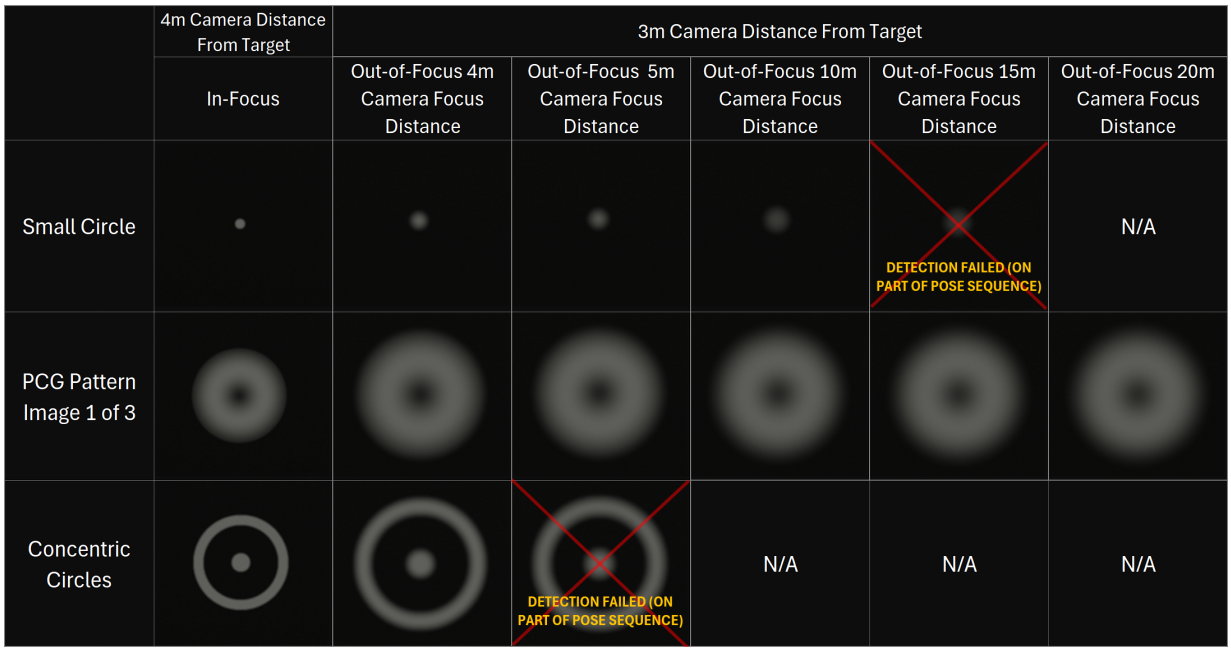
The goal of our final experiment was to gain further insight about the capabilities and limitations of small circles, concentric circles, and PCGs by building on the previous experimental work. To do this, we followed the method described in section 4.4.4 which is based on replicating the results of the previous robot guided lab experiment. We did this by copying the pose sequence of the robot's end-effector into the synthetic rendering environment. Figure 5.3 shows samples of the top leftmost circle from the 4×6 synthetic image grids of the small circles, PCGs, or concentric circle patterns that were tested. Note, that all of the circles that are shown in Figure 5.3 are only for the first pose out of 20 poses. From the robotic experiment that this is based on, we deliberately made the first pose calibration pattern plane orthogonal to the camera axis, with no perspective angle applied. In the lab, this helped with positioning and centering the target in the FOV. Since this is the case, all of the circles that are shown for this initial pose are not tilted, but later poses in the sequence are tilted in various directions that do not exceed 45° of rotation.

Table 5.3: Results for final synthetic experiment with lab experiment pose alignment and lighting condition replication. NOTE: All values shown are an average of f_x and f_y.

	In-focus 4m camera dist, 4m focus dist	Out-of-focus 3m camera dist, 4m focus dist	Out-of-focus 5m focus dist	Out-of-focus 10m focus dist	Out-of-focus 15m focus dist	Out-of-focus 20m focus dist
Small Circles ($f_x, f_{y\ \text{AVG}}$)	**8803.5456**	**8799.4637**	**8796.8263**	**8805.5990**	–	–
Corrected PCGs ($f_x, f_{y\ \text{AVG}}$)	**8790.9843**	**8794.1393**	**8807.8803**	**8803.4895**	**8793.6621**	**8798.6678**
Concentric Circles ($f_x, f_{y\ \text{AVG}}$)	**8783.8289**	**8796.4632**	–	–	–	–

In Figure 5.3, we can also see that the samples from the synthetic images were rendered with increasing levels of out-of-focus blur. The first two columns mirror the in and out-of-focus conditions of the robotic lab experiment. In the first column, the camera is set

Figure 5.3. shows the synthetically rendered patterns that were used in the final experiment. All of the sample images are from the top left circle of the 4×6 grid of each type of calibration pattern. Also, all of the circles that are shown are for pose 1 out of 20 (where no perspective angle has been applied yet). The left column is in-focus, and moving to the right is increasingly out-of-focus. The grids of these circles, and their respective image sequences were used to generate the results in Table 5.3.



to focus 4m into the distance, and is 4m away from the target, so it is in-focus. In the second column, the camera is still set to focus 4m into the distance, but it is moved closer to the target by 1m (and is 3m from the target), therefor, the images appear larger, and are out-of-focus.

The next four columns to the right, however, extend the out-of-focus conditions in a way that is only possible with synthetic rendering. This is due to real world FOV limitations. Specifically, all focus distances $\geq$ 5m were rendered using the same 3m camera distance from the calibration target as the previous out-of-focus case, but with fixed camera intrinsics and the camera focus distance extended further past the target, into the distance (from 5m up to 20m, in 5m increments).

If this had been a real-world case, it would not have been possible to bring the camera closer to the target, because the circles near the edge of the 4×6 grid would exceed the

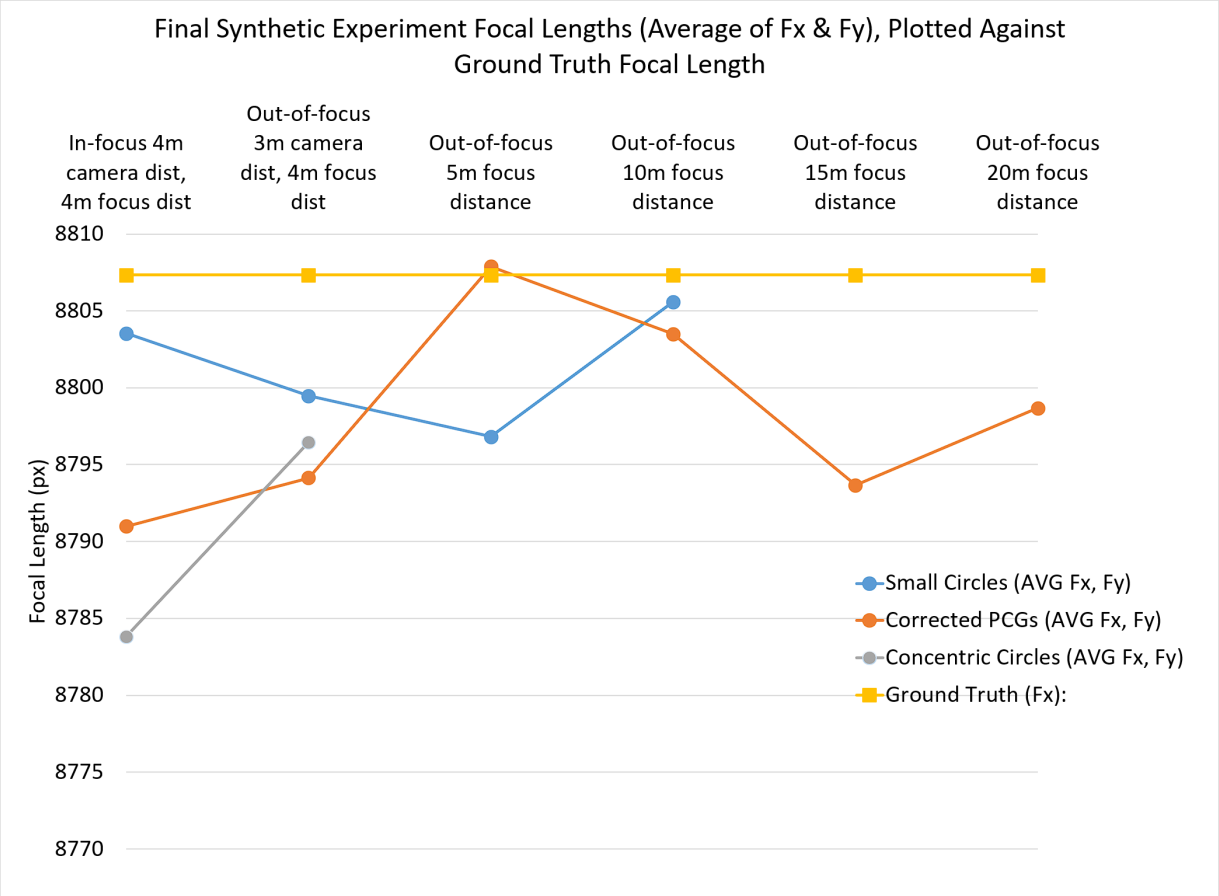
image FOV and become partially occluded or no longer visible. With Blender on the other hand, this is possible because although we render images based on a perspective pinhole camera model, using Blender's EEVEE render engine, depth of field (DoF), is handled as a post-processing operation, so it does not influence the camera model, and is therefor decoupled from the focal length. Table 5.3 shows the averaged (f_x, f_y) focal lengths for all the cases that are respective to Figure 5.3. Note that all PCG and concentric circle pattern cases use the concentric circle correction method from section 2.4, since the previous experiment in section 5.3 validated that this method works the best.

Figure 5.4 is a plot representing the focal lengths from Table 5.3, which includes the ground truth focal length for all of the synthetic pattern image sequences, which is 8807.339px. From this plot, we can see that for the initial in-focus case, the small circles are closest to the ground truth, whereas PCGs are next, and concentric circles are the worst. For the next out-of-focus (3m camera distance, 4m focus distance) setting, both concentric circles and small circles are closest to the ground truth, and PCGs improve as well, but are the worst out of the three.

For the next 5m focus distance setting, in Figure 5.4, we can see that the focal lengths recovered using PCGs slightly crosses, and almost matches the ground truth. From Table 5.3, we can see that this recovered focal length value is 8807.880, which is only 0.541px away from the ground truth of 8807.339px. In Figure 5.5, we can see that this equates to a percent difference of only 0.006%. At this point, the concentric circles are no longer detectable. The most difficult images to detect concentric circles from appeared to be from the last two poses at the end of the pose sequence. These two images were noticeably darker due to calibration target angle and simulated overhead lighting. This lighting difference was the same across all three of the patterns (PCG, small circles, concentric circles), but only concentric circles were adversely affected.

The small circle method uses fixed blob detection parameters across all poses and relies on OpenCV's find-circle-grid function, whereas the PCG detection method and concentric circles rely on our ellipse fitting pipeline from section 4.2.2. Using this pipeline, concentric circles were only detectable when we added individual morphology parameters that were hand tuned for each pose. This contrasts with PCGs where this level of fine tuning was not needed. This is because the PCG wrapped phase images are always a reconstruction of the original images (I_1 to I_3), with consistent pixel intensity, meaning that only the initial process of generating wrapped phase images is sensitive to lighting. Since there are three images to work with that cover the intensity spectrum throughout the area of the circles, there is more pixel information to work with, and therefor greater robustness to lighting than the concentric circle method.

Figure 5.4. shows the average of f_x and f_y for the recovered focal lengths for each of the in and out-of-focus cases, for small circles, PCGs, and concentric circles. These patterns and focus cases are plotted against the ground truth focal length of 8807.339px which was common to all of the synthetic Blender scenes that generated the image sequences from which the focal lengths were recovered.



At 10m out-of-focus, we obtain close to ground truth results for both PCGs and small circles, and from Figure 5.5, we can see that PCGs are 0.044% difference from the ground truth, and small circles are even better at only 0.020%. At 15m small circles are no longer detectable, even after numerous attempts to tune blob detection parameters. There appears to be limitations in the range of operation of OpenCV's blob detector when dealing with varying pixel intensities across a grid pattern, and for small circles to continue to be detectable, individual parameters would need to be hand tuned, or set adaptively, for every

Figure 5.5. shows a plot of the percent differences between the recovered average (of f_x, and f_y), focal lengths and the ground truth focal length.

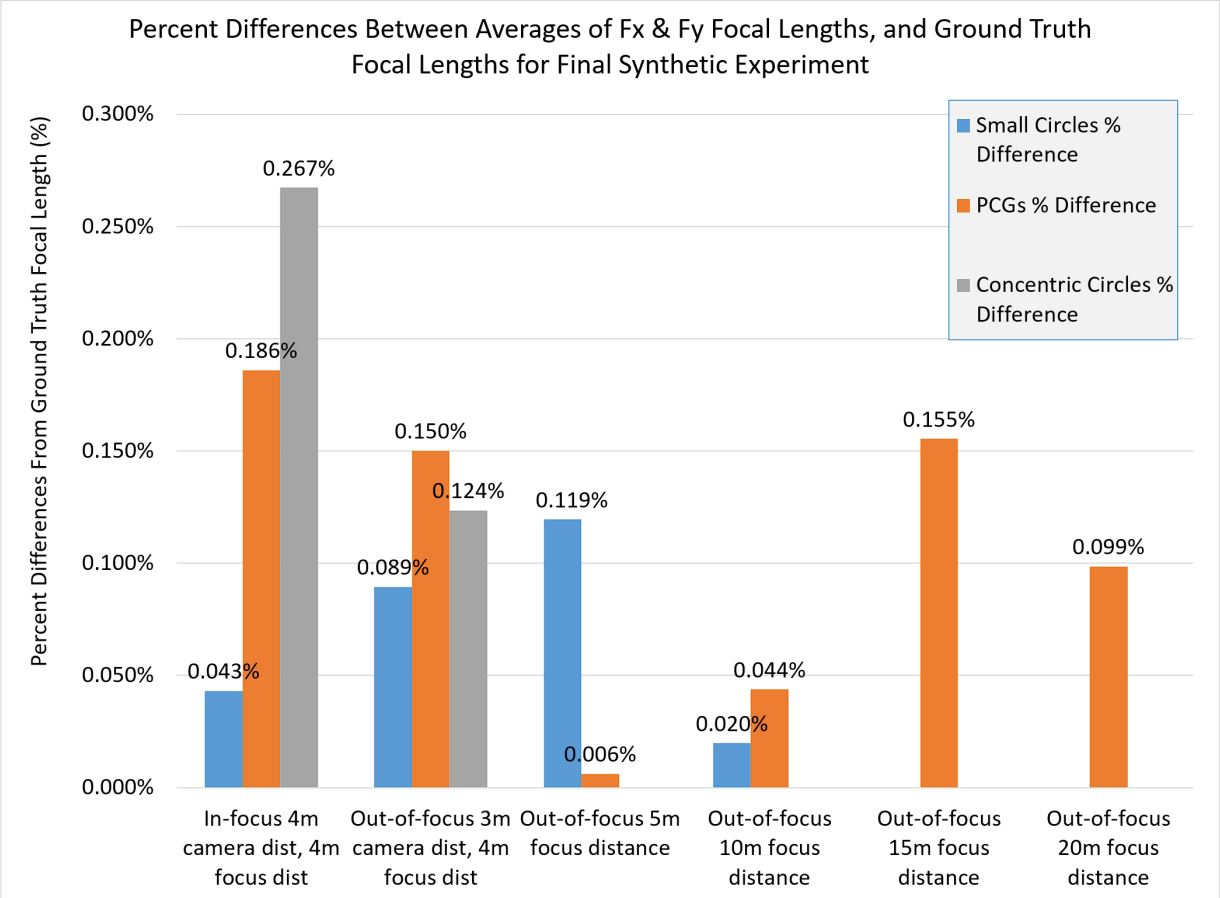


image. Using our ellipse fitting pipeline, and continuing to use only one set of canny edge detection parameters across all images in each sequence, PCGs continue to be detectable all the way up until 20m, where the percent differences vary from 0.155% for 15m, to 0.099% at 20m.

In addition to rendering increasingly blurred images within the FOV, synthetic rendering has shown the added advantage of maintaining image spatial resolution. This is achieved by progressively blurring the calibration patterns without changing their size. In contrast, real-world scenarios don't allow for this, as images of the same pattern end up being captured with circles of different radii due to changes in distance. Although a camera's focus ring could be adjusted to increase blur, this would change the effective focal length

of the camera-lens system, and calibration would recover a respectively different result.

This distinctive feature of synthetic rendering not only enhances the amount of data that can be collected for a given test setup, but also allows for comparison of blur cases based on the same pattern area, and relative number of pixels (other than the spread of pixels from blurring). This is something that helps to remove variability when comparing patterns, and isn't possible in real-world experiments.

Later, it was also discovered that the camera sensor width of 10.9mm that had been entered into the Blender scenes was inadvertently based on the effective sensor diagonal based on the real Basler acA1920-150uc camera. Based on this value, the effective ground truth focal length of the virtual pinhole camera model for f_x and f_y is 8807.339px, which correctly coincides with the calibration recovery results, and again validates our synthetic rendering and focal length recovery methods. If the correct sensor width of 9.22mm (calculated based on aspect ratio and the effective diagonal of 10.9mm) was used, then the ground truth focal length would be 10412.147px (see Appendix A for calculations). This likely explains why the focal lengths from the real-world experiment were higher, and in the range of 9800 - 9900 px. Also, the difference between 10412px and the real world recovered range $\sim$ 9850px is likely explained by the fact that it is not possible to know the real-world effective camera system focal length due to the analog setting of the focus ring.

Since the synthetic images were scaled to match the real images, there may have been a small distance offset that was not known or accounted for from the real world, between the robot base and camera. This would allow the images to look very similar without the possibility of their being a humanly perceptible difference in the camera's perspective depth that would relate to the differences in focal lengths. However, since the images were scaled similarly, and it is not possible to know the real-world effective camera system focal length, this experiment proved to be no less valuable, because it shows a close relative comparison between the real world and what could happen if blur conditions could be tested further for each calibration test pattern. This could in essence shed light on how longer focal length cameras would work with different calibration patterns without having to physically iterate through them in real life.

As a final comparison, we can look at the percent differences calculated in the real-world experiment 5.3 and our approximated synthetic re-creation. For comparison, Table 5.4 shows the real-world results vs. the synthetic re-creation, showing percent differences between in and out-of-focus cases, for both small circles and PCGs. Note that the real-world focal lengths are averages of (f_x, and f_y) from 5.2. On average (between small circles and PCGs), the percent differences of focused to out-of-focus focal lengths from the real-world are over 6$\times$ higher than the average percent differences from the synthetic re-

Table 5.4: Comparison of real-world results vs. synthetic re-creation of real-world experiment showing percent differences between in and out-of-focus cases for PCGs and small circles. Lower percent differences are better. Here, we can see that PCGs are affected by real world noise, but can theoretically perform better than small circles when subjected to out-of-focus blur.

		In-focus 4m camera dist, 4m focus dist	Out-of-Focus 3m	Percent Difference
Real World:	Small Circles (AVG f_x, f_y)	9903.627174	9926.779806	**0.2335%**
	Corrected PCGs (AVG f_x, f_y)	9898.585495	9927.571513	**0.2924%**
Synthetic (Based on Real World):	Small Circles (AVG f_x, f_y)	8803.545608	8799.463668	**0.0464%**
	Corrected PCGs (AVG f_x, f_y)	8790.98432	8794.139312	**0.0359%**

creation. Also, PCGs perform better than small circles in the synthetic case than in the real world. Although the differences are small, this underscores that it is important to account for the effects of noise as well as the accuracy of ellipse fitting techniques. This is evident because the main visible image differences between the real and synthetic images (shown in Figure 4.15 from section 4.4.4), are that there is noticeable camera exposure noise, as well as a pattern pixelation effect due to the dithering process of the E-Ink display.

5.5 Results Summary

In this chapter we outlined the results of three experiments, which iterated between synthetic and real-world experimental setups. The first experiment confirmed the efficacy of our overall synthetic rendering pipeline, and PCG calibration method. The second and third experiments were directly related, as they were configured the same way, and highlighted the differences between real world and synthetic results.

Using percent difference as our main comparison metric, we highlighted the trade-offs, advantages, and disadvantages between different ellipse fitting cases, as well as comparisons between small circles, PCGs, and concentric circle patterns.

In our last experiment, we directly compared our real-world results to the synthetic re-construction to better understand the magnitude of differences between the results.

Our overall conclusions from these results are that the PCG method performed extremely well under ideal conditions but suffered from reduced accuracy when subjected to real world noise, which we hypothesize is owing to their increased complexity and the efficacy of chosen ellipse fitting methods. However, the main advantage that was observed is that PCGs continue to be detectable after small circles, and long after concentric circles. This appears to be due to their increased pixel area from combined images, which adds robustness to extreme blur and variations in lighting conditions.

Chapter 6

Conclusions and Recommendations

This work aims to find the most accurate methods of control point localization for out-of-focus camera calibration. Towards this search, we reviewed a large volume of literature, and attempted to validate previous work from literature on PCG calibration target patterns, by comparing them to small circle grid patterns, and later concentric circles. Initial comparisons showed that PCGs were largely invariant to blur. However, lab experiments with real world noise showed that their accuracy was marginally worse than comparable small circles. Further experiments were conducted using synthetically generated images that closely mirrored real world images taken in the lab. Unlike from most related work, this was possible because of our efforts to preserve target pose consistency by using a robotic manipulator. The goal of this synthetic work was to preserve some real-world variability such as pose alignment and lighting, while also removing other real-world noise such as camera noise and grayscale display dithering. The results showed that when synthetically approximating real-world conditions, PCGs returned smaller differences in in-focus vs. out-of-focus focal lengths than small circles. However, when the out-of-focus intensity was extended beyond real world conditions, PCGs had more variability, than for small circles, but in one out-of-focus case, also returned the closest estimate to the ground truth. PCGs also continued to be detectable under the most blurred conditions.

We conclude that PCGs are the most robust pattern to variations in lighting and high intensity out-of-focus blur, as they proved to be easier to detect as out-of-focus blur increased, and when particular poses reflected difficult lighting conditions. This is believed to be due to their larger combined image pixel area, and consistently re-produced wrapped phase images. PCGs are therefor validated to be favorable for calibration of longer focal length cameras, and for calibration requirements in space constrained environments where lens distortion is minimal.

After further experimentation using synthetic images of concentric circles, it is clear that ellipse fitting methods play a vital role in the accuracy of control point localization for both concentric circles and for PCGs. In this work the Canny edge detector is used as the basis for ellipse fitting and center point localization, but detectability of ellipses and quality of detections prove to be sensitive, and are contingent on the Canny edge detector parameters that are chosen. From here, further work can be undertaken to determine if adaptive thresholding techniques could improve robustness, and if better ellipse fitting techniques, such as structured forests edge detection can enhance any of the phase-based circular pattern, or circle-based methods that were tested. Fundamentally different localization methods such as the technique based on conservations of gradient flows (from Calvet *et al.* [19] mentioned in section 3.5.1) could also be examined to determine if more accurate and robust control point localization is possible.

We also therefor conclude that there are trade-offs in calibration pattern selection that are necessary to consider when pairing with a respective use case. PCGs are generally more complicated to implement, require more images to be captured, and require more processing steps. This greater number of steps can lead to the heightened probability of numerical stack-up errors, which in tern can lead to less consistent results. Small circles on the other hand work well with existing circle detectors and should be used in more general cases. Although our work does not consider the effect of distortion, as per Mallon and Whelan [59], circles with a radius of 10px are stated to be sufficient for reducing the bias of lens distortion, whereas PCG patterns with their larger radiuses can only be worse when distortion is considered. This is another reason for using small circles in most general cases, however, Mallon and Whelan [59] do not specifically address high levels of out-of-focus blur in their work. This is why PCGs should be considered for very high focal length applications, where little distortion is present and when special attention is placed on parameter tuning, ellipse fitting methods, step validation, and numerical accuracy.

Since square phase-based methods were outside of the scope of our work, but did show some initially promising results, we also recommend testing the code from Schmalz *et al.* [3, 68] as a starting point, to determine if square phase-based methods are worth further consideration.

We also believe that our synthetic validation pipeline and experimentation method has the potential to closely approximate real-world conditions and is valuable for testing long focal length calibration patterns under various conditions. The ability to recover ground truth intrinsic parameters is invaluable for observing extremely small variations between calibration patterns, relative to a ground truth when recovering intrinsics with high accuracy. We have also demonstrated some special capabilities of synthetic rendering that extends results beyond real-world testing, and can isolate variables such as real-world

changes in pattern spatial resolution, and removal or control of noise to allow for closer comparisons between different patterns. With some refinement, these testing methods could be extended further to test a larger number of variables more rapidly, and with even closer to real-world results.

We also found that comparisons using percent differences are valuable since experiments from different scenes may return different magnitudes of pixel focal lengths, but it is the relative difference between recovered in and out-of-focus focal lengths that provides an indication of overall accuracy.

Finally, although our work focused on deterministic non-learning-based algorithms, we would like to acknowledge the role of learned methods such as deep neural networks and other machine learning methods in camera calibration. Although there are only a few learned methods that relate directly to camera calibration from out-of-focus images, our general takeaway is that learned methods can be valuable for encoding certain parts of the calibration, or control point detection process. Although it is possible to implement learned methods deterministically, larger and more advanced models rely on elements of stochasticity to perform well. Also, end-to-end learned calibration methods can be more difficult to control and implement effectively.

Although learned methods have not yet been proven to increase accuracy when relating to our work, we believe that they have the potential to improve the extensibility, and ease of use for aspects of out-of-focus calibration. Therefor, we recommend further exploration of learned methods (such as structured forests edge detection) for tasks such as ellipse fitting, or even to replace the entire control point detection and localization process. This however would only be effective provided that a given learned method is trained correctly on adequate variations of ground truths for desired conditions. We also believe that some extension of our synthetic rendering and ground truth recovery pipeline could be valuable in this regard, since it could be adapted to perform training dataset synthesis and data augmentation.

Table of Copyright Content Reuse Permissions and Licenses

Figure	Table	Citation No.	Publisher	Permission, Invoice No. or RightsLink License No.
2.3		[52]	HAL Open Science	Permission email
2.4		[77]	Elsevier	5873930308736
3.4		[26]	Elsevier	5873100004458
3.5		[38]	IEEE	The IEEE does not require individuals working on a thesis to obtain a formal reuse license.
3.6		[78]	Optica Publishing Group	Invoice # 2263132, for the use of: Figures 1, 10 and Table 1 from [78]
	3.1			
3.7				
3.8		[77]	Elsevier	5873930308736
3.9		[24]	Elsevier	5873921430784
3.10				
3.11				
	3.3			
3.12		[90]	Springer Nature	5875531099349
3.14		[23]	Springer Nature	5875531364412
3.15		[85]	Optica Publishing Group	Invoice # 2263132, for the use of: Figure 1 from [85]
3.16		[46]	Elsevier	5873920499397
3.17				
3.18				
3.19				
3.20				
3.21				
	3.4			
3.22		[16]	IEEE	The IEEE does not require individuals working on a thesis to obtain a formal reuse license.
3.23		[53]		
3.24		[64]		
3.25		[74]	Springer Nature	5875540019944

Licenses

License No. 5873930308736

ELSEVIER LICENSE
TERMS AND CONDITIONS

Sep 22, 2024

This Agreement between Ryan Schmalenberg ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

License Number	5873930308736
License date	Sep 21, 2024
Licensed Content Publisher	Elsevier
Licensed Content Publication	Optics and Lasers in Engineering
Licensed Content Title	Out-of-focus color camera calibration with one normal-sized color-coded pattern
Licensed Content Author	Yuwei Wang,Bolin Cai,Keyi Wang,Xiangcheng Chen
Licensed Content Date	Nov 1, 2017
Licensed Content Volume	98
Licensed Content Issue	n/a
Licensed Content Pages	6
Start Page	17
End Page	22

Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	2
Format	electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No
Title of new work	Camera Calibration from Out-of-Focus Images
Institution name	University of Waterloo
Expected presentation date	Oct 2024
Portions	Figures 2, and 4
The Requesting Person / Organization to Appear on the License	Ryan Schmalenberg
Requestor Location	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Publisher Tax ID	GB 494 6272 12
Total	0.00 CAD

License No. 5873100004458

ELSEVIER LICENSE
TERMS AND CONDITIONS

Sep 22, 2024

This Agreement between Ryan Schmalenberg ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

License Number	5873100004458
License date	Sep 20, 2024
Licensed Content Publisher	Elsevier
Licensed Content Publication	Pattern Recognition Letters
Licensed Content Title	Robust control point estimation with an out-of-focus camera calibration pattern
Licensed Content Author	Hyunseok Choi,Ho-Gun Ha,Hyunki Lee,Jaesung Hong
Licensed Content Date	Mar 1, 2021
Licensed Content Volume	143
Licensed Content Issue	n/a
Licensed Content Pages	7
Start Page	1
End Page	7

Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Format	electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No
Title of new work	Camera Calibration from Out-of-Focus Images
Institution name	University of Waterloo
Expected presentation date	Oct 2024
Portions	Figure 7, on page 5 of 7
The Requesting Person / Organization to Appear on the License	Ryan Schmalenberg
Requestor Location	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Publisher Tax ID	GB 494 6272 12
Total	0.00 CAD

License No. 5873921430784

ELSEVIER LICENSE
TERMS AND CONDITIONS

Sep 22, 2024

This Agreement between Ryan Schmalenberg ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

License Number	5873921430784
License date	Sep 21, 2024
Licensed Content Publisher	Elsevier
Licensed Content Publication	Optics and Lasers in Engineering
Licensed Content Title	Camera calibration with global LBP-coded phase-shifting wedge grating arrays
Licensed Content Author	Xiangcheng Chen,Xiaokai Song,Jun Wu,Yongxin Xiao,Yajun Wang,Yuwei Wang
Licensed Content Date	Jan 1, 2021
Licensed Content Volume	136
Licensed Content Issue	n/a
Licensed Content Pages	1
Start Page	106314
End Page	0

Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	4
Format	electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No
Title of new work	Camera Calibration from Out-of-Focus Images
Institution name	University of Waterloo
Expected presentation date	Oct 2024
Portions	Figures 1, 2, 10 and Table 1
The Requesting Person / Organization to Appear on the License	Ryan Schmalenberg
Requestor Location	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Publisher Tax ID	GB 494 6272 12
Total	0.00 CAD

License No. 5875531099349

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Sep 24, 2024

This Agreement between Ryan Schmalenberg ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5875531099349
License date	Sep 24, 2024
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	A Neural Network-Based Camera Calibration Method for Mobile Robot Localization Problems
Licensed Content Author	Anmin Zou, Zengguang Hou, Lejie Zhang et al
Licensed Content Date	Jan 1, 2005
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	figures/tables/illustrations
Number of figures/tables/ illustrations	2

Will you be translating?	no
Circulation/distribution	5000 - 9999
Author of this Springer Nature content	no
Title of new work	Camera Calibration from Out-of-Focus Images
Institution name	University of Waterloo
Expected presentation date	Oct 2024
Portions	Figures 1, and 2 on page 279 (3 of 8)
The Requesting Person / Organization to Appear on the License	Ryan Schmalenberg
Requestor Location	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Billing Type	Invoice
Billing Address	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Total	0.00 CAD

License No. 5875531364412

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Sep 24, 2024

This Agreement between Ryan Schmalenberg ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5875531364412
License date	Sep 24, 2024
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	CCDN: Checkerboard Corner Detection Network for Robust Camera Calibration
Licensed Content Author	Ben Chen, Caihua Xiong, Qi Zhang
Licensed Content Date	Jan 1, 2018
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1

Will you be translating?	no
Circulation/distribution	5000 - 9999
Author of this Springer Nature content	no
Title of new work	Camera Calibration from Out-of-Focus Images
Institution name	University of Waterloo
Expected presentation date	Oct 2024
Portions	Figure 1 on page 3
The Requesting Person / Organization to Appear on the License	Ryan Schmalenberg
Requestor Location	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Billing Type	Invoice
Billing Address	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Total	0.00 CAD
Terms and Conditions	

License No. 5873920499397

ELSEVIER LICENSE
TERMS AND CONDITIONS

Sep 22, 2024

This Agreement between Ryan Schmalenberg ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

License Number	5873920499397
License date	Sep 21, 2024
Licensed Content Publisher	Elsevier
Licensed Content Publication	Measurement
Licensed Content Title	Feature points extraction of defocused images using deep learning for camera calibration
Licensed Content Author	Junzhou Huo,Zhichao Meng,Haidong Zhang,Shangqi Chen,Fan Yang
Licensed Content Date	Jan 1, 2022
Licensed Content Volume	188
Licensed Content Issue	n/a
Licensed Content Pages	1
Start Page	110563
End Page	0

Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	6
Format	electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No
Title of new work	Camera Calibration from Out-of-Focus Images
Institution name	University of Waterloo
Expected presentation date	Oct 2024
Portions	Figure 1, 2, 6, 13, 14, and Table 5
The Requesting Person / Organization to Appear on the License	Ryan Schmalenberg
Requestor Location	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Publisher Tax ID	GB 494 6272 12
Total	0.00 CAD

License No. 5875540019944

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Sep 24, 2024

This Agreement between Ryan Schmalenberg ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5875540019944
License date	Sep 24, 2024
Licensed Content Publisher	Springer Nature
Licensed Content Publication	The Visual Computer
Licensed Content Title	Mono-spectrum marker: an AR marker robust to image blur and defocus
Licensed Content Author	Masahiro Toyoura et al
Licensed Content Date	Dec 20, 2013
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1

Will you be translating?	no
Circulation/distribution	5000 - 9999
Author of this Springer Nature content	no
Title of new work	Camera Calibration from Out-of-Focus Images
Institution name	University of Waterloo
Expected presentation date	Oct 2024
Portions	Figure 2, on page 1037 (3 of 10)
The Requesting Person / Organization to Appear on the License	Ryan Schmalenberg
Requestor Location	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Billing Type	Invoice
Billing Address	Mr. Ryan Schmalenberg 200 University Avenue West Waterloo, ON N2L 3G1 Canada Attn: Mr. Ryan Schmalenberg
Total	0.00 CAD
Terms and Conditions	

Figure 2.3 Reuse Permission Email

Ryan Schmalenberg

From: Mathieu LABUSSIERE [redacted]
Sent: September 23, 2024 2:57 AM
To: Ryan Schmalenberg
Subject: Re: URGENT: Request for Copyright Permissions for Image Reuse in Thesis / Dissertation
Attachments: cameramodel.pdf

Dear Ryan,

Of course, I have no problem for you using this figure. Just make sure to cite the manuscript! :)

Please find enclosed, the pdf version of the figure which is of better quality.

I'm glad you enjoyed reading the manuscript!

Best regards,

Mathieu LABUSSIERE
[redacted]

[redacted]

[redacted]

[redacted]

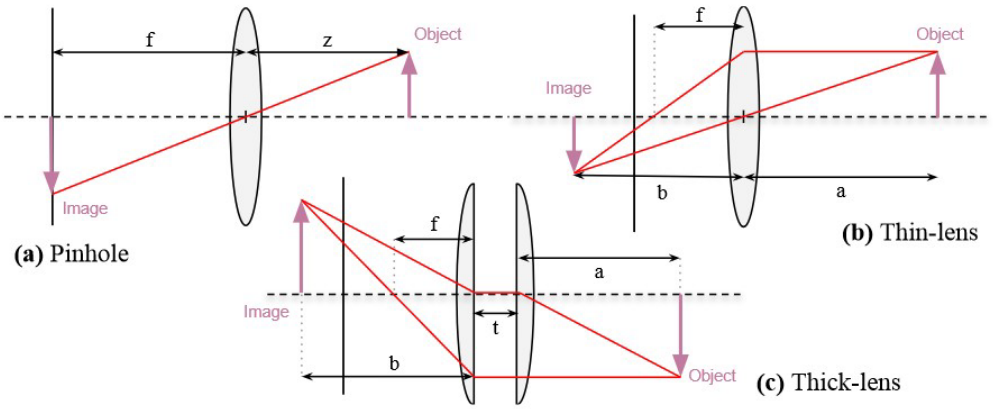
De: "Ryan Schmalenberg" [redacted]
À: "mathieu labussiere" [redacted]
Envoyé: Vendredi 20 Septembre 2024 21:44:52
Objet: URGENT: Request for Copyright Permissions for Image Reuse in Thesis / Dissertation

Hello Dr. Labussière,

I am a graduate student at the University of Waterloo in Ontario Canada, and I am preparing my thesis for deposit in UWSpace, the university's institutional repository. I understand that you are the copyright holder for your thesis:
Leveraging blur information in plenoptic cameras - Application to calibration and metric depth estimation
https://hal.science/tel-03554252/file/LABUSSIERE_Mathieu_manuscrit.pdf

I would like permission to include Figure 3.1: "Examples of camera models" (see image below) in my thesis, which will be made openly available as part of my thesis, titled, *Camera Calibration from Out-of-Focus Images*. Proper citation will be included with the reproduction of this figure.

1



Please let me know what your process is for providing permission, and if you need any further information from me.

If you do not hold the copyright for this material, or the right to grant this type of permission, I would greatly appreciate any information you can provide to me regarding the rights holder(s), including any contact information.

Thank you for considering this request, and thank you for your excellent thesis, I enjoyed reading it and learning from it!

Ryan Schmalenberg

MASc. Candidate

Vision and Image Processing Lab

Systems Design Engineering

University of Waterloo
Waterloo, Ontario

Optica Publishing Group Permission Email

Ryan Schmalenberg

From: Optica Publishing Group Copyright <copyright@optica.org>
Sent: September 23, 2024 3:17 PM
To: Ryan Schmalenberg
Subject: RE: URGENT: Request for Copyright Permissions for Thesis / Dissertation Reuse {2917}

--reply above this line--

Dear Ryan,

I believe my colleague just forwarded you a receipt for your invoice just now, which will have an invoice number and confirmation that it was paid. Optica typically does not send out a formal copyright permission document and considers this email to be sufficient authorization for the described use of the requested material under the specified terms. A PDF license or certificate is not needed.

For your convenience, I will add a final permissions statement below, but please let me know if you would need an actual document and I can put one together.

For the use of figures 1, 10 and Table 1 from Yuwei Wang, Xiangcheng Chen, Jiayuan Tao, Keyi Wang, and Mengchao Ma, "Accurate feature detection for out-of-focus camera calibration," Appl. Opt. 55, 7964-7971 (2016) and figure 1 from Jing Zhang, Bin Luo, Zhuolong Xiang, Qican Zhang, Yajun Wang, Xin Su, Jun Liu, Lu Li, and Wei Wang, "Deep-learning-based adaptive camera calibration for various defocusing degrees," Opt. Lett. 46, 5537-5540 (2021):

> Optica Publishing Group considers your requested use of its copyrighted material to be Fair Use under United States Copyright Law. We request that a complete citation of the original material be included in any publication.

While your publisher should be able to provide additional guidance, we prefer the below citation formats:

> For citations in figure captions:
>
> [Reprinted/Adapted] with permission from [ref #] © Optical Society of America [or Optica Publishing Group, as applicable]. (Please include the full citation in your reference list)
>
> For images without captions:
>
> Journal Vol. #, first page (year published) An example: Opt. Express 27, 1164 (2019)

Please let me know if you have any questions.

Kind Regards,

Cassie Travers

Kind Regards,

Cassie Travers

Cassie Travers
23 September 2024
Authorized Agent, Optica Publishing Group

Hello,

I am a graduate student at the University of Waterloo, and I am preparing my thesis for deposit in UWSpace, the university's institutional repository.

I understand that you are the copyright holder for the following articles:

- Accurate feature detection for out-of-focus camera calibration
  https://opg.optica.org/ao/abstract.cfm?uri=ao-55-28-7964
- Deep-learning-based adaptive camera calibration for various defocusing degrees
  https://opg.optica.org/ol/abstract.cfm?uri=ol-46-22-5537

I would like permission to include Figures 1, 2, and 12 as well as Table 1 from "Accurate feature detection for out-of-focus camera calibration", as well as Figure 1 from "Deep-learning-based adaptive camera calibration for various defocusing degrees".

All of these images and the table will be made openly available as part of my thesis, tentatively titled, *Camera Calibration from Out-of-Focus Images*. Proper citation will be included with the reproduction of this figure.

Please let me know what your process is for providing permission, and if you need any further information from me.

Thank you for considering this request,

Ryan Schmalenberg
MASc. Candidate
Vision and Image Processing Lab
Systems Design Engineering

University of Waterloo
Waterloo, Ontario

References

[1] Camera Calibration - MATLAB & Simulink. Available at https://www.mathworks.com/help/vision/camera-calibration.html.

[2] Eigenvalues, Eigenvectors and Generalized Schur Decomposition. Available at https://www.netlib.org/lapack/lug/node56.html.

[3] Friedrich Alexander Universitat Erlange Nurnberg Department of Computer Science Pattern Recognition Software - Camera Calibration with Active Targets. Available at http://www5.informatik.uni-erlangen.de/research/software/index.html.

[4] Image Processing Toolbox Documentation. Available at https://www.mathworks.com/help/images/index.html.

[5] OpenCV: Camera Calibration and 3D Reconstruction. Available at https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html.

[6] Polynomial eigenvalue problem - MATLAB polyeig. Available at https://www.mathworks.com/help/matlab/ref/polyeig.html.

[7] scipy.linalg.eig — SciPy v1.13.1 Manual. Available at https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.eig.html.

[8] Visual Servoing Platform: Tutorial: How to use Blender to generate simulated data for model-based tracking experiments. Available at https://visp-doc.inria.fr/doxygen/visp-3.4.0/tutorial-tracking-mb-generic-rgbd-Blender.html.

[9] Matrix representation of conic sections, September 2023. Page Version ID: 1176437094. Available at https://en.wikipedia.org/w/index.php?title=Matrix_representation_of_conic_sections&oldid=1176437094.

[10] Abdullah Abuolaim and Michael S. Brown. Defocus Deblurring Using Dual-Pixel Data. In *European Conference on Computer Vision (ECCV)*, online conference, August 2020. Springer.

[11] Masashi Baba, Masayuki Mukunoki, and Naoki Asada. A Unified Camera Calibration Using Geometry and Blur of Feature Points. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 816–819, Hong Kong, August 2006. ISSN: 1051-4651.

[12] Mark R. Banham and Aggelos K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2):24–41, 1997.

[13] Tyler Bell, Jing Xu, and Song Zhang. Method for out-of-focus camera calibration. *Applied Optics*, 55(9):2346–2352, 2016.

[14] Stuart Bennett and Joan Lasenby. ChESS - Quick and Robust Detection of Chessboard Features, January 2013. arXiv:1301.5491 [cs].

[15] Hauke Brunken and Clemens Gühmann. Deep learning self-calibration from planes. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 11433, pages 114333L–114333L–11. SPIE, 2020. ISSN: 0277-786X.

[16] Talha Hanif Butt and Murtaza Taj. Camera Calibration Through Camera Projection Loss. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2649–2653, Singapore, May 2022. ISSN: 2379-190X.

[17] Bolin Cai, Yuwei Wang, Keyi Wang, Mengchao Ma, and Xiangcheng Chen. Camera Calibration Robust to Defocus Using Phase-Shifting Patterns, 2017. Issue: 10 Publication Title: Sensors Volume: 17.

[18] Bolin Cai, Yuwei Wang, Jiajun Wu, Mengyu Wang, Fei Li, Mengchao Ma, Xiangcheng Chen, and Keyi Wang. An effective method for camera calibration in defocus scene with circular gratings. *Optics and Lasers in Engineering*, 114:44–49, 2019.

[19] L. Calvet, P. Gurdjos, C. Griwodz, and S. Gasparini. Detection and Accurate Localization of Circular Fiducials under Highly Challenging Conditions. In - *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 562–570, Las Vegas, NV, USA, June 2016.

[20] Mike Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(1):58–63, 1976.

[21] Marcela Carvalho, Bertrand Le Saux, Pauline Trouvé-Peloux, Andrés Almansa, and Frédéric Champagnat. Deep Depth from Defocus: how can defocus blur improve 3D estimation using dense neural networks?, September 2018. arXiv:1809.01567 [cs].

[22] Konstantin Chaika, Anton Filatov, Artyom Filatov, and Kirill Krinkin. Automatic Wheels and Camera Calibration for Monocular and Differential Mobile Robots, 2021. Issue: 13 Publication Title: Applied Sciences Volume: 11.

[23] Ben Chen, Caihua Xiong, and Qi Zhang. CCDN: Checkerboard corner detection network for robust camera calibration. In *International Conference on Intelligent Robotics and Applications*, pages 324–334, Newcastle, NSW, Australia, August 2018. Springer.

[24] Xiangcheng Chen, Xiaokai Song, Jun Wu, Yongxin Xiao, Yajun Wang, and Yuwei Wang. Camera calibration with global LBP-coded phase-shifting wedge grating arrays. *Optics and Lasers in Engineering*, 136:106314, 2021.

[25] Xinyu Chen, Ying Hu, Zi Ma, Shuanghe Yu, and Yuqing Chen. The location and identification of concentric circles in automatic camera calibration. *Optics & Laser Technology*, 54:185–190, 2013.

[26] Hyunseok Choi, Ho-Gun Ha, Hyunki Lee, and Jaesung Hong. Robust control point estimation with an out-of-focus camera calibration pattern. *Pattern Recognition Letters*, 143:1–7, 2021.

[27] Wikipedia contributors. Radon transform, June 2024. Page Version ID: 1230609747 Available at https://en.wikipedia.org/w/index.php?title=Radon_transform&oldid=1230609747.

[28] Xiaodong Cun and Chi-Man Pun. Defocus Blur Detection via Depth Distillation, July 2020. arXiv:2007.08113 [cs].

[29] Ankur Datta, Jun-Sik Kim, and Takeo Kanade. Accurate camera calibration using iterative refinement of control points. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1201–1208, Kyoto, Japan, 2009.

[30] Simon Donné, Jonas De Vylder, Bart Goossens, and Wilfried Philips. MATE: Machine learning for adaptive calibration template detection. *Sensors*, 16(11):1858, 2016.

[31] Alexander Duda and Udo Frese. Accurate Detection and Localization of Checkerboard Corners for Calibration. In *BMVC*, page 126, Newcastle upon Tyne, United Kingdom, 2018.

[32] Ahmed ElSheikh, Bassam A. Abu-Nabah, Mohammad O. Hamdan, and Gui-Yun Tian. Infrared Camera Geometric Calibration: A Review and a Precise Thermal Radiation Checkerboard Target. *Sensors*, 23(7), 2023.

[33] David Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Pearson, 2012.

[34] Peter Fuersattel, Sergiu Dotenco, Simon Placht, Michael Balda, Andreas Maier, and Christian Riess. OCPAD — Occluded checkerboard pattern detector. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, Lake Placid, NY, USA, 2016.

[35] Andreas Geiger, Frank Moosmann, Ömer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *2012 IEEE International Conference on Robotics and Automation*, pages 3936–3943, St Paul, MN, USA, 2012.

[36] Benyamin Ghojogh, Fakhri Karray, and Mark Crowley. Eigenvalue and Generalized Eigenvalue Problems: Tutorial, May 2023. arXiv:1903.11240 [cs, stat] version: 3.

[37] Nikhil Gupta, Hilda Faraji, Daan He, and Ghanshyam Rathi. Robust online estimation of the vanishing point for vehicle mounted cameras. In *2011 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, Beijing, China, September 2011. ISSN: 2378-928X.

[38] Hyowon Ha, Yunsu Bok, Kyungdon Joo, Jiyoung Jung, and In So Kweon. Accurate camera calibration robust to defocus using a smartphone. In *Proceedings of the IEEE International conference on computer vision*, pages 828–836, Santiago, Chile, 2015.

[39] Hyowon Ha, Michal Perdoch, Hatem Alismail, In So Kweon, and Yaser Sheikh. Deltille Grids for Geometric Camera Calibration. In - *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5354–5362, Venice, Italy, 2017.

[40] Haifei Huang, Hui Zhang, and Yiu-ming Cheung. The common self-polar triangle of concentric circles and its application to camera calibration. In *2015 IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 4065–4072, Boston, MA, USA, June 2015. IEEE.

[41] Fei Hao, Jinjiang Su, Jingjing Shi, Chaohan Zhu, Jiatong Song, and Yuntao Hu. Conic tangents based high precision extraction method of concentric circle centers and its application in camera parameters calibration. *Scientific Reports*, 11(1):20686, 2021.

[42] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge books online. Cambridge University Press, 2 edition, 2003.

[43] Yannick Hold-Geoffroy, Dominique Piche-Meunier, Kalyan Sunkavalli, Jean-Charles Bazin, Francois Rameau, and Jean-Francois Lalonde. A Perceptual Measure for Deep Single Image Camera and Lens Calibration. *IEEE transactions on pattern analysis and machine intelligence*, PP:1–12, 2023.

[44] Xiaowei Hu, Guijin Wang, Jinnan Wang, Pengfei Sun, Jingtao Fan, Feng Chen, and Yiyuan Xie. A Robust and Accurate Calibration Method for Out-of-focus Camera. *Electronic Imaging*, 2018:1–5, 2018.

[45] Lei Huang, Qican Zhang, and Anand Asundi. Camera calibration with active phase target: Improvement on feature detection and optimization. *Optics letters*, 38(9):1446–1448, 2013. Publisher: Optical Society of America.

[46] Junzhou Huo, Zhichao Meng, Haidong Zhang, Shangqi Chen, and Fan Yang. Feature points extraction of defocused images using deep learning for camera calibration. *Measurement*, 188:110563, 2022.

[47] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, December 2017.

[48] Neel Joshi. Defocus Blur. In Katsushi Ikeuchi, editor, *Computer Vision: A Reference Guide*, pages 171–173. Springer US, Boston, MA, 2014.

[49] Ali Karaali and Claudio Rosito Jung. Edge-Based Defocus Blur Estimation With Adaptive Scale Selection. *IEEE Transactions on Image Processing; IEEE Trans Image Process*, 27(3):1126–1137, 2018. ISBN: 1057-7149.

[50] Beomseok Kim, Hyeongseok Son, Seong-Jin Park, Sunghyun Cho, and Seungyong Lee. Defocus and Motion Blur Detection with Deep Contextual Features. *Computer Graphics Forum*, 37(7):277–288, 2018. ISBN: 0167-7055.

[51] Daniel. Kressner. *Numerical methods for general and structured eigenvalue problems*. Lecture notes in computational science and engineering, 46. Springer, Berlin, 2005.

[52] Mathieu Labussière. *Leveraging blur information in plenoptic cameras: Application to calibration and metric depth estimation*. Theses, Université Clermont Auvergne, December 2021.

[53] Jinwoo Lee, Hyunsung Go, Hyunjoon Lee, Sunghyun Cho, Minhyuk Sung, and Junho Kim. CTRL-C: Camera Calibration TRansformer With Line-Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16228–16237, Montreal, QC, Canada, October 2021.

[54] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative Filter Adaptive Network for Single Image Defocus Deblurring. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2034–2042, Los Alamitos, CA, USA, June 2021. IEEE Computer Society.

[55] Kang Liao, Lang Nie, Shujuan Huang, Chunyu Lin, Jing Zhang, Yao Zhao, Moncef Gabbouj, and Dacheng Tao. Deep Learning for Camera Calibration and Beyond: A Survey. *arXiv:2303.10559*, 2023.

[56] Yaroslava Lochman, Kostiantyn Liepieshov, Jianhui Chen, Michal Perdoch, Christopher Zach, and James Pritts. BabelCalib: A Universal Approach to Calibrating Central Cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15253–15262, Montreal, QC, Canada, October 2021.

[57] Manuel López, Roger Mari, Pau Gargallo, Yubin Kuang, Javier Gonzalez-Jimenez, and Gloria Haro. Deep Single Image Camera Calibration With Radial Distortion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2019-, pages 11809–11817, Long Beach, CA, USA, 2019. IEEE. ISSN: 1063-6919.

[58] Xueqin Lü, Lingzheng Meng, Liyuan Long, and Peisong Wang. Comprehensive improvement of camera calibration based on mutation particle swarm optimization. *Measurement*, 187:110303, 2022. ISBN: 0263-2241.

[59] John Mallon and Paul F. Whelan. Which pattern? Biasing aspects of planar calibration patterns and detection methods. *Pattern recognition letters*, 28(8):921–930, 2007. Place: Amsterdam Publisher: Elsevier B.V.

[60] Fahim Mannan and Michael S. Langer. Blur calibration for depth from defocus. In *2016 13th Conference on Computer and Robot Vision (CRV)*, pages 281–288, Victoria, BC, Canada, 2016. IEEE.

[61] Simon Placht, Peter Fürsattel, Etienne Assoumou Mengue, Hannes Hofmann, Christian Schaller, Michael Balda, and Elli Angelopoulou. ROCHADE: Robust Checkerboard Advanced Detection for Camera Calibration. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 766–779, Zurich, Switzerland, 2014. Springer International Publishing.

[62] Francois Rameau, Jinsun Park, Oleksandr Bailo, and In So Kweon. MC-Calib: A generic and robust calibration toolbox for multi-camera systems. *Computer Vision and Image Understanding*, 217:103353, 2022.

[63] Martin Rufli, Davide Scaramuzza, and Roland Siegwart. Automatic detection of checkerboards on blurred and distorted images. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3121–3126, Nice, France, 2008. IEEE.

[64] Karim Samaha, Georges Younes, Daniel Asmar, and John Zelek. Learned Intrinsic Auto-Calibration From Fundamental Matrices. In *2022 19th Conference on Robots and Vision (CRV)*, pages 224–232, Toronto, ON, Canada, May 2022.

[65] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In - *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, page 45, New York, New York, USA, 2006.

[66] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A Toolbox for Easily Calibrating Omnidirectional Cameras. In - *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701, Beijing, China, 2006.

[67] Ryan Schmalenberg and John Zelek. Out-of-Focus Camera Calibration. *Journal of Computational Vision and Imaging Systems*, 9(1):50–53, April 2024.

[68] Christoph Schmalz, Frank Forster, and Elli Angelopoulou. Camera Calibration: Active versus Passive Targets. *Optical Engineering*, 50, November 2011.

[69] Johannes Schneider and Michalis Vlachos. A Survey of Deep Learning: From Activations to Transformers, February 2023. arXiv:2302.00722 [cs].

[70] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. MIT Press, 2012.

[71] Murali Subbarao, Tse-Chung Wei, and Gopal Surya. Focused image recovery from two defocused images recorded with different camera settings. *IEEE Transactions on Image Processing*, 4(12):1613–1628, 1995.

[72] Huixuan Tang. *Modeling and Analysis of Optical Blur for Everyday Photography*. Thesis, University of Toronto, June 2018. Accepted: 2018-07-18T19:02:41Z.

[73] Masahiro Toyoura, Haruhito Aruga, Matthew Turk, and Xiaoyang Mao. Detecting Markers in Blurred and Defocused Images. In - *2013 International Conference on Cyberworlds*, pages 183–190, Yokohama, Japan, 2013.

[74] Masahiro Toyoura, Haruhito Aruga, Matthew Turk, and Xiaoyang Mao. Mono-spectrum marker: an AR marker robust to image blur and defocus. *The Visual Computer*, 30(9):1035–1044, 2014.

[75] Roger Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE journal of robotics and automation*, 3(4):323–344, 1987.

[76] Vladimir Vezhnevets. Graphics & Media lab The Laboratory of Computer Graphics at Dep. of CS MSU: OpenCV calibration object detection, 2005. Available at https://www.graphicon.ru/oldgr/en/research/calibration/opencv.html.

[77] Yuwei Wang, Bolin Cai, Keyi Wang, and Xiangcheng Chen. Out-of-focus color camera calibration with one normal-sized color-coded pattern. *Optics and Lasers in Engineering*, 98:17–22, 2017.

[78] Yuwei Wang, Xiangcheng Chen, Jiayuan Tao, Keyi Wang, and Mengchao Ma. Accurate feature detection for out-of-focus camera calibration. *Applied Optics*, 55(28):7964–7971, 2016.

[79] Yuwei Wang, Yajun Wang, Lu Liu, and Xiangcheng Chen. Defocused camera calibration with a conventional periodic target based on Fourier transform. *Optics letters*, 44(13):3254–3257, 2019. Publisher: Optical Society of America.

[80] Carl J. Wenning. Student Lab Handbook (Percent Difference and Percent Error). Available at http://www.phy.ilstu.edu/slh/Percent%20Difference%20Error.pdf.

[81] Marta Wilczkowiak, Peter Sturm, and Edmond Boyer. Using geometric constraints through parallelepipeds for calibration and 3D modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence; IEEE Trans Pattern Anal Mach Intell*, 27(2):194–207, 2005.

[82] Scott Workman, Connor Greenwell, Menghua Zhai, Ryan Baltenberger, and Nathan Jacobs. DEEPFOCAL: A method for direct focal length estimation. In *2015 IEEE International Conference on Image Processing (ICIP)*, volume 2015-, pages 1369–1373, Quebec City, QC, Canada, 2015. IEEE. ISSN: 1522-4880.

[83] Shourui Yang, Miao Liu, Shibin Yin, Yin Guo, Yongjie Ren, and Jigui Zhu. An improved method for location of concentric circles in vision measurement. *Measurement*, 100:243–251, 2017.

[84] B. W. Zhang, Y. F. Li, and S. Y. Chen. Concentric-circle-based camera calibration. *IET image processing*, 6(7):870–876, 2012. Publisher: Institution of Engineering and Technology.

[85] Jing Zhang, Bin Luo, Zhuolong Xiang, Qican Zhang, Yajun Wang, Xin Su, Jun Liu, Lu Li, and Wei Wang. Deep-learning-based adaptive camera calibration for various defocusing degrees. *Opt. Lett.*, 46(22):5537–5540, November 2021. Publisher: Optica Publishing Group.

[86] Quanbing Zhang and Yanyan Gong. A Novel Technique of Image-Based Camera Calibration in Depth-from-Defocus. In - *2008 First International Conference on Intelligent Networks and Intelligent Systems*, pages 483–486, Wuhan, China, 2008.

[87] Zhengyou Zhang. A flexible new technique for camera calibration, 2000. ISBN: 1939-3539 Issue: 11 Pages: 1330-1334 Publication Title: - IEEE Transactions on Pattern Analysis and Machine Intelligence Volume: 22.

[88] Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence; IEEE Trans Pattern Anal Mach Intell*, 26(7):892–899, 2004.

[89] Zhengyou Zhang. Weak Perspective Projection. In Katsushi Ikeuchi, editor, *Computer Vision: A Reference Guide*, pages 877–883. Springer US, Boston, MA, 2014.

[90] Anmin Zou, Zengguang Hou, Lejie Zhang, and Min Tan. A Neural Network-Based Camera Calibration Method for Mobile Robot Localization Problems. In *ADVANCES IN NEURAL NETWORKS - ISNN 2005, PT 3, PROCEEDINGS*, volume 3498 of

Lecture Notes in Computer Science, pages 277–284, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISSN: 0302-9743 Issue: III.

APPENDICES

Appendix A

Blender Focal Length Recovery and Lighting

A.1 Example Calculation of Focal Lengths from Blender Perspective Camera Model

The following steps outline a basic example of the calculations for recovering the ground truth focal lengths for synthetically rendered images from Blender scenes. Figure A.1 shows the perspective camera model and depth of field (DoF) settings.

To find f_x, f_y from K:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

we can follow:

$$\begin{aligned} f_x &= f_{mm} s_x \\ f_y &= f_{mm} s_y \end{aligned} \tag{A.2}$$

where s_x, s_y are the number of pixels per mm of the sensor. Given an image horizontal resolution of 1920px and a sensor width of 10.9mm:

$\therefore s_x = 1920/10.9 = 176.146 px/mm$

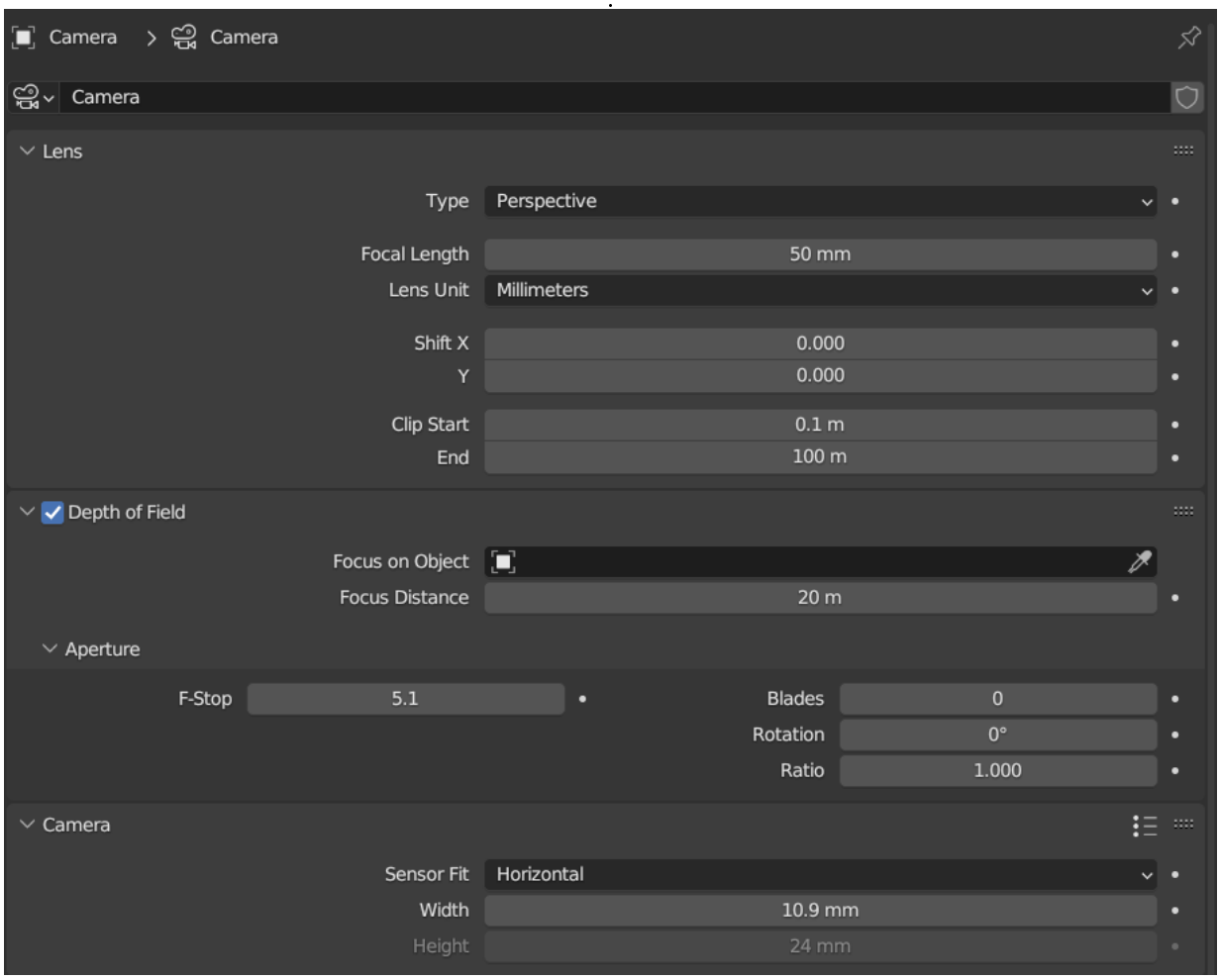
In this case, $s_x = s_y$

With a camera focal length in Blender specified as 50mm, based on (A.2) we can then multiply s_x & s_y by 50mm.

$\therefore f_x = 176.146 \times 50 = \mathbf{8807.339px}$

NOTE: $f_x = f_y$

Figure A.1. shows the camera settings panel in Blender where the focal length is specified in mm, and the focus distance can be set with depth of field turned on. Note that a 10.9mm sensor width was specified but the height of 24mm has no effect on the camera. With sensor fit set to "Horizontal", the actual sensor height is calculated independently based on the render aspect ratio and specified sensor width. More information about Blender's Depth of Field can be found here.



A.2 Synthetic Approximation of Lighting Conditions

Below are images of the lighting conditions in the real A.2 and synthetic experiment A.3 using Blender version 3.4.

Figure A.2. shows the lighting setup in the RoboHub lab from the real-world experimental setup where six spotlights were illuminated along with area lighting above the robot guided calibration target.



Figure A.3. shows the Blender scene for the final synthetic experiment with spotlights and area lighting approximating the real-world lab setup. Note that the orange object is the camera bounding box with camera FOV.

