GENERIC CAMERA CALIBRATION FOR OMNIFOCUS IMAGING, DEPTH ESTIMATION
AND A TRAIN MONITORING SYSTEM

BY

AVINASH KUMAR

DISSERTATION
Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:
Professor Emeritus Narendra Ahuja, Chair
Professor Thomas S. Huang
Professor Stephen E. Levinson
Professor Minh N. Do
ABSTRACT

Calibrating an imaging system for its geometric properties is an important step toward understanding the process of image formation and devising techniques to invert this process to decipher interesting properties of the imaged scene. In this dissertation, we propose new optically and physically motivated models for achieving state-of-the-art geometric and photometric camera calibration. The calibration parameters are then applied as input to new algorithms in omnifocus imaging, 3D scene depth from focus and machine vision based intermodal freight train analysis.

In the first part of this dissertation, we present new progress made in the areas of camera calibration with application to omnifocus imaging and 3D scene depth from focus and point spread function calibration. In camera calibration, we propose five new calibration methods for cameras whose imaging model can represented by ideal perspective projection with small distortions due to lens shape (radial distortion) or misaligned lens-sensor configuration (decentering). In the first calibration method, we generalize pupil-centric imaging model to handle arbitrarily rotated lens-sensor configuration, where we consider the sensor tilt to be about the physical optic axis. For such a setting, we derive an analytical solution to linear camera calibration based on collinearity constraint relating the known world points and measured image points assuming no radial distortion. Our second method considers a much simpler case of Gaussian thin-lens imaging model along with non-frontal image sensor and proposes analytical solution to the linear calibration equations derived from collinearity constraint. In the third method, we generalize radial alignment constraint to non-frontal sensor configuration and derive analytical solution to the resulting linear camera calibration equations. In the fourth method, we propose the use of focal stack images of a known checkerboard scene to calibrate cameras having non-frontal sensor. In the fifth method, we show that radial distortion is a result of changing entrance pupil location as a function of incident image rays and propose a collinearity based camera calibration method under this imaging model. Based on this model, we propose a new focus measure for omnifocus imaging and apply it to compute 3D scene depth from focus. We then propose a point spread function calibration method which computes the point spread function (PSF) of a CMOS image
sensor using Hadamard patterns displayed on an LCD screen placed at a fixed distance from
the sensor.

In the second part of the dissertation, we describe a machine vision based train monitoring
system, where we propose a motion-based background subtraction method to remove back-
ground between the gaps of an inter-modal freight train. The background subtracted image
frames are used to compute a panoramic mosaic of the train and compute gap length in pix-
els. The gap length computed in metric units using the calibration parameters of the video
camera allows for analyzing the fuel efficiency of loading pattern of the given inter-modal
freight train.
Dedicated to the memory of my beloved sister Sweety
To my parents Uday and Rekha for their blessings
To my wife Priti for her love and support
To my beautiful daughter Avishka
I would like to thank my advisor Professor Narendra Ahuja for his continuous guidance and support all through my PhD research. Without his patient and encouraging words this dissertation would not have been possible.

I would like to thank Professor Thomas S. Huang, Professor Steven E. Levinson and Professor Minh N. Do for being in my committee and giving valuable suggestions.

I would like to thank John M. Hart whom I have known for almost nine years now, for always having a smile on his face. I really admire his resilient work ethics when we traveled to far off locations for our railway project in winter and summer and he would never give up till the planned task for that day was finished.

I would also like to thank all the past and present members of my lab, Computer Vision and Robotics Lab, with whom I have interacted through these years including Xianbiao Shu, Sanketh Shetty, Emre Akbas, Bernard Ghanem, Tim Cheng, Esther Resendiz, Myra Nam, Qingxiong Yang, Abhishek Singh, Shengnan Wang and Huiguang Yang. I would specially like to thank Jia-Bin Huang for his critical reviews in improving of my papers before submission deadlines. They all made our lab a great place to work.

I am also thankful for having Jayash Koshal, Vijitashwa Pandey, Hitesh Bindra, Sourabh Bhattacharya, Akash Kushal, Himanshu Arora, Debashish Das in addition to many others as my friends who made my stay pleasant and helping me in the thick and thin of my stay here in Urbana-Champaign.

I would not have embarked on this long journey, if not for my undergraduate advisor Prof. C. V. Jawahar. He really got me interested in the field of computer vision and motivated me to go for a PhD. I would also like to thank my undergraduate friends Arjun Kumar, Vardhman Jain, Sachin Rawat, Uday K. Visesh, Vishal Dwivedi, Jagmohan Singh who helped me during my PhD years in many different ways.

I would like to thank my wife Priti for bearing through my ups and downs over this long period of time, being patient and always encouraging me in difficult times. She has been my pillar of strength. I would also like to thank my new-born daughter Avishka for bringing a special joy in my life.
Last but not the least, this milestone would never have come to fruition without the unwavering love, support and blessings of my parents, Mr. Uday K. Singh and Mrs. Rekha Singh and my beloved sister Archana Kumari. Today would not have been possible without their vision and many sacrifices.
# Table of Contents

LIST OF TABLES ................................................................. x

LIST OF FIGURES ............................................................... xi

CHAPTER 1  INTRODUCTION .................................................... 1
  1.1  Generic Camera Calibration and Applications ........................... 1
  1.2  Train Monitoring System ................................................. 3

I  Generic Camera Calibration and Applications .......................... 4

CHAPTER 2  GENERALIZED PUPIL-CENTRIC IMAGING AND LINEAR
  NON-FRONTAL CAMERA CALIBRATION ..................................... 5
  2.1  Introduction ............................................................. 5
  2.2  Related Work .......................................................... 8
  2.3  Pupil-Centric and Thin-Lens Imaging ................................ 9
  2.4  Example Pupil-Centric Lens Model ................................. 18
  2.5  Non-Frontal Projection Equations ................................... 20
  2.6  Collinearity Based Linear Calibration Equation .................... 22
  2.7  Analytical Calibration ................................................ 25
  2.8  Finding the Center of Radial Distortion ............................ 34
  2.9  Nonlinear Refinement ................................................. 36
  2.10  Experiments with Synthetic Data ................................... 36
  2.11  Experiments with Real Images ...................................... 38
  2.12  Discussion ............................................................ 42

CHAPTER 3  GAUSSIAN IMAGING: LINEAR NON-FRONTAL SENSOR
  CALIBRATION ..................................................................... 44
  3.1  Introduction ............................................................. 44
  3.2  Previous Work .......................................................... 44
  3.3  Modeling Non-Frontal Camera Calibration ............................ 46
  3.4  Linear Camera Calibration Equation ................................ 54
  3.5  Determining Calibration Parameters ................................ 56
  3.6  Image Distortion ....................................................... 66
  3.7  Nonlinear Minimization ............................................... 67
II  Train Monitoring System  163

CHAPTER 9  MOTION-BASED BACKGROUND SUBTRACTION AND PANORAMIC MOSAICING FOR FREIGHT TRAIN ANALYSIS  .  164
  9.1  Introduction  .  164
  9.2  Previous Work  .  167
  9.3  Motion-Based Background Subtraction  .  168
  9.4  Panoramic Mosaic Generation  .  171
  9.5  Gap Detection  .  173
  9.6  Load Classification  .  173
  9.7  Gap Length Estimation  .  174
  9.8  Results  .  176
  9.9  Discussion  .  177

CHAPTER 10  CONCLUSION AND FUTURE WORK  .  179

REFERENCES  .  180
LIST OF TABLES

2.1 Cinegon 1.4/8 mm lens parameters ................................. 18
2.2 Calibration results on real data ....................................... 39
2.3 Standard deviation of final estimates from methods A, B, C, D .... 39

3.1 Ground truth calibration parameters for synthetic experiments .... 68
3.2 $\sigma = 0$ ................................................................. 68
3.3 $\sigma = 0.01$ ............................................................. 69
3.4 $\sigma = 0.1$ ............................................................... 70
3.5 $\sigma = 0.5$ ............................................................... 70
3.6 $\sigma = 1.0$ ............................................................... 71

4.1 Calibration estimates using the two techniques ........................ 89

5.1 Estimated calibration parameters and their standard deviation for
one image ................................................................. 103
5.2 Estimated calibration parameters and their standard deviation for
five images ............................................................... 104

6.1 Calibration results on real data (TL = Thin-lens, PC = Pupil-centric,
ME = Moving entrance, NF = Non-frontal, F = Frontal) ............... 121
6.2 Standard deviation of calibration parameters shown in Table 6.1 .... 122
LIST OF FIGURES

2.1 Gaussian model of image formation. ........................................ 10
2.2 Pupil-Centric model of image formation. ................................. 12
2.3 Lens data-sheet values for Cinegon 1.4/8 mm lens. ....................... 19
2.4 Coordinate systems for non-frontal camera calibration. ............... 20
2.5 Hyperbolic relationship of \((\alpha, F)\) given (a) \(\kappa < 0\) (b) \(\kappa > 0\). .................................................. 33
2.6 Left: blue dot \((239.3, 330.8)\) is analytical estimate mean \((P^*_p)\), green dot \((236.3, 329.3)\) is laser calibrated and magenta dot \((239.3, 330.6)\) is nonlinear result initialized by the laser center. It is close to analytical estimate. Right: real calibration images (best viewed in color). ....................... 35
2.7 Absolute error, standard deviation vs. noise level (pixels) for various calibration parameters in \(U\) (best viewed in color). ................................. 37
2.8 Distribution of mean centered \(667 \times 10\) detected points. ............ 40
2.9 (a) Straight line fitting error to undistorted images using calibration parameters from technique A, B, C, D. (b, c, d) A non-frontal camera and the calibration setup (best viewed in color). ........................................ 41
2.10 (a) Omnifocus image \([5]\). (b) Depth from focus \([10]\). (c, d, e) 3D textured renderings of scene (best viewed in color). ................................. 42
3.1 Coordinate systems for non-frontal camera calibration. .............. 45
3.2 Number of points used for linear calibration (a) all points (b) central points. ................................................................. 71
4.1 Coordinate systems for camera calibration. ............................... 74
4.2 (a) Imaging model for RAC \([14]\). (b) An illustration of RAC not holding true in real images, when the sensor maybe non-frontal with respect to the lens plane. ................................................................. 75
4.3 Illustration of generalized radial alignment constraint (gRAC). ........ 76
4.4 (Left) Ground truth image formation. (Right) (a) Solution A: \((t_x, R_1, \lambda_1, t_{z1})\). (b) Solution B: \((t_x, R_2, \lambda_2, t_{z2})\). (c) Solution C: \((-t_x, R_2, -\lambda_1, t_{z1})\). (d) Solution D: \((-t_x, R_1, -\lambda_2, t_{z2})\). Solutions C and D can be rejected based on \(\lambda\) being negative. The better solution between A and B is selected by analyzing radial distortion model fitting error. ....................... 84
4.5 Relative error vs. noise (in pixels) using gRAC on synthetic data. .... 88
5.1 Defocus image formation. ....................................................... 92
5.2 Focal stack acquisition by (a) moving sensor along the optic axis and (b) rotating a camera with tilted sensor about the optic center [10]. On comparison of the imaging geometry in both cases we observe that the images formed on sensor location 1, 2 and 3 are similar. Thus, both systems can be used interchangeably to generate similar focal stack images. In this chapter, we use the configuration (b).

5.3 Geometric image formation.

5.4 Input: (top row) input focal stack images, (middle row) corresponding pinhole focal stack used for computing registration parameters to warp input focal stack (see Section 5.3.1). Output: (bottom row) Synthetic omnifocus focal stack obtained after omnifocus imaging (Section 5.3.2) and dewarping (Section 5.3.3). The camera was rotated from left to right. The inset shows details of intensity distribution at one single image location across image sequences in each row.

5.5 (a) Input focal stack images of top row of Figure 5.4 are registered to a global coordinate system (see Section 5.3.1). A fixed pixel location is selected in each of the four images with inset showing the zoomed in intensity inside. Due to registration, they are aligned. (b) Omnifocus image obtained from the registered focal stack on the left (see Section 5.3.2).

5.6 (a) Blurred focal stack image and (b) blur circle estimates from optimal $U^*$ using Equation 5.2 (inset shows details). The blurring of the CB corner looks consistent with the radius of the estimated Gaussian blur kernel (best seen in color).

6.1 Example of entrance pupil, exit pupil and aperture stop [42].

6.2 (a) No-distortion: aperture stop coincides with the principal point $O$. (b, c) Barrel and pin-cushion distortion due to change in stop location.

6.3 Pupil-centric imaging model for thick lenses.

6.4 Angle of incident light ray moves the entrance pupil leading to different amounts of radial distortion: (a) small incident angle (b) large incident angle.

6.5 Geometry of proposed calibration model.

6.6 Moving entrance pupil vs. angle of incidence ray. The plot is based on $(\epsilon_1 = -5.304, \epsilon_2 = 6.474)$ computed from calibration results from our proposed method as shown in the last column of Table 6.1.

6.7 Various calibration models which have been used in literature. (a) Category 1/Classical method: thin-lens imaging, frontal sensor, radial and de-centering distortion model. (b) Category 2: thin-lens imaging, moving entrance pupil, non-frontal sensor, radial distortion model. (c) Category 3: pupil-centric imaging, fixed entrance pupil, non-frontal sensor radial distortion. (d) Proposed calibration model: pupil-centric imaging, moving entrance pupil, non-frontal sensor.

6.8 Lens data-sheet values for Cinegon 1.4/8 mm lens.

6.9 Depth dependent undistortion (best viewed in color).

7.1 Hadamard LCD display and capture.
7.2 (a) LCD refresh rate. (b) Lens-less CMOS image with stripes. (c) At optimal integration time of the CMOS sensor, the captured image does not have any stripes. ................................................................. 130

7.3 Gradient vs. coarse integration time at (a) brightness 50 and (b) brightness 35. The minimum occur at time periods which are multiple of LCD refresh rate frequency. ................................................................. 131

7.4 (Left) Linear relationship between number of photons and the mean raw intensity value at optimal settings. (Right) Experimental setup. ................. 133

7.5 Anisotropic map for 640 × 640 LCD pixels. ........................................... 134

7.6 Experimental setup for calibrating the point spread function. .................. 135

7.7 Computed PSF for a camera with a lens. ................................................. 135

8.1 (a-f) Focal stack images. The goal is to design a focus measure which detects the best focused image (image (c)). .................................................. 138

8.2 Synthetic example: (a) focused image consisting of an intensity edge. (b) Zoomed out small window near intensity edge has no gradients. (c) Artificially blurred image of (a). (d) Zoomed out patch (intensity scaled between [0 255] for better visualization) located at the same location as in (a). This window contains significant gradients causing errors in gradient based focus measure. .................................................. 139

8.3 (a) Focused (b) Blurred with isotropic Gaussian blur of σ = 2.47. (c-d) Red indicates pixel locations where focus measures based on (c) variance [50], (d) energy of gradient [45] (in a 5 × 5 window) is higher for image (b) compared to the corresponding location in image (a). These regions lie near intensity edges. .................................................. 139

8.4 Multifocus imaging geometry. Focused (middle) and defocused (left, right). 140

8.5 Plot of focus measure values vs focal stack image frame index for pillbox (P) and Gaussian point spread function (PSF) (G). Since, the slope of G is more than P, the focus measure curve corresponding to Gaussian PSF finds the optimal focus frame index '4' more unambiguously as compared to the one using pillbox PSF. .................................................. 144

8.6 The red rays correspond to the formation of a set of focal stack images on sensor plane located along the x-axis, with the focused image being formed at x_f. The blue rays correspond to hypothesized set of focal stack images assuming x_k is focused and blurring the observed image at x_k. But, this blurring in turn generates the green set of rays, as the image at x_k was already defocused. .................................................. 146

8.7 Depth of field geometry for a tilted image sensor with minimum and maximum radial depth ranges required to be captured in focus. ................. 149

8.8 Two consecutive sensor rotations with θ_{opt}. ....................................... 151
8.9 Sensor tilt is 4.2 degrees. (a) Radial distance $d_{\text{max}} = 2$ ft, optimal sensor rotation $\theta_{\text{opt}} = 2.5981$ degrees compactly packs consecutive sharply focused areas. (b) Detailed view of the compact packing. (c) 47 images are required to capture a field of view of 120 degrees bounded by optical axes for the first and last frame.

8.10 Simulated optimal rotation angles for different depth ranges. (a) $d_{\text{min}} = 1$ ft, $d_{\text{max}} = 3$ ft, $\theta_{\text{opt}} = 2.565$ degrees; (b) $d_{\text{min}} = 2$ ft, $d_{\text{max}} = 3$ ft, $\theta_{\text{opt}} = 2.565$ degrees; (c) $d_{\text{min}} = 1$ ft, $d_{\text{max}} = 5$ ft, $\theta_{\text{opt}} = 2.539$ degrees; (d) $d_{\text{min}} = 3$ ft, $d_{\text{max}} = 5$ ft, $\theta_{\text{opt}} = 2.539$ degrees.

8.11 (a) $d_{\text{min}} = 1$ ft, $d_{\text{max}} = 9$ ft, $\theta_{\text{opt}} = 2.5219$ degrees computed from our method results in (b) gaps in SFA for consecutive frames. In order to cover these gaps, $\theta_{\text{opt}}$ needs to be reduced, resulting in acquisition of larger number of image frames to cover a fixed field of view.

8.12 Omnidfocus image of a planar scene with comparison on focus frame selection using generative and gradient based techniques. The artifacts due to gradient methods are indicated in zoomed out windows in black arrow, where (a) energy of gradient [45], (b) gray level distribution [46], (c) sum modified Laplacian [44] and (d) variance.

8.13 (left) Computed omnifocus image of a textured scene. (middle) Comparison of applying focus measure using generative and gradient based technique. The artifacts due to gradient based techniques is clearly visible (yellow arrow). (right) Estimated coarse and refined (using volumetric graph cuts) depth map of the scene (blue indicates close and red indicates far; see colorbar on left).

8.14 Normalized focus measure around an intensity edge using various techniques. Only the proposed technique and [46] succeed in finding the best focused frame 3 (encircled with solid lines). All focus measures were calculated in a $7 \times 7$ window around a pixel.

8.15 Normalized focus measure for various focus measure in a window containing sharp intensity edge. All the measures peak for the correct focused frame.

8.16 The leftmost image is the region whose ground truth focus frame index is known. The other images represent the absolute difference between the focus frame index by different focus measures and the ground truth focus frame index. The numbers in the bottom represent the percentage of erroneous predictions. The proposed generative focus measure has the best performance.

8.17 Teddy dataset: (top left) shows an image of the objects in the scene, the camera location and a scale to empirically measure the distance of various objects in the scene; (top middle) shows the omnifocus image of the teddy bear and other objects behind the bear; (top right) shows the medial filtered ($11 \times 11$) 2D depth estimate of the scene; (bottom row) shows two 3D views of the teddy bear.
8.18 Rocks dataset: (left) shows the setting of various geometric objects in the scene. The objects consist of a flat planar board, a prism object and more generic shaped objects in the form of rocks; (middle top) shows the omnifocus image of the scene; (right top) shows the medial filtered (11 $\times$ 11) 2D depth map; (middle and right bottom) shows two 3D views of this dataset.  

8.19 Cylinder dataset: (left) shows a cylindrical container and a flat board as the geometric objects in the scene; (middle top) shows the omnifocus image; (top right) shows the median filtered (11 $\times$ 11) 2D depth map of the cylinder; (middle and right bottom) shows two 3D view of the cylinder.

9.1 Different load types: (a) rail car, (b-f) different kinds of loads, including (b) double stack with upper and lower stacks of same length, (c, d) double stack with upper and lower stack of different lengths, (e) single stack and (f) trailer.

9.2 (a) Consecutive image frames of an intermodal train video with moving trees visible in the background. (b) Background removed using our technique. (c) Orthographic mosaic of the train.

9.3 Validating the hypothesized velocity $v$ at two image patches: red (dashed boundary) belonging to foreground and orange (solid boundary) belonging to background.

9.4 (a) Input image. (b) foreground cost $C_p$. (c) Extracted foreground.

9.5 Panoramic mosaic of an intermodal freight train with (a) trailers (b) double stack containers.

9.6 Detection of double stack containers.

9.7 Different types of gaps: (a) different length double stack and single stack pair and a pair of same length double stacks, (b) same length double stack and trailer pair.

9.8 Histogram of gap type distribution for a given intermodal freight train.

9.9 (a) Background subtraction mask on different illumination conditions. The background mask in proposed technique is missing at the boundaries as those regions were either not present or are lost in previous and next image frames respectively. (b) Orthographically projected panoramic mosaics. They can be concatenated from left to right to obtain the complete mosaic.

9.10 Histogram of different types of erroneous and accurate gap detection estimates after background removal.
In this dissertation, we focus on two different problems. The first part of the dissertation focuses on the problem of generic camera calibration and its application to novel omnifocus imaging and 3D scene depth estimation algorithms. The second part of the dissertation presents a train monitoring machine vision system which analyzes a real-world intermodal freight train to compute the gaps between the loads of the train. The novel contribution of this part of the dissertation is a new motion background subtraction method to remove background between the gaps of the loads.

1.1 Generic Camera Calibration and Applications

Camera calibration is one of the relevant and well-researched areas in the field of computer vision over the past few decades. Its importance stems from the fact that a calibrated camera can be used for various computer vision tasks like metric 3D reconstruction from cues like stereo and focus, image undistortion, camera pose estimation etc. A camera calibration approach is typically divided into two stages. In the first stage, which is called linear calibration, an imaging constraint, e.g. collinearity of a corresponding known scene and measured image point, is used to obtain an initial estimate of the calibration parameters. In the second stage, a nonlinear optimization of these parameters along with image distortion parameters is done. The initialization of this optimization is done from the first stage calibration. Most of the previous calibration works have focused on devising new imaging constraints and linearly calibrating them, while assuming standard thin-lens/Gaussian imaging conditions. Also, the image distortion model for radial and decentering distortion has not been physically motivated.

In this dissertation, we show that a more explicit modeling of image formation based on camera optics can lead to improvements in parameter estimation accuracy. Based on the new model, we propose four different camera calibration techniques which employ different calibration imaging constraints and the input calibration data and show that each of these
methods lowers the mean pixel re-projection error obtained by the current state-of-the-art
camera calibration techniques. We apply our calibration method to focal stack image com-
posing for omnifocus imaging and 3D scene depth estimation. In addition to geometric
calibration, we also present a technique to calibrate the point spread function of a typical
optical imaging system. We also propose new omnifocus imaging and 3D scene depth esti-
mation algorithm based on these calibration methods. The relevant chapters are organized
in the dissertation as follows:

• Chapter 2: In this chapter, we propose a generalized pupil-centric imaging model of
image formation which differs from the traditional thin-lens based imaging model. We
show that along with a non-frontal imaging sensor geometry, this model gives lower
mean pixel re-projection error than state-of-the-art techniques. Specifically, we use
collinearity based imaging constraint to derive the linear calibration equations and
then propose an analytical solution to solve these calibration equations. This work has
partially been presented in [1].

• Chapter 3: In this chapter, we consider the special case of Gaussian imaging and a non-
frontal imaging sensor for doing camera calibration. We derive the linear calibration
equation and then propose an analytical solution to these equations.

• Chapter 4: In this chapter, we generalize the pupil-centric imaging model along with
non-frontal sensor model to the radial alignment based imaging constraint. We pro-
pose and formulate the generalized radial imaging constraint and derive an analytical
solution to the problem of linear calibration in this setting. This work has partially
been presented in [2].

• Chapter 5: In this chapter, we use focal stack images of a checkerboard under pupil-
centric imaging and non-frontal sensor model to do camera calibration. The focal stack
images are captured from a camera with intentionally tilted sensor. The intentional
tilt of the sensor encodes a unique defocus distribution on the captured checkerboard
images. We use this focal stack to first do omnifocus imaging and then synthesize
sharply focused checkerboard images. Then, we minimize both the geometric as well
as the image blur error over the calibration parameters. This work has partially been
presented in [3].

• Chapter 6: In this chapter, we investigate the geometric/optical understanding of the
occurrence of radial distortion. While all prior methods employ an infinite series model
of radial distortion, we reason the use of this model from a physical point of view. We
show that radial distortion happens due to distortions in the shape of the imaging lens and that these distortions can be easily captured by assuming that the entrance pupil location is a function of the incoming ray directions. We thus show that unlike prior state-of-the-art camera calibration methods [4], which model entrance pupil location and radial distortion together, modeling only the entrance pupil locations suffices and gives lower mean pixel re-projection error.

- Chapter 7: In this chapter, we present a method to calibrate the point spread function of an imaging system, kept at a fixed distance from an LCD screen, using Hadamard image patterns displayed on the screen. We present calibrating the light distribution of the LCD screen and a radiometric calibration of the image sensor.

- Chapter 8: In this chapter, we present a new omnifocus imaging algorithm which differs from standard gradient based methods. We also present a method to compute the optimal number of focal stack images to capture, such that a given radial range of distances in-front of the camera can be captured in focus in at least one of the images. We also give results of a depth from focus algorithm which employs the omnifocus image and the calibration results to compute depth. This work has partially been presented in [5].

1.2 Train Monitoring System

We have written a machine vision based train monitoring system whose goal is to capture the video of an intermodal freight train, remove the background, classify the container types and then compute the length of gaps between the loads. The difficult problem we faced here was background subtraction across different times of the year. Such a setting required a background subtraction method which is robust to the scene illumination and background motion of trees and clouds in the sky.

- Chapter 9: In this chapter, we present a motion based background removal method. Specifically, we hypothesize a train speed in pixel shifts per frame using the motion of the wheels of the train. Then we compute the validity of the pixel motion for small image patches in the current image frame given the hypothesized velocity. Finally, we define a cost function which would maximize to 1 for foreground image patches and be less than 0 for the background. This work has partially been presented in [6, 7, 8, 9].
Part I

Generic Camera Calibration and Applications
CHAPTER 2

GENERALIZED PUPIL-CENTRIC IMAGING AND LINEAR NON-FRONTAL CAMERA CALIBRATION

2.1 Introduction

Camera calibration estimates intrinsic (physical) and extrinsic (pose) parameters of a camera with respect to a known world coordinate system. In a camera, occasionally, due to manufacturing limitations, lens and sensor plane may be slightly tilted with respect to each other. In other cases, a significant amount of tilt may be desired for unconventional imaging, e.g. obtaining a perfectly focused image of a plane not perpendicular to the direction of viewing, controlling orientation of camera’s depth of field profile (Scheimpflug principle), computing an omnifocus image [5] or estimating scene depth using depth from focus [10]. Such a camera with a tilted sensor is called non-frontal [10], as opposed to an ideal frontal camera whose lens and sensor planes are parallel. Typically all lens-sensor configurations can be considered as non-frontal with frontal being a special case. Conventionally, there are two main approaches to model non-frontal sensor:

1. Implicit: The “effect” of sensor tilt on an image is modeled as decentering distortion [11] about an effective axis normal to the sensor and passing through the camera’s center of projection. As sensor and lens planes are not parallel, this axis is different from the optic axis, about which only radial distortion exists. It has been analytically shown that decentering modeling is approximate and only holds for small amounts of sensor tilt. In fact, for small tilts, a standalone radial distortion model about optic axis is approximately equivalent to a combination of decentering and radial distortion model about the effective axis [12]. Most conventional calibration methods [13, 14, 15, 16] follow this model. We denote all calibration parameters except decentering parameters in this model as the set $U$.

2. Explicit: It is assumed that the lens optic axis coincides with the z-axis of the lens coordinate system and the sensor coordinate system has its origin at the intersection of optic axis and sensor plane, defined as the center of radial distortion (CoD) henceforth.
The sensor non-frontalness is explicitly modeled by a $3 \times 3$ rotation matrix $R$ [4]. Although, this results in an increased number of calibration parameters as $(U, R)$, it is a physically meaningful model of lens-sensor configuration and can be used for sensor tilt estimation via camera calibration. Such a tilt estimate can be used for depth estimation [10] and omnifocus imaging [5]. Thus, in this chapter, we follow this model for non-frontal camera calibration.

Next, we discuss various aspects of non-frontal calibration which are improved upon in this chapter for achieving higher accuracy in calibration parameter estimates.

### 2.1.1 Rotation Model

As the lens is planar and symmetric about optic axis, only two Euler angles corresponding to sensor rotation about the $x$-axis and $y$-axis of the lens coordinate system are sufficient to model $R$.

### 2.1.2 Imaging Model

#### Prior

If the sensor is physically tilted only about a single lens axis (e.g. $y$-axis), [17] showed that tilt estimates were more accurate in a pupil-centric imaging model compared to a thin-lens imaging model. They derived equations, in terms of pupil-centric parameter set $L$ to map calibration results obtained from thin-lens model to pupil-centric model and vice versa. Thus, one could simply do thin-lens calibration, apply the equations and obtain pupil-centric parameters.

#### Proposed

But for real cameras with arbitrary sensor tilt, these equations do not generalize directly. Instead, an affine transformation using $L$ is first applied to known checkerboard world points which are then input to a thin-lens calibration framework. The thin-lens calibration results can then be linearly transformed back to obtain pupil-centric estimates. Thus, we first derive generalized pupil centric imaging for arbitrary sensor tilt and obtain a mapping from pupil-centric to a geometrically equivalent thin-lens imaging model.
2.1.3 Calibration Technique

Typical camera calibration consists of three steps:

1. Assuming no lens distortion, compute a $3 \times 4$ perspective projection matrix whose elements encode calibration parameters via a system of nonlinear equations \([18]\).

2. Analytically solve these equations to obtain initial calibration parameter estimates \([19, 20, 15]\).

3. Nonlinearly refine these estimates taking lens distortion into account.

Prior

For implicitly modeled non-frontal sensors \([15, 16, 13]\), decentering parameters are estimated in step 2, which renders an initial estimation of $U$ in steps 1 and 2 an easy task. But, in calibration of explicitly modeled non-frontal sensors using generalized pupil-centric imaging, the number of calibration parameters increases to $(U, R, L)$, thus leading to two effects. First, the system of nonlinear equations in step 1 becomes under-determined. Second, the nonlinearity between elements of $(U, R, L)$ encoded in the projection matrix of step 1 becomes too complex causing step 2 to be difficult and non-trivial.

Proposed

To handle these, we append a novel pupil-centric constraint to the system of nonlinear calibration equations in step 1 and assume that two parameters in $(U, R, L)$ corresponding to CoD are known. This makes step 2 a well-constrained problem. Then we propose an new analytical solution for finding actual calibration parameters in step 2. This solution is then used in a computational framework to find an optimal estimate of CoD, such that a separate novel radial alignment based constraint is minimized (Section 2.8). Given the optimal CoD, the analytical solution is then used to reliably initialize the nonlinear minimization step 3. Finally, we also show that calibrated estimates of entrance pupil location and optical focal length of the camera 2.8 can also be achieved in the analytical framework via the pupil-centric constraint. Conventionally, these have been done optically \([17]\).
2.2 Related Work

Camera calibration is a well-researched area and a number of techniques exist depending on the imaging constraint being used. The work by [14] uses a radial alignment constraint whereas [15, 13] use collinearity constraint. Another constraint known as the homography constraint [16] between the object and the image plane has also been used for calibration. The work presented in this chapter is closest to the work done by Gennery [4] which proposed a calibration framework used on NASA’s Mars Exploration Rover Mission and proposes critical improvements to their work. To the best of our knowledge, this is the only work that considers explicit non-frontal parameterization. Following are some major differences of our work with theirs. First, they assume that angles of incidence and exitance of a light ray at the entrance and exit pupil are equal, which is incorrect [17], while we incorporate a generalized pupil-centric imaging which models the exact relationship between these rays. Second, they initialize nonlinear refinement of $U$ heuristically and set $R$ as an identity matrix, which can lead to local optima, while we analytically derive initial estimates leading to increased confidence in attaining a solution hopefully close to global optima. Third, they include decentering distortion as a part of nonlinear optimization, which is redundant for an explicit non-frontal model [12] and can lead to instable results. Fourth, we focus on cameras with a small field of view and thus assume that entrance-pupil is fixed (central) while they assume a general scenario of varying entrance-pupil (non-central) location (typical to fish-eye lenses).

Another explicit sensor tilt estimation technique was proposed in [21] using image defocus, but they do not show how to compute $R$ from their tilt parameterization. The major contributions in this chapter are as follows:

1. We generalize pupil-centric imaging from single axis to to arbitrarily rotated non-frontal sensors. This leads to a new mapping from pupil-centric to a geometrically equivalent thin-lens imaging. Thus, non-frontal calibration under pupil-centric imaging can be done in a thin-lens framework.

2. We derive the linear projection equations in terms of $(U, R, L)$ for the equivalent thin-lens imaging and the nonlinear relationship among calibration parameters encoded by these projection equations.

3. We propose an analytical solution to solve $(U, R)$ and two parameters of $L$, namely entrance pupil and optical focal length by incorporating a novel pupil-centric constraint
and assuming that CoD parameters in $U$ are known.

4. We develop a new radial alignment like constraint (similar to [14] based on analytical solution to estimate the center of radial distortion (CoD). A similar technique but using a different constraint has earlier been proposed by Scaramuzza et al. [22] for catadioptric cameras and by Tardif and Sturm [23] for cameras with radially symmetric distortion.

2.3 Pupil-Centric and Thin-Lens Imaging

We derive the generalized non-frontal thin-lens and pupil-centric imaging equations and the mapping between them. The coordinate systems (CS) used in this chapter are defined as:

- **World coordinate system ($C_W$):** A set of known world points is defined in this coordinate system e.g. one of the corners of a two-dimensional checkerboard could be defined as the origin of this CS with the xy plane corresponding to the checkerboard surface.

- **Lens coordinate system ($C_{H_1}, C_{H_2}$):** A coordinate system located at primary ($H_1$) and secondary ($H_2$) principal planes with their origin located at the intersection of the optic axis and the planes.

- **Entrance pupil coordinate system ($C_{E_n}$):** This coordinate system is parallel to $C_{H_1}$ but is centered at the center of entrance pupil where the optic axis intersects the entrance pupil plane.

- **Sensor coordinate system ($C_S$):** This coordinate system is located on the imaging sensor with origin at the intersection of optic axis and sensor plane.

- **Image coordinate system ($C_I$):** The measured image pixels are described in this coordinate system.

As the pupil-centric model requires the location of principal planes as a parameter, we will model thin-lens as a Gaussian thick-lens since both are geometrically equivalent. The notation $X_Y$ represents point $X$ in coordinate system $Y \in \{C_W, C_{H_1}, C_{H_2}, C_{E_n}, C_S, C_I\}$. A superscript of $T$ implies transpose of the underlying vector or matrix.
2.3.1 Gaussian Model of Image Formation

Consider Figure 2.1. Let \( P_{C_{H_1}} = (x_l, y_l, z_l) \) be a world point located in a coordinate system \( C_{H_1} \) whose origin \( O_1 \) is at the intersection of the optical axis and the first principal plane \( H_1 \).

Under the Gaussian thick-lens model of image formation, a light ray from \( P_{C_{H_1}} \) which passes through the first nodal point \( O_1 \) is called the principal ray. It is assumed that it comes out from the second nodal point \( O_2 \) at the same angle at which it entered \( O_1 \), i.e. \( \theta_{in} = \theta_{out} \). This ray is then incident on a non-frontal sensor at location \( P_{C_S}^g \).

![Figure 2.1: Gaussian model of image formation.](image-url)

The image plane has a local coordinate system \( C_S \) centered at the location where the optical axis intersects the image plane. Since, the image plane is non-frontal, \( C_S \) can be described as being translated by a translation \( Tg_{H_2}^S \) from the coordinate system \( C_{H_2} \) lying on the rear principal plane \( H_2 \) and rotated by the matrix transformation \( R_{H_2}^S \), where each of these transformations can be given as

\[
R_{H_2}^S = \begin{bmatrix}
  r_{11} & r_{12} & r_{13} \\
  r_{21} & r_{22} & r_{23} \\
  r_{31} & r_{32} & r_{33}
\end{bmatrix};
Tg_{H_2}^S = \begin{bmatrix}
  0 \\
  0 \\
  \lambda_g
\end{bmatrix}
\] (2.1)

Here, \( \lambda_g \) is the translation along the physical optic axis.

Assuming ideal perspective projection, the coordinates of image point \( P_{C_S}^g \) formed from the corresponding world point ray from \( P_{C_{H_1}} \) can be derived in terms of the parameters of \( (R_{H_2}^S, \lambda_g) \) as follows. Before the derivation, we note that a Gaussian thick-lens model is geometrically equivalent to a thin-lens imaging model, as the principal planes \( H_1 \) and \( H_2 \) can coincide with each other since \( \theta_{in} = \theta_{out} \) holds. This results in \( O_1 = O_2 = O_w \), where \( O_w = (0, 0, 0) \) in \( C_{H_1} \). We thus also have \( R_{H_1}^S = R_{H_2}^S \). The world point \( P_{C_{H_1}} \) and the origin \( O_w \) in coordinate system \( C_{H_1} \) can then be represented in \( C_S \) as \( P_{C_{H_1}}^S \) and \( O_w^S \) respectively.
where,

\[
\begin{bmatrix}
P_{C_{H_1}}^S \\
1
\end{bmatrix} = \begin{bmatrix}
R_{H_2}^S & R_{H_2}^S T g_{H_2}^S \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
P_{C_{H_1}} \\
1
\end{bmatrix} = \begin{bmatrix}
R_{H_2}^S (P_{C_{H_1}} + T g_{H_2}^S) \\
1
\end{bmatrix}
\]  
\tag{2.2}

\[
\begin{bmatrix}
O_w^S \\
1
\end{bmatrix} = \begin{bmatrix}
R_{H_2}^S T g_{H_2}^S \\
1
\end{bmatrix}
\]  
\tag{2.3}

The ray of light joining \( P_{C_{H_1}}^S \) and \( O_w^S \) intersects the image plane at \( P_{C_{S}}^g \). Thus, the point \( P_{C_{S}}^g \) in \( C_S \) can be obtained by solving for scale parameter \( \beta \) as:

\[
P_{C_{S}}^g = P_{C_{H_1}}^S + \beta (O_w^S - P_{C_{H_1}}^S)
= \beta O_w^S + (1 - \beta) P_{C_{H_1}}^S
= \beta (R_{H_2}^S T g_{H_2}^S) + (1 - \beta)(R_{H_2}^S P_{C_{H_1}} + R_{H_2}^S T g_{H_2}^S)
= R_{H_2}^S T g_{H_2}^S + (1 - \beta) R_{H_2}^S P_{C_{H_1}}
\]  
\tag{2.4}

where,

\[
R_{H_2}^S T g_{H_2}^S = \begin{bmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
r_{31} & r_{32} & r_{33}
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
\lambda_g
\end{bmatrix}
= \lambda_g
\begin{bmatrix}
r_{13} \\
r_{23} \\
r_{33}
\end{bmatrix}
\]  
\tag{2.5}

\[
R_{H_2}^S P_{C_{H_1}} = \begin{bmatrix}
T_{1}^T P_{C_{H_1}} \\
T_{2}^T P_{C_{H_1}} \\
T_{3}^T P_{C_{H_1}}
\end{bmatrix}; \quad T_i^T = \begin{bmatrix}
r_{i1} & r_{i2} & r_{i3}
\end{bmatrix}
\]  
\tag{2.6}

Since \( P_{C_{S}}^g \) lies on the image sensor plane, \( \beta \) can be obtained by simplifying \( P_{C_{S}}^g \) derived in Equation 2.4 using Equation 2.6 and then equating the resulting \( z \) coordinate to 0 as shown in Equation 2.7.

\[
\lambda_g r_{33} + (1 - \beta) r_{3}^T P_{C_{H_1}} = 0
\]

\[
\implies \beta = 1 + \frac{\lambda_g r_{33}}{r_{3}^T P_{C_{H_1}}}
\]  
\tag{2.7}

Using Equation 2.7 in Equation 2.4 yields \( P_{C_{S}}^g \) as,

\[
P_{C_{S}}^g = R_{H_2}^S T g_{H_2}^S + (1 - \beta) R_{H_2}^S P_{C_{H_1}}
= R_{H_2}^S T g_{H_2}^S + \left( -\frac{\lambda_g r_{33}}{r_{3}^T P_{C_{H_1}}} \right) R_{H_2}^S P_{C_{H_1}}
\]
\[
\begin{align*}
\begin{bmatrix}
\lambda_gr_{13} \\
\lambda_gr_{23} \\
\lambda_gr_{33}
\end{bmatrix} + \begin{pmatrix}
-\lambda_gr_{33}r_3^TP_{CH_1}^T \\
\lambda_gr_{33}r_3^TP_{CH_1}^T \\
0
\end{pmatrix}
\begin{bmatrix}
r_1^TP_{CH_1} \\
r_2^TP_{CH_1} \\
r_3^TP_{CH_1}
\end{bmatrix}
\end{align*}
\]

\[
= \begin{bmatrix}
\lambda_gr_{13} - \lambda_gr_{33}r_3^TP_{CH_1}^T \\
\lambda_gr_{23} - \lambda_gr_{33}r_3^TP_{CH_1}^T \\
0
\end{bmatrix}
\]

(2.8)

The above relationship can be simplified to obtain the \( x \) and \( y \) component of \( P_{CS}^g \) in terms of world points \( P_{CH_1} \) as

\[
P_{CS}^g(x) = \frac{\lambda_g(-r_{22}x_l + r_{21}y_l)}{r_{31}x_l + r_{32}y_l + r_{33}z_l}
\]

(2.9)

\[
P_{CS}^g(y) = \frac{\lambda_g(r_{12}x_l - r_{11}y_l)}{r_{31}x_l + r_{32}y_l + r_{33}z_l}
\]

(2.10)

If it is assumed that the image sensor is only tilted only about \( y \)-axis of coordinate system \( C_{H_1} \), then the above relationship can be simplified to the ones derived in [17].

**2.3.2 Pupil-Centric Model of Image Formation**

The pupil-centric model of image formation is shown in Figure 2.2. The world point \( P_{CH_1} = (x_l, y_l, z_l) \) is given in the coordinate system \( C_{H_1} \) attached to first principal plane \( H_1 \). We obtain the image location \( P_{CS}^p \) formed by the principal ray from the world point \( P_{CH_1} \) formed on the image plane located at distance \( \lambda_p \) from \( H_2 \) and rotated with respect to \( C_{H_2} \) by an amount represented by rotation matrix \( R_{H_2}^S \). These variables are same as assumed
while analyzing the image formation using a Gaussian thick-lens model as described in Section 2.3.1.

In this model, the principal ray responsible for image formation is the ray which passes through the center of the entrance pupil \( E_n = (0, 0, a_n) \) defined in coordinate system \( C_{H_1} \) and impinges on the front principal plane \( H_1 \) at \( h_1 \). This point can be obtained in terms of parameter \( \beta \) as follows:

\[
\begin{align*}
  h_1 &= P_{C_{H_1}} + \beta(E_n - P_{C_{H_1}}) \\
  &= \beta E_n + (1 - \beta)P_{C_{H_1}} \\
  &= \beta \begin{bmatrix} 0 \\ 0 \\ a_n \end{bmatrix} + (1 - \beta) \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix}
\end{align*}
\]  
(2.11)

The constant \( \beta \) can be computed as follows. Since \( h_1 \) lies on the \( xy \) plane, the \( z \) coordinate of \( h_1 \) is 0. Thus, we have

\[
\beta a_n + (1 - \beta)z_l = 0 \\
\beta = \frac{z_l}{z_l - a_n}
\]  
(2.12)

Replacing \( \beta \) from Equation 2.12 into Equation 2.11 we get

\[
\begin{bmatrix}
  x_l \\
  y_l \\
  z_l
\end{bmatrix}
\]  
(2.13)

The principal ray after reaching front principal plane \( H_1 \) exits from the back principal plane \( H_2 \) at location \( h_2 \). Let \( H_2 \) be obtained by translating \( H_1 \) by a value \( d \) which denotes the distance between the two principal planes \( H_1 \) and \( H_2 \). Thus, \( h_2 \) can be obtained from Equation 2.13 as,

\[
\begin{bmatrix}
  \frac{-a_n x_l}{z_l - a_n} \\
  \frac{-a_n y_l}{z_l - a_n} \\
  d
\end{bmatrix}
\]  
(2.14)

Let,

\[
x_1 = \frac{-a_n x_l}{z_l - a_n}
\]  
(2.15)
\[ y_1 = \frac{-a_n y_l}{z_l - a_n} \quad (2.16) \]
\[ z_1 = d \quad (2.17) \]

After \( h_2 \), the direction of principal ray is governed by the location of the exit pupil \( E_x = (0, 0, a_x) \) (in \( C_{H_1} \)) as it should either pass or appear to pass through \( E_x \) (as is illustrated in Figure 2.2). The translation transformation from \( C_{H_2} \) to \( C_S \) can be represented as

\[
T_{p_{H_2}}^S = \begin{bmatrix} 0 \\ 0 \\ -d + \lambda_p \end{bmatrix} \quad (2.18)
\]

where, it is assumed that \( d \) is -ve in the \( C_{H_1} \) coordinate system. Now, given \( T_{p_{H_2}}^S \) and the rotation transformation \( R_{H_2}^S \) between \( C_{H_2} \) and \( C_S \) we can obtain \( h_2 \) and \( E_x \) in the coordinate system \( C_S \) attached to the image plane and represented as \( h_2^S \) and \( E_x^S \) respectively.

\[
h_2^S = R_{H_2}^S (h_2 + T_{p_{H_2}}^S) \quad (2.19)
\]
\[
E_x^S = R_{H_2}^S (E_x + T_{p_{H_2}}^S) \quad (2.20)
\]

Using Equations 2.19 and 2.20, the line joining \( h_2^S \) and \( E_x^S \) can be expressed in terms of a variable \( \beta \) as:

\[
\overrightarrow{E_x h_2^S} = E_x^S + \beta (h_2^S - E_x^S)
\]
\[
= R_{H_2}^S \left[ E_x + T_{p_{H_2}}^S + \beta (h_2 - E_x) \right]
\]
\[
= R_{H_2}^S \begin{bmatrix} 0 \\ 0 \\ a_x - d + \lambda_p \end{bmatrix} + \beta \begin{bmatrix} x_1 \\ y_1 \\ z_1 - a_x \end{bmatrix}
\]
\[
= R_{H_2}^S \begin{bmatrix} \beta x_1 \\ \beta y_1 \\ (a_x - d + \lambda) + \beta (z_1 - a_x) \end{bmatrix} \quad (2.21)
\]

To find the image location \( P_{C_S}^p \) on the image sensor, the \( z \)-coordinate of \( \overrightarrow{E_x h_2^S} \) is set to 0 and solved for \( \beta \) yielding

\[
\beta = \frac{r_{33}(a_x - d + \lambda_p)}{a_x r_{33}^2 X_1} \quad (2.22)
\]
where,

\[
X_1 = \begin{bmatrix}
    x_1 \\
    y_1 \\
    z_1
\end{bmatrix}
\]

\[
r^T_3 = [r_{31} \ r_{32} \ r_{33}]
\]

Thus,

\[
P_{CS}^p = \begin{bmatrix}
    r_{11}\beta x_1 + r_{12}\beta y_1 + r_{13}(a_x - d + \lambda_p - \beta a_x) \\
    r_{21}\beta x_1 + r_{22}\beta y_1 + r_{23}(a_x - d + \lambda_p - \beta a_x) \\
    0 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
    \beta(r_{11}x_1 + r_{12}y_1 + r_{13}z_1) + r_{13}(a_x - d + \lambda_p - \beta a_x) \\
    \beta(r_{21}x_1 + r_{22}y_1 + r_{23}z_1) + r_{23}(a_x - d + \lambda_p - \beta a_x) \\
    0
\end{bmatrix}
\]

(2.23)

Now we simplify the \( x \)-component of \( P_{CS}^p \) as follows,

\[
\beta(r_{11}x_1 + r_{12}y_1 + r_{13}z_1) + r_{13}(a_x - d + \lambda_p - \beta a_x)
\]

\[
= \frac{\beta(r^T_1X_1) + r_{13}(a_x - d + \lambda_p - \beta a_x)}{a_xr_{33} - r^T_3X_1}
\]

\[
= \frac{(a_x - d + \lambda_p)(x_1(r_{11}r_{33} - r_{13}r_{31}) + y_1(r_{12}r_{33} - r_{13}r_{32}))}{a_xr_{33} - r^T_3X_1}
\]

(Using \( R_{H_2}^{-1} = R_{H_2}^T \))

\[
= \frac{(a_x - d + \lambda_p)[r_{22}x_1 - r_{21}y_1]}{a_xr_{33} - r^T_3X_1}
\]

(2.24)

Since the entrance pupil \( E_n \) and exit pupil \( E_x \) are conjugate to each other, they are related by the thin-lens equation. We need to obtain the coordinates of \( E_x \) in a right-hand coordinate system which is centered at \( O_2 \) and has its \( z \)-axis toward the image plane. Let the new coordinates of \( E_x \) in this coordinate system be denoted as \( E'_x \). We have

\[
E'_x = \begin{bmatrix}
    R_{y_1}(\pi) & 0 \\
    0 & 1
\end{bmatrix} \begin{bmatrix}
    I & T_1 \\
    0 & 1
\end{bmatrix} \begin{bmatrix}
    E_x \\
    1
\end{bmatrix}
\]

(2.25)
where, \( R_{yl}(\pi) \) is a clockwise rotation of \( C_{H_1} \) about the axis \( y_l \) by \( \pi \) radians, \( T_1 \) is the translation of \( C_{H_1} \) from principal plane \( H_1 \) to \( H_2 \) and \( E_x = (0, 0, a_x) \). Thus we have,

\[
R_{yl}(\pi) = \begin{bmatrix}
-1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & -1
\end{bmatrix} ; \quad T_1 = \begin{bmatrix}
0 \\
0 \\
-d
\end{bmatrix}
\] (2.26)

Simplifying Equation 2.25 using Equation 2.26 we get,

\[
E'_x = R_{yl}(\pi)(E_x + T_1)
\]

\[
= \begin{bmatrix}
-1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & -1
\end{bmatrix} \left( \begin{bmatrix}
0 \\
0 \\
a_x
\end{bmatrix} + \begin{bmatrix}
0 \\
0 \\
-d
\end{bmatrix} \right)
\]

\[
= \begin{bmatrix}
0 \\
0 \\
-(a_x - d)
\end{bmatrix}
\] (2.27)

Applying the thin-lens equation to \( E_n \) and \( E'_x \),

\[
\frac{1}{a_n} + \frac{1}{-(a_x - d)} = \frac{1}{F}
\]

\[
a_x - d = \frac{a_n F}{F - a_n}
\] (2.28)

Using \( (a_x - d) \) from Equation 2.28 and \((x_1, y_1, z_1)\) from Equations 2.15, 2.16 and 2.17 into Equation 2.24 we get the \( x \)-component of \( P^n_{C_S} \) as follows

\[
P^n_{C_S}(x) = \frac{(a_x - d + \lambda_p)[r_{22}x_1 - r_{21}y_1]}{-r_{31}x_1 - r_{32}y_1 + r_{33}(a_x - d)} - r_{31}
\]

\[
\left( \frac{a_n F}{F - a_n} + \lambda \right) \begin{bmatrix}
r_{22} \left( -\frac{a_n x_l}{z_l - a_n} \right) - r_{21} \left( -\frac{a_n y_l}{z_l - a_n} \right)
\end{bmatrix}
\]

\[
= \left( \frac{F}{F - a_n} \right) \left( a_n + \lambda_p - \frac{a_n \lambda_p}{F} \right) \begin{bmatrix}
-r_{22}x_l + r_{21}y_l \left( \frac{a_n}{z_l - a_n} \right)
\end{bmatrix}
\]

\[
r_{31} \left( \frac{a_n}{z_l - a_n} \right) x_l + r_{32} \left( \frac{a_n}{z_l - a_n} \right) y_l + r_{33} \left( \frac{a_n F}{F - a_n} \right)
\] (2.29)
Dividing Nr and Dr by \( \left( \frac{F}{F-a_n} \right) \left( \frac{a_n}{z_l-a_n} \right) \) yields,

\[
P^p_{CS}(x) = \frac{(a_n + \lambda_p - \frac{a_n \lambda_p}{F}) \left( -r_{22} x_l + r_{21} y_l \right)}{r_{31} \left( \frac{F-a_n}{F} \right) x_l + r_{32} \left( \frac{F-a_n}{F} \right) y_l + r_{33} (z_l - a_n)} \tag{2.30}
\]

Applying a similar analysis as shown from Equations 2.24-2.29 for the \( x \)-component of \( P^p_{CS} \) in Equation 2.23 to the \( y \) component of \( P^p_{CS} \) in Equation 2.23 yields:

\[
P^p_{CS}(y) = \frac{(a_n + \lambda_p - \frac{a_n \lambda_p}{F}) \left( r_{12} x_l - r_{11} y_l \right)}{r_{31} \left( \frac{F-a_n}{F} \right) x_l + r_{32} \left( \frac{F-a_n}{F} \right) y_l + r_{33} (z_l - a_n)} \tag{2.31}
\]

Now, multiplying and dividing the numerator of Equations 2.30 and 2.31 with \( \left( \frac{F-a_n}{F} \right) \), the final form of \( P^p_{CS} \) is obtained as:

\[
P^p_{CS}(x) = \frac{\left( \lambda_p + \frac{a_n F}{F-a_n} \right) \left( -r_{22} \left( \frac{F-a_n}{F} \right) x_l + r_{21} \left( \frac{F-a_n}{F} \right) y_l \right)}{r_{31} \left( \frac{F-a_n}{F} \right) x_l + r_{32} \left( \frac{F-a_n}{F} \right) y_l + r_{33} (z_l - a_n)} \tag{2.32}
\]

\[
P^p_{CS}(y) = \frac{\left( \lambda_p + \frac{a_n F}{F-a_n} \right) \left( r_{12} \left( \frac{F-a_n}{F} \right) x_l - r_{11} \left( \frac{F-a_n}{F} \right) y_l \right)}{r_{31} \left( \frac{F-a_n}{F} \right) x_l + r_{32} \left( \frac{F-a_n}{F} \right) y_l + r_{33} (z_l - a_n)} \tag{2.33}
\]

Let

\[
\alpha = \frac{F-a_n}{F} \tag{2.34}
\]

### 2.3.3 Mapping Pupil-Centric to Thin-Lens

The thin-lens and pupil-centric model are geometrically equivalent if \( P^g_{CS} = P^p_{CS} \) (Figures 2.1 and 2.2). Comparing Equations 2.9 and 2.32 with Equations 2.10 and 2.33 we get the following mapping. If

\[
\lambda_g = \lambda_p + \frac{a_n}{\alpha} \tag{2.35}
\]

where

\[
\frac{a_n}{\alpha} = O_2 E_x \tag{2.36}
\]
and if $P_{C_{H_1}} = (x_l, y_l, z_l)$ is transformed by $A_{pg}$ where,

\[
A_{pg} = \begin{bmatrix}
\alpha & 0 & 0 \\
0 & \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}
\]  

(2.37)

and translated along the optic axis by $-a_n$ resulting in new world coordinates of $P_{C_{E_n}} = (x^g_l, y^g_l, z^g_l)$ as

\[
x^g_l = \alpha x_l \\
y^g_l = \alpha y_l \\
z^g_l = z_l - a_n
\]  

(2.38)

then, the pupil-centric model is geometrically equivalent to a thin-lens model. This implies that thin-lens calibration with transformed world point $P_{C_{E_n}}$ (Equation 2.38) will yield estimates for rotation $R$ and parameters $\lambda_g, a_n, \alpha$. Given these parameters and from Equation 2.35, $\lambda_p$ can be obtained. Next, we obtain projection equations for equivalent thin-lens model which will be used for analytical calibration.

2.4 Example Pupil-Centric Lens Model

Table 2.1: Cinegon 1.4/8 mm lens parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Labels (Figure 2.3)</th>
<th>Value (in mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focal length</td>
<td>$H_1 F_1, H_2 F_2$</td>
<td>8.2</td>
</tr>
<tr>
<td>Front focal length</td>
<td>$L_1 F_1$</td>
<td>11.7</td>
</tr>
<tr>
<td>Back focal length</td>
<td>$L_2 F_2$</td>
<td>12.6</td>
</tr>
<tr>
<td>Principal/nodal point separation</td>
<td>$H_1 H_2$</td>
<td>20.9</td>
</tr>
<tr>
<td>Distance from front glass vertex to entrance pupil</td>
<td>$L_1 E_n$</td>
<td>13.4</td>
</tr>
<tr>
<td>Distance from rear glass vertex to exit pupil</td>
<td>$L_2 E_x$</td>
<td>27.0</td>
</tr>
</tbody>
</table>

In our experiments, we use a Cinegon 1.4/8 mm lens [24], whose specifications are described in Table 2.1 and shown in Figure 2.3. From these specifications, we will derive various pupil-centric parameters of $a_n, a_x, d$ and $F$. We have,

\[
a_n = \overrightarrow{H_1 E_n'} \text{ (Figure 2.3)}
\]
In order to verify the calculated values, we plug the values in Equations 2.39-2.42 into
Equation 2.28, which is derived from the thin-lens constraint.

\[ a_x - d = 10.5 \text{ mm } + 20.9 \text{ mm} \]
\[ = 31.4 \text{ mm} \]
\[ \frac{a_n F}{F - a_n} = 1.7 \text{ mm} \]
\[ = 31.35 \text{ mm} \]

which are approximately equal.

2.5 Non-Frontal Projection Equations

In this section, we derive the non-frontal projection equations for a thin-lens imaging model derived from a pupil-centric imaging model (Section 2.3.3). These equations encode the calibration parameters which we want to estimate. See Figure 2.4 for the imaging setting. Here, a world point \( P_{cw} = (X, Y, Z) \) is imaged at pixel location \( P_{ci} = (I, J) \). Assuming collinearity of these two points, they can be related in terms of various calibration parameters as described in Sections 2.5.1-2.5.5.
2.5.1 Transformation from $C_W$ to $C_{H_1}$

Given a $3 \times 3$ rotation matrix $S_W^{H_1} = (s_{ij} : 1 \leq (i, j) \leq 3)$ and $3 \times 1$ translation vector $T_W^{H_1} = (t_x, t_y, t_z)$ between $C_W$ and $C_{H_1}$, $P_{C_W}$ can be represented in $C_{H_1}$ as $P_{C_{H_1}} = (x_l, y_l, z_l)$,

$$
\begin{bmatrix}
  x_l \\
y_l \\
z_l \\
1
\end{bmatrix} = \begin{bmatrix}
  S_W^{H_1} & T_W^{H_1} \\
  0 & 1
\end{bmatrix} \begin{bmatrix}
P_{C_W} \\
1
\end{bmatrix} = \begin{bmatrix}
s_{11}X + s_{12}Y + s_{13}Z + t_x \\
s_{21}X + s_{22}Y + s_{23}Z + t_y \\
s_{31}X + s_{32}Y + s_{33}Z + t_z \\
1
\end{bmatrix}
$$

(2.43)

2.5.2 Pupil-Centric to Gaussian ($C_{H_1}$ to $C_{E_n}$)

Applying the equivalence relationship from Section 2.3.3, $P_{C_{H_1}}$ is transformed to $P_{C_{E_n}} = (x_l^g, y_l^g, z_l^g)$, thus converting a pupil-centric calibration problem to a thin-lens one as:

$$
\begin{bmatrix}
x_l^g \\
y_l^g \\
z_l^g \\
1
\end{bmatrix} = \begin{bmatrix}
\alpha & 0 & 0 & 0 \\
0 & \alpha & 0 & 0 \\
0 & 0 & 1 & -a_n \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x_l \\
y_l \\
z_l \\
1
\end{bmatrix} = \begin{bmatrix}
\alpha x_l \\
\alpha y_l \\
z_l - a_n \\
1
\end{bmatrix}
$$

(2.44)

$$
\begin{bmatrix}
\alpha s_{11}X + \alpha s_{12}Y + \alpha s_{13}Z + \alpha t_x \\
\alpha s_{21}X + \alpha s_{22}Y + \alpha s_{23}Z + \alpha t_y \\
s_{31}X + s_{32}Y + s_{33}Z + t_z - a_n \\
1
\end{bmatrix}
$$

(2.45)

2.5.3 Transformation from $C_{E_n}$ to $C_S$

Given a $3 \times 3$ rotation matrix $R_{E_n}^S = R_{H_2}^S = (r_{ij} : 1 \leq (i, j) \leq 3)$ and $3 \times 1$ translation vector $T_{E_n}^S = (0, 0, \lambda_g)$ between $C_{E_n}$ and $C_S$, $P_{C_{E_n}}$ can be represented in $C_S$ as $P_{C_S} = (x_s, y_s, z_s)$ where

$$
\begin{bmatrix}
x_s \\
y_s \\
z_s \\
1
\end{bmatrix} = \begin{bmatrix}
R_{E_n}^S & 0 \\
0 & 1
\end{bmatrix} \begin{bmatrix}
I & T_{E_n}^S \\
0 & 1
\end{bmatrix} \begin{bmatrix}
P_{C_{E_n}} \\
1
\end{bmatrix} = \begin{bmatrix}
r_{11}x_l^g + r_{12}y_l^g + r_{13}z_l^g + r_{13}\lambda_g \\
r_{21}x_l^g + r_{22}y_l^g + r_{23}z_l^g + r_{23}\lambda_g \\
r_{31}x_l^g + r_{32}y_l^g + r_{33}z_l^g + r_{33}\lambda_g \\
1
\end{bmatrix}
$$

(2.46)

where, $R_{E_n}^S = R_{H_2}^S$ since $xy$ planes of $C_{H_1}, C_{H_2}, C_{E_n}$ are parallel to each other and the optic axis is the common $z$-axis (Figure 2.2), $I$ is a $3 \times 3$ identity matrix. $\lambda_g$ is the distance
between the origin of $C_E$ and $C_S$ measured along the optic axis/z-axis of $C_E$. The center of projection is located at $E_n$. Let its coordinates in $C_S$ can be obtained as $O_{CS} = (r_{13}\lambda_g, r_{23}\lambda_g, r_{33}\lambda_g)$.

2.5.4 Central Perspective Projection

Given the world point $P_{CS}$ and location $O_{CS}$, the intersection of the image ray connecting these two points on the non-frontal sensor can be computed as $P_{nf} = (x_{nf}, y_{nf})$, where

$$
\begin{bmatrix}
  x_{nf} \\
  y_{nf}
\end{bmatrix} =
\begin{bmatrix}
-\lambda_g(r_{22}x_g^0 - r_{21}y_g^0) \\
\frac{1}{\lambda_g(r_{12}x_g^0 - r_{11}y_g^0)} \\
\frac{1}{\lambda_g(r_{31}x_g^0 + r_{32}y_g^0 + r_{33}z_g^0)}
\end{bmatrix}
$$

(2.47)

2.5.5 Metric to Pixels ($C_S$ to $C_I$)

Next, $P_{nf}$ can be converted to image coordinates $P_{CI} = (I, J)$ via pixel sizes ($s_x, s_y$) and CoD $(I_0, J_0)$ as:

$$
I = \frac{x_{nf}}{s_x} - I_0
$$

(2.48)

$$
J = \frac{y_{nf}}{s_y} - J_0
$$

(2.49)

where, the skew in $C_I$ is assumed to be 0 [13, 25]. Ignoring noise and image distortion, the predicted image coordinates $P_{CI}$ will correspond to the actual measured image coordinates. Thus, assuming that the world point $P_{CW} = (X, Y, Z)$ and the corresponding image point $P_{CI} = (I, J)$ are known, we can simplify the collinearity relationship given by Equations 2.48 and 2.49 to obtain the linear non-frontal calibration equation.

2.6 Collinearity Based Linear Calibration Equation

In this section, we derive the linear calibration equation for non-frontal sensor model using the collinearity based constraint as in Equations 2.48 and 2.49. We have,

$$
I + I_0 = -\frac{\lambda_g}{s_x} \frac{(r_{22}x_i^0 - r_{21}y_i^0)}{(r_{31}x_i^0 + r_{32}y_i^0 + r_{33}z_i^0)}
$$

$$
\Rightarrow (I + I_0)(r_{31}x_i^0 + r_{32}y_i^0 + r_{33}z_i^0) = -\lambda_{gx}(r_{22}x_i^0 - r_{21}y_i^0)

\left(\lambda_{gx} = \frac{\lambda_g}{s_x}\right)
$$

22
\(\implies (I_{r31} + I_0 r_{31} + \lambda_{gx} r_{22}) x_i^\theta_A + (I_{r32} + I_0 r_{32} - \lambda_{gx} r_{21}) y_i^\theta_B + (I_{r33} + I_0 r_{33}) z_i^\theta_C = 0 \quad (2.50)\)

Simplifying term A above using \((x_i^\theta_A, y_i^\theta_B, z_i^\theta_C)\) from Equation 2.45 we get,

\[(I_{r31} + I_0 r_{31} + \lambda_{gx} r_{22}) x_i^\theta = (I_{r31} + I_0 r_{31} + \lambda_{gx} r_{22})(\alpha s_{11} X + \alpha s_{12} Y + \alpha s_{13} Z + \alpha t_x)\]
\[= X(I_0 a r_{31} s_{11} + \lambda_{gx} a r_{22} s_{11}) + Y(I_0 a r_{31} s_{12} + \lambda_{gx} a r_{22} s_{12}) + Z(I_0 a r_{31} s_{13} + \lambda_{gx} a r_{22} s_{13})\]
\[+ I X(a r_{31} s_{11}) + I Y(a r_{31} s_{12}) + I Z(a r_{31} s_{13})\]
\[+ I (a r_{31} t_x) + (I_0 a r_{31} t_x + \lambda_{gx} a r_{22} t_x) \quad (2.51)\]

Simplifying term B above using \((x_i^\theta_A, y_i^\theta_B, z_i^\theta_C)\) from Equation 2.45 we get,

\[(I_{r32} + I_0 r_{32} - \lambda_{gx} r_{21}) y_i^\theta = (I_{r32} + I_0 r_{32} - \lambda_{gx} r_{21})(\alpha s_{21} X + \alpha s_{22} Y + \alpha s_{23} Z + \alpha t_y)\]
\[= X(I_0 a r_{32} s_{21} + \lambda_{gx} a r_{22} s_{21}) + Y(I_0 a r_{32} s_{22} - \lambda_{gx} a r_{21} s_{22}) + Z(I_0 a r_{32} s_{23} - \lambda_{gx} a r_{21} s_{23})\]
\[+ I X(a r_{32} s_{21}) + I Y(a r_{32} s_{22}) + I Z(a r_{32} s_{23})\]
\[+ I (a r_{32} t_y) + (I_0 a r_{32} t_y - \lambda_{gx} a r_{21} t_y) \quad (2.52)\]

Simplifying term C above using \((x_i^\theta_A, y_i^\theta_B, z_i^\theta_C)\) from Equation 2.45 we get,

\[(I_{r33} + I_0 r_{33}) z_i^\theta = (I_{r33} + I_0 r_{33})(s_{31} X + s_{32} Y + s_{33} Z + t_z - a_n)\]
\[= X(I_0 r_{33} s_{31}) + Y(I_0 r_{33} s_{32}) + Z(I_0 r_{33} s_{33})\]
\[+ I X(r_{33} s_{31}) + I Y(r_{33} s_{32}) + I Z(r_{33} s_{33})\]
\[+ I (r_{33} (t_z - a_n)) + (I_0 r_{33} (t_z - a_n)) \quad (2.53)\]

Thus Equation 2.50 can now be written in terms of Equations 2.51, 2.52 and 2.53 as:

\[X(I_0 (a r_{31} s_{11} + a r_{32} s_{21} + r_{33} s_{31}) + \lambda_{gx} a (r_{22} s_{11} - r_{21} s_{21}))\]
\[+ Y(I_0 (a r_{31} s_{12} + a r_{32} s_{22} + r_{33} s_{32}) + \lambda_{gx} a (r_{22} s_{12} - r_{21} s_{22}))\]
\[+ Z(I_0 (a r_{31} s_{13} + a r_{32} s_{23} + r_{33} s_{33}) + \lambda_{gx} a (r_{22} s_{13} - r_{21} s_{23}))\]
\[+ I X(a r_{31} s_{11} + a r_{32} s_{21} + r_{33} s_{31})\]
\[+ I Y(a r_{31} s_{12} + a r_{32} s_{22} + r_{33} s_{32})\]
\[+ I Z(a r_{31} s_{13} + a r_{32} s_{23} + r_{33} s_{33})\]
\[ + I(a_{31}t_x + a_{32}t_y + r_{33}(t_z - a_n)) \]
\[ + (I_0(a_{31}t_x + a_{32}t_y + r_{33}(t_z - a_n)) + \lambda_{gy}\alpha(r_{22}t_x - r_{21}t_y)) = 0 \tag{2.54} \]

Using the above expansion, the collinearity relationship in Equation 2.49 can also be simplified to obtain

\[ X(J_0(a_{31}s_{11} + a_{32}s_{21} + r_{33}s_{31}) + \lambda_{gy}\alpha(r_{11}s_{21} - r_{12}s_{11})) \]
\[ + Y(J_0(a_{31}s_{12} + a_{32}s_{22} + r_{33}s_{32}) + \lambda_{gy}\alpha(r_{11}s_{22} - r_{12}s_{12})) \]
\[ + Z(J_0(a_{31}s_{13} + a_{32}s_{23} + r_{33}s_{33}) + \lambda_{gy}\alpha(r_{11}s_{23} - r_{12}s_{13})) \]
\[ + JX(a_{31}s_{11} + a_{32}s_{21} + r_{33}s_{31}) \]
\[ + JY(a_{31}s_{12} + a_{32}s_{22} + r_{33}s_{32}) \]
\[ + JZ(a_{31}s_{13} + a_{32}s_{23} + r_{33}s_{33}) \]
\[ + J(a_{31}t_x + a_{32}t_y + r_{33}(t_z - a_n)) \]
\[ + (J_0(a_{31}t_x + a_{32}t_y + r_{33}(t_z - a_n)) - \lambda_{gy}\alpha(r_{12}t_x - r_{11}t_y)) = 0 \tag{2.55} \]

For each world and image point correspondence, we get two linear collinearity constraints parameterized by 14 intrinsic and extrinsic calibration parameters. Given \( N \) correspondences, the two equations can be stacked for each of them to obtain the following matrix equation

\[ A_{N \times 11}Q_{11 \times 1} = b_{N \times 1} \tag{2.56} \]

where,

\[ \begin{bmatrix} X_i & Y_i & Z_i & 0 & 0 & 0 & I_iX_i & I_iY_i & I_iZ_i & 1 & 0 \\ 0 & 0 & 0 & X_i & Y_i & Z_i & J_iX_i & J_iY_i & J_iZ_i & 0 & 1 \end{bmatrix} \begin{bmatrix} Q_1 \\ \vdots \\ Q_{11} \end{bmatrix} = \begin{bmatrix} -I_i \\ -J_i \end{bmatrix} \tag{2.57} \]

And the linear coefficients \( Q \) are parameterized as follows

\[ Q_1 = \frac{I_0(\alpha s_{11}r_{31} + \alpha s_{21}r_{32} + s_{31}r_{33}) + \lambda_{gy}\alpha(r_{22}s_{11} - r_{21}s_{21})}{a_{31}t_x + a_{32}t_y + r_{33}(t_z - a_n)} \tag{2.58} \]
\[ Q_2 = \frac{I_0(\alpha s_{12}r_{31} + \alpha s_{22}r_{32} + s_{32}r_{33}) + \lambda_{gy}\alpha(r_{22}s_{12} - r_{21}s_{22})}{a_{31}t_x + a_{32}t_y + r_{33}(t_z - a_n)} \tag{2.59} \]
\[ Q_3 = \frac{I_0(\alpha s_{13}r_{31} + \alpha s_{23}r_{32} + s_{33}r_{33}) + \lambda_{gy}\alpha(r_{22}s_{13} - r_{21}s_{23})}{a_{31}t_x + a_{32}t_y + r_{33}(t_z - a_n)} \tag{2.60} \]
where,

$$\lambda_{gx} = \frac{\lambda_g}{s_x}$$

$$\lambda_{gy} = \frac{\lambda_g}{s_y}$$

$$\alpha = F - a_n$$

The set of calibration parameters $U$ now becomes

$$U = \{s_{ij} (3 \text{ angles}), r_{ij} (2 \text{ angles}) : 1 \leq (i, j) \leq 3, t_x, t_y, t_z, \lambda_g, s_x, I_0, J_0, a_n, \alpha \} \quad (2.69)$$

where, $(s_{ij}, t_x, t_y, t_z)$ are the extrinsic parameters, $(\lambda_g, s_x, I_0, J_0)$ are the intrinsic parameters and $(a_n, \alpha)$ are the pupil-centric parameters.

### 2.7 Analytical Calibration

In this section we propose our technique for computing $U$ given Equations 2.58-2.68. We observe that there are 11 equations from collinearity constraint but 14 unknown calibration parameters (size of $U$). Later, we augment these equations with a pupil-centric constraint equation, which increases the number of constraints to 12. In order to get a unique solution
to the set of 12 equations, two parameters need to be pre-calibrated or known beforehand. We select \((I_0, J_0)\) as the known set of parameters as they are coupled with the sensor tilt and cannot be determined uniquely from collinearity based calibration. They can either be calibrated using a laser based auto-collimation method [26] or can be computed separately using a cost function different from collinearity constraint as shown later in Section 2.8.

For rotation, we use the convention that it rotates the \(x\), \(y\) and \(z\)-axes sequentially in a clockwise direction while looking toward the origin. We assume \(S_W^H\) (Equation 2.43) and \(R_S^H\) (Equation 2.46) follow this convention and are composed of Euler angles \((\theta, \phi, \psi)\) and \((\rho, \sigma, \tau)\) respectively.

**Lemma 1** For a rotation matrix \(R\), its inverse and transpose are equal. Thus we get nine equations relating the elements of \(R\) as follows,

\[
R = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \text{ then, } \begin{bmatrix} (ei - hf) & (ch - bi) & (bf - ce) \\ (gf - di) & (ai - cg) & (dc - af) \\ (dh - eg) & (bg - ah) & (ae - bd) \end{bmatrix} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}
\]

We also define a \(3 \times 3\) matrix:

\[
R_{SW} = \alpha (R_{En}^{-1} S_{pg} W_{SW})
\]

\[
= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}
\]

\[
= \begin{bmatrix} r_{11}s_{11} + r_{12}s_{21} + \alpha r_{13}s_{31} \\ r_{21}s_{11} + r_{22}s_{21} + \alpha r_{23}s_{31} \\ r_{31}s_{11} + r_{32}s_{21} + \alpha r_{33}s_{31} \end{bmatrix} \begin{bmatrix} r_{12}s_{11} + r_{12}s_{22} + \alpha r_{13}s_{32} \\ r_{22}s_{11} + r_{22}s_{22} + \alpha r_{23}s_{32} \\ r_{32}s_{11} + r_{32}s_{22} + \alpha r_{33}s_{32} \end{bmatrix} \begin{bmatrix} r_{13}s_{11} + r_{13}s_{23} + \alpha r_{13}s_{33} \\ r_{23}s_{11} + r_{23}s_{23} + \alpha r_{23}s_{33} \\ r_{33}s_{11} + r_{33}s_{23} + \alpha r_{33}s_{33} \end{bmatrix}
\]

whose \((i, j)^{th}\) element is

\[
rs_{ij} = r_{i1}s_{1j} + r_{i2}s_{2j} + \alpha r_{i3}s_{3j}
\]

The rotation matrix is defined as rotating the coordinate system clockwise when viewing
toward the origin, e.g. the rotation matrix $R_{En}^S(\rho, \sigma, \tau)$ can be expressed as:

$$R_{En_x}^S(\rho) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho) & -\sin(\rho) \\ 0 & \sin(\rho) & \cos(\rho) \end{bmatrix}$$ (2.74)

$$R_{En_y}^S(\sigma) = \begin{bmatrix} \cos(\sigma) & 0 & \sin(\sigma) \\ 0 & 1 & 0 \\ -\sin(\sigma) & 0 & \cos(\sigma) \end{bmatrix}$$ (2.75)

$$R_{En_z}^S(\tau) = \begin{bmatrix} \cos(\tau) & -\sin(\tau) & 0 \\ \sin(\tau) & \cos(\tau) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ (2.76)

Combining the rotations, we get $R_{En}^S(\rho, \sigma, \tau)$ as:

$$R_{En}^S(\rho, \sigma, \tau) = R_{En_x}^S(\rho) \cdot R_{En_y}^S(\sigma) \cdot R_{En_z}^S(\tau) = \begin{bmatrix} \cos(\sigma) \cos(\tau) \sin(\sigma) \cos(\tau) - \cos(\rho) \sin(\tau) & \sin(\rho) \sin(\tau) + \cos(\rho) \sin(\sigma) \cos(\tau) \\ \cos(\sigma) \sin(\tau) \cos(\rho) \cos(\sigma) + \sin(\rho) \sin(\sigma) \sin(\tau) & \cos(\rho) \sin(\sigma) \sin(\tau) - \sin(\rho) \cos(\sigma) \\ -\sin(\rho) \cos(\sigma) & \cos(\rho) \cos(\sigma) \end{bmatrix}$$ (2.77)

If the rotation between lens and sensor about the $z$-axis is assumed to be 0, due to the rotational symmetry of the lens, then $\tau = 0$. The rotation matrix $R_{En}^S$ becomes:

$$R_{En}^S(\rho, \sigma, 0) = \begin{bmatrix} \cos(\sigma) & \sin(\rho) \sin(\sigma) & \cos(\rho) \sin(\sigma) \\ 0 & \cos(\rho) & -\sin(\rho) \\ -\sin(\rho) \cos(\sigma) & \cos(\rho) \cos(\sigma) \end{bmatrix}$$ (2.78)

Comparing $R_{En}^S$ from Equation 2.78 and $R_{En}^S = (r_{ij} : 1 \leq (i,j) \leq 3)$ we get

$$r_{21} = 0$$ (2.79)

Similarly, for the extrinsic rotation $S_{W}^{H_i}$ composed of Euler rotation angles ($\theta, \phi, \psi$) we get

$$S_{W}^{H_i}(\theta, \phi, \psi) =$$

27
\[
\begin{bmatrix}
\cos(\phi) \cos(\psi) & \sin(\theta) \sin(\phi) \cos(\psi) - \cos(\theta) \sin(\psi) & \sin(\theta) \sin(\psi) + \cos(\theta) \sin(\phi) \cos(\psi) \\
\cos(\phi) \sin(\psi) & \cos(\theta) \cos(\psi) + \sin(\theta) \sin(\phi) \sin(\psi) & \cos(\theta) \sin(\phi) \sin(\psi) - \sin(\theta) \cos(\psi) \\
-\sin(\phi) & \sin(\theta) \cos(\phi) & \cos(\theta) \cos(\phi)
\end{bmatrix}
\]

(2.80)

2.7.1 Analytical Decomposition of Projection Matrix

We denote known quantities in bold letters and sign ambiguities with a “∗” appended to their known absolute value. We begin by finding the sign of \( r_{33} \) in \( R_{in}^S \) as follows. Since rotation between lens and sensor is usually \( \leq \frac{\pi}{2} \) for real cameras, we can deduce that \( r_{33} = \cos(\rho) \cos(\sigma) \geq 0 \) from Equation 2.78. Now, let

\[
D = \alpha r_{31} t_x + \alpha r_{32} t_y + r_{33} (t_z - a_n)
\]

(2.81)

denote the denominator in Equations 2.58-2.68. From Equations 2.64-2.66 we have

\[
D^2 = \frac{\alpha^2 + (1 - \alpha^2)r_{33}^2}{Q_7^2 + Q_8^2 + Q_9^2}
\]

(2.82)

Next, given that \((I_0, J_0)\) are known, we can reduce Equations 2.58-2.66 to derive five equations in terms of five unknown calibration parameters \((\lambda_{gx}, \lambda_{gy}, r_{13}, r_{23}, \alpha)\) as shown next:

**Equation 1:** From Equations 2.58, 2.14, 2.60, 2.64, 2.65, 2.66 and 2.71 we have

\[
\begin{align*}
D^2(Q_3 Q_8 - Q_2 Q_9) &= \lambda_{gx} \alpha r_{33} [rs]_{21} \\
D^2(Q_1 Q_9 - Q_3 Q_7) &= \lambda_{gx} \alpha r_{33} [rs]_{22} \\
D^2(Q_2 Q_7 - Q_1 Q_8) &= \lambda_{gx} \alpha r_{33} [rs]_{23}
\end{align*}
\]

(2.83)

(2.84)

(2.85)

Squaring and adding both sides of Equations 2.85, 2.83 and 2.84 we get

\[
\frac{\lambda_{gx}^2 \alpha^2 r_{33}^2 (1 + (\alpha^2 - 1)r_{23}^2)}{D^4} = \frac{(Q_3 Q_8 - Q_2 Q_9)^2 + (Q_1 Q_9 - Q_3 Q_7)^2 + (Q_2 Q_7 - Q_1 Q_8)^2}{A}
\]

(2.86)

**Equation 2:** From Equations 2.61, 2.62, 2.63, 2.64, 2.65, 2.66 and 2.71:

\[
\begin{align*}
D^2(Q_5 Q_9 - Q_6 Q_8) &= \lambda_{gy} \alpha r_{33} [rs]_{11} \\
D^2(Q_6 Q_7 - Q_4 Q_9) &= \lambda_{gy} \alpha r_{33} [rs]_{12}
\end{align*}
\]

(2.87)

(2.88)
\[ D^2(Q_4Q_8 - Q_5Q_7) = \lambda_{yy}\alpha r_{33}[rs]_{13} \]  

(2.89)

Squaring and adding both sides of Equations 2.87, 2.88 and 2.89 we get

\[ \frac{\lambda_{yy}a^2r_{33}^2(1 + (\alpha^2 - 1)r_{13}^2)}{D^1} = \left(\frac{(Q_5Q_9 - Q_6Q_8)^2 + (Q_6Q_7 - Q_4Q_9)^2 + (Q_4Q_8 - Q_5Q_7)^2}{2}\right) \]

(2.90)

**Equation 3:** From Equations 2.58, 2.14, 2.60, 2.64, 2.65 and 2.66 we have

\[ DM_1 = \lambda_{yx}(r_{21}s_{21} - r_{22}s_{11});\ M_1 = I_0Q_7 - Q_1 \]  

(2.91)

\[ DM_2 = \lambda_{yx}(r_{21}s_{22} - r_{22}s_{12});\ M_2 = I_0Q_8 - Q_2 \]  

(2.92)

\[ DM_3 = \lambda_{yx}(r_{21}s_{23} - r_{22}s_{13});\ M_3 = I_0Q_9 - Q_3 \]  

(2.93)

Squaring and adding both sides of Equations 2.91, 2.92 and 2.93 we get

\[ \frac{\lambda_{yx}a^2(1 - r_{23}^2)}{D^2} = \frac{M_1^2 + M_2^2 + M_3^2}{M} \]  

(2.94)

**Equation 4:** From Equations 2.61-2.66 we have

\[ DP_1 = \lambda_{yy}(r_{12}s_{11} - r_{11}s_{21});\ P_1 = J_0Q_7 - Q_4 \]  

(2.95)

\[ DP_2 = \lambda_{yy}(r_{12}s_{12} - r_{11}s_{22});\ P_2 = J_0Q_8 - Q_5 \]  

(2.96)

\[ DP_3 = \lambda_{yy}(r_{12}s_{13} - r_{11}s_{23});\ P_3 = J_0Q_9 - Q_6 \]  

(2.97)

Squaring and adding both sides of Equations 2.95-2.97 we get

\[ \frac{\lambda_{yy}a^2(1 - r_{13}^2)}{D^2} = \frac{P_1^2 + P_2^2 + P_3^2}{P} \]  

(2.98)

**Equation 5:** From Equations 2.58, 2.61, 2.14 2.62, 2.60 and 2.63 we get

\[ I_0\lambda_{yy}(r_{11}s_{21} - r_{12}s_{11}) - J_0\lambda_{yx}(r_{22}s_{11} - r_{21}s_{21}) = \frac{D(I_0Q_4 - J_0Q_1)}{\alpha} \]  

(2.99)

\[ I_0\lambda_{yy}(r_{11}s_{22} - r_{12}s_{12}) - J_0\lambda_{yx}(r_{22}s_{12} - r_{21}s_{22}) = \frac{D(I_0Q_5 - J_0Q_2)}{\alpha} \]  

(2.100)

\[ I_0\lambda_{yy}(r_{11}s_{23} - r_{12}s_{13}) - J_0\lambda_{yx}(r_{22}s_{13} - r_{21}s_{23}) = \frac{D(I_0Q_6 - J_0Q_3)}{\alpha} \]  

(2.101)
Squaring and adding Equations 2.99, 2.100 and 2.101 we get

\[
\frac{\alpha^2}{D^2} \left( I_0^2 \lambda_g^2 + J_0^2 \lambda_g^2 - (I_0 \lambda_g r_{13} + J_0 \lambda_g r_{23})^2 \right) = (2.102)
\]

\[
\frac{(I_0 Q_4 - J_0 Q_1)^2 + (I_0 Q_5 - J_0 Q_2)^2 + (I_0 Q_6 - J_0 Q_3)^2}{T} (2.103)
\]

2.7.2 Solving for \( r_{13}, r_{23}, \lambda_{gx}, \lambda_{gy}, \alpha, D \)

From Section 2.7.1 we have the following six equations to be solved:

\[
\frac{\lambda_{gx} \alpha^2 r_{23}^2}{D^4} (1 + (\alpha^2 - 1) r_{23}^2) = A \tag{2.104}
\]

\[
\frac{\lambda_{gy} \alpha^2 r_{13}^2}{D^4} (1 + (\alpha^2 - 1) r_{13}^2) = B \tag{2.105}
\]

\[
\frac{\lambda_{gx} \alpha^2}{D^2} (1 - r_{23}^2) = M \tag{2.106}
\]

\[
\frac{\lambda_{gy} \alpha^2}{D^2} (1 - r_{13}^2) = P \tag{2.107}
\]

\[
\frac{\alpha^2}{D^2} \left( I_0^2 \lambda_g^2 + J_0^2 \lambda_g^2 - (I_0 \lambda_g r_{13} + J_0 \lambda_g r_{23})^2 \right) = T \tag{2.108}
\]

\[
\frac{1 - (1 - \alpha^2)(r_{13}^2 + r_{23}^2)}{D^2} = L \tag{2.109}
\]

Multiplying Equation 2.106 and Equation 2.107 we get an equation in terms of \( r_{13}, r_{23} \):

\[
\frac{\lambda_{gx} \lambda_{gy} \alpha^4}{D^4} (1 - r_{13}^2)(1 - r_{23}^2) = MP \tag{2.110}
\]

\[
\Rightarrow (\lambda_{gx} \lambda_{gy})^2 = \frac{MP D^4}{\alpha^4(1 - r_{13}^2)(1 - r_{23}^2)} \tag{2.111}
\]

Equation 2.108 can be expanded to

\[
\frac{\alpha^2}{D^2} I_0^2 \lambda_g^2 (1 - r_{13}^2) + J_0^2 \lambda_g^2 (1 - r_{23}^2) - 2I_0 J_0 \lambda_g \lambda_g r_{13} r_{23} = T \tag{2.112}
\]

Substituting Equations 2.106, 2.107 and 2.111 into Equation 2.112 we get

\[
\frac{I_0^2 P + J_0^2 M - T}{2I_0 J_0} = \pm r_{13} r_{23} \sqrt{\frac{MP}{(1 - r_{13}^2)(1 - r_{23}^2)}}
\]
\[
\begin{align*}
\Rightarrow & \quad \frac{r_{13}^2 r_{23}^2}{(1 - r_{13}^2)(1 - r_{23}^2)} = \frac{U}{M} \\
& \quad (2.113)
\end{align*}
\]

Substituting Equation 2.106 into Equation 2.104 we get

\[
\begin{align*}
& \quad \frac{Mr_{33}^2}{D^2(1 - r_{23}^2)}(1 + (\alpha^2 - 1)r_{23}^2) = A \\
& \quad \Rightarrow \frac{r_{33}^2}{D^2} \left(1 + \frac{\alpha^2 r_{23}^2}{1 - r_{23}^2}\right) = \frac{A}{M} \\
& \quad \Rightarrow \frac{1 - r_{13}^2 - r_{23}^2}{D^2} \left(1 + \frac{\alpha^2 r_{23}^2}{1 - r_{23}^2}\right) = \frac{A}{M} \\
& \quad \Rightarrow \frac{1 - r_{13}^2 - r_{23}^2 + \frac{\alpha^2 r_{23}^2(1 - r_{13}^2 - r_{23}^2)}{D^2}}{D^2(1 - r_{23}^2)} = \frac{A}{M} \\
& \quad \Rightarrow \frac{1 - r_{13}^2 - r_{23}^2 + \frac{\alpha^2 r_{23}^2}{D^2} - \frac{\alpha^2 r_{13}^2 r_{23}^2}{D^2(1 - r_{13}^2)}}{D^2(1 - r_{23}^2)} = \frac{A}{M} \\
& \quad (2.114)
\end{align*}
\]

Applying a similar analysis by substituting Equation 2.107 into Equation 2.105 we get

\[
\begin{align*}
& \quad \frac{1 - r_{13}^2 - r_{23}^2 + \frac{\alpha^2 r_{13}^2}{D^2}}{D^2} - \frac{\alpha^2 r_{13}^2 r_{23}^2}{D^2(1 - r_{13}^2)} = \frac{B}{P} \\
& \quad (2.115)
\end{align*}
\]

We observe from Equations 2.109, 2.113, 2.114 and 2.115 that we have four equations in four variables namely \((r_{13}^2, r_{23}^2, \alpha^2, D^2)\), which can be solved to get closed-form solutions (using symbolic algebra packages e.g. *Mathematica* [27]) as:

\[
\begin{align*}
& \quad r_{13}^2 = \frac{(V - 1)V \pm \sqrt{(G-L)(H-L)(V-1)^2V}}{H-L} \\
& \quad (V - 1)^2 \\
& \quad (2.116) \\
& \quad r_{23}^2 = \frac{(V - 1)V \pm \sqrt{(G-L)(H-L)(V-1)^2V}}{G-L} \\
& \quad (V - 1)^2 \\
& \quad (2.117) \\
& \quad \alpha^2 = \frac{G^2V \mp (H + L(V - 1)) \left(\sqrt{(G-L)(H-L)(V-1)^2V} \mp HV\right)}{V \left(G^2 + (H + L(V - 1))^2 + 2G(H + L(V - 1) - 2HV)\right)} \\
& \quad \mp \frac{G \left(\sqrt{(G-L)(H-L)(V-1)^2V} \pm V(L + 2HV - LV)\right)}{V \left(G^2 + (H + L(V - 1))^2 + 2G(H + L(V - 1) - 2HV)\right)} \\
& \quad (2.118)
\end{align*}
\]

Since \(r_{33}\) is positive, we get a unique solution for \(r_{33}\). Then, we solve for \(\lambda_{gx}, \lambda_{gy}, D^2\) uniquely and \(r_{13}^*, r_{23}^*\) with sign ambiguity.
Computing $D^2$

Using Equation 2.82 as $r_{33}^2, \alpha^2$ are known.

Computing $\lambda_{gx}, \lambda_{gy}$

Given $r_{13}^2, \alpha^2, D^2$ and $\lambda_{gx} > 0$ (sensor is behind the lens), Equation 2.94 can be used to compute $\lambda_{gx}$ uniquely. Similarly, using Equation 2.98, $\lambda_{gy}$ can be determined.

Determining $\text{sign}(\alpha)$

From $\alpha^2$, the magnitude of $\alpha$ is determined. The $\text{sign}(\alpha)$ can be determined by verifying the following two conditions derived from known values of $(a_x, d)$ and the constraint that $F > 0$ for a converging lens. Let $\kappa = d - a_x$. Then,

- **Condition 1**: If $\kappa < 0$, then $\text{sign}(\alpha) = \text{“+”}$, as from Figure 2.5(a), $F > 0$ only when $\alpha \in [0, 1)$.

- **Condition 2**: If $\kappa > 0$, then $\text{sign}(\alpha) = -\text{sign}(a_n)$, where $\text{sign}(a_n)$ can be obtained from lens data-sheet.

**Proof:** Using Equation 2.7 we get

$$
\frac{1}{a_n} + \frac{1}{\kappa} = \frac{1}{F}; \quad F = \frac{a_n}{1 - \alpha}
$$

They can be solved for $(a_n, F)$ in terms of $\kappa$ and $\alpha$ as:

$$
F = \kappa \left( \frac{-\alpha}{1 - \alpha} \right) \quad (2.120)
$$

$$
a_n = -\kappa * \alpha \quad \text{(pupil-centric constraint)} \quad (2.121)
$$

Equation 2.120 is a rectangular hyperbola in $(F, \alpha)$ with asymptotes $F = \kappa$ and $\alpha = 1$ as shown in Figure 2.5(a,b) for $\kappa < 0$ and $\kappa > 0$ respectively. From Figure 2.5(a), if $\kappa < 0$, then $F > 0$ only when $\alpha \in [0, 1)$. Hence, **Condition 1** follows. Similarly, from Figure 2.5(b) if $\kappa > 0$, then $F > 0$ only when $\alpha \notin [1, 0)$, which implies $\alpha$ can take both “+” and “−” sign. But, from Equation 2.121, since $\kappa > 0$, we have that $\text{sign}(\alpha)$ is opposite to $\text{sign}(a_n)$. Here, we assume that $\text{sign}(a_n)$ is known from the lens data-sheet. Hence, **Condition 2** follows.

Note that we only need $\text{sign}(a_n)$ as prior knowledge which is a much weaker constraint.
than knowing $a_n$, which in-fact we can now compute from Equation 2.121. $F$ can also be computed from Equation 2.120.

Computing the Third Row of $S_{W}^{H_1}$

From Equations 2.58-2.65 and uniquely determined $\lambda_{gx}, \lambda_{gy}, r_{33}$ and $D^2$, we get

$$s_{31} = \frac{D^2((Q_2Q_6 - Q_3Q_5) + I_0(Q_5Q_9 - Q_6Q_8) + J_0(Q_3Q_8 - Q_2Q_9))}{\lambda_{gx}\lambda_{gy}\alpha^2r_{33}}$$  \hspace{1cm} (2.122)

$$s_{32} = \frac{D^2((Q_3Q_4 - Q_1Q_6) + I_0(Q_6Q_7 - Q_4Q_9) + J_0(Q_1Q_9 - Q_3Q_7))}{\lambda_{gx}\lambda_{gy}\alpha^2r_{33}}$$  \hspace{1cm} (2.123)

$$s_{33} = \frac{D^2((Q_1Q_5 - Q_2Q_4) + I_0(Q_4Q_8 - Q_5Q_7) + J_0(Q_2Q_7 - Q_1Q_8))}{\lambda_{gx}\lambda_{gy}\alpha^2r_{33}}$$  \hspace{1cm} (2.124)

Computing the First and Second Rows of $RS_{W}^{S}$

The first row of $RS_{W}^{S}$: $(rs_{11}, rs_{12}, rs_{13})$ can be determined from Equations 2.87-2.89 as $D^2$, $r_{33}, \lambda_{gy}$ and $\alpha$ are known. Similarly, the second row of $RS_{W}^{S}$: $(rs_{21}, rs_{22}, rs_{23})$ can be determined from Equations 2.83-2.85.

Computing $R_{E_n}^{S}, S_{W}^{H_1}$

Since $R_{E_n}^{S}$ is parameterized as $(\rho, \sigma, \tau = 0)$, we have that $r_{21} = \cos(\rho)\sin(\tau) = 0$. Also, since $r_{13} = \cos(\rho)\sin(\sigma)$ and $r_{23} = -\sin(\rho)$, $R_{E_n}^{S}$ can be uniquely determined if $(r_{13}, r_{23})$ are known uniquely. But until now, we have determined $(r_{13}, r_{23})$ with sign ambiguity leading to four possible solutions of $R_{E_n}^{S}$. To solve this ambiguity, we assume that correct signs of $(r_{13}, r_{23})$ are known. Then $r_{22} = \cos(\rho), r_{23} = -\sin(\rho), r_{21} = 0$ are known. Given the
second row of $RS_W^S$ and the known third row of $S_W^{H_1}$, Equation 2.73 can be used to obtain
the following constraint,

$$rs_{21} = r_{22}s_{21} + \alpha r_{23}s_{31}$$ (2.125)

which can be uniquely solved for $s_{21}$. Forming similar linear equations for $rs_{22}$ and $rs_{23}$
in Equation 2.73, $s_{22}$ and $s_{23}$ can be determined uniquely. Thus, the 2nd row of $S_W^{H_1}$ is
determined. By taking the cross-product of the second and third rows of $S_W^{H_1}$ (computed
uniquely earlier), the first row and thus $S_W^{H_1}$ can be determined. Thus, for all four signed
solutions of $(r_{13}, r_{23})$, we get four solutions to $R_{E_n}^S$ and $S_W^{H_1}$ leading to four solutions to
$RS_W^S$ in Equation 2.126, where $\alpha$ is known uniquely. But as shown before, the first and
second rows of $RS_W^S$ have already been uniquely computed by a different constraint. Thus,
by comparing the first two rows of known unique solution and four predicted solutions of
$RS_W^S$, we find optimal $R_{E_n}^S$ and $S_W^{H_1}$ as the one with minimum Frobenius norm.

$$\begin{bmatrix}
  r_{11} & r_{12} & r_{13} \\
  0 & r_{22} & r_{23} \\
  r_{31} & r_{32} & r_{33}
\end{bmatrix}
R_{E_n}^S(\rho, \sigma, 0)
\begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & \alpha
\end{bmatrix}
\begin{bmatrix}
  s_{11} & s_{12} & s_{13} \\
  s_{21} & s_{22} & s_{23} \\
  s_{31} & s_{32} & s_{33}
\end{bmatrix}
= \begin{bmatrix}
  rs_{11} & rs_{12} & rs_{13} \\
  rs_{21} & rs_{22} & rs_{23} \\
  rs_{31} & rs_{32} & rs_{33}
\end{bmatrix}
\alpha A_p^\alpha^{-1}
S_W^{H_1}(\theta, \phi, \psi)$ (2.126)

Computing $D$

Since $R_{E_n}^S, S_W^{H_1}$ are known uniquely Equation 2.64 can be used to determine $D$.

Computing $T_W^{H_1}$

Equations 2.67, 2.68 can be solved to get $(t_x, t_y)$ uniquely and $t_z$ can then be determined
from Equation 2.82.

2.8 Finding the Center of Radial Distortion

In this section, we determine the principal point $P_p = (I_0, J_0)$, which is assumed to be
known in the analytical decomposition presented in Section 2.7.1. An accurate estimate of
$P_p$ is critical as radial distortion is centered at this point and origin of $C_S$ lies here. We
propose a computational technique to find $P_p$ based on the radial alignment constraint [14]. Specifically, different points in the image are hypothesized as valid candidates for $P_p$ and $U$ is obtained via analytical calibration (Section 2.7). Then, $U$ is used to project $i^{th}$ world point $P_{Cw}^i$ onto a frontal sensor to get sensor point $P_{CS}^i$ as well as de-warp the non-frontal observed image point $P_{CI}^i$ onto the frontal sensor as $P_{CS}'^i$. For optimal $P_p^*$, the two predicted frontal sensor points $P_{CS}^i$ and $P_{CS}'^i$ should be radially aligned about $P_p^*$. Thus, we can define the following error function over $N$ observations:

$$P_p^* = \arg\min_{P_p} \sum_{i=1}^{N} \cos^{-1} \left[ \frac{(P_{CS}^i - P_p)^T(P_{CS}'^i - P_p)}{\|P_{CS}^i - P_p\|\|P_{CS}'^i - P_p\|} \right]$$ (2.127)

which minimizes the angle between the ray $\overrightarrow{P_pP_{CS}^i}$ and $\overrightarrow{P_pP_{CS}'^i}$. $P_p$ is iteratively selected from a square window around the image center (half of image height and width) and the cost (Equation 2.127) is computed at each of these locations. Figure 2.6, shows $P_p^*$ obtained using our approach (red dots) on 11 input calibration images from a real dataset (Section 2.11). The mean of these points (blue dot) is chosen as the final optimal $P_p$. $P_p$ obtained from a laser-centering method [26] (green dot) and after a nonlinear optimization process (magenta dot) is close to analytical estimate.
dot) initialized by the laser-centering method is also shown. From Figure 2.6, mean $P'_p$ (blue dot) computed from our method is quite close to the nonlinearly optimized result.

2.9 Nonlinear Refinement

The analytical solution in Section 2.7 is used to initialize the nonlinear optimization of $U$ taking radial distortion into account. This step is optimized over 13 parameters in $U$ excluding $F$. We fix $F$ from our analytical solution. An undistorted image point $X_u = (x_u, y_u)$ radially distorts to $X_d = (x_d, y_d)$ in $C_S$ as $X_d = X_u + g(X_u, k_1, k_2)$ where $(k_1, k_2)$ are radial distortion parameters and

$$g(X_u, k_1, k_2) = \left[ x_u(k_1 r_u^2 + k_2 r_u^4) \quad y_u(k_1 r_u^2 + k_2 r_u^4) \right]$$ (2.128)

with $r_u = \sqrt{x_u^2 + y_u^2}$. The iterative nonlinear refinement of $(U, k_1, k_2)$ is initialized using analytical $U$ form Section 2.5 and $(k_1 = 0, k_2 = 0)$. Given initial calibration parameters, $i^{th}$ world point $P_{C_W}^i$ is projected onto a frontal sensor to get $X_u^i$ and then distorted to get $X_d^i$. This point is then projected back to a non-frontal sensor using $(R_{S_n}^2)^{-1}$ and pixel coordinates $P_{C_I}^i$ are obtained. The optimal $U^*$ is obtained by minimizing the sum of re-projection error between $P_{C_I}^i$ and observed image coordinates $P_{C_I}^o$ for $N$ observations using Levenberg-Marquardt optimization [13].

$$U^* = \text{argmin}_U \sum_{i=1}^{N} \| P_{C_I}^i - P_{C_I}^o(U, k_1, k_2) \|_2^2$$ (2.129)

2.10 Experiments with Synthetic Data

The analytical calibration proposed in Section 2.7.1 assumes that no image noise or distortion and exact projection of world to image points, but both are present in real images. Therefore, it's necessary to evaluate the robustness of our technique to these factors as follows.

2.10.1 Data Generation

A non-frontal camera is simulated with $U$: $(\rho, \sigma, \tau) = (2.0, 4.0, 0.0)$ degrees; $(\lambda_{px}, \lambda_{py}) = (8.4, 8.4)$ mm; $(I_0, J_0) = (240, 320)$ pixels; $(a_n, a_x, d, F) = (10.3, 28.8, 0.6, 16.2)$ mm;
\((k_1, k_2) = (0.0022, -0.000013); (\theta, \phi, \psi) = (0.1, 43.3, 0.0)\) degrees; \((t_x, t_y, t_z) = (-65.0, -41.0, 102.2)\) mm. The synthetic 3D world points \(P_{CW}\) are simulated and image points \(P_C\) are obtained (Section 2.5). Gaussian noise with 0 mean and standard deviation \(\mu = \{.01, .02, .03, .04, .05, .1, .2, \ldots, 1\}\) pixels is added to the synthesized points. Then, at each of these noise levels, analytical calibration estimates are obtained (Section 2.7.1) for all calibration parameters except \((I_0, J_0)\) which are assumed to be known (equal to \((240, 320)\) in this case). This experiment is repeated for 100 trials. Figure 2.7 shows the results for various calibration parameters. The x-axis denotes the variation of noise and y-axis plots the mean of the difference between ground truth and estimates. The vertical bars represent the standard deviation of the estimates. As can be observed, for lower noise levels and the amount of distortion we applied, the analytical estimates are close to ground truth values with low standard deviation e.g. for \(\lambda_{px}\), at 0.01 pixels noise, the standard deviation of the estimate is 0.08 pixels. But, for 1 pixel noise, the standard deviation is 16.6 pixels. Our corner detection on real data (Section 2.11) has standard deviation of \(\approx 0.011\) pixels.

![Figure 2.7: Absolute error, standard deviation vs. noise level (pixels) for various calibration parameters in \(U\) (best viewed in color).](image-url)
2.11 Experiments with Real Images

We calibrate a non-frontal camera using a checkerboard pattern and four calibration methods, namely: (A) generalized pupil-centric model initialized by our analytical solution and (B) basic initialization (C) Gennery [4], (D) decentering distortion model [13]; and based on re-projection error and undistortion accuracy show that method (A) proposed in this paper outperforms all other techniques.

2.11.1 Camera Setup

We use a AVT Marlin F-033C camera fitted with 1/2 inch Sony CCD sensor and C-mount Schneider Cinegeon 1.4/8 mm Compact lens. The sensor is intentionally tilted with respect to the lens by $\approx 5^\circ$ (Figure 2.9(b)). The captured images have a resolution of $640 \times 480$ pixels.

2.11.2 Checkerboard

A custom made precise glass checkerboard (Figure 2.9(c)) with $20 \times 20$ squares, each having dimension $5 \times 5$ mm is used to model known world points. The checkerboard corner location accuracy on the glass surface is $\pm 0.001$ mm.

2.11.3 3D Data Acquisition

A 2.5D dataset is collected by moving the CB along its surface normal (Figure 2.9(d)) by $(0, 0.5, \cdots, 4.5)$ mm (ten depths) and at each position an image is captured. The complete 2.5D dataset is collected three times and the images averaged to handle random errors in CB positioning. The camera is then moved to a different location and another 2.5D data is collected. Likewise, eleven 2.5D data sets are collected (see Figure 2.6 (right) sample images from this dataset). The CB corners are detected using MATLAB Bouquet’s toolbox [25]. The corner detection accuracy is $\approx 0.011$ pixels along both image directions and is computed using the technique from [20].
Table 2.2: Calibration results on real data

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>$s_y$</td>
<td>1.0002</td>
<td>$s_x$</td>
<td>0.9981</td>
<td>1.00</td>
<td>0.9989</td>
<td>1.00</td>
<td>1.0049</td>
</tr>
<tr>
<td></td>
<td>$\lambda_p$</td>
<td>8.742</td>
<td></td>
<td>8.738</td>
<td>8.198</td>
<td>8.128</td>
<td>8.198</td>
<td>8.154</td>
</tr>
<tr>
<td>Center of distortion</td>
<td>$I_0$</td>
<td>239.27</td>
<td>$J_0$</td>
<td>330.82</td>
<td>240.0</td>
<td>330.78</td>
<td>239.27</td>
<td>335.47</td>
</tr>
<tr>
<td></td>
<td>$k_1$</td>
<td>-</td>
<td>$k_2$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Radial distortion</td>
<td>$p_1$</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Decentering</td>
<td>$p_2$</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Non-frontal</td>
<td>$\rho$</td>
<td>-0.075</td>
<td>$\sigma$</td>
<td>3.874</td>
<td>0.113</td>
<td>0.0</td>
<td>0.246</td>
<td>-</td>
</tr>
<tr>
<td>Entrance pupil</td>
<td>$a_n$</td>
<td>6.911</td>
<td></td>
<td>6.817</td>
<td>6.5</td>
<td>6.437</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Optical focal length $F$</td>
<td></td>
<td>8.503</td>
<td></td>
<td>8.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Re-projection error</td>
<td></td>
<td>-</td>
<td></td>
<td>0.0504</td>
<td>0.0959</td>
<td>-</td>
<td>0.3807</td>
<td>0.0534</td>
</tr>
</tbody>
</table>

Table 2.3: Standard deviation of final estimates from methods A, B, C, D

<table>
<thead>
<tr>
<th>Calibration method</th>
<th>$\lambda_{px}$</th>
<th>$\lambda_{py}$</th>
<th>$I_0$</th>
<th>$J_0$</th>
<th>$\rho$</th>
<th>$\sigma$</th>
<th>$a_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.007219</td>
<td>0.003638</td>
<td>0.013715</td>
<td>0.011904</td>
<td>0.002366</td>
<td>0.003788</td>
<td>0.001428</td>
</tr>
<tr>
<td>B</td>
<td>0.016658</td>
<td>0.012721</td>
<td>0.026964</td>
<td>0.024149</td>
<td>0.005372</td>
<td>0.007684</td>
<td>0.002743</td>
</tr>
<tr>
<td>C</td>
<td>0.026557</td>
<td>0.007716</td>
<td>0.080238</td>
<td>0.078268</td>
<td>0.005930</td>
<td>0.006513</td>
<td>-</td>
</tr>
<tr>
<td>D</td>
<td>0.031462</td>
<td>0.028355</td>
<td>0.027996</td>
<td>0.030721</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

2.11.4 Corner Detection Accuracy

The corner detection is done using the technique presented and implemented in the MATLAB calibration toolbox [25]. In order to verify the accuracy of corner detection, a set of $n = 10$ gray scale images each of size $480 \times 640$ are captured in intervals of few seconds with everything else in the setup remaining the same. A total of $m = 667$ corner points are detected in each of the $n$ images. Owing to various sources of noise, the detected corners will also vary in each image. In order to estimate (as we have limited data) the first- and second-order statistics of detected points, the following procedure is applied. Let a corner point $j$ in image $i$ be denoted as $p_{ij} = (x_{ij}, y_{ij})$ and let $\bar{p}_i = \frac{1}{n} \sum_{j=1}^{n} p_{ij}$ denote the mean of the detected points for each detected corner. Similarly, assuming that covariance matrix is same for all the detected points, we have the covariance matrix $C$ as

$$C = \frac{1}{m(n-1)} \sum_{i=1}^{m} \sum_{j=1}^{n} (p_{ij} - \bar{p}_i)(p_{ij} - \bar{p}_i)^T \quad (2.130)$$
Figure 2.8, shows all the 667 points in 10 images along with the $3\sigma$ error ellipse computed using covariance $C$. The figure shows that close to 99.11% of points lie inside the $3\sigma$ ellipse, thus validating that the corner detection indeed follows an approximately Gaussian distribution. Also, the ellipse is almost close to a circle indicating that the $C$ is approximately diagonal with the standard deviation of point detection in $x$ and $y$ being $\sigma_x = 0.011426$ and $\sigma_y = 0.011412$ respectively.

Figure 2.8: Distribution of mean centered $667 \times 10$ detected points.

2.11.5 Re-projection Error

Table 2.2 shows the calibration results of intrinsic parameters in $U$ and the re-projection error using methods A, B, C and D from the camera and the 2.5D data captured above. Column A (left) shows the results from analytical method proposed in Section 2.7. Since, there are 11 2.5D datasets captured from different camera orientations, we have 11 analytical solutions for intrinsic parameters. We select the one which has closest $(a_n, F)$ values with those in lens data-sheet. Also, we select points close to our computed principal point for this step where distortion is expected to be least. Column A (right) shows final refined parameters. In Column B, we assume the same imaging model as Column A, but instead
use initial parameters from the lens data-sheet and assume tilt is 0. This is the default initialization. In Column C, we present the results from implementation of Gennery [4]. First, their imaging model assumes angle of incidence and exit at the entrance pupil is same which is not accurate [17]. Second, they optimize over decentering distortion which is redundant [12] and third, their initialization is default. In order to do a basic test of their calibration method, we avoided optimization over decentering distortion and initialized the sensor tilt and the principal point with our analytical initialization which is more accurate than using default initialization (Column C (left)). For other parameters we used default initialization. Column C (right) shows the results after nonlinear refinement. The obtained re-projection error is 0.38 pixels which is highest implying the calibration from method C is inaccurate. Lastly, Column D shows results of applying decentering [13] calibration. Comparing the re-projection error from last row of Table 2.2 and standard deviation of estimated parameters from Table 2.3, we observe that proposed method (A) achieves the lowest re-projection error of 0.0504 with lowest parameter variance across all methods. The sensor tilt is computed as $\approx 5.3^\circ$ and used for omnifocus imaging (Figure 2.10(a)) and depth from focus (Figure 2.10(b)).

![Figure 2.9: (a) Straight line fitting error to undistorted images using calibration parameters from technique A, B, C, D. (b, c, d) A non-frontal camera and the calibration setup (best viewed in color).](image)

2.11.6 Undistortion Error

Since accurate camera calibration would lead to accurate image undistortion using the obtained calibration parameters, we use this metric to compare various imaging model and calibration techniques. Specifically, we take the calibration results of technique A, B, C and
D and undistort 11 CB images. Then, we detect all the corners in the undistorted image using Bouguet’s MATLAB toolbox [25] and fit vertical and horizontal lines to detected corner points using orthogonal line fitting regression and compute the mean line fitting error as a metric for straightness of undistorted CB lines. Figure 2.9 shows the results for all 11 images (x-axis) with methods A, B, C, D and their refined estimates from Table 2.2. As we see, the error is least (red solid line) for majority of the images undistorted by the method A proposed in this chapter. The error (magenta dotted line) is maximum for the technique (C) [4]. The second-best performance is by decentering based approach (green dotted line).

2.11.7 Application

The sensor tilt is used for omnifocus imaging [5] and scene depth estimation [10] (Figure 2.10(a-e)).

Figure 2.10: (a) Omnifocus image [5]. (b) Depth from focus [10]. (c, d, e) 3D textured renderings of scene (best viewed in color).

2.12 Discussion

We have generalized the pupil-centric imaging for an arbitrary non-frontal sensor parameterized by two rotation angles. We have presented a set of new mappings between pupil-centric
and thin-lens model. We have derived an analytical calibration technique assuming known principal point to obtain other calibration estimates on the equivalent thin-lens model. This technique is then used to computationally estimate the correct principal point based on a radial alignment metric.
CHAPTER 3

GAUSSIAN IMAGING: LINEAR NON-FRONTAL SENSOR CALIBRATION

3.1 Introduction

In Chapter 2, we applied an optically accurate model of image formation for achieving camera calibration, namely pupil-centric model and assumed a tilted image sensor to do calibration. In this chapter, we relax the assumption of pupil-centric imaging, while still keeping the non-frontal sensor model and derive the method for camera calibration using collinearity constraint. This analysis may be useful for cameras whose entrance and exit pupil centers coincide with the front and back principal planes. Thus, we focus on Gaussian imaging model of image formation, a tilted sensor model and only radial distortion as the governing factors for image formation. We have the image formation and calibration setting as shown in Figure 3.1.

3.2 Previous Work

The earliest work on linear frontal camera calibration can be traced back to the DLT method [18], which used the collinearity of the 3D points and their 2D image projections to formulate a set of linear equations which could be stacked together to form the matrix equation $Ax = b$. The elements of $x$ consist of five intrinsic parameters, namely focal length, image scale, two coordinates of principal point and skew between image axes and six extrinsic parameters, namely three translation and three rotation parameters. Thus there are 11 parameters in total.

It can be noted that the vector $x$ is nothing but the stacked elements of a $3 \times 3$ or $3 \times 4$ projection matrix depending on the fact that a coplanar object or a 3D object is being used as a calibration object respectively. If the projection matrix is of the size $3 \times 4$, which allows for extraction of 11 camera calibration parameters with one being a scale parameter, a number of decomposition techniques have been proposed. See [20] for a number of references. Thus,
using a 3D object for calibration allows for extraction of all the 11 calibration parameters. Our work for non-frontal camera calibration falls in this category, where a 3D object is used as calibration object leading to 14 calibration parameters (11 + 3 rotation angle parameters of sensor tilt) encoded in a $3 \times 4$ projection matrix. In the case of 2D calibration, where multiple images of a 2D calibration target are used for calibration, the projection matrix is of the form $3 \times 3$, which allows for extraction of only eight calibration parameters taking scale into account, under the assumption that any set of three other parameters are already known or calibrated from other techniques. A number of different techniques have been proposed by [20] to enable decomposition in such a situation. In addition to such decomposition techniques for DLT matrices, there have been other methods of linear calibration which do not use DLT based formulation. Tsai [14], used the parallelism between the line joining principal point on the image plane with the corresponding distorted and undistorted image points and the line joining the 3D world point to its projection on the optical axis to define a constraint called the radial alignment constraint. This constraint allowed for a linear solution to a number of calibration parameters without the formulation of any collinearity based constraints. But, only radial distortion was considered in the calibration procedure. Weng et al. [15], proposed a decomposition technique for linear calibration and then applied an alternating minimization technique over calibration parameters and the distortion coefficients. More recently, Zhang [16] and Sturm and Maybank [28] have used homography based techniques on multiple images of 2D calibration targets to formulate constraints between the calibration parameters, which are then solved by specific decomposition techniques. In non-frontal camera calibration Louhichi et al. [29, 30] have proposed a technique.

Figure 3.1: Coordinate systems for non-frontal camera calibration.
but they do not talk about the parameterization of the rotation matrix nor do they suggest a good parameter initialization method. And incorrect initializations may lead to inaccurate calibration. Decomposition techniques based on collinearity and coplanarity conditions have also been used for pose estimation in a generic camera calibration setting [31] and for geolocalization [32]. Next, we present the various coordinate systems required for calibration in Gaussian imaging setting.

3.3 Modeling Non-Frontal Camera Calibration

For non-frontal camera calibration, the four coordinate systems are defined (see Figure 3.1) as,

1. World coordinate system \((C_W)\): A coordinate system in which the known 3D control points are defined.

2. Lens coordinate system \((C_L)\): A coordinate system attached to the optic center of the camera such that the \(z\)-axis of this coordinate system coincides with the optical axis of the camera. This axis intersects the CCD sensor at some location \(S\).

3. Sensor coordinate system \((C_S)\): A coordinate system whose origin is located at \(S\) on the sensor plane and the plane formed by its \(x\)- and \(y\)-coordinate axes lies on the CCD sensor plane be at right angles.

4. Image coordinate system \((C_I)\): A coordinate system in which the observed image points corresponding to known 3D points are defined.

Given a 3D point \(P_w\) defined in \(C_W\), and its corresponding image point \(P_i\) defined in \(C_I\), the mapping between these two points can be defined using a series of unknown transformations between all the four coordinate systems. These unknown transformations are in turn related to the unknown camera calibration parameters. The set of unknown calibration parameters will be denoted as \(U\). Thus, formulating the transformation between various coordinate axes is the first step toward deriving the unknown camera calibration parameters.

3.3.1 Transformation from \(C_W\) to \(C_L\)

Let a known 3D point in \(C_W\) be denoted as \(P_w = (X_w, Y_w, Z_w)\). In the lens coordinate system \(C_L\), this point is denoted as \(P_l = (x_l, y_l, z_l)\) and is related to \(P_w\) via the following
rotation and translation transformation relating corresponding homogeneous coordinates.

\[
\begin{bmatrix}
x_l \\
y_l \\
z_l \\
1
\end{bmatrix}
= T^L_W S^L_W
\begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix}
\]  

(3.1)

where \(T^L_W\) is the unknown 4 \times 4 translation matrix from \(C_W\) to \(C_L\) given as

\[
T^L_W = \begin{bmatrix}
1 & 0 & 0 & t_x \\
0 & 1 & 0 & t_y \\
0 & 0 & 1 & t_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(3.2)

and \(S^L_W\) is the unknown 4 \times 4 rotation matrix from \(C_W\) to \(C_L\) given as

\[
S^L_W = \begin{bmatrix}
s_{11} & s_{12} & s_{13} & 0 \\
0 & s_{22} & s_{23} & 0 \\
s_{31} & s_{32} & s_{33} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(3.3)

Thus Equation 3.1 can be expanded in terms of Equation 3.2 and Equation 3.3 to denote the world point \(P_w\) in \(C_L\) as:

\[
\begin{bmatrix}
x_l \\
y_l \\
z_l \\
1
\end{bmatrix}
= \begin{bmatrix}
s_{11}X_w + s_{12}Y_w + s_{13}Z_w + t_x \\
s_{21}X_w + s_{22}Y_w + s_{23}Z_w + t_y \\
s_{31}X_w + s_{32}Y_w + s_{33}Z_w + t_z \\
1
\end{bmatrix}
\]  

(3.4)

The set of unknown calibration parameters is

\[U = \{(s_{ij} : 1 \leq (i, j) \leq 3), t_x, t_y, t_z\}\]  

(3.5)
3.3.2 Transformation from $C_L$ to $C_S$

Let $R_L^S$ be the unknown $4 \times 4$ rotational transformation between $C_L$ and $C_S$ and given as

$$R_L^S = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$  \hspace{1cm} (3.6)$$

In conventional frontal calibration, $R_L^S$ is assumed to be a $4 \times 4$ identity matrix, whereas for a non-frontal calibration two more angle variables are added to the set of calibration parameters $U$. Let $\lambda$ be the amount of translation along negative $z$-axis of $C_L$ where the physical optic axis intersects the image sensor plane at $S$. We assume that the origin of $C_S$ lies at $S$. Thus, the translation transformation between $C_L$ and $C_S$ can be described by $T_L^S$ as

$$T_L^S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \lambda \\ 0 & 0 & 0 & 1 \end{bmatrix}$$  \hspace{1cm} (3.7)$$

Using Equations 3.4, 3.6 and 3.7 the world point $P_w$ can now be expressed in $C_S$ as $P_s = (x_s, y_s, z_s)$ given as:

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = R_L^S T_L^S \begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \lambda \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l + \lambda \\ 1 \end{bmatrix}$$
The updated set of calibration parameters \( U \) is
\[
U = \{ s_{ij} : 1 \leq (i, j) \leq 3, t_x, t_y, t_z, r_{ij} : 1 \leq (i, j) \leq 3, \lambda \} \tag{3.9}
\]

The optic center of the lens lies at the origin of \( C_L \) and can be expressed in \( C_S \) as \( P_{os} = (x_{os}, y_{os}, z_{os}) \) where,
\[
\begin{bmatrix}
x_{os} \\
y_{os} \\
z_{os} \\
1
\end{bmatrix}
= R^S_L T^S_L \begin{bmatrix}
0 \\
0 \\
0 \\
1
\end{bmatrix}
\]
\[
= \begin{bmatrix}
r_{11} & r_{12} & r_{13} & 0 \\
r_{21} & r_{22} & r_{23} & 0 \\
r_{31} & r_{32} & r_{33} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & \lambda \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
0 \\
1
\end{bmatrix}
\]
\[
= \begin{bmatrix}
r_{13} \lambda \\
r_{23} \lambda \\
r_{33} \lambda \\
1
\end{bmatrix} \tag{3.10}
\]

In the next section, we derive the projection equations for this setting.

### 3.3.3 Linear Projection Equations

Assuming no distortion and noise and under collinearity constraint, the world point \( P_w \) and can now be projected onto the image sensor. We take a coordinate geometry approach to derive this projection.

Let us consider the equation of a line in three dimensions connecting two points \( P_1 = (x_1, y_1, z_1) \) and \( P_2 = (x_2, y_2, z_2) \) as
\[
\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} \tag{3.11}
\]
If the two points $P_1$ and $P_2$ correspond to the known 3D world point ($P_s$) and the optic center of the lens ($P_{os}$) (all being expressed in $C_S$) respectively, then Equation 3.11 denotes a ray of light which originates at the 3D point, passes through the optic center and intersects the image sensor on a point denoted as $P_{nf}$ (see Figure 3.1). $P_{nf}$ is the image of $P_s$. Let $P_{nf} = (x_{nf}, y_{nf})$. Then, by substituting $z = 0$ in Equation 3.11 we get

$$\frac{x_{nf} - x_s}{x_{os} - x_s} = \frac{y_{nf} - y_s}{y_{os} - y_s} = \frac{0 - z_s}{z_{os} - z_s}$$

(3.12)

from which the location of $P_{nf}$ on the non-frontal sensor can be computed as:

$$\begin{bmatrix} x_{nf} \\ y_{nf} \end{bmatrix} = \begin{bmatrix} \frac{z_{os} x_s - x_s z_{os}}{z_{os} - z_s} \\ \frac{z_{os} y_s - y_s z_{os}}{z_{os} - z_s} \end{bmatrix}$$

(3.13)

Simplifying $x_{nf}$ using Equations 3.8 and 3.10, we get

$$x_{nf} = \frac{[r_{31} x_t + r_{32} y_t + r_{33} z_t + r_{33} \lambda][r_{13} \lambda] - [r_{11} x_t + r_{12} y_t + r_{13} z_t + r_{13} \lambda][r_{33} \lambda]}{[r_{31} x_t + r_{32} y_t + r_{33} z_t + r_{33} \lambda] - [r_{33} \lambda]}$$

$$= \frac{\lambda[r_{13} r_{31} - r_{11} r_{33}] x_t + \lambda[r_{13} r_{32} - r_{12} r_{33}] y_t}{[r_{31} x_t + r_{32} y_t + r_{33} z_t]}$$

$$= -\frac{\lambda[r_{22} x_t - r_{21} y_t]}{[r_{31} x_t + r_{32} y_t + r_{33} z_t]}$$

(3.14)

Similarly,

$$y_{nf} = \frac{[r_{31} x_t + r_{32} y_t + r_{33} z_t + r_{33} \lambda][r_{23} \lambda] - [r_{21} x_t + r_{22} y_t + r_{23} z_t + r_{23} \lambda][r_{33} \lambda]}{[r_{31} x_t + r_{32} y_t + r_{33} z_t + r_{33} \lambda] - [r_{33} \lambda]}$$

$$= \frac{\lambda[r_{23} r_{31} - r_{21} r_{33}] x_t + \lambda[r_{23} r_{32} - r_{22} r_{33}] y_t}{[r_{31} x_t + r_{32} y_t + r_{33} z_t]}$$

$$= \frac{\lambda[r_{12} x_t - r_{11} y_t]}{[r_{31} x_t + r_{32} y_t + r_{33} z_t]}$$

(3.15)

### 3.3.4 Transformation from $C_S$ to $C_I$

The predicted image point $P_{nf}$ on the nonfrontal sensor are measured in metric units e.g. mm. In order to convert them to image coordinate system $C_I$ so that they can be expressed in pixels, a transformation needs to be applied to $P_{nf}$. This transformation accounts of a possible skewness of the $C_I$ coordinate system encoded in an angle parameter $\alpha$ by which the vertical axis of $C_I$ is tilted with respect to the vertical axis of $C_S$, the size of each pixel denoted as $s_x$ and $s_y$ along the axes of $C_I$ and the location of the origin of $C_S$ in $C_I$ denoted
as \((I_0, J_0)\). Thus, the image point \(P_p = [I_p, J_p]^T\) in \(C_I\) corresponding to point \(P_{nf}\) due to the
world point \(P_w\) can be given as

\[
\begin{bmatrix}
I_p \\
J_p
\end{bmatrix} = \begin{bmatrix}
\frac{x_{nf} - y_{nf} \tan(\alpha)}{s_x} - I_0 \\
\frac{y_{nf} \sec(\alpha)}{s_y} - J_0
\end{bmatrix}
\]

(3.16)

where \(x_{nf}\) and \(y_{nf}\) are given by Equations 3.14 and 3.15. If it is assumed that there is no
distortion and noise, then the projected image coordinates \(P_p\) will correspond exactly to the
observed image coordinate \(P_i = (I, J)\). This allows us to further simplify each of the two
equations in Equation 3.16 in terms of given observations of image coordinates \(P_i\) and world
points \(P_w\) as follows. The first equation is

\[
I + I_0 = -\frac{\lambda}{s_x} \frac{[(r_{22}x_l - r_{21}y_l) + (r_{12}x_l - r_{11}y_l) \tan \alpha]}{r_{31}x_l + r_{32}y_l + r_{33}z_l}
\]

\[= [I + I_0][r_{31}x_l + r_{32}y_l + r_{33}z_l] = -\lambda x[(r_{22} + r_{12} \tan \alpha)x_l - (r_{21} + r_{11} \tan \alpha)y_l] \]

(3.17)

\[= [I r_{31} + I_0 r_{31} + \lambda x (r_{22} + r_{12} \tan \alpha)]x_l + [I r_{32} + I_0 r_{32} - \lambda x (r_{21} + r_{11} \tan \alpha)]y_l \]

Simplifying term A above using Equation 3.4,

\[= [I r_{31} + I_0 r_{31} + \lambda x (r_{22} + r_{12} \tan \alpha)][s_{11}X_w + s_{12}Y_w + s_{13}Z_w + t_x] \]

Simplifying term B above using Equation 3.4,

\[= [I r_{32} + I_0 r_{32} - \lambda x (r_{21} + r_{11} \tan \alpha)][s_{21}X_w + s_{22}Y_w + s_{23}Z_w + t_y] \]
\[ IZ_w[r_{32}s_{23}] + Z_w[I_0 r_{32}s_{23} - \lambda_x(r_{21} + r_{11}\tan \alpha)s_{23}] + \\
I[r_{32}t_y] + [I_0 r_{32}t_y - \lambda_x(r_{21} + r_{11}\tan \alpha)t_y] \]  
(3.19)

Simplifying term C above using Equation 3.4,

\[ \implies [I r_{33} + I_0 r_{33}][s_{31}X_w + s_{32}Y_w + s_{33}Z_w + t_z] \]

\[ \implies IX_w[r_{33}s_{31}] + X_w[I_0 r_{33}s_{31}] + IY_w[r_{33}s_{32}] + Y_w[I_0 r_{33}s_{32}] + \\
IZ_w[r_{33}s_{33}] + Z_w[I_0 r_{33}s_{33}] + I[r_{33}t_z] + [I_0 r_{33}t_z] \]  
(3.20)

Combining Equations 3.18, 3.19 and 3.20, we get a simplified version of Equation 3.17 as

\[ X_w[I_0(r_{31}s_{11} + r_{32}s_{21} + r_{33}s_{31}) + \lambda_x((r_{22}s_{11} - r_{21}s_{21}) + \tan \alpha(r_{12}s_{11} - r_{11}s_{21}))) + \\
Y_w[I_0(r_{31}s_{12} + r_{32}s_{22} + r_{33}s_{32}) + \lambda_x((r_{22}s_{12} - r_{21}s_{22}) + \tan \alpha(r_{12}s_{12} - r_{11}s_{22}))) + \\
Z_w[I_0(r_{31}s_{13} + r_{32}s_{23} + r_{33}s_{33}) + \lambda_x((r_{22}s_{13} - r_{21}s_{23}) + \tan \alpha(r_{12}s_{13} - r_{11}s_{23}))) + \\
IX_w[r_{31}s_{11} + r_{32}s_{21} + r_{33}s_{31}] + \\
Y_w[r_{31}s_{12} + r_{32}s_{22} + r_{33}s_{32}] + \\
IZ_w[r_{31}s_{13} + r_{32}s_{23} + r_{33}s_{33}] + \\
I[r_{31}t_x + r_{32}t_y + r_{33}t_z] + \\
[I_0(r_{31}t_x + r_{32}t_y + r_{33}t_z) + \lambda_x((r_{22} + r_{12}\tan \alpha)t_x - (r_{21} + r_{11}\tan \alpha)t_y)] = 0 \]  
(3.21)

The second equation relating the y coordinates in Equation 3.16 can be simplified as follows:

\[ J + J_0 = \frac{\lambda_y[(r_{12}x_l - r_{11}y_l)] \sec \alpha}{[r_{31}x_l + r_{32}y_l + r_{33}z_l]} \]

\[ \implies [J + J_0][r_{31}x_l + r_{32}y_l + r_{33}z_l] = \lambda_y[(r_{12} \sec \alpha)x_l - (r_{11} \sec \alpha)y_l] \left( \text{let } \lambda_y = \frac{\lambda}{s_y} \right) \]

\[ \implies [Jr_{31} + J_0 r_{31} - \lambda_y r_{12} \sec \alpha]x_l + [Jr_{32} + J_0 r_{32} + \lambda_y r_{11} \sec \alpha]y_l + [Jr_{33} + J_0 r_{33}]z_l = 0 \]  
(3.22)

Simplifying term A above using Equation 3.4,

\[ \implies [Jr_{31} + J_0 r_{31} - \lambda_y r_{12} \sec \alpha][s_{11}X_w + s_{12}Y_w + s_{13}Z_w + t_x] \]

\[ \implies JX_w[r_{31}s_{11}] + X_w[J_0 r_{31}s_{11} - \lambda_y r_{12}s_{11} \sec \alpha] + \\
JY_w[r_{31}s_{12}] + Y_w[J_0 r_{31}s_{12} - \lambda_y r_{12}s_{12} \sec \alpha] + \\
JZ_w[r_{31}s_{13}] + Z_w[J_0 r_{31}s_{13} - \lambda_y r_{12}s_{13} \sec \alpha] + \\
J[r_{31}t_x] + [J_0 r_{31}t_x] \]

52
Simplifying term B above using Equation 3.4,

\[ \implies [Jr_{32} + J_0r_{32} + \lambda_y r_{11} \sec \alpha] [s_{21} X_w + s_{22} Y_w + s_{23} Z_w + t_y] \]

\[ \implies JX_w [r_{32} s_{21}] + X_w [J_0 r_{32} s_{21} + \lambda_y r_{11} s_{21} \sec \alpha] + \\
JY_w [r_{32} s_{22}] + Y_w [J_0 r_{32} s_{22} + \lambda_y r_{11} s_{22} \sec \alpha] + \\
JZ_w [r_{32} s_{23}] + Z_w [J_0 r_{32} s_{23} + \lambda_y r_{11} s_{23} \sec \alpha] + \\
J[r_{32} t_y] + [J_0 r_{32} t_y + \lambda_y r_{11} t_y \sec \alpha] \] (3.24)

Simplifying term C above using Equation 3.4,

\[ \implies [Jr_{33} + J_0r_{33}] [s_{31} X_w + s_{32} Y_w + s_{33} Z_w + t_z] \]

\[ \implies JX_w [r_{33} s_{31}] + X_w [J_0 r_{33} s_{31}] + JY_w [r_{33} s_{32}] + Y_w [J_0 r_{33} s_{32}] + \\
JZ_w [r_{33} s_{33}] + Z_w [J_0 r_{33} s_{33}] + J [r_{33} t_z] + [J_0 r_{33} t_z] \] (3.25)

Combining Equations 3.23, 3.24 and 3.25, we get a simplified version of Equation 3.22 as

\[ X_w [J_0 (r_{31} s_{11} + r_{32} s_{21} + r_{33} s_{31}) + \lambda_y \sec \alpha (r_{11} s_{21} - r_{12} s_{11})] + \\
Y_w [J_0 (r_{31} s_{12} + r_{32} s_{22} + r_{33} s_{32}) + \lambda_y \sec \alpha (r_{11} s_{22} - r_{12} s_{12})] + \\
Z_w [J_0 (r_{31} s_{13} + r_{32} s_{23} + r_{33} s_{33}) + \lambda_y \sec \alpha (r_{11} s_{23} - r_{12} s_{13})] + \\
JX_w [r_{31} s_{11} + r_{32} s_{21} + r_{33} s_{31}] + \\
JY_w [r_{31} s_{12} + r_{32} s_{22} + r_{33} s_{32}] + \\
JZ_w [r_{31} s_{13} + r_{32} s_{23} + r_{33} s_{33}] + \\
J[r_{31} t_x + r_{32} t_y + r_{33} t_z] + \\
[J_0 (r_{31} t_x + r_{32} t_y + r_{33} t_z) + \lambda_y \sec \alpha (r_{11} t_y - r_{12} t_x)] = 0 \] (3.26)

Typically \( s_x \) is known from the camera manual and is assumed to be given. The set of unknown calibration parameters can now be updated to:

\[ U = \{ (s_{ij} : 2 \leq (i, j) \leq 1), t_x, t_y, t_z, \{ r_{ij} : 2 \leq (i, j) \leq 1 \}, \lambda, s_y, \alpha, I_0, J_0 \} \] (3.27)

The extrinsic rotation matrix can be encoded with three angle parameters and the sensor rotation can be encoded using two angle parameters. One of the intrinsic parameters in
\( \lambda, s_x, s_y \) needs to be assumed to fix the scale of the complete calibration system. We will assume that \( s_y \) is known to us and also modify the two calibration parameters \( \lambda, s_x \) to \( \lambda_x = \frac{\lambda}{s_x} \) and \( \lambda_y = \frac{\lambda}{s_y} \). Thus the final set of unknown calibration is of size 13 and has the following elements

\[
U = \left[ \{s_{ij} : 2 \leq (i, j) \leq 1\}, t_x, t_y, t_z, \{r_{ij} : 2 \leq (i, j) \leq 1\}, \lambda_x, \lambda_y, \alpha, I_0, J_0 \right]
\] (3.28)

The first six parameters in \( U \) are extrinsic parameters and the next seven are the intrinsic parameters of the imaging system.

### 3.4 Linear Camera Calibration Equation

This section describes the linear calibration equation and its relation to calibration parameters \( U \). Combining Equations 3.4, 3.8, 3.10 and 3.13 into Equation 3.16 gives the image pixel locations \((I, J)\) (defined in \( C_I \)) of a known world point \( P_w \) (defined in \( C_W \)) in terms of \( U \). As shown in Equations 3.21 and 3.26), each corresponding world point \( P_w \) and its image \((I, J)\), we get two equations corresponding to two coordinates of the image point. Each of the two equations in Equation 3.16 can be rearranged as a linear equation in terms of: \((X_w, Y_w, Z_w, I, J)\). Thus, if there are \( N \) world point-image point correspondences, \( 2N \) linear equations can be obtained. These equations can be stacked together to form a matrix equation of the form \( A P = 0 \) as shown in Equation 3.29, where \( A \) is a matrix of size \( 2N \times 12 \) (Equation 3.30) and \( P \) is a column vector of size \( 12 \times 1 \). Each row of \( A \) is made up of known observations of corresponding world and image points: \((X_w, Y_w, Z_w) < - > (I, J)\). Each row of \( P \), which has one element, is a nonlinear function of a subset of unknown calibration parameters in \( U \). Since \( A \) is full-rank, its rank is 12. Thus, we have the homogeneous equation

\[
A_{2N \times 12} P_{12 \times 1} = 0
\] (3.29)

The corresponding inhomogeneous linear equation can be obtained by dividing \( P_i \) by \( P_{12} \) to obtain \( A'Q = b \), where
where the elements of new vector $Q$ is defined as:

$$Q_i = \frac{P_i}{P_{12}} \forall (i = 1, \ldots, 11)$$

and $b$ is the sign reversed rightmost column of original matrix $A$ in Equation 3.29.

$$b = \begin{bmatrix}
-I_1 \\
-J_1 \\
\vdots \\
-I_N \\
-J_N
\end{bmatrix}_{2N \times 1}$$

By simplifying Equations 3.21 and 3.26, it can be shown that the elements of $Q$ are nonlinearly related to the unknown calibration parameters $U$ as:

$$Q_1 = \frac{I_0(r_{31}s_{11} + r_{32}s_{21} + r_{33}s_{31}) + \lambda_x((r_{22}s_{11} - r_{21}s_{21}) + \tan \alpha(r_{12}s_{11} - r_{11}s_{21}))}{r_{31}t_x + r_{32}t_y + r_{33}t_z}$$

$$Q_2 = \frac{I_0(r_{31}s_{12} + r_{32}s_{22} + r_{33}s_{32}) + \lambda_x((r_{22}s_{12} - r_{21}s_{22}) + \tan \alpha(r_{12}s_{12} - r_{11}s_{22}))}{r_{31}t_x + r_{32}t_y + r_{33}t_z}$$

$$Q_3 = \frac{I_0(r_{31}s_{13} + r_{32}s_{23} + r_{33}s_{33}) + \lambda_x((r_{22}s_{13} - r_{21}s_{23}) + \tan \alpha(r_{12}s_{13} - r_{11}s_{23}))}{r_{31}t_x + r_{32}t_y + r_{33}t_z}$$

$$Q_4 = \frac{J_0(r_{31}s_{11} + r_{32}s_{21} + r_{33}s_{31}) + \lambda_y \sec \alpha(r_{11}s_{21} - r_{12}s_{11})}{r_{31}t_x + r_{32}t_y + r_{33}t_z}$$
\[ Q_5 = J_0(r_{31}s_{12} + r_{32}s_{22} + r_{33}s_{32}) + \lambda_y \sec \alpha (r_{11}s_{22} - r_{12}s_{12}) \quad (3.37) \]

\[ Q_6 = J_0(r_{31}s_{13} + r_{32}s_{23} + r_{33}s_{33}) + \lambda_y \sec \alpha (r_{11}s_{23} - r_{12}s_{13}) \quad (3.38) \]

\[ Q_7 = \frac{r_{31}s_{11} + r_{32}s_{21} + r_{33}s_{31}}{r_{31}t_x + r_{32}t_y + r_{33}t_z} \quad (3.39) \]

\[ Q_8 = \frac{r_{31}s_{12} + r_{32}s_{22} + r_{33}s_{32}}{r_{31}t_x + r_{32}t_y + r_{33}t_z} \quad (3.40) \]

\[ Q_9 = \frac{r_{31}s_{13} + r_{32}s_{23} + r_{33}s_{33}}{r_{31}t_x + r_{32}t_y + r_{33}t_z} \quad (3.41) \]

\[ Q_{10} = I_0 + \frac{\lambda_x((r_{22} + r_{12} \tan \alpha)t_x - (r_{21} + r_{11} \tan \alpha)t_y)}{r_{31}t_x + r_{32}t_y + r_{33}t_z} \quad (3.42) \]

\[ Q_{11} = J_0 + \frac{\lambda_y \sec \alpha (r_{11}t_y - r_{12}t_x)}{r_{31}t_x + r_{32}t_y + r_{33}t_z} \quad (3.43) \]

where \( \lambda_x = \frac{\lambda}{s_x} \) and \( \lambda_y = \frac{\lambda}{s_y} \). Since \( \mathbf{A}' \) is full rank and known along with \( \mathbf{b} \), \( Q \) can be computed in least squares sense using SVD based techniques. Section 3.5 describe our new technique to analytically find the calibration parameters \( \mathbf{U} \) from \( Q \) given Equations 3.33-3.43.

### 3.5 Determining Calibration Parameters

#### 3.5.1 Preliminaries

**Rotation Convention**

The rotation convention when rotating the coordinate system in 3D about the \( x \)-, \( y \)- and \( z \)-axes is clockwise direction when looking towards the origin. This results in the following standard rotation matrices when rotating the \( x \)-, \( y \)- and \( z \)-axes by \( (\rho, \sigma, \tau) \) degrees respectively,

\[
R_x(\rho) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\rho) & -\sin(\rho) \\
0 & \sin(\rho) & \cos(\rho)
\end{bmatrix}
\]

\[
R_y(\sigma) = \begin{bmatrix}
\cos(\sigma) & 0 & \sin(\sigma) \\
0 & 1 & 0 \\
-\sin(\sigma) & 0 & \cos(\sigma)
\end{bmatrix}
\]
Thus any 3D rotation matrix $R_z(\tau)R_y(\sigma)R_x(\rho)$ can be written as

$$
\begin{bmatrix}
\cos(\sigma) \cos(\tau) & \sin(\sigma) \sin(\tau) \cos(\rho) - \cos(\sigma) \sin(\rho) \\
-\sin(\sigma) \sin(\tau) & \cos(\tau) \cos(\sigma) + \sin(\sigma) \sin(\rho) \\
\cos(\sigma) \sin(\tau) & \cos(\tau) \sin(\sigma) - \sin(\sigma) \cos(\rho) \\
\end{bmatrix}
$$

(3.44)

**Lemma 2** Inverse and Transpose Equivalence for Rotation Matrices: Another important relation which we use in our derivation is that for any orthonormal matrix, e.g. a rotation matrix $R$, we have $R^{-1} = R^T$. If $R = [abc; def; ghi]$, then we have the following relationship

$$
\begin{bmatrix}
(\text{ei} - hf) & (\text{ch} - bi) & (bf - ce) \\
(gf - di) & (ai - cg) & (dc - af) \\
(dh - eg) & (bg - ah) & (ae - bd) \\
\end{bmatrix} =
\begin{bmatrix}
a & d & g \\
b & e & h \\
c & f & i \\
\end{bmatrix}
$$

(3.45)

The equivalence of each element in the above equality is critical to our decomposition technique.

Defining the Combined Rotation Matrix $RS^S_W$

We denote the product of the rotation matrices relating $C_W$ to $C_L: S^L_W$ (Equation 3.3) and $C_L$ to $C_S: R^S_L$ (Equation 3.6) as $R^S_W$:

$$
RS^S_W = R^S_L S^L_W
$$

(3.46)
whose \((i, j)^{th}\) element is

\[ r_{sj} = r_{i1}s_{1j} + r_{i2}s_{2j} + r_{i3}s_{3j} \]  

(3.47)

### 3.5.2 Decomposition

It was shown in Equation 3.28, that the calibration parameters set \(U\) consists of five unknown rotation angles, two for rotation matrix \(R^S_L\) and three for rotation matrix \(S^L_W\). But in our technique, instead of extracting five rotation angles, we will extract the elements of the rotation matrices \(R^S_L\) and \(S^L_W\) and then decompose it into Euler angles to get the angles. We follow the convention that all the known quantities will be represented in bold letters for ease of following the derivation. Also, any quantity which is computed with sign ambiguity will be marked with “*”. Before explaining the technique, we need to make assumptions about some of the calibration parameters which will be trivially true for a real-world camera.

The first assumption is that the amount of rotation \(R^S_L\) between the lens coordinate system \(C_L\) and the sensor coordinate system \(C_S\) is small. Thus, letting \((\alpha, \beta, \gamma)\) denote the angular components of \(R^S_L\), it is expected that \(0 \leq (|\alpha|, |\beta|, |\gamma|) \leq \frac{\pi}{2}\) or equivalently taking cosines: \(1 \geq (\cos(\alpha), \cos(\beta), \cos(\gamma)) \geq 0\). From this condition and assuming \(R^S_L\) to be of the general form of a rotation matrix (Equation 3.44), the sign of \(r_{33}\) from Equation 3.6 can be obtained as positive,

\[ r_{33} = \cos(\beta) \cos(\alpha) \geq 0 \]  

(3.48)

The 11 Equations 3.33-3.43 relate 13 calibration parameters in the set \(U\) (Equation 3.28). Thus, at least 11 parameters can be extracted from these equations after assuming that two of them are known. In this chapter, we chose to assume that the principal point/center of radial distortion \((I_0, J_0)\) is known and can be computed by using some other technique, e.g. [33].

Now, let us denote the common denominator in Equations 3.33-3.43 as \(D\),

\[ r_{31}t_x + r_{32}t_y + r_{33}t_z = D \]  

(3.49)

Squaring and adding the left-hand side (LHS) and right-hand side (RHS) of Equations 3.39, 3.40 and 3.41 leads to

\[ Q_7^2 + Q_8^2 + Q_9^2 \]
\[
\left( r_{31}s_{11} + r_{32}s_{21} + r_{33}s_{31} \right)^2 + \left( r_{31}s_{12} + r_{32}s_{22} + r_{33}s_{32} \right)^2 + \left( r_{31}s_{13} + r_{32}s_{23} + r_{33}s_{33} \right)^2 \\
= \frac{1}{D^2} \quad \text{(using Equation 3.49 and orthonormality of third row of } RS^S \text{.)}
\]

Thus,

\[
D^* = \pm \frac{1}{\sqrt{Q_7^2 + Q_8^2 + Q_9^2}}
\] (3.50)

The sign of \( D^* \) will be resolved later as most of the later derivations will depend on \( D^2 \). Using Equation 3.41, \( rs_{33} \) is obtained as follows

\[
Q_9 = \frac{s_{13}r_{31} + s_{23}r_{32} + s_{33}r_{33}}{r_{31}t_x + r_{32}t_y + r_{33}t_z} \\
\Rightarrow Q_9 = \frac{rs_{33}}{D^*} \\
\Rightarrow rs_{33}^* = Q_9 D^*
\] (3.51)

The sign of \( rs_{33}^* \) is ambiguous due to its dependence on \( D^* \) whose sign will be derived later.

Next we determine the scale parameter \( \lambda_x \). Using Equations 3.33, 3.34, 3.39 and 3.40 and applying Lemma 2 described in Equation 3.45, the following relationship can be obtained

\[
D^2(Q_1Q_8 - Q_2Q_7) = -\lambda_x r_{33}[rs_{23} + rs_{13} \tan \alpha]
\] (3.52)

Below we show the steps of this derivation which will be critical for obtaining similar equations useful for our decomposition process.

\[
D^2(Q_1Q_8 - Q_2Q_7) \\
= \lambda_x [(r_{22}s_{11} - r_{21}s_{21})(s_{12}r_{31} + s_{22}r_{32} + s_{32}r_{33}) - (r_{22}s_{12} - r_{21}s_{22})(s_{11}r_{31} + s_{21}r_{32} + s_{31}r_{33})] \\
+ \lambda_x \tan \alpha [(r_{12}s_{11} - r_{11}s_{21})(s_{12}r_{31} + s_{22}r_{32} + s_{32}r_{33}) \\
- (r_{12}s_{12} - r_{11}s_{22})(s_{11}r_{31} + s_{21}r_{32} + s_{31}r_{33})]
\]

\[
= \lambda_x [r_{21}r_{31}(s_{11}s_{22} - s_{12}s_{21}) + r_{21}r_{33}(s_{22}s_{31} - s_{21}s_{32}) \\
+ r_{22}r_{32}(s_{11}s_{22} - s_{12}s_{21}) + r_{22}r_{33}(s_{11}s_{32} - s_{12}s_{31})] \\
+ \lambda_x \tan \alpha [r_{11}r_{31}(s_{11}s_{22} - s_{12}s_{21}) + r_{12}r_{32}(s_{11}s_{22} - s_{12}s_{21}) + r_{12}r_{33}(s_{11}s_{32} - s_{12}s_{31}) \\
+ r_{11}r_{33}(s_{22}s_{31} - s_{21}s_{32})]
\]
\[ = \lambda_x [r_{21}r_{31}s_{33} - r_{21}r_{33}s_{13} + r_{22}r_{32}s_{33} - r_{22}r_{33}s_{23}] + \lambda_x \tan \alpha [r_{11}r_{31}s_{33} + r_{12}r_{32}s_{33} - r_{12}r_{33}s_{23} - r_{11}r_{33}s_{13}] \]

(applying Lemma 2 to the elements of \( S^L_W \))

\[ = \lambda_x [s_{33}(r_{21}r_{31} + r_{22}r_{32}) - r_{21}r_{33}s_{13} - r_{22}r_{33}s_{23}] + \lambda_x \tan \alpha [s_{33}(r_{11}r_{31} + r_{12}r_{32}) - r_{12}r_{33}s_{23} - r_{11}r_{33}s_{13}] \]

(orthogonality of rows of \( R^S_L \))

\[ = -\lambda_x r_{33}[r_{21}s_{13} + r_{22}s_{23} + r_{23}s_{33}] - \lambda_x r_{33} \tan \alpha [r_{11}s_{13} + r_{12}s_{23} + r_{13}s_{33}] \]

\[ = -\lambda_x r_{33}[r_{s_{23}} + r_{s_{13}} \tan \alpha] \quad \text{(from } RS^S_W \text{ in Equation 3.46)} \]

Applying similar algebraic manipulation on Equations 3.33, 3.34, 3.39 and 3.40 we have

\[ D^2(Q_2Q_9 - Q_3Q_8) = -\lambda_x r_{33}[r_{s_{21}} + r_{s_{11}} \tan \alpha] \quad \text{(3.53)} \]

\[ D^2(Q_3Q_7 - Q_1Q_9) = -\lambda_x r_{33}[r_{s_{22}} + r_{s_{12}} \tan \alpha] \quad \text{(3.54)} \]

Squaring and adding the LHS and RHS of Equations 3.52, 3.53 and 3.54 and applying the orthonormality of second row of \( RS^S_W \), we get

\[ D^4 \left[ (Q_2Q_9 - Q_3Q_8)^2 + (Q_1Q_9 - Q_3Q_7)^2 + (Q_1Q_8 - Q_2Q_7)^2 \right] \]

\[ = \lambda_x^2 r_{33}^2 [r_{s_{21}}^2 + r_{s_{22}}^2 + r_{s_{23}}^2] + \tan^2 \alpha (r_{s_{11}}^2 + r_{s_{12}}^2 + r_{s_{13}}^2)] \]

\[ = \lambda_x^2 r_{33}^2 [1 + \tan^2 \alpha] \]

\[ \implies D^4 A = \lambda_x^2 r_{33}^2 \sec^2 \alpha \]

\[ \implies r_{s_{33}}^2 = \frac{D^4 A}{\lambda_x^2 \sec^2 \alpha} \quad \text{(3.55)} \]

where, \( A \) is known because \( Q \) is known. Similarly using Equations 3.36, 3.37, 3.38, 3.39, 3.40 and 3.41, we have

\[ D^2(Q_6Q_8 - Q_5Q_9) = -\lambda_y r_{33} r_{s_{11}} \sec \alpha \quad \text{(3.56)} \]

\[ D^2(Q_4Q_9 - Q_6Q_7) = -\lambda_y r_{33} r_{s_{12}} \sec \alpha \quad \text{(3.57)} \]
\[ D^2(Q_5Q_7 - Q_4Q_8) = -\lambda_y r_{33} r_{13} \sec \alpha \quad (3.58) \]

Squaring and adding the LHS and RHS of Equations 3.56, 3.57 and 3.58 and applying the orthonormality of the first row of \( RS_W^s \) yields

\[
D^4 \left[ (Q_5Q_9 - Q_6Q_8)^2 + (Q_4Q_9 - Q_6Q_7)^2 + (Q_4Q_8 - Q_5Q_7)^2 \right] \\
= \lambda_y^2 r_{33}^2 \sec^2 \alpha [r_{11}^2 + r_{12}^2 + r_{13}^2] \\
\implies D^4 B = \lambda_y^2 r_{33}^2 \sec^2 \alpha 
\]

where \( B \) is known because \( Q \) is known. Eliminating \( r_{33} \) and \( D^4 \) from Equations 3.55 and 3.59 by taking ratios on both sides gives us a relation between \( \lambda_x \) and \( \lambda_y \) in terms of known values \( A \) and \( B \). Also note that \( \lambda_y \neq 0 \) since that would imply \( \lambda = 0 \) which is not possible for real-world imaging sensor. Thus, we have

\[
\frac{\lambda_x^2}{\lambda_y^2} = \frac{A}{B} \\
\implies \lambda_y^2 = \frac{B \lambda_x^2}{A} 
\]

Computing \( \lambda_x \)

Using Equations 3.33-3.41, we have

\[
D^*(Q_1 - I_0Q_7) = \lambda_x [r_{22}s_{11} - r_{21}s_{21}] + \tan \alpha (r_{12}s_{11} - r_{11}s_{21}) 
\]

\[
D^*(Q_2 - I_0Q_8) = \lambda_x [r_{22}s_{12} - r_{21}s_{22}] + \tan \alpha (r_{12}s_{12} - r_{11}s_{22}) 
\]

\[
D^*(Q_3 - I_0Q_9) = \lambda_x [r_{22}s_{13} - r_{21}s_{23}] + \tan \alpha (r_{12}s_{13} - r_{11}s_{23}) 
\]

\[
D^*(Q_4 - J_0Q_7) = \lambda_y \sec \alpha [(r_{11}s_{21} - r_{12}s_{11})] 
\]

\[
D^*(Q_5 - J_0Q_8) = \lambda_y \sec \alpha [(r_{11}s_{22} - r_{12}s_{12})] 
\]

\[
D^*(Q_6 - J_0Q_9) = \lambda_y \sec \alpha [(r_{11}s_{23} - r_{12}s_{13})] 
\]
where \(M_1, M_2, M_3, P_1, P_2, P_3\) are known. Combining Equations 3.61 and 3.64, we get

\[
\frac{M_1}{\lambda_x} + \frac{P_1 \sin \alpha}{\lambda_y} = r_{22}s_{11} - r_{21}s_{21}
\]  

(3.67)

Similarly combining Equations 3.62, 3.65, 3.63 and 3.66, we get

\[
\frac{M_2}{\lambda_x} + \frac{P_2 \sin \alpha}{\lambda_y} = r_{22}s_{12} - r_{21}s_{22}
\]  

(3.68)

\[
\frac{M_3}{\lambda_x} + \frac{P_3 \sin \alpha}{\lambda_y} = r_{22}s_{13} - r_{21}s_{23}
\]  

(3.69)

Squaring and adding the LHS and RHS of Equations 3.67, 3.68 and 3.69 and applying orthonormality of the second row of \(R_{LS}^L\) leads to

\[
\frac{M^2}{\lambda_x^2} + \frac{P^2}{\lambda_y^2} + \frac{MP}{\lambda_x \lambda_y} = r_{21}^2 + r_{22}^2
\]  

(3.70)

\[
\Rightarrow 1 - r_{23}^2 = \frac{M}{\lambda_x^2} + \frac{P \sin^2 \alpha}{\lambda_y^2} + 2 \sin \alpha \frac{MP}{\lambda_x \lambda_y}
\]  

(3.71)

where \(M, P, MP\) are known as they are derived from known quantities \(Q, D^2, I_0, J_0\).

Similarly, squaring and adding the LHS and RHS Equations 3.64, 3.65 and 3.66 and applying orthonormality of the first row of \(R_{LS}^L\) gives

\[
r_{11}^2 + r_{12}^2 = \frac{P}{\lambda_y^2} \cos^2 \alpha
\]  

(3.72)

\[
\Rightarrow 1 - r_{13}^2 = \frac{P \cos^2 \alpha}{\lambda_y^2}
\]  

(3.73)

where \(P\) is known as it is derived from known quantities \(Q, D^2, I_0, J_0\).

Adding both sides of Equations 3.71 and 3.73 and from the fact that \(r_{13}^2 + r_{23}^2 + r_{33}^2 = 1\), we get

\[
r_{33}^2 = \frac{M}{\lambda_x^2} + \frac{P}{\lambda_y^2} + 2 \sin \alpha \frac{MP}{\lambda_x \lambda_y} - 1
\]  

Substituting for \(r_{33}\) in terms of \(\lambda_x\) using Equation 3.55 and for \(\lambda_y\) in terms of \(\lambda_x\) using
Equation 3.60, we get
\[
\frac{D^4A \cos^2 \alpha}{\lambda_x^2} = \frac{PA}{B \lambda_x^2} + \frac{M}{\lambda_x^2} + 2 \sin \alpha \frac{MP}{\lambda_x^2} \sqrt{\frac{A}{B}} - 1
\]
from which \(\lambda_x\), which is positive, can be computed in terms of knowns \(M, P, A, B, D\) as
\[
\lambda_x = \sqrt{\frac{PA}{B} + M + 2 \sin \alpha MP \sqrt{\frac{A}{B} - D^4A \cos^2 \alpha}} \quad (3.74)
\]
Computing \(\lambda_y\)

Thus, \(\lambda_y\) can be computed from Equation 3.60 and 3.74,
\[
\lambda_y^2 = \frac{B}{A} \lambda_x^2
\]
\[
\lambda_y = \sqrt{\frac{MB}{A} + P + 2 \sin \alpha MP \sqrt{\frac{B}{A}} - D^4B \cos^2 \alpha} \quad (3.75)
\]
Computing \(r_{13}, r_{23}, r_{33}\)

Using \(\lambda_x\) in Equation 3.55 and the fact that \(r_{33} \geq 0\) from Equation 3.48, \(r_{33}\) can be computed as:
\[
r_{33} = \sqrt{\frac{AD^4}{\lambda_x^2}} \quad (3.76)
\]
Using Equations 3.71 and 3.73 and the obtained values of \(\lambda_x\) and \(\lambda_y\), \(r_{13}^*\) and \(r_{23}^*\) can be also be obtained but with sign ambiguity. This ambiguity will later be resolved using Equation 3.92.

Computing \(r_{22}\)

If the rotation between lens and sensor about the \(z\)-axis is assumed to be 0, due to the rotational symmetry of the lens, then \(\tau = 0\). The rotation matrix \(R_L^S\) becomes:
\[
R_L^S(\rho, \sigma, 0) = \begin{bmatrix}
\cos(\sigma) & \sin(\rho) \sin(\sigma) & \cos(\rho) \sin(\sigma) \\
0 & \cos(\rho) & -\sin(\rho) \\
-\sin(\sigma) & \sin(\rho) \cos(\sigma) & \cos(\rho) \cos(\sigma)
\end{bmatrix} \quad (3.77)
\]
Comparing $R^S_L$ from Equation 3.77 and $R^S_L = (r_{ij} : 1 \leq (i, j) \leq 3)$ in Equation 3.6, we get

$$r_{21} = 0$$

(3.78)

Since, $r_{23}$ is known and $r_{22} = \cos(\rho)$, which is positive, we get

$$r_{22} = \sqrt{1 - r_{23}^2}$$

(3.79)

Computing $r_{11}$

From Equations 3.77, 3.76 and 3.79 we have

$$r_{11} = \frac{r_{33}}{r_{22}}$$

(3.80)

Computing $s_{31}, s_{32}, s_{33}$

Using Equations 3.33-3.40, we get

$$s_{33} = \frac{D^2(Q_1Q_5 - Q_2Q_4) + I_0D^2(Q_4Q_8 - Q_5Q_7) + J_0D^2(Q_2Q_7 - Q_1Q_8)}{\lambda_x \lambda_y r_{33} \sec \alpha}$$

(3.81)

$$s_{32} = \frac{D^2(Q_3Q_4 - Q_1Q_6) + I_0D^2(Q_6Q_7 - Q_4Q_9) + J_0D^2(Q_1Q_9 - Q_3Q_7)}{\lambda_x \lambda_y r_{33} \sec \alpha}$$

(3.82)

$$s_{31} = \frac{D^2(Q_2Q_6 - Q_3Q_5) + I_0D^2(Q_5Q_9 - Q_6Q_8) + J_0D^2(Q_3Q_8 - Q_2Q_9)}{\lambda_x \lambda_y r_{33} \sec \alpha}$$

(3.83)

Computing $r_{s11}, r_{s12}, r_{s13}, r_{s21}, r_{s22}, r_{s23}$

Using Equations 3.56, 3.57 and 3.58 and the known values of $\lambda_y$ (Equation 3.75) and $r_{33}$ (Equation 3.76), we obtain the first row of $R^S_W$ as:

$$r_{s11} = \frac{D^2(Q_5Q_9 - Q_6Q_8)}{\lambda_y r_{33} \sec \alpha}$$

(3.84)

$$r_{s12} = \frac{D^2(Q_6Q_7 - Q_4Q_9)}{\lambda_y r_{33} \sec \alpha}$$

(3.85)

$$r_{s13} = \frac{D^2(Q_4Q_8 - Q_5Q_7)}{\lambda_y r_{33} \sec \alpha}$$

(3.86)
Similarly, from Equations 3.52, 3.53 and 3.54, we get the second row of $RS_W^S$,

$$rs_{21} = \frac{D^2(Q_2Q_9 - Q_3Q_8)}{-\lambda_xr_{33}}$$  (3.87)
$$rs_{22} = \frac{D^2(Q_1Q_9 - Q_3Q_7)}{\lambda_xr_{33}}$$  (3.88)
$$rs_{23} = \frac{D^2(Q_1Q_8 - Q_2Q_7)}{-\lambda_xr_{33}}$$  (3.89)

The third row of $RS_W^S$ can be determined without sign ambiguity, if the first two rows are known from Equations 3.84-3.89. Thus, the rotation transformation from $C_W$ to $C_S$ is known without any ambiguity. The sign of $D$ can be computed as follows. From Equation 3.51, we have $rs^*_{33} = Q_9D^*$ but $rs_{33}$ has also been determined separately without sign ambiguity, therefore we have for $sign(D)$

$$sign(D) = \frac{sign(rs_{33})}{sign(Q_9)}$$  (3.90)

Computing the Remaining Elements of $RS_L^S, S_W^L$

Since $RS_L^S$ is parameterized as $(\rho, \sigma, \tau = 0)$, we have that $r_{21} = \cos(\sigma)\sin(\tau) = 0$. Also, since $r_{13} = \cos(\rho)\sin(\sigma)$ and $r_{23} = -\sin(\rho)$, $RS_L^S$ can be uniquely determined if $(r_{13}, r_{23})$ are known uniquely. But until now, we have determined $(r_{13}, r_{23})$ with sign ambiguity leading to four possible solutions of $RS_L^S$. To solve this ambiguity, we assume that correct signs of $(r_{13}, r_{23})$ are known. Then $r_{22} = \cos(\rho)$, $r_{23} = -\sin(\rho)$, $r_{21} = 0$ are known. Given the second row of $RS_W^S$ and the known third row of $S_W^L$, Equation 3.47 can be used to obtain the following constraint,

$$rs_{21} = r_{22}s_{21} + r_{23}s_{31}$$  (3.91)

which can be uniquely solved for $s_{21}$. Forming similar linear equations for $rs_{22}$ and $rs_{23}$ in Equation 3.47, $s_{22}$ and $s_{23}$ can be determined uniquely. Thus, the 2nd row of $S_W^L$ is determined. By taking the cross-product of the second and third rows of $S_W^L$ (computed uniquely earlier), the first row and thus $S_W^L$ can be determined. Thus, for all four signed solutions of $(r_{13}, r_{23})$, we get four solutions to $RS_L^S$ and $S_W^L$ leading to four solutions to $RS_W^S$ in Equation 3.46. But as shown before, the first and second rows of $RS_W^S$ have already been uniquely computed by a different constraint. Thus, by comparing the first two rows of known unique solution and four predicted solutions of $RS_W^S$, we find optimal $RS_L^S$ and $S_W^L$ as
the one with minimum Frobenius norm.

\[
\begin{bmatrix}
n_{11} & n_{12} & n_{13} \\
n_{21} & n_{22} & n_{23} \\
n_{31} & n_{32} & n_{33}
\end{bmatrix}
\begin{bmatrix}
n_{11} & n_{12} & n_{13} \\
n_{21} & n_{22} & n_{23} \\
n_{31} & n_{32} & n_{33}
\end{bmatrix} =
\begin{bmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
r_{31} & r_{32} & r_{33}
\end{bmatrix}
\begin{bmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
r_{31} & r_{32} & r_{33}
\end{bmatrix}
\]

Lastly, \( R_S^S \) and \( S_L^S \) are modified to satisfy the orthonormality conditions by using the technique described in [34], and \( R_L^S = R_S^S \cdot S_L^S^{-1} \) is computed. The translation vector \( T = (tx, ty, tz) \) relating \( C_W \) and \( C_L \) given by Equation 3.2 can be computed as

\[
t_x = \frac{D\lambda_y(-\lambda_y r_{11}I_0 + r_{11}Q_{10}\lambda_y - \lambda_x J_0 r_{21} + \lambda_x Q_{11}r_{21})}{\lambda_y(-r_{21}r_{12} + r_{11}r_{22})}
\]

(3.93)

\[
t_y = \frac{D\lambda_x(-r_{22}\lambda_x J_0 - I_0\lambda_y r_{12} + Q_{10}\lambda_y r_{12} + r_{22}\lambda_x Q_{11})}{\lambda_x(-r_{21}r_{12} + r_{11}r_{22})}
\]

(3.94)

\[
t_z = \frac{D - r_{31}tx - r_{32}ty}{r_{33}}
\]

(3.95)

3.6 Image Distortion

Image distortion is a kind of geometric aberration, where the location of image point is deviated from ideal perspective projection. This happens due to a number of reasons, e.g.

1. Radial distortion occurs due to the imperfections in the spherical nature of the lenses being used for imaging, which results in ideal image points move radially outward (pincushion) or inward (barrel) on the image plane.

2. De-centering distortion results from imperfections in alignment of lenses in the lens-sensor assembly. This distortion moves the ideal points radially as well as tangentially [35]. But, since we explicitly model the lens-sensor tilt, we do not consider decentering distortion explicitly.

The widely accepted model for radial distortion [20, 13, 36, 16] can be expressed as

\[
X_d = X_u + g(X_u, \delta)
\]

where, \( g() \) is a function which models the distortion induced as a function of the undistorted points \( X_u = (x_u, y_u) \), distorted points \( X_d = (x_d, y_d) \) and \( \delta \) is a set of distortion parameters.
The function \( g() \) is given as

\[
g(X_u, \delta) = \begin{bmatrix} x_u(k_1r_u^2 + k_2r_u^4) \\ y_u(k_1r_u^2 + k_2r_u^4) \end{bmatrix}
\]  

(3.96)

where \( r_u = \sqrt{x_u^2 + y_u^2} \) and \( \delta = (k_1, k_2) \).

### 3.7 Nonlinear Minimization

The linear estimation technique described in Section 3.4 does not take camera noise and distortion into account while estimating the camera calibration parameters. Also, the linear estimation technique does not estimate any of the calibration parameters directly, rather it estimates various non-linear relationships between the calibration parameters which, in our case is the known vector \( \mathbf{Q} \) (see Equation 3.31), and the relationships are as given in Equations 3.33-3.43. Thus, in order to incorporate distortion parameters (Section 3.6) and noise, a nonlinear minimization needs to be applied on the actual calibration parameters [16], [36] over \( N \) corresponding world and image points.

\[
U^* = \operatorname*{argmin}_U \sum_{k=1}^{N} \left\| \begin{bmatrix} I_k \\ J_k \end{bmatrix} - \begin{bmatrix} I_k(U) \\ J_k(U) \end{bmatrix} \right\|_2^2
\]  

(3.97)

where \( I_k(U) \) and \( J_k(U) \) are the synthesized image points computed from current estimate of calibration parameters \( U \), taking radial distortion into account. The complete set of calibration parameters now becomes

\[
U^* = \{ S_W^S(\theta, \phi, \psi), t_x, t_y, t_z, R_L^S(\alpha, \beta), \lambda_x, \lambda_y, I_0, J_0, k_1, k_2 \}
\]  

(3.98)

where the center of radial distortion \((I_0, J_0)\) are assumed to be known. The nonlinear error function Equation 3.97 can be optimized using gradient descent if the initial estimates obtained in Section 3.5.2 are close to the global optima.

### 3.8 Synthetic Experiments

This section describes our results on synthetic experiments. A camera is simulated with the set of calibration parameters \( U^* \) and with corresponding values shown in Table 3.1. The world points are created on a \( 16 \times 12 \times 4 \) 3D grid where the grid separation is 5 mm in \( C_W \).
Table 3.1: Ground truth calibration parameters for synthetic experiments

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation angles between $C_W$ to $C_L(S^L_W)$</td>
<td>$\theta = -0.01835 \quad \phi = 2.7486 \quad \psi = 0.3332$</td>
</tr>
<tr>
<td>Translation between $C_W$ to $C_S(T^L_W)$</td>
<td>$t_x = 26.802 \quad t_y = -34.061 \quad t_z = 122.44$</td>
</tr>
<tr>
<td>Rotation angles between $C_L$ to $C_S(R^L_W)$</td>
<td>$\alpha = 0.21811 \quad \beta = 3.5672 \quad \gamma = 0$</td>
</tr>
<tr>
<td>Focal length in image pixels</td>
<td>$\lambda_x = 849.82 \quad \lambda_y = 849.82$</td>
</tr>
<tr>
<td>Image center</td>
<td>$I_0 = 236.33 \quad J_0 = 329.33$</td>
</tr>
<tr>
<td>Radial distortion parameters</td>
<td>$k_1 = -0.0021826 \quad k_2 = 2.5683e-005$</td>
</tr>
</tbody>
</table>

Table 3.2: $\sigma = 0$

<table>
<thead>
<tr>
<th>Relative error in estimation of various calibration parameters</th>
<th>All points for linear estimation</th>
<th>Central points for linear estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td></td>
<td>\lambda_x - \lambda_x^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\lambda_y - \lambda_y^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\alpha - \alpha^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\beta - \beta^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>R^S_W - R^S_W^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>T^L_W - T^L_W^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>k_1 - k_1^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>k_2 - k_2^*</td>
</tr>
</tbody>
</table>

The image points are then projected and distorted based on the calibration parameters in Table 3.1. The obtained image points are shown in Figure 3.2(a).

The linear estimation technique and the decomposition technique described in Section 3.4 and Section 3.5 respectively, assume that there is no noise in image point detection and the image points are not distorted. In order to alleviate the linear estimates from getting effected by distortion, one needs to choose image points for the linear equation $A^\prime_{2N\times 11}Q_{11\times 1} = b_{11\times 1}$ from a small window around the image center as distortion is least around the image center.

In order to verify this hypothesis, a synthetic experiment is done where Gaussian noise with $0$ mean and $\sigma$ standard deviation is added to the projected image points, where $\sigma = (0, .01, .1, .5, 1.0)$ pixels. For each set of experiments corresponding to a $\sigma$ corrupted image points, the linear estimates are obtained first. These estimates are calculated based on selecting two different sets of image points. The first set corresponds to all the image points (see Figure 3.2(a)) and the second set consists of selecting a small set of points around the image center (see Figure 3.2(b)) called as central points. Then for each of these set of points a non-linear refinement is done based on the technique described in Section 3.7.
Table 3.3: $\sigma = 0.01$

<table>
<thead>
<tr>
<th>Relative error in estimation of various calibration parameters</th>
<th>All points for linear estimation</th>
<th>Central points for linear estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Linear</td>
<td>Nonlinear</td>
</tr>
<tr>
<td>$\frac{</td>
<td></td>
<td>\lambda_x - \lambda'_x</td>
</tr>
<tr>
<td>$\frac{</td>
<td></td>
<td>\lambda_y - \lambda'_y</td>
</tr>
<tr>
<td>$\frac{</td>
<td></td>
<td>\alpha - \alpha'</td>
</tr>
<tr>
<td>$\frac{</td>
<td></td>
<td>\beta - \beta'</td>
</tr>
<tr>
<td>$\frac{</td>
<td></td>
<td>RS_{W} - RS_{W}'</td>
</tr>
<tr>
<td>$\frac{</td>
<td></td>
<td>T_{W} - T_{W}'</td>
</tr>
<tr>
<td>$\frac{</td>
<td></td>
<td>k_1 - k_1'</td>
</tr>
<tr>
<td>$\frac{</td>
<td></td>
<td>k_2 - k_2'</td>
</tr>
</tbody>
</table>

The error metric used to compare the obtained results with the ground truth calibration parameters is by taking the ratio of the norm of the error difference and the norm of the ground truth calibration parameter (see [15] for details) where lower values indicate better estimated parameters. The results of calibration with varying amounts of noise are shown in Tables 3.2-3.6. It can be seen that the selection of central set of points gives very good estimates for smaller noise levels e.g. $\sigma < 1$. For larger noise levels, the linear estimates are not good for central points. But, in both situations, the nonlinear estimates lead to smaller errors in estimated parameters.

3.9 Discussion

A linear calibration technique has been proposed for image sensors which have a tilt and which follow a thin-lens imaging model. This technique allows for estimating majority of the parameters. The correctness and accuracy of the proposed method has been verified using a number of synthetic experiments. Since, most imaging systems do not follow exact Gaussian imaging, we did not do any experiments with real data.
Table 3.4: $\sigma = 0.1$

<table>
<thead>
<tr>
<th>Relative error in estimation of various calibration parameters</th>
<th>All points for linear estimation</th>
<th>Central points for linear estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Linear</td>
<td>Nonlinear</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\lambda_x - \lambda'_x</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\lambda_y - \lambda'_y</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\alpha - \alpha'</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\beta - \beta'</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>RS^*_W - RS'_W</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>T^*_L - T'_L</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>k_1 - k'_1</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>k_2 - k'_2</td>
</tr>
</tbody>
</table>

Table 3.5: $\sigma = 0.5$

<table>
<thead>
<tr>
<th>Relative error in estimation of various calibration parameters</th>
<th>All points for linear estimation</th>
<th>Central points for linear estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Linear</td>
<td>Nonlinear</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\lambda_x - \lambda'_x</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\lambda_y - \lambda'_y</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\alpha - \alpha'</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\beta - \beta'</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>RS^*_W - RS'_W</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>T^*_L - T'_L</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>k_1 - k'_1</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>k_2 - k'_2</td>
</tr>
</tbody>
</table>
### Table 3.6: $\sigma = 1.0$

<table>
<thead>
<tr>
<th>Relative error in estimation of various calibration parameters</th>
<th>All points for linear estimation</th>
<th>Central points for linear estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Linear</td>
<td>Nonlinear</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\lambda_x - \lambda_x^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\lambda_y - \lambda_y^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\alpha - \alpha^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\beta - \beta^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>RS_{W} - RS_{W}^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>T_{W} - T_{W}^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>k_1 - k_1^*</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>k_2 - k_2^*</td>
</tr>
</tbody>
</table>

![All image points](image1.png) ![Central image points](image2.png)

**Figure 3.2:** Number of points used for linear calibration (a) all points (b) central points.
4.1 Introduction

In camera calibration, the radial alignment constraint (RAC) has been proposed as a technique to obtain closed-form solution to calibration parameters when the image distortion is purely radial about an axis normal to the sensor plane. But, in real images this normality assumption might be violated due to manufacturing limitations or intentional sensor tilt. A misaligned optic axis results in traditional formulation of RAC not holding for real images leading to calibration errors. In this chapter, we propose a generalized radial alignment constraint (gRAC), which relaxes the optic axis-sensor normality constraint by explicitly modeling their configuration via rotation parameters which form a part of camera calibration parameter set. We propose a new analytical solution to solve the gRAC for a subset of calibration parameters. We discuss the resulting ambiguities in the analytical approach and propose methods to overcome them. The analytical solution is then used to compute the intersection of optic axis and the sensor about which overall distortion is indeed radial. Finally, the analytical estimates from gRAC are used to initialize the nonlinear refinement of calibration parameters. Using simulated and real data, we show the correctness of the proposed gRAC and the analytical solution in achieving accurate camera calibration.

Camera calibration estimates the physical (intrinsic) properties of the camera and its pose (extrinsic) with respect to a known world coordinate system using known locations of 3D scene points and their measured image coordinates. Typically camera calibration is a two step procedure. In the first step, either all or a subset of unknown calibration parameters are linearly estimated by using a linear constraint, e.g. DLT [18], collinearity of a scene point and its image [15] under the assumption of no image distortion or image noise. In the second step, image distortion and noise are taken into account and calibration parameters are nonlinearly optimized [13]. This step is typically initialized by the calibration estimates obtained in the first step.

Assuming radial distortion as the major source of image distortion, Tsai [14] observed
that the location vectors of a scene point and its distorted image point should be radially aligned about the optic axis of the lens and thus their cross product must be zero. This was termed as the radial alignment constraint (RAC) and could be analytically solved for a subset of the calibration parameters. The major assumption of RAC was that the optic axis is normal to the sensor at the center of radial distortion (CoD) and was known a priori. Although, later it was shown that the RAC could itself be used to compute the CoD [33].

But in a generic imaging setting, the optic axis may not be normal to the image sensor due to manufacturing limitations in aligning lens elements or assembling lens-sensor planes exactly parallel to each other. Although, sometimes an intentional tilting of sensor can prove useful in obtaining slanted depth-of-field effects like tilt-shift imaging [37], omnifocus imaging [5] and depth from focus estimation [38]. Under such a setting where sensor is non-frontal to the lens, the RAC can be interpreted in the following two ways, both of which we show to be inaccurate: (1) RAC can be modeled about an \textit{effective} optic axis which is normal to the sensor at the location denoted as the principal point. But the total distortion about this point is a combination of radial and decentering [15] distortion and thus the world and distorted image point are not radially aligned. (2) If RAC is formulated about the physical optic axis, then even though the world and image point lie on the same 3D plane, they are not parallel to each other and thus are not radially aligned.

Thus, in this chapter we propose the generalized radial alignment constraint (gRAC) to handle the more generic case of sensor non-frontalness. We first model the lens-sensor configuration by an explicit rotation matrix about the optic axis [4] and include it as a part of intrinsic calibration parameter set. Second, the rotation parameters are used to project the observed image points on the non-frontal sensor on to a hypothesized frontal sensor assuming that the pixel size (in metric) are known a priori. The gRAC constraint is then derived for these frontal image points (Section 4.4) about the optic axis and the CoD. As this constraint is different than RAC [14], it requires a new analytical method to solve it for a subset of calibration parameters (Section 4.5). Third, the analytical technique is used to computationally estimate the CoD (Section 4.5.3). Section 4.3 describes the RAC from [14]. Section 4.2 describes the coordinate system and the generic lens-sensor configuration for which gRAC will be derived. Section 4.7 describes the results obtained on synthetic and real data.
4.2 Calibration Coordinate Systems

In this section, we describe the coordinate systems (CS) used in this chapter for the task of camera calibration. See Figure 4.1.

1. World coordinate system: In this coordinate system, the location of world points in metric units is known, e.g. corners of a checkerboard (CB) of known dimensions.

2. Image coordinate system: The observed image points are measured in pixels in this coordinate system.

3. Lens coordinate system: The origin of this coordinate system lies at the lens center (center of projection) and whose $z$-axis coincides with the optic axis. It has metric units.

4. Sensor coordinate system: The origin of this coordinate system is at the CoD and the $z$-axis coincides with the optic axis. The $xy$-plane of the coordinate system lies on the sensor surface. It has metric units.

![Coordinate systems for camera calibration.](image)

Figure 4.1: Coordinate systems for camera calibration.

4.3 Tsai’s Radial Alignment Constraint

In this section, we describe the radial alignment constraint as proposed in [14]. Consider Figure 4.2(a) which describes the coordinate system used in [14]. The lens and the sensor
coordinate system are assumed to be parallel to each other with a common z-axis \((z_l \text{ or } z_s)\) as the “effective” optic axis and \(O_s\) as the principal point. The image of world point \(P_w = (x_w, y_w, z_w)\) on the sensor is formed at \(P_d = (x_d, y_d)\). Assuming only radial lens distortion about \(O_s\), this point would ideally be imaged at \(P_u = (x_u, y_u)\) such that the triplet \(O_s, P_d, P_u\) are collinear. Let \(P_w\) be denoted as \(P_l = (x_l, y_l, z_l)\) in the lens coordinate system, Then the normal from \(P_l\) onto the “effective” optic axis will be incident at \(P_{oz} = (0, 0, z_l)\).

![Diagram](image.png)

Figure 4.2: (a) Imaging model for RAC [14]. (b) An illustration of RAC not holding true in real images, when the sensor maybe non-frontal with respect to the lens plane.

Then, the RAC says that the vector \(\overrightarrow{P_{oz}P_l}\) is radially aligned to the vector \(\overrightarrow{O_sP_d}\) or \(\overrightarrow{P_{oz}P_l}\parallel\overrightarrow{O_sP_d}\), as the two vectors are normal to the same line, namely “effective” optic axis and also lie on the same 3D plane formed by the points \(O_s, P_u, O_l\). Thus, we get the RAC constraint \(\overrightarrow{P_{oz}P_l} \times \overrightarrow{O_sP_d} = 0\), which is solved to obtain a subset of calibration parameters. Furthermore, assuming that radial distortion was symmetric about \(O_s\), the RAC constraint was also used to estimate the principal point \(O_s\) [33].

But, while the imaging model in [14] assumed that radial distortion was symmetric about “effective” optic axis, in reality this is inaccurate for real images. Here, radial distortion is symmetric about the physical optic axis which may not coincide with the former due to unintentional lens misalignment or intentional sensor tilt (see Section 4.1). Thus, a more generic image formation model is required (see Figure 4.2(b)) [4, 1] where the non-alignment of lens and sensor is explicitly modeled via a rotation matrix and the distorted \((P_d)\) and undistorted \((P_u)\) image points are radially aligned about the CoD \((O_p)\). It can be seen that the world point \(P_l\) lies on the 3D plane formed by triplets \(\{O_p, P_u, O_l\}\) (shown in blue in Figure 4.2(b)). In comparison, the 3D plane formed by \(\{O_s, P_d, O_l\}\) (shown in red in
Figure 4.2(b)) is different from the blue plane as \(O_p\) is not a part of this plane. For RAC to hold, the two vectors: \(\overrightarrow{O_sP_d}\) and \(\overrightarrow{P_lP_{oz}}\), should be radially aligned which constrains them to lie on the same plane. Since \(\overrightarrow{O_sP_d}\) belongs to red plane and \(P_l\) belongs to the blue plane which does not coincide with red plane, \(P_l\) is out of plane with respect to red plane. Thus the normal \(\overrightarrow{P_lP_{oz}}\) from \(P_l\) normally incident onto the “effective” optic axis (edge of red plane) can never be coplanar with \(\overrightarrow{O_sP_d}\), or traditional RAC [14] cannot hold. Thus, next we propose the gRAC for a generic non-frontal sensor model.

### 4.4 Generalized Radial Alignment Constraint (gRAC)

In this section, we derive the gRAC (see Figure 4.3). Here the lens and sensor planes are...
assumed to be not parallel to each other but related via a rotation transformation $R$ \([4, 1]\), about the optic axis. Under these settings, if a world point $P_w$ ($P_l$ in lens coordinate system) is imaged at $P_{nf}$ on the sensor (in sensor coordinate system), then as per RAC, $\overrightarrow{O_P P_{nf}}$ is not parallel to $\overrightarrow{O_P P_l}$. But, if the relative rotation $R$ between lens and sensor coordinate system is known, then the projected frontal image point $P_f$ of $P_{nf}$ on a hypothesized frontal sensor gets radially aligned with the world point $P_l$. In other words, we will have that $\overrightarrow{O_P P_f} \parallel \overrightarrow{O_P P_l}$.

In the following we will derive this constraint as a function of $R$ and then solve it to get a closed-form solution to a subset of calibration parameters including $R$.

We define $R$ as a rotation matrix which aligns the lens coordinate system with the sensor coordinate system and is parameterized by two Euler angles ($\rho, \sigma$) corresponding to clockwise rotations about its $x$- and $y$-axes respectively. The rotation of lens coordinate system about the $z$-axis is considered redundant as the lens is symmetric about its $z$-axis. Thus, the final Euler angle representation of rotation is $R(\rho, \sigma, 0)$ where:

$$R(\rho, \sigma, 0) = \begin{bmatrix} \cos(\sigma) & \sin(\rho) \sin(\sigma) & \cos(\rho) \sin(\sigma) \\ 0 & \cos(\rho) & -\sin(\rho) \\ -\sin(\sigma) & \sin(\rho) \cos(\sigma) & \cos(\rho) \cos(\sigma) \end{bmatrix} \quad (4.1)$$

Let $r_{ij}$ denoted the $i^{th}$ row and $j^{th}$ column entry of $R$. Next, we derive the gRAC by analyzing the geometric relationship between a given known 3D scene point $P_w$ and its corresponding observed distorted image point $P_{nf}$ as a function of various calibration parameters.

Consider the imaging configuration in Figure 4.3, where a known world point $P_w = (x_w, y_w, z_w)$ in world coordinate system gets imaged at the pixel location $P_c = (I, J)$ in image coordinate system. Let the world and the lens coordinate system be related by a rotation $S = (s_{ij} : 1 \leq (i,j) \leq 3)$ parameterized by Euler angles ($\theta, \phi, \psi$) and a $3 \times 1$ translation $T = (t_x, t_y, t_z)$. Then, $P_w$ can be expressed as $P_l = (x_l, y_l, z_l)$ in lens coordinate system, where $P_l = SP_w + T$. Thus,

$$\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \begin{bmatrix} s_{11}x_w + s_{12}y_w + s_{13}z_w + t_x \\ s_{21}x_w + s_{22}y_w + s_{23}z_w + t_y \\ s_{31}x_w + s_{32}y_w + s_{33}z_w + t_z \end{bmatrix} \quad (4.2)$$

Let the imaged point $P_c$ be expressed in sensor coordinate system as $P_{nf} = (x_{nf}, y_{nf})$, where

$$x_{nf} = (I + I_0)s_x, \quad y_{nf} = (J + J_0)s_y \quad (4.3)$$

77
and \((I_0, J_0)\) is the location of the CoD in pixels and \((s_x, s_y)\) are the pixel sizes (in metric units e.g. mm) along the \(x\)- and \(y\)-axes of sensor coordinate system.

Now, we compute the projection of \(P_{nf}\) on a hypothesized frontal sensor, so that a radial alignment constraint can be deduced between the frontal projected sensor point and the world point \(P_l\) expressed in lens coordinate system. Let the projected point on the frontal sensor be \(P_f = (x_d f, y_d f)\). Given the sensor tilt parameterized by rotation \(R\), the distance \(\lambda\) between the lens and frontal sensor coordinate system along the optic axis and the collinearity of center of projection \(O_l\), \(P_{nf}\) and \(P_f\), we get the coordinates of \(P_f\) as:

\[
\begin{bmatrix}
  x_d f \\
  y_d f
\end{bmatrix} = \begin{bmatrix}
  -\frac{(r_{11} x_d n_f + r_{21} y_d n_f) \lambda}{r_{13} x_d n_f + r_{23} y_d n_f - \lambda} \\
  -\frac{(r_{12} x_d n_f + r_{22} y_d n_f) \lambda}{r_{13} x_d n_f + r_{23} y_d n_f - \lambda}
\end{bmatrix}
\] (4.4)

Next, we project world point \(P_l\) on the optic axis to obtain \(P_{oz} = (0, 0, z_l)\). Then, we have that location vectors \(\overrightarrow{O_p P_f}\) and \(\overrightarrow{P_{oz} P_l}\) are coplanar lying on a plane formed by points \((O_p, P_f, P_l)\) and are also parallel and radially aligned to each other. From the radially aligned constraint, we have \(\overrightarrow{O_p P_f} \times \overrightarrow{P_{oz} P_l} = 0\), which given \(\overrightarrow{O_p P_f} = x_d f \hat{i} + y_d f \hat{j}\) and \(\overrightarrow{P_{oz} P_l} = x_l \hat{i} + y_l \hat{j}\) (both in lens sensor coordinate system) simplifies to the generalized radial alignment constraint (gRAC):

\[x_d f \cdot y_l = y_d f \cdot x_l \] (4.5)

If it is assumed that the subset
\[U_1 = (I_0, J_0, s_x, s_y)\] (4.6)

of calibration parameters is known, then \(P_{nf} = (x_d n_f, y_d n_f)\) can be computed using Equation 4.3. Given known \(P_{nf}\), Equation 4.4 can be used to obtain hypothesized frontal points \(P_f = (x_d f, y_d f)\) as a function of unknown calibration parameters \((R, \lambda)\). Also, using Equation 4.2, \(P_l = (x_l, y_l)\) can be obtained in terms of unknown extrinsic calibration parameters \((\theta, \phi, \psi, t_x, t_y, t_z)\) and known world points \(P_w = (x_w, y_w, z_w)\). Thus, Equation 4.5 can be simplified to obtain the linear equation \(A Q = b\), relating \(i^{th}\) world-image point observation as

\[
\begin{bmatrix}
  x_d n_f x_w & x_d n_f y_w & x_d n_f z_w & x_d n_f y_d n_f x_w & y_d n_f y_w & y_d n_f z_w
\end{bmatrix}
\begin{bmatrix}
  q_1 \\
  \vdots \\
  q_T
\end{bmatrix} = y_d n_f \] (4.7)
Here, \((A, b)\) are known, while \(Q = \{q_1, \cdots, q_7\}\) encodes seven calibration parameters denoted here as \(U_2\):

\[
U_2 = (\rho, \sigma, \theta, \phi, \psi, t_x, t_y) \tag{4.8}
\]

via the following nonlinear relationships:

\[
q_1 = \frac{r_{11}s_{21} - r_{12}s_{11}}{r_{22}t_x} \tag{4.9}
\]
\[
q_2 = \frac{r_{11}s_{22} - r_{12}s_{12}}{r_{22}t_x} \tag{4.10}
\]
\[
q_3 = \frac{r_{11}s_{23} - r_{12}s_{13}}{r_{22}t_x} \tag{4.11}
\]
\[
q_4 = \frac{r_{11}t_y - r_{12}t_x}{r_{22}t_x} \tag{4.12}
\]
\[
q_5 = \frac{-s_{11}}{t_x} \tag{4.13}
\]
\[
q_6 = \frac{-s_{12}}{t_x} \tag{4.14}
\]
\[
q_7 = \frac{-s_{13}}{t_x} \tag{4.15}
\]

As \((A, b)\) are known, Equation 4.7 can be solved in least squares sense given four or more observations of scene points to obtain an estimate of \(Q\). This estimate can be used to analytically solve the set of nonlinear relationships in Equations 4.9-4.15 to obtain \(U_2\) as we show in Section 4.5. It can be noted that in Tsai’s RAC [14], \(R\) was an identity matrix and their solution was derived based on this assumption. In the gRAC case, the derivations are comparatively more involved due to the inclusion of \(R\) parameter. For calibrating the remaining calibration parameters, namely

\[
U_3 = (\lambda, t_z) \tag{4.16}
\]

we adopt the technique of [14] as shown in Section 4.5. Thus, from Equations 4.6, 4.8 and 4.16, the final set of camera calibration parameters to be calibrated is

\[
U = \{U_1, U_2, U_3\}
\]
4.5 Analytical Solution to gRAC

In this section, we analytically solve Equations 4.9-4.15 for the seven calibration parameters $U_2$ (Equation 4.8) assuming that $U_1$ (Equation 4.6) is known. Later, we will show a technique similar to [33] and estimate $U_1$ given optimal estimates of $U_2$ applied to the gRAC based linear Equation 4.7. We use $|x|$ to denote that magnitude of $x$ without knowing the sign.

4.5.1 Stage 1: Determining Sign Ambiguous Estimates

Solving for $t_x$

Squaring and adding Equations 4.13-4.15 and from orthonormality of first row of extrinsic rotation matrix $S$ (Equation 4.2), $t_x$ can be computed with a sign ambiguity as

$$|t_x| = \frac{1}{\sqrt{q_5^2 + q_6^2 + q_7^2}}$$  \hspace{1cm} (4.17)

Solving for $s_{11}, s_{12}, s_{13}$

Given $t_x^*$, using Equations 4.13, 4.14 and 4.15, we get

$$s_{11} = -q_5 t_x \quad s_{12} = -q_6 t_x \quad s_{13} = -q_7 t_x$$  \hspace{1cm} (4.18)

Solving for $s_{21}, s_{22}, s_{23}$

Adding the product of Equations 4.9 and 4.13, Equations 4.10 and 4.14 and Equations 4.11 and 4.15 we get,

$$\frac{r_{12}}{r_{22}} = t_x^2 (q_1 q_5 + q_2 q_6 + q_3 q_7)$$  \hspace{1cm} (4.19)

Also, adding the squares of Equations 4.9, 4.10 and 4.11 and using the orthonormality of first and second rows of $S$, we obtain

$$\frac{r_{11}^2 + r_{12}^2}{r_{22}^2} = t_x^2 (q_1^2 + q_2^2 + q_3^2)$$  \hspace{1cm} (4.20)

$$\implies \frac{r_{11}^2}{r_{22}^2} = N t_x^2 - M^2 t_x^4$$  \hspace{1cm} (using Equation 4.19)  \hspace{1cm} (4.21)
As $r_{11} = \cos(\sigma) > 0$ and $r_{22} = \cos \rho > 0$ from Equation 4.1, the ratio $\frac{r_{11}}{r_{22}}$ from Equation 4.21 can be determined uniquely as

$$\frac{r_{11}}{r_{22}} = \sqrt{Nt_{x}^{2} - M^{2}t_{x}^{4}}$$  \hspace{1cm} (4.22)$$

Applying Equations 4.19 and 4.22 and Equations 4.13-4.15 to Equations 4.9-4.11 respectively we can solve for $s_{21}, s_{22}, s_{23}$ with sign ambiguity as (since $t_{x}$ is sign ambiguous)

$$s_{21} = \frac{(q_{1} - t_{x}^{2}Mq_{5})t_{x}}{P}$$  \hspace{1cm} (4.23)$$

$$s_{22} = \frac{(q_{2} - t_{x}^{2}Mq_{6})t_{x}}{P}$$  \hspace{1cm} (4.24)$$

$$s_{23} = \frac{(q_{3} - t_{x}^{2}Mq_{7})t_{x}}{P}$$  \hspace{1cm} (4.25)$$

Solving for $s_{21}, s_{22}, s_{23}$ Uniquely

Assuming a right-hand coordinate system, the cross product of the first (Equation 4.18) and second (Equations 4.23-4.25) row of $S$ can be used to determine the third row of $S$: $(s_{21}, s_{22}, s_{23})$. These estimates are unique as the it involves terms of $t_{x}^{2}$ which is greater than 0 and all other terms involving $q_{i}$ are uniquely known.

Solving for $t_{y}$

Applying Equations 4.19 and 4.22 to Equation 4.12, we get

$$t_{y} = \frac{(q_{4} + t_{x}^{2}M)t_{x}}{P}$$  \hspace{1cm} (4.26)$$

Solving for $\{r_{11}, \cdots, r_{33}\}$

The left-hand side (LHS) of Equation 4.19 and Equation 4.22 can be expressed in terms of Euler angle $(\rho, \sigma)$ via Equation 4.1, which expresses sensor rotation matrix $R$ in terms of its component Euler angles as follows

$$\frac{r_{12}}{r_{22}} = \frac{\sin \rho \sin \sigma}{\cos \rho} = t_{x}^{2}M$$  \hspace{1cm} (4.27)$$
and
\[ \frac{r_{11}}{r_{22}} = \frac{\cos \sigma}{\cos \rho} = P \] (4.28)

These two equations can be solved for \((\rho, \sigma)\) with a sign ambiguity to obtain
\[
\rho = \pm \cos^{-1} \left( \frac{L^2 + P^2 + 1 - \sqrt{(L^2 + P^2 + 1)^2 - 4P^2}}{2P^2} \right) \] (4.29)
\[
\sigma = \pm \sin^{-1} \left( \sqrt{1 - P^2 \cos^2(\rho)} \right) \] (4.30)

Although the individual signs of \((\rho, \sigma)\) are not known uniquely, the relative sign of \((\rho, \sigma)\) with respect to each other can be determined from the sign of \(L\) in Equation 4.27 as the denominator in Equation 4.27 is always positive \((\cos \rho > 0)\). The ambiguity here arises from the fact that gRAC is designed for a frontal coordinate system which is obtained by projecting the non-frontal sensor coordinates \(P_{nf}\) onto a frontal sensor to give \(P_f\). Since, this projection involves taking the cosine of tilt angles encoded in \(R\), it is many-to-one leading to sign ambiguity in analytical estimate of \((\rho, \sigma)\).

### 4.5.2 Stage 2: Determining the Sign of Estimates

In Section 4.5.1, we determined partial set of extrinsic and intrinsic parameters denoted here as \(U_e = \{S, t_x, t_y\}\) and intrinsic parameters denoted here as \(U_i = \{R\}\) respectively with sign ambiguity. While the sign ambiguity in determining \(U_e\) resulted from not knowing \(t_x\) uniquely in Equation 4.17, the ambiguity in \(U_i\) was inherent to the gRAC constraint due to many-to-one projection map from a non-frontal sensor configuration to a frontal sensor configuration. Next, we present a technique to retrieve the sign of \(t_x\) uniquely (similar but not same as in Tsai [14]), thus determining \(U_e\) uniquely. This is followed by a method to uniquely determine \(U_i\).

We also note that given all sign ambiguities in \(\{U_e, U_i\}\), there are four possible solution sets for \(\{U_e, U_i\}\), corresponding to the combinations: \(\text{sign}(t_x) = \pm\) and either \(\text{sign}(\rho, \sigma) = (+, +)/(-, -)\) or \(\text{sign}(\rho, \sigma) = (+, -)/(-, +)\). This is so as the relative sign of \((\rho, \sigma)\) is uniquely determined from \(\text{sign}(L)\) (Equation 4.27). We assume the two rotation matrices obtained from sign ambiguity of \((\rho, \sigma)\) are \(R_1\) and \(R_2\).
Determining $\lambda, t_z$ and the Sign of $t_x$ by Ignoring Lens Distortion

Let us redefine

$$u = r_{13} x_{df} + r_{23} y_{df}$$  \hspace{1cm} (4.31)$$
$$v = -r_{11} x_{df} - r_{21} y_{df}$$  \hspace{1cm} (4.32)$$

Then from Equation 4.4, we have $x_{df} = \frac{u\lambda}{u-\lambda}$. Also, if we ignore lens distortion, then world point $P_l$ and frontal image point $P_f$ can be related as

$$x_{df} = -\lambda \frac{x_l}{z_l}$$  \hspace{1cm} (4.33)$$

Replacing for $x_{df}$ we get

$$\frac{v\lambda}{u-\lambda} = -\lambda \frac{x_l}{w + t_z}$$  \hspace{1cm} (4.34)$$

where $w = s_{31} x_w + s_{32} y_w + s_{33} z_w$ from Equation 4.2. This equation can be simplified to set up the following linear equation,

$$\begin{bmatrix} -x_l & v \end{bmatrix} \begin{bmatrix} \lambda \\ t_z \end{bmatrix} = -ux_l - vw$$  \hspace{1cm} (4.35)$$

where, $(u, v)$ are functions of $R$ from Equations 4.31-4.32 and $(x_l, w)$ are functions of $t_x$ from Equations 4.2, 4.18 and 4.23-4.25.

Now, given multiple world-image point observations, Equation 4.35 can be solved for $(\lambda, t_z)$ using each of the four possible values of $\{U_e, U_i\}$. Graphically, the four possible solutions to $\{U_e, U_i, \lambda, t_z\}$ can be visualized in Figure 4.4, where on the left we have the ground truth imaging and on the right are the four imaging hypothesis labeled as A, B, C and D. As can be seen all four solutions satisfy the perspective (we had assumed no distortion earlier) imaging of $P_w$ to $P_f$ but each correspond to different calibration parameters. Based on this analysis, solution C and D can be rejected by checking the sign of $\lambda$ obtained from Equation 4.35 as $\lambda$ cannot be negative. The correct solution among A and B can be obtained by analyzing model fitting error for radial distortion coefficients as described next.
Figure 4.4: (Left) Ground truth image formation. (Right) (a) Solution A: \((t_x, R_1, \lambda_1, t_z_1)\). (b) Solution B: \((t_x, R_2, \lambda_2, t_z_2)\). (c) Solution C: \((-t_x, R_2, -\lambda_1, t_z_1)\). (d) Solution D: \((-t_x, R_1, -\lambda_2, t_z_2)\). Solutions C and D can be rejected based on \(\lambda\) being negative. The better solution between A and B is selected by analyzing radial distortion model fitting error.

Determining \(R\)

From Figure 4.4 (right, a-b), we observe that among the two solutions A and B, only solution A coincides with a rotation which will result in a frontal sensor parallel to the lens plane. This implies that the projected frontal points in A will fit the symmetric radial distortion model better than in B. For each set of calibration parameters in A and B, we first compute the radial distortion parameters of \((k_1, k_2)\) by solving the linear equation \(P_f - Q_f(1 + k_1 r^2 + k_2 r^4) = 0\) for a set of world-image point observations. Here \(Q_f = (x_f, y_f)\) is ideally projected frontal image sensor points, \(r^2 = x_f^2 + y_f^2\) and \(P_f = (xd_f, yd_f)\). The radial distortion model fitting error \(E_{rad}\) can then be obtained as:

\[
E_{rad} = P_f - Q_f(1 + k_1 r^2 + k_2 r^4)
\] (4.36)

The solution with least \(E_{rad}\) is selected, e.g. in Figure 4.4, solution A will get selected. Thus, \(R(\rho, \sigma), t_x, \lambda, t_z\) are estimated uniquely. Furthermore \(t_x\) can then be used to estimate
S uniquely from Equation 4.18 and Equations 4.23-4.25. Also, applying $t_x$ to Equation 4.26, $t_y$ can be estimated uniquely. Thus, we uniquely determine the calibration parameters $U_e = \{S, t_x, t_y\}$, $U_i = \{R\}$ and $U_3 = \{\lambda, t_z\}$, or in other words $\{U_2, U_3\}$ (from Equations 4.8 and 4.16). Next, we estimate the remaining calibration parameters of CoD $(I_0, J_0)$ in $U_1$.

4.5.3 Iterative Determination of CoD

The RAC [14] as well as the proposed gRAC are formulated in sensor coordinate system (metric), while the image measurements are in the image coordinate system (pixels). This requires conversion from pixels to metric domain as per Equation 4.3, which is a function of $U_1 = \{I_0, J_0, s_x, s_y\}$. Since, the gRAC has rank seven which is same as the size of $U_2$, there are no additional analytical constraints to determine $U_1$ completely. Thus, we first assume $(s_x, s_y)$ are known. Typically $s_y$ is already known [14, 15] as it defines the reference scale over which $\lambda, s_x$ are defined and $s_x$ can be obtained reliably from the sensor data-sheet.

If the principal point were same as the CoD $(I_0, J_0)$, [33] showed that the residual error in RAC (Section 4.3) when applied to measured image points on frontal sensor is quadratic with respect to error in the assumed location $(I_0, J_0)$. Thus, [33] proposed to compute $(I_0, J_0)$ by nonlinear minimization of residual RAC [14] error. But, in our imaging model (Section 4.4), the measured image points are on a non-frontal sensor plane. They need to be converted to frontal sensor coordinates requiring knowledge of $R$ (Equation 4.4). While $R$ can be computed from the analytical technique of Section 4.5, this technique in turn requires correct estimates of $(I_0, J_0)$. To solve this “chicken and egg problem”, we propose an iterative solution similar to [22] as follows:

1. Uniformly sample an image region for $(I_0, J_0)$.
2. For each hypothesized $(I_0, J_0)$ obtain gRAC (Equation 4.7).
3. Solve gRAC for $R$ (Section 4.5) and obtain frontal coordinates (Equation 4.4).
4. Nonlinearly minimize residual RAC [14] obtained from frontal coordinates to get optimal $(I_0^*, J_0^*)$.
5. Compute the difference error $E = abs(I_0 - I_0^*) + abs(J_0 - J_0^*)$, giving an estimate of how good the initial assumed $(I_0, J_0)$ was. Select the point with minimum error $E$. Stop if $E$ is less then a threshold, otherwise go to Step 6.
6. Refine the sampling around the selected point in Step 5 and repeat Steps 2, 3, 4 and 5.
Finally, the initial estimates obtained from gRAC are used as initialization for nonlinear refinement of parameters to obtain $U^*$. This process incorporates radial lens distortion and minimizes the re-projection error over all observed scene and image points.

### 4.6 gRAC for Pupil-Centric Imaging Model

The pupil-centric imaging model has been shown to be more accurate than generic thin-lens model for camera calibration [1]. For the case of gRAC, we have earlier assumed a thin-lens model in Section 4.3 and derived the analytical solution. As shown in Kumar and Ahuja [1], the known world points in lens coordinate system are related in pupil-centric and the thin-lens model by an affine matrix $A_{pg}$ and a translation along the optic axis by $-a_n$. Here $a_n$ is the entrance pupil location with respect to the front principal plane and $A_{pg}$ is given as:

$$
A_{pg} = \begin{bmatrix}
\alpha & 0 & 0 \\
0 & \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

where, if $F$ is the optical focal length, then

$$
\alpha = \frac{F - a_n}{F}
$$

If $P_i^p = (x_i^p, y_i^p, z_i^p)$ and $P_i^g = (x_i^g, y_i^g, z_i^g)$ denote a known world point described in a coordinate system located at the front principal coordinate system and a pinhole coordinate system respectively, then we have the following relationships

$$
x_i^g = \alpha x_i^p
$$

$$
y_i^g = \alpha y_i^p
$$

$$
z_i^g = z_i^p - a_n
$$

Since the radial alignment of distorted and undistorted points is invariant to the location of the center of projection on the optic axis, the analysis of gRAC can be readily adapted to the pupil-centric imaging model by simply modifying Equation 4.2 as per the relationships

86
in Equations 4.39-4.41:

\[
\begin{bmatrix}
x_l \\
y_l \\
z_l
\end{bmatrix} =
\begin{bmatrix}
\alpha(s_{11}x_w + s_{12}y_w + s_{13}z_w + t_x) \\
\alpha(s_{21}x_w + s_{22}y_w + s_{23}z_w + t_y) \\
s_{31}x_w + s_{32}y_w + s_{33}z_w + t_z - a_n
\end{bmatrix}
\]  (4.42)

Since, both the x- and y-components of \( P_l \) are modified by the same factor, the gRAC derived in Equation 4.5 is not affected, thus analytically showing that gRAC is independent of the location of the center of projection as mentioned earlier in this section. But, the pupil-centric modeling via \( \alpha \) and \( a_n \) is critical to accurately estimating calibration set \( U_3 \) as well as removing the ambiguity discussed in Section 4.5.2.

4.7 Experiments

We present and compare the results of proposed analytical solution to gRAC on synthetic distorted and real data with traditional RAC [14].

4.7.1 Synthetic Data

A camera was simulated with intrinsic parameters \( \lambda = 8.4 \text{ mm}, \rho = 0, \sigma = 4 \text{ degrees}, \)
\( s_x = 0.01, s_y = 0.01 \text{ mm}, I_0 = 240, J_0 = 320 \text{ pixels}, k_1 = 0.0021966, k_2 = -1.3001e^{-05} \) and extrinsic parameters \( \theta = 0.10, \phi = 43.31, \psi = 0.02 \text{ degrees}, t_x = -65.09, t_y = -41.04, t_z = 102.2 \text{ mm}. \) Synthetic world points \( P_w \) are generated and projected (Section 4.4) using simulated camera parameters to obtain image points. Then, Gaussian noise with standard deviation \{0.05, 0.1, \cdots, 1.0\} pixels is added to the synthesized image points to simulate measurement error. The gRAC constraint (Equation 4.7) is applied and analytical calibration estimates are computed. This procedure is repeated 100 times and the mean of all the trials is taken and compared with the ground truth data. Figure 4.5(a-d) shows the relative error(\%) in estimation of \( R(\rho, \sigma), S(\theta, \phi, \psi), t_x, t_y, t_z \) and \( \lambda \) respectively. The error bars in Figure 4.5 indicates the std. dev. in the estimation of respective calibration parameters. The relative error in parameter estimates increases with increasing noise. For lower noise levels, this error as well as the std. dev. is low for all calibration parameters. As, the measurement error in our real data is close to 0.11 pixels, the simulation gives confidence that for real data, gRAC based analytical solution should be robust to image noise.
### 4.7.2 Real Data

The camera used for calibration is a custom made AVT Marlin F-033C camera with sensor tilted $\approx 3-4$ degrees and acquiring $640 \times 480$ resolution images. The corners of a checkerboard (CB) calibration pattern with $20 \times 20$ squares of length 5 mm and positional accuracy of .001 mm are used as known 3D scene points. A 2.5D image data is captured by moving the CB along its surface normal and imaging each discrete CB position. A set of 11 such 2.5D datasets are captured by placing the camera at different locations in front of the CB. The corners in the acquired calibration images are computed using [25]. We compute calibration parameters by using RAC and gRAC and then refine them via nonlinear minimization. The results obtained are shown in Table 4.1. Comparing the re-projection errors in the last row of Table 4.1, we observe that calibration based on the analytical estimates obtained from gRAC leads to smaller re-projection error as compared to traditional RAC. The image center $(I_0, J_0)$ from analytical gRAC has been obtained using the technique proposed in Section 4.5.3. It can be seen that it is quite different from the one obtained by RAC [33] indicating that the optic axis is indeed not orthogonal to the lens and thus the sensor is tilted. The analytical tilt estimate from gRAC is 3.81 degrees and after refinement it is 4.23 degrees. The small difference arises since analytical solution ignores noise and is thus

---

**Figure 4.5:** Relative error vs. noise (in pixels) using gRAC on synthetic data.
sensitive to measurement errors.

Table 4.1: Calibration estimates using the two techniques

<table>
<thead>
<tr>
<th>Method</th>
<th>RAC [14]</th>
<th>gRAC (proposed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>analytical</td>
<td>nonlinear</td>
</tr>
<tr>
<td></td>
<td>analytical</td>
<td>nonlinear</td>
</tr>
<tr>
<td>$\lambda_x = \frac{\Delta}{s_x}$</td>
<td>829.57</td>
<td>823.64</td>
</tr>
<tr>
<td>$\lambda_y = \frac{\Delta}{s_y}$</td>
<td>833.63</td>
<td>827.67</td>
</tr>
<tr>
<td>Principal Point $J_0$</td>
<td>225.845</td>
<td>226.15</td>
</tr>
<tr>
<td>Radial $k_1$</td>
<td>$-0.0021$</td>
<td>$2.33e-05$</td>
</tr>
<tr>
<td>Radial $k_2$</td>
<td>$-0.0021$</td>
<td>$2.33e-05$</td>
</tr>
<tr>
<td>Radial $\rho$</td>
<td>$-0.49$</td>
<td>$0.13$</td>
</tr>
<tr>
<td>Radial $\sigma$</td>
<td>$3.81$</td>
<td>$4.23$</td>
</tr>
<tr>
<td>Re-projection Error</td>
<td>$0.082064$</td>
<td>$0.057119$</td>
</tr>
</tbody>
</table>

4.8 Discussion

In this chapter, we have proposed a generalized radial alignment constraint (gRAC) which takes possible misalignment between lens and sensor planes into account. We have developed an analytical solution to solve the gRAC constraint for a subset of calibration parameters. Then, we have shown that the center of radial distortion can also be computed based on the analytical solution using an iterative approach. Finally, we have shown that nonlinear calibration with gRAC initialization leads to lower re-projection error than RAC [14] based initialization.
CHAPTER 5

NON-FRONTAL CAMERA CALIBRATION USING FOCAL STACK IMAGERY

5.1 Introduction

A non-frontal camera has its lens and sensor plane misaligned either due to manufacturing limitations or an intentional tilting as in tilt-shift cameras. Under ideal perspective imaging, a geometric calibration of tilt is impossible as tilt parameters are correlated with the center of radial distortion (CoD). In other words, there are infinite combinations of CoD and sensor tilt parameters such that the perspective imaging equations are satisfied equally well. Previously, the non-frontal calibration problem (including sensor tilt estimation) has been solved by introducing extra constraints which an CoD should satisfy. In this paper, we propose an additional constraint which incorporates image blur/defocus present in non-frontal camera images into the calibration framework. Specifically, it has earlier been shown that a non-frontal camera rotating about its center of projection captures images with varying focus. This stack of images is referred to as a focal stack. Given a focal stack of a known checkerboard (CB) pattern captured from a non-frontal camera, we combine geometric re-projection error and image blur error computed from current estimate of sensor tilt as the calibration optimization criteria. We show that the combined technique outperforms geometry-only methods while also additionally yielding blur kernel estimates at CB corners.

Camera calibration is the task of estimating the intrinsic and extrinsic parameters of a camera imaging a 3D scene and capturing 2D images of this scene on the image sensor. The intrinsic parameters encode the physical characteristics of the camera and the extrinsic parameters determine the 3D pose of the camera with respect to a known world coordinate system. In this chapter, we focus on calibrating non-frontal cameras whose sensor and lens plane are not constrained to lie on the same plane. Typically, all cameras can be considered as non-frontal given that we take manufacturing limitations into account. Although, sometimes an intentionally tilted sensor can be used for tilt-shift photography, focal stack acquisition [10] and omnifocus imaging [5].

Previously, a number of techniques have been proposed for calibration of such cameras
including that of [11, 14, 15, 13, 16] which do not take sensor tilt into account and [4, 1, 2] which also estimate the sensor tilt. As these techniques are inherently geometric in nature as then minimize the pixel re-projection error to obtain calibration estimates, they will be referred to as geometric techniques henceforth.

Additionally, other image properties e.g. image blur, vignetting have also been used for camera calibration. In [39], a flat texture surface was imaged and the effect of vignetting in the captured image was used for calibration. In [40], geometric and blur properties of a lens were used to model the point spread function (PSF) of the lens and then the model parameters were optimized from the observed PSF images. In [41], blurred edges in a CB image were used to estimate the radius of a circular blur kernel as well as the location of the CB corner. These measurements were then compared with physically modeled predictions parameterized by the calibration parameters, and the resulting error minimized to obtain the optimal values of the parameters.

This chapter combines ideas from geometric and image blur based methods to achieve non-frontal camera calibration. While [41] focuses on the problem of detecting corners under unwanted blurring of the pinhole CB images, we propose that instead of treating blur as unwanted, a sequence of intentionally blurred images, in addition to traditional, sharp images, can provide useful constraints to handle some inherent ambiguities in calibrating cameras modeled as being non-frontal. These ambiguities pertain to the highly correlated parameters of sensor-tilt angle and CoD location and have been discussed and analyzed in [1, 2] who also provide geometric solutions. In this work, we propose an additional blurring constraint to solve this ambiguity problem. We leverage on the idea that the sensor tilted about the optic axis of the camera produces unique image blur pattern on the captured image of a scene from a non-frontal camera. Thus, if the image blur can be analyzed to uniquely estimate the sensor tilt and the CoD.

Now, geometric calibration requires sharp pinhole images while the blur constraint requires presence of image blur in the input calibration image data. While both constraints cannot be satisfied at the same time for an image, we propose to use a focal stack as an input. A focal stack has each scene point imaged with varying amounts of focus including zero blur in a sequence of images. Also, it has been shown in [10] that a non-frontal camera rotating about its optic center can be used to capture a focal stack. Thus, we have blurred images from a non-frontal camera. An omnifocus imaging technique [5] can be applied to the focal stack to compute a sharp focused image. This image is a representative of an ideal pinhole image and is used as an input to geometric calibration framework.

Thus, our proposed calibration technique takes a focal stack as input, warps and registers them with respect to each other such that a CB corner across the focal stack appears at the
same pixel location in a global image coordinate frame. The registered images are then used to compute an omnifocus image [5] from which sharp CB corners are detected [25]. Given the registration parameters, the omnifocus image is then warped back to each of the focal stack images to synthesize a focused focal stack which can now be used in a conventional geometric calibration framework [1] to obtain extrinsic and intrinsic parameter estimates. While the extrinsic parameter encodes the object distance of a particular CB corner, the intrinsic parameters of sensor tilt encodes the image distance of that particular CB corner. Thus, using the thin lens equation and a Gaussian blur kernel model, the parameters of the blur kernel can be computed and applied to the focused focal stack to synthesize a conventional focal stack. The synthetic conventional focal stack can then be compared with the observed conventional focal stack to give the image blur error. The combined geometric and image blur error is then used to optimize all calibration parameters including blur kernel parameters.

We describe the geometry of blurred image formation in Section 5.2.1. The focal stack acquisition technique is described in Section 5.2.2. Section 5.2.3 describes pinhole image formation in a non-frontal camera and the calibration parameters. Section 5.2.4 explains the computation of blur kernel from calibration parameters. The proposed calibration approach is described in Section 5.3. Finally, Section 5.4 shows the results and the efficacy of the proposed calibration technique on real data sets.

5.2 Image Formation, Modeling and Acquisition

5.2.1 Defocus Image Formation

![Defocus Image Formation Diagram](image_url)

Figure 5.1: Defocus image formation.
Figure 5.1 shows image formation with aperture wide open in a thin-lens setting where an object is imaged on an image sensor. From the thin lens law [42], we have that the distance $u$ of the object and the distance $v$ of its sharply focused image from the thin lens are conjugate to each other. This implies that they are related by the thin-lens equation

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{F}$$

(5.1)

where $F$ is the focal length of the lens. In the defocused camera setting, when the image sensor is moved to a distance $s_d$ from the lens which is different than $v$, the captured image of the object is blurred. If the object is a point source of light, then this image is commonly referred to as the point spread function (PSF). The shape of the PSF for spherical lenses is typically assumed to be circular with radius parameter $r$ and light intensity distribution $h(x, y)$ is assumed to be Gaussian [43]. A functional form of $h(x, y)$ can be obtained as follows:

$$r = \frac{D}{2} \left| \frac{s_d}{v} - 1 \right| \quad \text{and} \quad h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

(5.2)

where $D$ is aperture diameter of the lens and is assumed to be known a priori based on F-number [42], $\sigma \approx \frac{c}{\xi}$ where $c$ is camera dependent (empirically set to 0.3 in our experiments).

### 5.2.2 Focal Stack Images as Non-Frontal Camera Calibration Data

![Figure 5.2: Focal stack acquisition by (a) moving sensor along the optic axis and (b) rotating a camera with tilted sensor about the optic center [10]. On comparison of the imaging geometry in both cases we observe that the images formed on sensor location 1, 2 and 3 are similar. Thus, both systems can be used interchangeably to generate similar focal stack images. In this chapter, we use the configuration (b).](image)

A focal stack is a collection of focused and defocus images of a static scene captured by varying some camera setting before each image capture and then acquiring the image. These
settings could be e.g. relative distance between lens and sensor, object distances from the lens or aperture size. While calibrating a camera, it is usually assumed that the camera configuration is fixed. Thus the ideal way to capture a focal stack for non-frontal camera calibration is to translate the CB along an axis parallel to the optic axis of the lens [44]. The conjugate technique of keeping the CB fixed and moving the sensor along the optic axis (see Figure 5.2(a)) will also generate a similar focal stack. We will refer to it as the conventional focal stack acquisition method. In [10], a further abstraction of the conventional technique was proposed where the sensor of the camera was first tilted with respect to the lens plane. Then, this camera was rotated about the optic center. This resulted in a single scene point getting imaged with varying amounts of focus in each captured image (see Figure 5.2(b)). This happened because while rotation of the camera caused the object distance to change in each image, the sensor tilt at each rotation caused the image distance to change as well. Thus, by the thin-lens Equation 5.1, both $u$ and $v$ varied to generate defocus, while in the conventional case only image distance $v$ varied to generate defocus. Although tilted sensor camera [10] gives more freedom for focal stack acquisition, it has the same defocusing properties as conventional technique. Thus, without any loss of generality, the camera proposed in [10] can be used to acquire focal stack calibration data.

5.2.3 Geometric Imaging Model

Now, we describe the geometric image formation image projection equations as a function of various calibration parameters for a non-frontal camera [10]. Compared to conventional cameras with frontal sensor, the only added calibration parameter for non-frontal cameras are two Euler angle rotation parameters of the image sensor [4]. For calibration, we consider the following four coordinate systems (see Figure 5.3):

1. World coordinate system ($C_W$) located on one of the corners of the CB pattern.

2. Lens coordinate system ($C_L$) centered at the pinhole projection with its $z$-axis aligned with the optic axis and the $xy$-plane parallel to the lens plane.

3. Sensor coordinate system ($C_S$) located on the image sensor with origin at the location where the optic axis intersects with the sensor plane.

4. Image coordinate system ($C_I$) in which the observed image points are defined.

Assuming noiseless and distortion less imaging, we consider an object point $P_w = (X_w, Y_w, Z_w)$ in $C_W$ and its corresponding measured image point $P_m = (I, J)$ in $C_I$. These two points can be mapped in terms of various camera calibration parameters as follows.
If $C_W$ and $C_L$ are related by a three Euler angle parameter rotation matrix $S = (s_{ij} : 1 \leq (i, j) \leq 3)$ and translation $T = (t_x, t_y, t_z)$, then $P_W$ can be expressed in $C_L$ as $P_{CL} = (x_l, y_l, z_l)$:

$$
\begin{bmatrix}
    x_l \\
    y_l \\
    z_l \\
    1
\end{bmatrix} =
\begin{bmatrix}
    S & T \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    P_W \\
    1
\end{bmatrix}
= 
\begin{bmatrix}
    s_{11}X + s_{12}Y + s_{13}Z + t_x \\
    s_{21}X + s_{22}Y + s_{23}Z + t_y \\
    s_{31}X + s_{32}Y + s_{33}Z + t_z \\
    1
\end{bmatrix}
$$

(5.3)

Let $C_S$ be translated by $T_\lambda = (0, 0, \lambda)$ along the optical axis from the origin of $C_L$ and rotated by rotation matrix $R$ parameterized by two Euler angles $[4]$. $P_L$ can then be expressed in the coordinate system $C_S$ as

$$
\begin{bmatrix}
    x_s \\
    y_s \\
    z_s \\
    1
\end{bmatrix} =
\begin{bmatrix}
    R & 0 \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    I & T_\lambda \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    P_L \\
    1
\end{bmatrix}
$$

Figure 5.3: Geometric image formation.
where \( R = (r_{ij} : 1 \leq (i,j) \leq 3) \). Since \( R \) is parameterized by two Euler angles, we note that \( r_{21} = 0 \) in Equation 5.4. Given \( P_S \) and ignoring distortion, the intersection of the light ray, connecting \( P_S \) and the origin of \( C_L \), with the non-frontal sensor plane can be obtained using simple coordinate geometry. This is the point on the sensor where the ideal image of the scene point is formed. Let this point be denoted as \( P_{nf} = (x_{nf}, y_{nf}) \). It can then be transformed to obtain final predicted image coordinates \( P_p = (I_p, J_p) \) as:

\[
I_p = \frac{x_{nf}}{s_x} s_x - I_0; \quad J_p = \frac{y_{nf}}{s_y} s_y - J_0
\]  

(5.5)

where \((I_0, J_0)\) are the location of origin of \( C_S \) in image coordinates and \((s_x, s_y)\) are the size of the pixel in distance metric. Under ideal noiseless and distortion less imaging \( P_m = P_p \). Finally, the set of 12 calibration parameters which need to be estimated are:

\[
U = \begin{bmatrix}
S \hfill \begin{array}{c}
\text{3 Euler ang.}
\end{array}
\hfill T \hfill \begin{array}{c}
\text{3 Euler ang.}
\end{array}
\hfill R \hfill \begin{array}{c}
\text{2 Euler ang.}
\end{array}
\hfill \lambda, s_x, I_0, J_0
\end{bmatrix}
\]  

(5.6)

where we assume that \( s_y \) is given to us as \( s_y \) decides the reference scale with respect to which calibration parameters are estimated.

### 5.2.4 Integrating Geometric and Blur Cues for Calibration

Now, from the viewpoint of camera calibration, it can be seen that the blur parameters in Equation 5.2 are dependent on camera calibration parameters of Section 5.2.3 as follows. Given the current estimate of calibration parameters \((S, T)\) the checkerboard corners in lens coordinate system \( C_L \) are known from Equation 5.3. Thus, the \( u \) parameter is determined for each CB corner. As optical focal length \( F \) is assumed to be given, \( v \) can be computed from lens equation Equation 5.1. Also, given the current calibration estimate of \( R, \lambda \), the distance of the a measured CB corner point \( P_m \) from the lens coordinate system \( C_L \) can be computed. This amounts to computing \( s_d \) (Figure 5.1). Given that \( D \) is already known before calibration and \( v, s_d \) has been computed, the blur radius \( r \) in Equation 5.2 can be determined. This can be used to compute the current estimate of blur kernel \( h(x, y) \) from
Equation 5.2 as a function of calibration parameters $U$ and known parameters $F, D$.

5.3 Proposed Approach

The proposed approach consists of the following steps. Since, we are given only the focal stack images which are blurred, there is no guarantee that one of the images will have all the CB corners in focus. Thus, we need to combine all the focal stack images and obtain an omnifocus [5] image where all the CB corners are imaged in focus. The corners in this all focused image can then be reliably measured. But, in order to obtain an omnifocus image, the focal stack images need to be registered first in a global coordinate system which is addressed in Section 5.3.1. The omnifocus image is then created in this global coordinate system as discussed in Section 5.3.2. Since this image is created in a global coordinate
system, it needs to be warped back to align with input focal stack images. This is discussed in Section 5.3.3. Finally, for every input CB focal stack image, we obtain a corresponding sharp omnifocus image. We call this as focused focal stack. The two stacks are then used for combined geometric and blur cue based non-frontal camera calibration.

5.3.1 Registration of Focal Stack

The focal stack acquisition in Section 5.2.2 is dependent on the rotation of the non-frontal camera about a single viewpoint, which is the optic center $O_L$ (Figure 5.3). This has the benefit that consecutive focal stack images are related by a homography relation and can be registered. But, achieving a highly accurate unique viewpoint is a difficult task and there is always some parallax error between various camera poses due to which single viewpoint based homography does not exist and images are not registered accurately. The next best option is to compute planar homography between images of planar checkerboard pattern and then register images. But, due to presence of image distortion and blurring of features in consecutive focal stack images, the planar homography is not accurate enough to register images. Due to all these factors and the fact that even small inaccuracies in pairwise homography can accumulate into large errors while registering multiple images to a single image leads us to seek other techniques of image registration.

Since accurate registration is absolutely necessary for combining focal stack images and computing the omnifocus image, we treat registration as a preprocessing step done using conventional geometric camera calibration. Under this setting, the aperture of the non-frontal camera (Figure 5.2(b)) is closed to maximum possible and then is rotated about $O_L$ in same increments as were used to obtain focal stack images. This enables us to capture pinhole checkerboard images which are sharp and geometrically same as focal stack as shown in bottom row of Figure 5.4 (middle row). We call this pinhole focal stack. The pinhole focal stack is then used to do a conventional calibration using [25]. While, this calibration assumes that the sensor and the lens are parallel, the obtained parameters can be used obtain a numerical map $M$ which projects all pinhole focal stack images to one global coordinate system where they are registered. The registration map $M$ can then be used to register the actual focal stack images. The resulting registered focal stack images are shown in Figure 5.5. Since the registered images have blurring due to interpolation, they are not used for any image processing. Rather, given a pixel location in the registered images, the map $M$ is used to revert back to the original focal stack and that image is processed.
Figure 5.5: (a) Input focal stack images of top row of Figure 5.4 are registered to a global coordinate system (see Section 5.3.1). A fixed pixel location is selected in each of the four images with inset showing the zoomed in intensity inside. Due to registration, they are aligned. (b) Omnifocus image obtained from the registered focal stack on the left (see Section 5.3.2).

5.3.2 Computation of Omnifocus Image

Given the set of registered focal stack images, with each pixel being imaged in focus in at least one of the input images, an omnifocus image of the CB can be created [45, 44, 46]. Such an image denoted here as $I_{\text{omni}}$ has all the scene depths imaged in focus. see Figure 5.5(b) for the omnifocus image obtained for the registered input data in Figure 5.5(a). This image is obtained by computing a focus measure [5] at each pixel location across the focal stack. The focus measure computes the high frequency information in a window around a particular pixel location for all images in the focal stack. The image index which maximizes this measure is then selected as the image from which the focused pixel intensities are selected. Also, note that the focus measure is not computed on the registered images, rather on the actual focal stack images via the mapping $M$ in Section 5.3.1.
5.3.3 Dewarping and Corner Detection

Given the pixel wise mapping $M$ between registered and input focal stack images (Section 5.3.1), the omnifocus image $I_{\text{omni}}$ obtained in Section 5.3.2, can be dewarped to input focal stack images with the only difference that the dewarped images are now focused. The dewarping technique requires the location of corresponding CB corner locations in $I_{\text{omni}}$ and the input focal stack images. The CB corners of focal stack are assumed to be same as that of the corresponding pinhole focal stack images (Section 5.3.1). The CB corners in $I_{\text{omni}}$ are computed by a corner detection technique [25]. Now, since the input focal stack images have radial distortion and the registered images do not have any distortion, a linear mapping e.g. affine or homography is insufficient for accurate dewarping. Thus, we apply a thin plate spline [47] technique to dewarp $I_{\text{omni}}$ into focused version of input focal stack images. These images are geometrically aligned to pinhole images shown in bottom row of Figure 5.4, but differ in image blur. This difference arises from the fact that pinhole focal stack images had negligible Seidel aberrations (spherical aberration, coma, astigmatism) while these are inherently present in focused focal stack obtained by dewarping omnifocus image obtained from wide open aperture. Once the images are dewarped, the corners of the checkerboard are detected and stored. Thus, we have the geometric and blur information both obtained from focal stack images.

5.3.4 Combined Calibration Error Function

Given the current estimate of calibration parameters $U$ (Equation 5.6), the complete calibration error $E_t$ is defined as a sum of geometric re-projection error $E_g$ and the blur error $E_p$. Since images have distortion, the calibration parameter set $U$ is appended with radial distortion terms. It was observed in [4] that if the sensor is modeled as non-frontal, then radial distortion about the optic center is sufficient to model all distortions. The radial distortion model consists of two calibration parameters: $(k_1, k_2)$ relating a pair of distorted $P_S^d = (x_{sd}, y_{sd})$ and undistorted $P_S^u = (x_{su}, y_{su})$ image points in $C_S$ as $P_S^d = P_S^u + g(P_S^u, \delta(k_1, k_2))$, where

$$g(P_S^u, \delta(k_1, k_2)) = \left[x_{su}(k_1 r_{su}^2 + k_2 r_{su}^4), y_{su}(k_1 r_{su}^2 + k_2 r_{su}^4)\right]\ (5.7)$$

where $r_{su} = \sqrt{x_{su}^2 + y_{su}^2}$. Thus, the complete set of calibration parameters is

$$U = (S, T, R, \lambda_{px}, \lambda_{py}, I_0, J_0, a_n, k_1, k_2)\ (5.8)$$

where $\lambda_{px} = \frac{\lambda_p}{s_x}$ and $\lambda_{py} = \frac{\lambda_p}{s_y}$. 

100
5.3.5 Total Calibration Error

Let us assume that we have the estimate of calibration parameters as $U$. We also know the dimension and the location of corners on the CB in the world coordinate system $C_W$. The observations consists of two sets of data: (a) input focal stack which has CB images with varying amounts of blur, (b) a synthesized set of focused focal stack which has been generated by integrating (a) into an omnifocus image and dewarping the omnifocus image back to geometrically align with (a). The combined observation set is now used to design a combination of geometric and blur based error function which should minimize for optimal calibration parameters $U^*$ as described below.

Geometric Error

Given the current estimate $U$ of calibration parameters and the $k^{th}$ known world point $P^k_W$, Equation 5.5 gives the predicted image point $P^k_p = (I^k_p, J^k_p)$ on the image sensor. Additionally, we have measured image coordinates $P^k_m = (I^k_m, J^k_m)$ of CB corners from the focused focal stack images obtained after dewarping of omnifocus image (see Section 5.3.3). The geometric error $E^k_g(U)$ for the $k^{th}$ observed CB corner as a function of $U$ can be defined as:

$$E^k_g(U) = (I^k_p(U) - I^k_m)^2 + (J^k_p(U) - J^k_m)^2 \quad (5.9)$$

Blurring Error

Given current estimate $U$, the blur kernel $h^k(x, y)$ at the $k^{th}$ CB corner across all the focused focal stack images can be computed as explained in Section 5.2.4. Also, we have the actual focal stack images (top row in Figure 5.4) as input. Thus, at each $k^{th}$ geometric corner location $P^k_p = (I^k_p, J^k_p)$ in each focused focal stack image, a square window of size $n \times n$ denoted as $W_f$ around the corner can be blurred using $h^k(I_m, J_m)$ and compared with the corresponding observed $n \times n$ blurred corner $W_b$ in the captured focal stack, then the blur error can be defined as:

$$E_p^k(U) = 1 - NCC(W_b^k, W_f^k \ast h^k(I_m, J_m)) \quad (5.10)$$

where $NCC$ denotes the normalized cross correlating [48] between the two patches and its value lies between $-1$ and $1$, where $1$ denotes higher correlation. The use of $NCC$ was justified as it is robust to intensity changes due to lens vignetting effects in the captured focal stack.
Thus, the total geometric and blur based error for all CB corner locations for all the images in the focal stack is

\[ E_t(U) = \sum_{k=1}^{N} E^k_g(U) + E^k_p(U) \] (5.11)

where \( N \) is the total number of CB corners in all the input focal stack images. This error is then minimized as:

\[ U^* = \arg\min_{U} E_t(U) \] (5.12)

to get the optimal calibration parameters \( U^* \) using Levenberg-Marquardt algorithm [49].

5.4 Experiments

5.4.1 Data Acquisition

The data is acquired using a custom made AVT Marlin camera, fitted with 1/2 inch Sony CCD sensor tilted by \( \approx 3-4 \) degrees with respect to the lens plane and a C-mount Schneider Cinegon 1.4/8 mm compact lens. The acquired image resolution is \( 640 \times 480 \). This camera is first centered empirically and the images of the checkerboard (CB) are captured by rotating the camera with a closed aperture (pinhole focal stack acquisition) and a wide aperture (focal stack imaging). In total, five images are captured in each setup without changing any other experimental conditions. A sample of four images is shown in Figure 5.4 (top row). The CB is custom made to get high positional accuracy of the corners. The size of each square in the CB is \( 5 \times 5 \) mm.

5.4.2 Results: Real Data

We estimate \( U^* \) using two methods and show that the proposed geometric and blur based approach clearly outperforms geometric-only calibration by comparing \( E_t(U^*) \) and the variance of \( U^* \). The two methods are: (A) Conventional: Geometric calibration using pinhole focal stack images where \( E_t(U) = \sum_{k=1}^{N} E^k_g(U) \) and (B) Proposed: Geometric and Blur, where the CB corners are taken from focused focal stack images and blurred images are from input focal stack (\( E_t(U) \)).
Table 5.1: Estimated calibration parameters and their standard deviation for one image

<table>
<thead>
<tr>
<th>Method</th>
<th>method (A)</th>
<th>method (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$U^*$</td>
<td>standard deviation</td>
</tr>
<tr>
<td>$\lambda_x = \frac{\Delta}{s_x}$</td>
<td>847.812</td>
<td>4.007</td>
</tr>
<tr>
<td>$\lambda_y = \frac{\Delta}{s_y}$</td>
<td>845.380</td>
<td>3.389</td>
</tr>
<tr>
<td>CoD</td>
<td>$l_0$</td>
<td>239.753</td>
</tr>
<tr>
<td></td>
<td>$J_0$</td>
<td>330.773</td>
</tr>
<tr>
<td>Radial</td>
<td>$k_1$</td>
<td>$-0.0023$</td>
</tr>
<tr>
<td>distortion</td>
<td>$k_2$</td>
<td>$5.7e-05$</td>
</tr>
<tr>
<td>Tilt (deg)</td>
<td>$\cos^{-1}(r_{33})$</td>
<td>0.750</td>
</tr>
<tr>
<td>Re-projection</td>
<td>$E_g$</td>
<td>0.044911</td>
</tr>
</tbody>
</table>

The calibration estimates $U^*$ and their corresponding standard deviation along with total geometric re-projection error, computed from methods (A) and (B) are shown in Table 5.1 and Table 5.2 for one and five images in the focal stack. From Table 5.1, it is observed that the re-projection error $E_g(U^*)$ using proposed method (B) is much less than those obtained from method (A). Similarly, the standard deviation of the obtained estimates is least for the proposed technique. Also, comparing the estimates of tilt of the sensor, it is found that method (A) estimates the tilt to be 0.75 degrees which is very different from the specifications provided by the manufacturer. But the estimates from method (B) which is 4.260 degrees is much closer to the camera specifications. The tilt is a critical parameter in depth from focus/defocus techniques [10].

Next we analyze the results of using more images for calibration. In this case also, the best performer with respect to total error and standard deviation of estimated parameters is method (B). We also observe that for the proposed technique, only one image is sufficient to get accurate results. Finally, in Figure 5.6, we plot the estimated blur circle $r$ (Equation 5.2(b)) obtained from the calibration results of Table 5.2 using method (B), where the inset shows the blurred image and the predicted blur radius in detail. Since, the sensor is tilted, the left part of the image in Figure 5.6(a) is focused while the right part is defocused. This behavior is clearly observed by the estimated blur kernel sizes at the CB locations in Figure 5.6(a), where the size increase from left to right along the image horizontal.

5.5 Discussion

In this chapter, we have presented a framework for non-frontal camera calibration using geometric properties and image blur given a focal stack. We have shown improved results
Figure 5.6: (a) Blurred focal stack image and (b) blur circle estimates from optimal $U^*$ using Equation 5.2 (inset shows details). The blurring of the CB corner looks consistent with the radius of the estimated Gaussian blur kernel (best seen in color).

Table 5.2: Estimated calibration parameters and their standard deviation for five images

<table>
<thead>
<tr>
<th>Method</th>
<th>method (A)</th>
<th>method (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$U^*$</td>
<td>standard deviation</td>
</tr>
<tr>
<td>$\lambda_x = \frac{\lambda}{s_x}$</td>
<td>848.535</td>
<td>1.633</td>
</tr>
<tr>
<td>$\lambda_y = \frac{\lambda}{s_y}$</td>
<td>847.062</td>
<td>1.579</td>
</tr>
<tr>
<td>CoD</td>
<td>$I_0^*$</td>
<td>238.804</td>
</tr>
<tr>
<td></td>
<td>$J_0^*$</td>
<td>331.596</td>
</tr>
<tr>
<td>Radial distortion</td>
<td>$k_1$</td>
<td>$-0.0022$</td>
</tr>
<tr>
<td></td>
<td>$k_2$</td>
<td>$5.2e-05$</td>
</tr>
<tr>
<td>Tilt (deg) $\cos^{-1}(r_33)$</td>
<td>2.487</td>
<td>2.856</td>
</tr>
<tr>
<td>Re-projection</td>
<td>$E_g$</td>
<td>0.045093</td>
</tr>
</tbody>
</table>

using the proposed technique in terms of the estimation error and the variance of the estimated parameters. The future work would be focused on making this technique independent of pinhole data by devising accurate image registration techniques to handle problems like parallax, radial distortion etc.
6.1 Introduction

In this chapter, we consider the classical problem of calibrating a small field of view camera whose imaging model consists of an ideal perspective projection followed by an additional small distortion of the image rays before they form the image. The overall distortion on the image plane is typically modeled as a combination of radial and decentering terms, which are typically infinite series involving terms of image coordinates. As this distortion modeling is approximate in nature, some of the prior work has looked into more accurate physical modeling. For the case of decentering distortion, a non-frontal sensor model has previously been developed and shown to improve calibration accuracy. But a similar physical modeling of radial distortion has not been considered as a part of camera calibration. We hypothesize in this chapter that radial distortion can be physically explained as occurring due to the variation in the location of the entrance pupil (image of aperture stop) for image rays incident from different directions in a pupil-centric imaging setting. Thus, both are geometrically equivalent. The state of the art [4] in incorporating moving entrance pupil into camera calibration also employs traditional radial distortion correction, which we show using our analysis to be redundant. We show that a new camera calibration technique based on a combination of pupil-centric model, moving entrance pupil and non-frontal sensor, without traditional radial and decentering modeling, outperforms state-of-the-art results in small field of view camera calibration methods in terms of pixel re-projection error.

Camera calibration entails estimation of intrinsic and extrinsic parameters of a camera given a set of known world points and their measured image coordinates on the image plane [11, 14, 15, 13, 16]. The intrinsic parameters model the geometric properties of the image sensor and the extrinsic properties model the pose of the camera in a known coordinate system.

Camera calibration depends on the imaging model that governs the camera optics. A small field of view camera typically follows a perspective projection model, where as a small focal
length camera might follow a fish-eye projection model. The reader can review [4] to get an overview of different projection models including perspective and fish-eye. In this chapter, we focus on small field of view cameras following perspective projection. Typically, image formation in such cameras is never strictly perspective due to departure in ideal behavior of camera optics causing image rays to deviate from ideal projection. This variation in optical behavior is sometimes intentional to alleviate the effect of chromatic/achromatic lens aberrations [42] and sometimes unintentional, e.g. small lens-sensor misalignment due to manufacturing limitations. The geometric effects of such deviations distort straight lines in the real world which get imaged as curved lines. Traditionally, image distortion has been modeled as a combination of radial distortion [11] to model radially outward movement of ideal image points and decentering distortion to model keystone like distortions owing to small lens-sensor misalignment [15, 35]. Both of these distortion models are in the form of an infinite series function of ideal image coordinates [15, 16, 13]. Since, they are not physically motivated, the models are approximate in nature. Some prior work has focused on explicit physical modeling of decentering effects by introducing rotation matrix parameterization for lens-sensor tilt [4]. Kumar and Ahuja [1] have obtained lower pixel re-projection errors by combining this rotation formulation to a pupil-centric imaging model.

In this chapter, we propose that radial distortion on the image plane can also be geometrically modeled by assuming a moving entrance pupil for various rays incident on the lens in a pupil-centric imaging setting. Previously, Gennery [4] had combined moving entrance pupil with radial distortion modeling, which this work shows to be redundant.

Our main contributions in this chapter are:

1. The idea of the equivalence of moving entrance pupil and radial distortion as explained in Section 6.2. The impact of this is that we are able to give a physically more meaningful model of radial distortion in images.

2. We propose a new calibration algorithm based on these ideas which gives minimal pixel re-projection error compared to all existing state-of-the-art calibration methods for perspective cameras. See Section 6.3.

3. We also present a survey of various other calibration algorithms and their underlying imaging model related assumptions and show how the proposed model is new and differs from them in Section 6.4.

4. Finally, in Section 6.5, we compare our calibration results with those obtained by these methods and show that we get least re-projection error without increasing the
parameterization of the calibration problem, i.e. we use the same number of parameters to compare all the algorithms.

6.2 The New Insight on Radial Distortion

**Notations:** See Figure 6.1. The entrance pupil is defined as the image of the aperture stop as seen from an axial point on the object through lens elements preceding the aperture stop. Similarly, the exit pupil is the image of the apertures stop as seen from an axial point on the image plane through lens elements between the aperture stop and the image plane. For the purposes of camera calibration, most of the prior works assume these points to be fixed, except [4] which model the fact that entrance-pupil could vary depending on the incident ray direction.

![Figure 6.1: Example of entrance pupil, exit pupil and aperture stop [42].](image)

Our main hypothesis is that if it is assumed that the entrance pupil of the imaging system is never fixed, then the radial deviation of image rays from ideal perspective projection can be explained by the motion of the entrance pupil. Thus, we get a geometric modeling of the observed radial distortion on the image plane. In fact, we show from quantitative results on real data (Table 6.1) that the traditional calibration model [4] of having both entrance pupil movement and radial distortion is redundant.
6.2.1 Conventional Explicit Radial Distortion Modeling

Let an ideal perspective projection of a world point on the image plane be \( P_p = (x_p, y_p) \). Due to distortion, this point actually appears at location \( P_d = (x_d, y_d) \). If there is no decentering of lens-sensor, then these two points can be related by two radial distortion parameters \((k_1, k_2)\) as

\[
\begin{bmatrix}
  x_d \\
  y_d 
\end{bmatrix}
= (1 + k_1 \cdot r_p^2 + k_2 \cdot r_p^4)
\begin{bmatrix}
  x_u \\
  y_u 
\end{bmatrix}
\] (6.1)

where \( r_p = \sqrt{x_u^2 + y_u^2} \) is the radial distance of the ideal image point from the center of radial distortion. This is the conventional modeling of radial distortion on the image plane [11, 15, 13, 16].

6.2.2 Thin-Lens Imaging Model: Stop Location and Radial Distortion

We consider a thin lens setting as shown in Figure 6.2, where the chief ray is responsible for image formation and the stop location varies in each of the figures. We note from Hecht [42],

![Figure 6.2](image)

Figure 6.2: (a) No-distortion: aperture stop coincides with the principal point \( O \). (b, c) Barrel and pin-cushion distortion due to change in stop location.

that the location of the aperture stop invariably causes radial distortion (see Chapter 6.
in [42]) as compared to an ideal image formed by a principal ray. Here, the principal ray is defined as the ray which passes through the intersection of the principal planes and the optic axis. Such rays do not deviate. The introduction of an aperture stop can block the principal ray causing the chief ray to become the main image forming ray. This in turn leads to deviation of chief ray from going on a straight path causing barrel or pin-cushion distortion as shown in Figure 6.2 (b, c). The ideal case, where the principal ray coincides with the chief ray, is termed as orthoscopic projection.

6.2.3 Thick Lenses Imaging Model: Stop Location and Radial Distortion

Now, we extend the analysis from Hecht [42] to the case of thick lenses, which are a more practical representation of the imaging systems typically used and needed to be calibrated. For such lenses, we first describe the pupil-centric imaging geometry.

![Figure 6.3: Pupil-centric imaging model for thick lenses.](image)

Figure 6.3 shows the pupil-centric imaging model which is parameterized by a set of lens parameters. These parameters include the location of the entrance pupil $E_n$, the location of exit pupil $E_x$ and the front and back principal planes $H_1, H_2$ respectively and are assumed to be known apriori from lens manufacturer specifications. The chief ray which is the image forming ray passes through $E_n$ and exits through $E_x$, making an angle of $\theta_{in}$ at $E_n$ and $\theta_{out}$ at $E_x$. It was shown in [17], that $\theta_{in}$ and $\theta_{out}$ are related as

$$\tan(\theta_{in}) = \frac{F - a_x}{a_x} \tan(\theta_{out}) \quad \text{(6.2)}$$

where $F$ is the optical focal length of the system. If $y_o$ an $y_i$ are object and image heights respectively and $z_o$ an $z_i$ are object and image distances from $E_n$ and $E_x$ respectively, then
we have \( \tan(\theta_{in}) = \frac{y_i}{z_o} \) and \( \tan(\theta_{out}) = \frac{y_i}{z_i} \). The transverse magnification \( M_T \) [42] of the system can then be computed from Equation 6.2 as:

\[
M_T = \frac{y_i}{y_o} = \frac{F - a_x}{F} \cdot \frac{z_i}{z_o}
\]  

(6.3)

If \( z_o \) and \( z_i \) are fixed, then \( M_T \) is fixed as \( a_x \) and \( F \) are fixed. Thus, we can conclude that for fixed lens parameters there is a constant magnification factor relating all object points on a fixed plane at \( z_0 \) and the captured image points. But, we know from observed images captured from a thick lens, that magnification is not constant across the image plane, rather it either increases or decreases as the image points move far from the center of the image. Or in other words, the magnification \( M_T \) is radially varying causing observed radial distortion of image points. Thus, we can give the following hypothesis.

Figure 6.4: Angle of incident light ray moves the entrance pupil leading to different amounts of radial distortion: (a) small incident angle (b) large incident angle.
6.2.4 Our Hypothesis: Moving Entrance Pupil Explains Radial Distortion in Pupil-Centric Imaging

The only way to explain this phenomenon of varying magnification/radial distortion over the image plane is to assume that all of the ray geometries shown in Figure 6.2(a,b,c) occur simultaneously. But, this would imply that there are multiple aperture stop locations in the imaging system at the same time. Since, this is physically impossible, we hypothesize that in the case of a thick lens, the entrance pupil varies its location monotonically depending on the angle $\theta_{in}$ of incidence of the incoming ray. Thus, for small angles of incidence (Figure 6.4(a)), the image projection would behave like an orthoscopic projection (similar to thin-lens scenario of Figure 6.2(a)) leading to almost perspective imaging/less distortion. This hypothesis agrees with the observation that there is almost no radial distortion at regions near the image center, where $\theta_{in}$ is small. As the incident ray angle $\theta_{in}$ increases (Figure 6.4(b)), the entrance pupil location changes and the imaging system behaves similar to that of either of Figure 6.2(b, c). This again confirms to the observation that radial distortion is higher toward the periphery of the image. Thus, we can conclude that the movement of the entrance pupil as a function of incident ray angle $\theta_{in}$ is a predominant geometric reason for the occurrence of radial distortion on the image plane.

It must be noted that the movement of entrance pupil is a known phenomenon for fish-eye lenses and has been included as a part of camera calibration by Gennery [4], but in his work he also included the traditional radial distortion correction [11] as an additional step. Comparatively, we have shown from our previous analysis that moving entrance pupil in itself manifests the physical effect of radially distorting ideal perspective image points and thus is not required as a part of calibration. We believe that Gennery [4] was not able to analyze this, since the imaging model he considers in his paper is not pupil-centric, i.e. the paper assumed $\theta_{in} = \theta_{out}$ in Figure 6.3, which incidentally is a common assumption in most of the previous calibration literature [15, 13, 16].

Next, based on our hypothesis, we present our new camera calibration model.

6.3 Proposed Camera Calibration Modeling

The proposed camera calibration model assumes pupil-centric imaging with moving entrance pupil location to model the path of image forming chief rays which project a known world point onto the image plane. Based on the earlier analysis of Section 6.2, we exclude traditional radial distortion correction. We also discard modeling of decentering distortion and assume that the physical decentering of image sensor plane with respect to the real principal
plane (see Figure 6.5) can be modeled via a rotation matrix. Such a modeling [1] was earlier shown to be more accurate than decentering model [15, 13] for camera calibration as it could handle generic lens-sensor misalignments.

Our complete calibration model is shown in Figure 6.5, where a known world point location \( P_w \) is imaged to a pixel location \( P_i \) on the image plane based on the calibration parameter estimates and \( \hat{P}_i \) is the measured image coordinate. The coordinates of these points and the transformation between them are defined in terms of four right-hand coordinate systems namely:

- **World coordinate system (\( C_W \)):** The known world points are defined in this coordinate system, e.g. a coordinate system lying on the surface of a known flat checkerboard pattern as in our work.

- **Entrance pupil coordinate system (\( C_{E_n} \)):** The origin of this coordinate system lies at the entrance pupil \( E_n \) of the imaging system. Since, in our case \( E_n \) is assumed to be moving with respect to the incident image ray (Section 6.2), we define the origin of \( C_{E_n} \) to correspond to the location where the incidence rays with \( \theta_{in} \approx 0 \) intersect the optic axis. This location can typically be obtained from the lens data-sheet as a signed distance from the front principal plane.

- **Sensor coordinate system (\( C_S \)):** The \( xy \)-plane of this coordinate system lies on the image sensor with the origin lying at the intersection of the optic axis and the sensor plane. This intersection point will be called as center of distortion (CoD) henceforth.

- **Image coordinate system (\( C_I \)):** The measured pixel values are described in this coordinate system. In this chapter, we assume it lies at the top-right corner of the image sensor.

The distances are measured in pixel values in \( C_I \) and in metric (e.g. mm) in the remaining coordinate systems \( C_W, C_{E_n}, C_S \).

Given multiple world and image point correspondences: \( P_w \leftrightarrow \hat{P}_i \), the goal of camera calibration is to estimate the transformation between all the above four coordinate systems. The transformations in turn encode the intrinsic and extrinsic calibration parameters of the camera.

See Figure 6.5. Let the world point be \( P_w = (X, Y, Z) \) and the measured image point be \( \hat{P}_i = (\hat{I}, \hat{J}) \). Let the signed distance of \( E_n \) from the front principal plane \( (H_1) \) be \( a_n \) and of exit pupil \( E_x \) from the back principal plane \( (H_2) \) be \( a_x \). Then, assuming no noise, we have the following transformations relating \( P_w \) and \( \hat{P}_i \).
6.3.1 Transformation from $C_W$ to $C_{E_n}$

Let $S = \{s_{ij} : 1 \leq (i, j) \leq 3\}$ be the rotation matrix and $T = (t_x, t_y, t_z)$ be the translation between these two coordinate systems. Then, $P_w$ can be expressed as $P_l = (x_l, y_l, z_l)$ in $C_{E_n}$ as:

$$
\begin{bmatrix}
  x_l \\
  y_l \\
  z_l
\end{bmatrix} =
\begin{bmatrix}
  s_{11} & s_{12} & s_{13} \\
  s_{21} & s_{22} & s_{23} \\
  s_{31} & s_{32} & s_{33}
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z
\end{bmatrix} +
\begin{bmatrix}
  t_x \\
  t_y \\
  t_z
\end{bmatrix}
$$

(6.4)

6.3.2 Incorporating Entrance Pupil Movement

Let the ray from $P_w$ be incident on the optical axis at $E_n(\theta)$ at an angle of $\theta$. Let the distance between $E_n$ and $E_n(\theta)$ be $\sigma(\theta)$, where we use the model for $\sigma(\theta)$ from [4] as:

$$
\sigma(\theta) = \left(\frac{\theta}{\sin(\theta)} - 1\right) (\epsilon_1 + \epsilon_2 \theta^2 + \ldots)
$$

(6.5)
Here \((\epsilon_1, \epsilon_2)\) are the pupil movement parameters. As the distance moved by the pupil \(\sigma(\theta)\) and \(\theta\) are dependent on each other they cannot be determined independently. But, a simple trigonometric equation can be derived in terms of \(P_l\) and \(\theta\) [4] as:

\[
z_l \sin(\theta) - \sqrt{x_l^2 + y_l^2} \cos(\theta) - (\theta - \sin(\theta))(\epsilon_1 + \epsilon_2 \theta^2) = 0 \tag{6.6}
\]

and can be solved iteratively for \(\theta\) using Newton-Raphson method using initial value of \(\theta\) as \(\arctan\left(\frac{\sqrt{x_l^2 + y_l^2}}{z_l}\right)\) and current estimates of calibration parameters \(S, T, \epsilon_1\) and \(\epsilon_2\). An example plot of the variation of entrance pupil center as incident ray angle \(\theta\) changes is shown in Figure 6.6 for the lens used in our experiments. As can be seen, the default \(E_n\) provided by the manufacturer has been calibrated for incident rays with \(\theta \approx 50\) degrees.

![Figure 6.6: Moving entrance pupil vs. angle of incidence ray. The plot is based on \((\epsilon_1 = -5.304, \epsilon_2 = 6.474)\) computed from calibration results from our proposed method as shown in the last column of Table 6.1.](image)

### 6.3.3 Computing Incident Ray Intersection with Front Principal Plane \(H_1\)

Given the computed ray direction \(\theta\), the incident ray from \(P_w\) intersects the front principal plane \(H_1\) at some location \(q_1\). Using simple coordinate geometry, the coordinates of \(q_1\) can
be computed as

\[ q_1 = \frac{-(a_n + \sigma(\theta))}{z_l - \sigma(\theta)} \begin{bmatrix} x_l \\ y_l \end{bmatrix} \] (6.7)

Since the point of incidence on the front and back principal planes are same, we have that \( q_1 = q_2 \).

### 6.3.4 Computing the Exitance Ray from \( E_x \) to \( q_2 \)

As is known from pupil-centric geometry that the exit ray must appear to come from exit pupil \( E_x \), the intersection of this ray with the non-frontal sensor plane can be computed. We assume that the image sensor is non-frontal in nature and is rotated with respect to \( H_2 \) by a two parameter rotation matrix \( R(\alpha, \beta) \), where \((\alpha, \beta)\) are right-hand Euler angles of rotation about the \( x \)- and \( y \)- axis of coordinate system at \( H_2 \) which will align \( H_2 \) with the non-frontal sensor \( C_S \) [1]. We also assume that the origin of the non-frontal sensor is located at a perpendicular distance of \( \lambda \) from \( H_2 \).

The exit pupil location in back principal plane can be obtained as \( E_x = (0, 0, a_x) \), where \( a_x \) is a pupil-centric parameter obtained from the lens data-sheet. Now, both \( E_x \) and \( q_2 \) can be obtained in terms of \( C_S \) as:

\[ E_x^{CS} = R \begin{bmatrix} 0 \\ 0 \\ a_x + \lambda \end{bmatrix} \] (6.8)

\[ q_2^{CS} = R \begin{bmatrix} q_2(x) \\ q_2(y) \\ \lambda \end{bmatrix} \] (6.9)

The intersection of this ray with the image sensor can then be obtained using simple coordinate geometry as \( P_s = (x_s, y_s) \).

### 6.3.5 Obtaining the Image Coordinates

The image sensor points are in metric units (e.g. mm), and need to be transformed to pixel coordinates in image coordinate system \( C_I \). If the pixel sizes are \( s_x \) and \( s_y \) and the location of origin of \( C_S \) in \( C_I \) is \((I_0, J_0)\) pixels, then we can obtain the predicted image pixel coordinates
\( P_i = (I, J) \) as

\[
\begin{bmatrix}
    I \\
    J
\end{bmatrix} = \begin{bmatrix}
    \frac{x_i}{sx} - I_0 \\
    \frac{y_i}{sy} - J_0
\end{bmatrix}
\] (6.10)

We refer to \((I_0, J_0)\) as the center of distortion (CoD). The predicted image point \(P_i(U)\) is thus obtained as a nonlinear function of a set of 14 calibration parameters \(U\) which need to be estimated. These parameters compose the world to image plane transformations described earlier and can be enumerated as:

\[ U = \{S, T, R(\alpha, \beta), s_x, \lambda, I_0, J_0, \epsilon_1, \epsilon_2\} \] (6.11)

### 6.3.6 Linear and Nonlinear Optimization

In this section, we describe the calibration algorithm implementation. The calibration is done in two stages (1) initial linear estimation and (2) final nonlinear refinement using the estimated parameters from linear estimation. For stage (1), we assume that the entrance pupil is fixed and used the analytical technique in [1] to get the linear estimate. As this linear estimation technique was derived for 3D scene points, we adapt their method for 2D scene points. The final nonlinear estimation is done using Levenberg-Marquardt optimization [13] by minimizing the pixel re-projection error over all \(N\) world-image point correspondences as:

\[
U^* = \arg \min_U \sum_{n=1}^{N} ||\hat{P}_i^n - P_i^n (U)||_2^2
\] (6.12)

where \(U^*\) is the final optimal results. The results in the columns of Table 6.1 for various calibration techniques correspond to \(U^*\) from respective methods.

### 6.4 Comparison with Other Calibration Algorithms

In this section, we describe prominent existing state-of-the-art techniques in camera calibration with which we would be comparing our proposed model’s calibration accuracy in the results section (Section 6.5). These methods vary with respect to each other in terms of the imaging model, the distortion model and the orientation of the image sensor. We next
explain these criteria and various models which fall under these criteria.

- Imaging model: The imaging model describes the image formation from the world point to the image point. There are two types of imaging model being employed in camera calibration. The basic thin-lens model assumes that the incident and exiting rays responsible for image formation by the optical system are principal rays which pass through the optic center of the system of lenses and are parallel to each other [11, 14, 15, 13, 16]. The second model called as the pupil-centric model assumes that the image rays responsible for image formation are the chief rays which enter the imaging system at the entrance pupil and appear to exit from the exit pupil [42, 17]. In this model, the location of the entrance pupil can either be assumed to be fixed [1] or it can be assumed to be moving [4].

- Distortion model: Real imaging systems behave far from ideal perspective projection and are often accompanied by some amount of visible distortion on the image plane where straight lines in real world are imaged as curves. The distortion can be modeled as a combination of radial and decentering distortion [11, 14, 15, 13, 16]. The model is basically an infinite series function of the ideal image points.

- Orientation of the image sensor: Many times the imaging surface may not be normal to the physical optic axis of the lens system due to manufacturing limitations or sometimes to achieve special effects, e.g. tilt-shift effect. Traditionally, it has been assumed that there exists an effective optic axis which is normal to the image sensor plane. This is referred to as a frontal sensor [15, 13] model. Recently, it has been proposed that calibration can be designed about the physical optic axis by assuming that the sensor is non-frontal with respect to the lens ($H_2$) plane. The non-frontalness can be modeled as a two parameter rotation matrix relating the lens plane and the image sensor plane. This is called as non-frontal sensor modeling [4, 1].

Based on the above criteria, we can classify many prior camera calibration techniques into the following three categories, shown graphically in Figure 6.7.

1. Category 1: See Figure 6.7(a) for the calibration model in this category. A number of existing calibration methods fall in this category including those proposed in Weng et al. [15], Heikkilä and Silvén [13], Zhang [16]. In this category, imaging is assumed to be thin-lens, the sensor is frontal and image distortion is modeled as a combination of explicit radial and decentering distortion.
2. Category 2: See Figure 6.7(b) for the complete model. Here the imaging model is thin-lens and the image forming incident and the exiting rays are incident at the entrance pupil location instead of the optic center. But, the entrance pupil is not fixed and is assumed to be moving depending on the incident ray angle. The image sensor is assumed to be non-frontal, i.e. calibration is modeled about the physical optic axis. Any observed image distortion is modeled as explicitly being radial about the physical optic axis. This model has been proposed by Gennery [4].

3. Category 3: See Figure 6.7(c) for this model. The imaging model here is pupil-centric with incident ray entering the lens system at the entrance pupil and exiting the system at the exit pupil. The entrance pupil is assumed to be fixed. The image sensor is assumed to be non-frontal and it is shown that sensor non-frontalness compensates for decentering distortion adjustment typically done in Category 1 techniques. The only distortion that is modeled is radial distortion using traditional infinite series formulation. This method has been proposed by Kumar and Ahuja [1].
In comparison to all these methods, our proposed method does not fall in any of these categories as our imaging model is pupil-centric, assumes moving entrance pupil and the image sensor is assumed to be non-frontal and we do not propose to explicitly model the image distortion based on our analysis on equivalence of moving entrance pupil and observed radial distortion. We also incorporate non-frontal sensor model as it physically corresponds to decentering distortion effects and is more robust for large sensor tilts.

6.5 Results

6.5.1 Calibration Data

The calibration data consists of a precisely constructed glass checkerboard with $5 \times 5$ mm squares. Since the checkerboard is transparent, it is back lit to generate white and black squares on the captured calibration images. The checkerboard is fixed at a location and a camera with a tilted image sensor is used to capture a set of five images of the checkerboard from different viewpoints. A tilted sensor camera is useful in validating the non-frontal modeling in [4, 1]. The corners from the checkerboard images are detected using MATLAB Bouguet’s toolbox [25]. The accuracy of corner detection is separately calculated using the method of [20] and is found to be $\approx 0.011$ pixels.

6.5.2 Camera Specifications

We use an AVT Marlin F033C camera with a custom made image sensor which has been slightly tilted by about 3 degrees. This camera is fitted with a Cinegon $1.4/8.2$ mm lens. The data-sheet [24] of the lens provides the pupil-centric parameters of the camera. In Figure 6.8, we show the various parameters provided by the manufacturer. Out of these numbers, the two numbers which we use in our calibration are the distance of the entrance pupil from the front principal plane denoted as $H_1E_n(a_n)$ and the distance of the exit pupil from the rear/back principal plane denoted as $H_2E_x(a_x)$. Simple computations from Figure 6.8 lead to $a_n = 6.5$ mm and $a_x = 31.4$ mm. We use these values in our calibration method in Equation 6.7 and Equation 6.8.
Distance Sign Convention
L₁ = Front Glass Vertex
L₂ = Rear Glass Vertex
Eₐ = Entrance Pupil
Eₓ = Exit Pupil
F = Front Focal Point
F’ = Rear Focal Point
H₁ = Front Principal Plane
H₂ = Rear Principal Plane
Principal planes
H₁Eₐ = 6.5 mm
H₂Eₓ = 31.4 mm

Figure 6.8: Lens data-sheet values for Cinegon 1.4/8 mm lens.

6.5.3 Analysis of Calibration Results

Here, we present the results of proposed calibration method and compare the pixel re-projection error with the representative algorithms for each of the calibration categories mentioned in Section 6.4. The results for these prior techniques and our current method (last column) are shown in Table 6.1. The different imaging conditions have been abbreviated in the caption of Table 6.1.

It can be seen that with the same number of parameters in all the calibration methods, our new method is able to achieve minimum re-projection error of 0.075673 pixels. The second column of this table corresponds to the implementation of Heikkilä and Silvén [13]. Here, the estimate of the CoD (218.647, 330.477) corresponds to the principal point on the image plane where the effective optic axis is normal to the sensor. In the third and fourth columns, we shown the results obtained from a Category 2 calibration method of Gennery [4]. Here, we implement two variations of their method. The method in column labeled as GenneryA uses a thin-lens imaging model with radial distortion and non-frontal sensor and a fixed entrance pupil location. The fixed entrance pupil allows us to conduct calibration over the same number of intrinsic calibration parameters, namely 8, as in our proposed method. The re-projection error here is higher than all other methods. The estimate of sensor tilt of
≈ 0.424 degrees is also far from the known lens specifications. The method in GenneryB is a full implementation of [4] where we have thin-lens imaging, moving entrance pupil, non-frontal sensor and radial distortion parameterization. The entrance-pupil model adds two more calibration parameters making the number of intrinsic parameters 10. Thus, the re-projection error of GenneryB is less than GenneryA since we have used more number of parameters, yet it is more than our proposed method in last column. In the fifth column, we implement the calibration method of Kumar and Ahuja [1] from Category 3, where pupil-centric imaging model is used along with non-frontal sensor and radial distortion. Since decentering is encompassed in non-frontal sensor model, it is not calculated. The re-projection error of 0.075817 pixels is the second best here and the sensor tilt estimate is close to lens specifications. Finally, the sixth and the last column presents the calibration results from the proposed method in this paper. There is no explicit radial or decentering distortion modeling. We observe that the re-projection error of 0.075673 pixels is minimal for our case across all compared methods. The standard deviation of a set of intrinsic calibration parameters is also shown in Table 6.2. The deviation in the estimate of center of distortion is minimal in our case. An interesting observation is that the standard deviation of sensor tilt angle \( \beta \) for Kumar and Ahuja [1] is 0.230 which is close to the difference of \( \beta \) estimates of 3.278 degree and 3.051 degree obtained by them and our current method. Thus, our estimates of sensor tilt (\( \alpha, \beta \)) have better confidence levels.

Table 6.1: Calibration results on real data (TL = Thin-lens, PC = Pupil-centric, ME = Moving entrance, NF = Non-frontal, F = Frontal)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>distortion model</td>
<td>Radial+Decentering</td>
<td>Radial</td>
<td>Radial</td>
<td>Radial</td>
<td>–</td>
</tr>
<tr>
<td>imaging Model</td>
<td>TL</td>
<td>TL+ME</td>
<td>PC</td>
<td>PC+ME</td>
<td></td>
</tr>
<tr>
<td>sensor orientation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scale</td>
<td>( \frac{\lambda_p}{f} ) (mm)</td>
<td>1.0003</td>
<td>1.0005</td>
<td>1.0005</td>
<td>0.9988</td>
</tr>
<tr>
<td></td>
<td>( \lambda_p ) (mm)</td>
<td>8.240</td>
<td>8.383</td>
<td>8.358</td>
<td>8.6496</td>
</tr>
<tr>
<td>principal point([13])</td>
<td>( I_0 )</td>
<td>218.647</td>
<td>223.61</td>
<td>223.85</td>
<td>229.66</td>
</tr>
<tr>
<td>center of distortion([4, 1])</td>
<td>( J_0 )</td>
<td>330.477</td>
<td>327.46</td>
<td>327.30</td>
<td>332.21</td>
</tr>
<tr>
<td>radial distortion</td>
<td>( k_1 )</td>
<td>-0.0019</td>
<td>-1.9e-03</td>
<td>-4.3e-03</td>
<td>-0.002</td>
</tr>
<tr>
<td></td>
<td>( k_2 )</td>
<td>.000034</td>
<td>4.2e-05</td>
<td>3.8e-05</td>
<td>.00004</td>
</tr>
<tr>
<td>entrance-pupil movement</td>
<td>( \epsilon_1 )</td>
<td>–</td>
<td>–</td>
<td>8.122</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>( \epsilon_2 )</td>
<td>–</td>
<td>–</td>
<td>15.178</td>
<td>–</td>
</tr>
<tr>
<td>decentering</td>
<td>( p_1 )</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>distortion</td>
<td>( p_2 )</td>
<td>.000015</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>image Sensor rotation</td>
<td>( \alpha )</td>
<td>–</td>
<td>0.105</td>
<td>0.114</td>
<td>–0.424</td>
</tr>
<tr>
<td>(degrees)</td>
<td>( \beta )</td>
<td>–</td>
<td>0.424</td>
<td>0.439</td>
<td>3.278</td>
</tr>
<tr>
<td>re-projection Error</td>
<td>0.077315</td>
<td>0.078539</td>
<td>0.077878</td>
<td>0.075817</td>
<td>0.075673</td>
</tr>
</tbody>
</table>
Table 6.2: Standard deviation of calibration parameters shown in Table 6.1

<table>
<thead>
<tr>
<th>Calibration Method</th>
<th>$\lambda_{px}$</th>
<th>$\lambda_{py}$</th>
<th>$I_0$</th>
<th>$J_0$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heikkilä and Silvén [13]</td>
<td>0.955</td>
<td>0.944</td>
<td>0.457</td>
<td>0.387</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>GenneryA [4]</td>
<td>0.131</td>
<td>0.127</td>
<td>0.559</td>
<td>0.625</td>
<td>0.029</td>
<td>0.027</td>
</tr>
<tr>
<td>GenneryB [4]</td>
<td>0.173</td>
<td>0.170</td>
<td>0.411</td>
<td>0.431</td>
<td>0.022</td>
<td>0.023</td>
</tr>
<tr>
<td>Kumar and Ahuja [1]</td>
<td>0.674</td>
<td>0.650</td>
<td>1.042</td>
<td>0.336</td>
<td>0.038</td>
<td>0.230</td>
</tr>
<tr>
<td>Ours</td>
<td>1.176</td>
<td>1.180</td>
<td>0.288</td>
<td>0.239</td>
<td>0.031</td>
<td>0.083</td>
</tr>
</tbody>
</table>

6.6 Image Undistortion

In our paper, an ideal undistorted image is the one which is formed when all the image rays from the scene pass through the same entrance-pupil location and thus have a fixed magnification as given by Equation 6.3. This entails predicting the intersection of red dotted line, corresponding to the image ray from $P_w$ which passes through the ideal entrance pupil location $E_n$ in Figure 6.9, with the image sensor plane. But, the depth of $P_w$ is along the actual distorted image ray (solid blue line in Figure 6.9), is not known. Thus, the location of ideal undistorted point becomes depth dependent (solid green and blue lines in Figure 6.9) and can not be predicted accurately from our calibration model. But, empirically we have

Figure 6.9: Depth dependent undistortion (best viewed in color).
observed that the variation in the position of undistorted image point (corresponding to solid blue line in Figure 6.9) as a function of its depth is very small. Thus, we propose to use a fixed scene depth for all image points in the scene and then obtain undistorted image points.

6.7 Discussion

In this chapter, we have hypothesized a physical model of radial distortion and shown that it occurs due to the movement of entrance pupil for different incident ray directions. All prior calibration methods, including those which incorporate moving entrance pupil, also add explicit radial distortion correction to complete the image projection model. We show via analysis and experiments that explicit radial distortion modeling is not needed and in-fact we obtain lesser re-projection error by considering only the motion of the entrance pupil. We finally propose that the best re-projection results for a fixed number of calibration parameters are obtained by considering a pupil-centric imaging, moving entrance pupil and a non-frontal sensor.
CHAPTER 7

POINT SPREAD FUNCTION CALIBRATION USING HADAMARD MATRIX AND LCD SCREEN

7.1 Introduction

This chapter presents a Hadamard matrix basic linear algebraic approach to calibrate the point spread function (PSF) of an imaging system. Given the PSF of the system and an acquired image under same imaging conditions, a non-blind deconvolution algorithm can be applied to recover the actual image. Traditional methods of computing the PSF include recording the response of the imaging system to an input point source of light, and repeating this procedure for all possible locations of the input point source location. Typically, an LCD screen is used where a single LCD pixel is lit while all pixels on the LCD screen are off. We show in this paper that the same goal of computing PSF can be achieved by projecting a Hadamard matrix based pattern on the LCD screen, where we have one distinct benefit: more number of LCD pixels in a Hadamard matrix based projection pattern are ON, leading to more input light for the imaging device and thus higher signal-to-noise (SNR) ratio in the captured images as those obtained by traditional unit impulse images.

There are four major contributions in this chapter are:

1. A Hadamard matrix based calibration technique is presented to compute the sensing matrix of an imaging system in Section 7.3.

2. The refresh rate of the LCD screen is not synchronized with the integration time of the CMOS sensor. This results in the captured image being spatially modulate by alternating black and white strips. We propose a solution, where we find optimal integration times where the lines would disappear. The reasoning behind our method is shown in Section 7.4.

3. The linear algebra of image formation is only valid if the number of photons being emitted by the LCD screen is directly proportional to the final digital number (DN) stored by the raw image at each pixel location. This is also termed as radiometric calibration which can be done traditionally using an OECF chart. Typically, the
linearity relationship does not hold if the pixels are saturated, i.e. the pixel well on each CMOS sensor is completely filled. To handle this, we modify LCD brightness, integration time and gain settings such that a linear relationship exists between number of photons and raw image intensity (DN) as shown in Section 7.5.

4. The brightness distribution in a half sphere around an LCD pixel is not isotropic. This means that an imaging device spatially localized near the LCD will not receive equal amounts of brightness from all LCD pixels. This in turn implies that the relationship between mathematical matrix values and the number of photons resulting from displaying the matrix will be different for a matrix index at the center and the one at the corner. Thus a map of the size of the display matrix needs to be computed when when pre-multiplied with the mathematical matrix will result in the imaging device receiving equal number of photons from displayed matrix irrespective of the index of the displayed matrix element. This is discussed in Section 7.6.

7.2 Image Formation

In 2D, image formation in an imaging system can be expressed as:

\[ y = t \ast x \]  \hspace{1cm} (7.1)

where \( y \) is the image captured on the sensor, \( x \) is the unknown sharp image, \( t \) is the point spread function (PSF)/blur kernel of the imaging system. This equation can be expressed in a linear algebraic form as

\[ Y = TX + N \]  \hspace{1cm} (7.2)

where \( Y \) and \( X \) are column-wise vectorized forms of \( y \) and \( x \) respectively and \( T \) is a block toeplitz matrix whose columns are shifted versions of the 2D blur kernel \( t \). This matrix will be called as the sensing matrix. Also \( N \) is the Gaussian noise at each observed pixel value in \( Y \). Given a fixed setting of scene and camera, the matrix \( T \) can be precomputed and used to compute \( X \) given \( Y \). The next section presents a Hadamard matrix based technique to compute \( T \) which can be applied to any optical imaging system.
7.3 Computation of Sensing Matrix $T$

Let the size of a single captured image $Y$ formed by the imaging system be $(m_y, n_y)$. Also, the input scene $X$ is assumed to be a square planar scene of size $(m_x, m_x)$. From Equation 7.1, since $Y$ and $X$ are vectorized, the size of $T$ matrix is $(m_y n_y, m_x m_x)$. It is also assumed that each entry of input stimulus matrix $X$ correspond to a single LCD pixel.

7.3.1 Unit Impulse Images

The conventional way to compute $T$ is to record the response of the imaging system exposed to unit impulses corresponding to lighting each pixel of an LCD while all other pixels are off. This can be achieved by creating a matrix $X$ of size $(m_x^2, m_x^2)$ assigning

$$X = I_{(m_x^2, m_x^2)}$$  \hspace{1cm} (7.3)

Each column of $X$ can now be reshaped into a square matrix $X$ of size $(m_x, m_x)$ such that only one element of $X$ is 1 (ON). This matrix is then displayed on the LCD screen and an image $Y(m_y, n_y)$ is captured. By displaying all the columns of $X$ and capturing, vectorizing and column-wise stacking of the resulting observed images $Y$, an observation matrix $Y$ can be created. Thus, from Equations 7.2 and 7.3 we have

$$Y(m_y n_y, m_x^2) = T(m_y n_y, m_x^2)I(m_x^2, m_x^2) + N(m_y n_y, m_x^2)$$  \hspace{1cm} (7.4)

Thus, the sensing matrix $T$ can be obtained as

$$T = Y - N$$  \hspace{1cm} (7.5)

7.3.2 Hadamard Patterns

A Hadamard matrix is a square matrix whose entries are either +1 or −1 and the rows of the matrix are mutually orthogonal. Let $H_n$ denote a square hadamard matrix of size $n \times n$. This is called as Hadamard matrix of order $n$, e.g.

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$  \hspace{1cm} (7.6)
We use the following Hadamard matrix property

\[ H_n H_n^T = nI_n \]  \hspace{1cm} (7.7)

to devise a calibration protocol which results in better SNR for the T matrix. Let X be a Hadamard matrix of order \( m_x^2 \) as:

\[ X = H_{m_x^2} \]  \hspace{1cm} (7.8)

Then, each column of X can be reshaped into a square matrix A of size \((m_x^2, m_x^2)\). Since the first column of any Hadamard matrix is all 1, the matrix A formed by the first column will have a higher energy than the matrices formed by all other columns. To handle this, the sign of all entries of \( H_{m_x^2} \) are randomly flipped as follows:

\[ H_f(m_x^2) = C_{m_x^2}H_{m_x^2} \]  \hspace{1cm} (7.9)

where \( C_{m_x^2} \) is a diagonal matrix whose diagonal entries are randomly flipped to 1 or \(-1\). It can be seen that the Hadamard property of Equation 7.7 is preserved for \( H_f \) since \( C_{m_x^2}C_{m_x^2}^T = I_{m_x^2} \) and

\[ H_f(m_x^2)H_f(m_x^2)^T = C_{m_x^2}(H_{m_x^2}H_{m_x^2}^T)C_{m_x^2}^T \]
\[ = C_{m_x^2}(m_x^2 I_{m_x^2})C_{m_x^2}^T \]
\[ = m_x^2 I_{m_x^2} \]  \hspace{1cm} (7.10)

When \( H_f \) is used for \( X \), the imaging Equation 7.2 becomes

\[ Y_{(m_y,n_y,m_x^2)} = T_h(m_y,n_y,m_x^2)H_f(m_x^2) + N_{(m_y,n_y,m_x^2)} \]  \hspace{1cm} (7.11)

The sensing matrix \( T_h \) can be computed by post-multiplying \( H_f \) on both sides of Equation 7.11 and then applying the Hadamard property of Equation 7.10 as follows

\[ YH_f^T = T_hH_fH_f^T + NH_f^T \]
\[ = n_x^2 T_h + NH_f^T \]
\[ T_h = \frac{YH_f^T}{n_x^2} - \frac{NH_f^T}{m_x^2} \]  \hspace{1cm} (7.12)

Assuming that the entries of noise matrix N have a Gaussian distribution of \( \mathcal{N}(0, \sigma^2) \), the elements of matrix \( \frac{N}{m_x^2} \) have variance \( \frac{\sigma^2}{m_x^2} \). Since \( H_f \) has entries +1 and −1, each element of the matrix in the second term of Equation 7.12 is a linear sum of the elements of \( \frac{N}{m_x^2} \).
As a result, the variance of each element of $\frac{\mathbf{NH}^T}{m^2_x}$ is $\frac{\sigma^2_x}{m^2_x}$ as $\mathbf{H}^T$ is of order $m^2_x$. Thus, $\mathbf{T}_h$ obtained by Hadamard matrix method has imaging noise of lesser variance. Another way to understand this is that in each exposure of a reshaped column of $\mathbf{H}$, more number of LCD pixels are ON as compared to a single ON LCD pixel in unit impulse images (Section 7.3.1). Thus, the Signal to Noise ratio of captured images in each column of $\mathbf{Y}$ is better.

Displaying Hadamard Patterns

The Hadamard matrix $\mathbf{H}$ is made of $+1$ and $-1$ and the LCD can display only 1 (ON) or 0 (OFF). Let $\mathbf{A}$ be the square matrix of size $m^2_x$ obtained by reshaping each column of $\mathbf{H}$. To obtain the image $\mathbf{Y}$ corresponding to the stimulus $\mathbf{A}$, two a new matrix $\mathbf{A}_{LCD}$ is created which has $-1$ entries replaced by 0. Then $\mathbf{A}_{LCD}$ and $1 - \mathbf{A}_{LCD}$ are displayed on the LCD in succession and the difference of two captured images is stored as the corresponding column of $\mathbf{Y}$. Thus, the response to matrix $\mathbf{A}$ having $+1$ and $-1$ entries is obtained.

7.3.3 Generating Huge Hadamard Matrix Efficiently

The length of a row/column of $\mathbf{H}$ determines the size of the scene for which we want to compute the sensing matrix $\mathbf{T}$, e.g. if we assume that we are only imaging a screen which is $p \times p$ LCD pixels wide, then the size of $\mathbf{H}$ matrix has to be $(p^2, p^2)$. If $p = 640$, then $\mathbf{H}$ is of order 409600. The generation of such a matrix of size $409600 \times 409600$ with a 1 byte representation for each entry in Matlab requires $\approx 157$ GB of RAM. Thus, for calibration, we need to generate each col of $\mathbf{H}$ of length $m^2_x$, apply the sign randomization transformation of Equation 7.9, reshape into a square matrix $\mathbf{A}$ of size $(m_x, m_x)$, create corresponding $\mathbf{A}_{LCD}$ and $1 - \mathbf{A}_{LCD}$ display matrices and capture two images whose difference is stored as the $j^{th}$ column of $\mathbf{Y}$. In order to generate each $(i, j)$ element of a large $\mathbf{H}$ matrix, we can use the Hadamard property that the Kronecker product of Hadamard matrices of order $m$ and $n$ is a Hadamard matrix of order $mn$, $\mathbf{H}_{mn} = \mathbf{H}_m \otimes \mathbf{H}_n$. After some computation, we get the following relationship relating indices of $\mathbf{H}_m, \mathbf{H}_n, \mathbf{H}_{mn}$:

$$\mathbf{H}_m(I, J) = \mathbf{H}_m \left( \left\lfloor \frac{I}{n} \right\rfloor, \left\lfloor \frac{J}{n} \right\rfloor \right) \times$$

$$\mathbf{H}_n \left( I - n \left\lfloor \frac{I}{n} \right\rfloor - 1, J - n \left\lfloor \frac{J}{n} \right\rfloor - 1 \right)$$

(7.13)
The above relationship can be applied recursively to small Hadamard matrices and entries of a large Hadamard matrix (order a multiple of 4) can be created, e.g:

\[
H_{409600} = H_{32} \otimes H_{20} \otimes H_{32} \otimes H_{20}
\]  

(7.14)

The complete process is shown in Figure 7.1. In order for the algebraic relation form Equation 7.11 to be actually able to accurately model the physical imaging process, a number of pre-calibration steps needs to be done. These steps involve calibrating the LCD display and the CMOS sensor such that

1. The CMOS integration time should be synchronized with the LCD refresh rate to avoid alternating dark and white stripes in the captured image (Section 7.4).

2. The number of photons emitted from the LCD screen have to be linearly related to the captured raw calibration image intensity (Section 7.5).

3. The anisotropic light distribution on an LCD screen needs to be corrected so that an "ON" LCD pixel on the center and corner of the screen result in equal number of photons toward the CMOS sensor (Section 7.6).
7.4 Synchronization of LCD Screen and CMOS Integration Time

An LCD screen typically refreshes at 60 Hz. This implies that it goes through 1 ON-OFF cycle 60 times in a second. If a CMOS sensor integrates the light from such an LCD screen for time period $t$, then the captured image is modulated by alternating black and white horizontal stripes. This happens because each CMOS pixel integrates the light for same period of time but the start and end time for each pixel is different. Thus, different pixels are exposed to different parts of the LCD refresh rate cycle. This is shown in Figure 7.2(a).

If the integration time is a multiple of the fundamental time period of the LCD refresh rate,

then no matter what the start integration and end integration time of a pair of pixels are, they end up integrating the same amount of light. Thus, the integration time can be set to be the multiple of this time period. In order to observe this and find the optimal integration time, we select a range of discrete integration times $I_t = \{t_0, \cdots, t_n\}$ where $t_n$ is such that the pixels are not saturated. The gain $g$ is set to 1 for each channel of the GRGB bayer pattern over the CMOS sensor. Then for each integration time $t_i$ from the set $I_t$ and for each channel, the raw intensity values are summed along the horizontal direction resulting in a 1D signal whose length is equivalent to the number of rows in the image $I_{t_i}$. An error function $E_{t_i}$ is then defined over this signal, which computes the sum of derivative over the signal. Since, a smooth image should not have any stripes, an ideal integration time $t_{opt}$ would yield very low derivatives and hence a low sum. An example plot of integration time $I_t$ and the error values $E_{t_i}$ are shown for each of the Gr, R, Gb, B channel in Figure 7.3.

![Diagram](image)

Figure 7.2: (a) LCD refresh rate. (b) Lens-less CMOS image with stripes. (c) At optimal integration time of the CMOS sensor, the captured image does not have any stripes.
Figure 7.3: Gradient vs. coarse integration time at (a) brightness 50 and (b) brightness 35. The minimum occur at time periods which are multiple of LCD refresh rate frequency.

7.5 Linear Relationship between Photons and Digital Number (DN)

For Equation 7.11 to hold in practice, there has to be linear relationship between the number of photons emitted by the LCD screen and the recorded intensity on the raw image. Also, it has to be made sure that the integration time of the CMOS sensor $T_{ex}$ and the the gain of each channel is such that there is no saturation of the pixels. Thus, we need to modify the brightness level $B$ of the LCD monitor, the integration time $T_{ex}$ and the gain $G_i$ of the $i^{th}$ color channel such that there is a linearity between the number of photons. We model
the number of photons acquired by each CMOS well as
\[ N = G_i \cdot k \cdot T_{ex} \cdot B \]  
(7.15)

where \( N \) is the observed intensity and \( k \) encodes the quantum efficiency. The gain \( G_i = 1 \).

For a fixed brightness \( B \), gain \( G_i = 1 \) for each color channel, the integration time \( T_{ex} \) is varied and the exposure time which results in saturation of the image starts is computed. This can be done by observing the plot of Figure 7.3(a) and finding the integration time after which the plot becomes flat, e.g. \( T_{ex} = 400 \). Thus, we have the LCD and the CMOS settings such that the CMOS well is full. For calibration, we want to fill the well by only 70% full to avoid saturation. Thus, our optimal captured intensity \( N^{opt} \) is

\[ N^{opt} = 0.7N = G_i \cdot k \cdot T_{ex} \cdot 0.7B \]  
(7.16)

Thus, the brightness \( B \) is reduced by 70%. In our experiments, \( B = 50 \), so for linearity to hold the new brightness of the LCD is changed to \( B^{opt} = 35 \). The optimal integration time is recomputed with \( B^{opt} = 35 \), \( G_i = 1 \) for all color channels and the obtained plot is shown in Figure 7.3(b). The new \( T_{ex}^{opt} \) in Equation 7.16 is selected as the farthest minima in this plot as it corresponds to synchronized time between LCD and CMOS (Section 7.4). Finally, the gain \( G_i \) are modified to \( G_i^{opt} \) for each channel such that \( N^{opt} \approx 2400 \) for each color channel.

In order to test if the linearity relation between the number of photons and the final intensity \( N^{opt} \) exists, a simple experiment was done. An LED light source with DC power supply was kept in front of the CMOS sensor. The integration time and gain of each channel was set to the optimal values computed above at \( T_{ex}^{opt} \) and \( G_i^{opt} \) respectively. Next, the current supply was varied leading to varying brightness of the LED such that \( N^{opt} \approx 2400 \) for each color channel of the captured image. Under these settings, there are two ways to verify the linearity. The first is to vary the current supply by constant increments and record the mean intensity of the captured image. Then check if the mean intensity of 2400 lies on a linear curve. But, in this case their might not be a linearity relationship between the varying current and the increase in the LED brightness or the number of photons. Thus, we adopt a different approach where the current supply is kept constant i.e. the LED brightness is fixed, but the integration time of \( T_{ex}^{opt} \) is varied by fixed amounts and an image is captured. The mean value \( N^{opt} \) is computed for each of these images. By varying the exposure time by constant amounts, we lineary vary the number of photons captured by the CMOS sensor. If there is a linearity between the number of photons and \( N^{opt} \), then the computed curve will
be linear around the $N_{opt} = 2400$. As seen in Figure 7.4, this indeed holds.

![Graph showing linear relationship between number of photons and the mean raw intensity value at optimal settings.](image)

Figure 7.4: (Left) Linear relationship between number of photons and the mean raw intensity value at optimal settings. (Right) Experimental setup.

### 7.6 Anisotropic LCD Correction

The brightness distribution on an LCD screen is not isotropic. In order to compensate for that, a map/matrix $C$ needs to be computed such that when the matrix $CA$ is displayed on the LCD screen, the number of photons emitted are uniform from all LCD screen pixels being imaged for calibration. This mask can be computed as:

$$C = \frac{\text{mean}(Y)H_f^T}{m_x^2}$$  \hspace{1cm} (7.17)

An example $C$ computed for a 640 × 640 sized LCD pixels at a resolution of 10 × 10 pixels is shown in Figure 7.5.

### 7.7 Experiments and Results

The calibration setup is as shown in Figure 7.6, where we set up an LCD monitor and use an Aptina MT9J003 sensor. The setup is placed in a dark room to attenuate the effect of extraneous light sources in PSF computation. In order to save time, we display a 10 × 10 block of pixels as ON on the LCD screen. Thus, the PSF is computed at a coarser scale as shown in Figure 7.7.
7.8 Discussion

In this chapter, we have presented a point spread function calibration method using an LCD screen and for a CMOS sensor and Hadamard patterns. We have proposed solutions to a number of calibration challenges in this setting.
Figure 7.6: Experimental setup for calibrating the point spread function.

Figure 7.7: Computed PSF for a camera with a lens.
CHAPTER 8
GENERATIVE FOCUS MEASURE

8.1 Introduction

Given a stack of registered images acquired using a range of focus settings (focal stack images), we propose a new focus measure to identify the most focused image. Although, most of the chapter is concerned with the new focus measure, for evaluation purposes, we will present it in the context of an application to generating omnifocus images. An omnifocus image is the composite image in which each pixel is selected form the frame in the stack in which it appears to be in best focus. Conventional focus measures usually maximize some measure of image gradient in a window. They tend to fail when one of the edges of the window lies near the boundary of an intensity edge, or the pixel is near other complex edge patterns. This leads to the misidentification of the focused frame and formation of artifacts in omnifocused image. Our proposed measure does not attempt to identify the focused frame by calculating the degree of defocus, like the gradient based methods. Rather, it hypothesizes that a specific frame is in focus and then validates or rejects this hypothesis by recreating the defocused frames in the vicinity, and comparing them with the observed defocused frames. This forward generative process leads to correct focus frame selection in regions where typical measures fail. This is because the conventional measures try to identify the focused frame from its distorted version which is the result of a complex convolution process. This involves a backward estimation for a many-to-one transformation. On the other hand, the generation of defocused frames from a hypothesized focused frame is more accurate since it involves applying an operator in the forward direction. We analytically show that under ideal imaging conditions, the proposed focus measure is unimodal in nature. This makes the search for the best focused image unambiguous. We evaluate our focus measure by generating omnifocus images from real focal stack images, and show that it performs better than all the existing focus measures.

Conventional cameras have limited depth of field (DoF), i.e. at a time they can focus only on a fixed range of depths depending on the camera setting used for image capture.
Due to this, scene depths outside the DoF limits are imaged in defocus and much of the high-frequency details pertaining to edges and corners of the objects at these depths is lost. This could be critical for various low-level vision algorithms relying on pixel level image information e.g. segmentation. Additionally, partially focused images are not visually pleasing. Thus, capturing a scene with extended DoF and generating an omnifocus (omni = all) image of the scene has been a popular research area in computer vision [50, 44, 10, 46] and optics [51]. With the recent affordability of DSLR cameras and smart phones fitted with better cameras, there has been a renewed interest toward omnifocus imaging [52, 53, 54] and related areas of refocusing [55, 56], optimal number of images to capture subject to reduced noise and defocus [57] and exposure [58] for extended DoF capture etc. In this chapter, we focus on the problem of omnifocus imaging using focal stack images [50, 56] which use a focus measure [45, 44, 46] metric to quantify the amount by which an image pixel is in focus. This metric is then used to predict the best focused pixel across the focal stack for each pixel location. We analyze a small but critical drawback in the design of existing focus measure techniques which can lead to artifacts near intensity edges (not depth edges) located on locally planar surface. We thereafter propose a new focus measure to handle this problem and show results on real data. A detailed review of existing omnifocus imaging techniques including the ones employing focus measure criteria is presented next.

8.2 Previous Work

Most of the previous work in omnifocus imaging can be broadly divided into two categories:

1. Computational Cameras/Single Image: The camera optics is modified to acquire an image with depth invariant blur and a single deconvolution is used to obtain omnifocus image [51, 54]. Levin et al. [59] capture a single image using coded aperture and use calibrated blur kernels to simultaneously obtain depth and an omnifocus image.

2. Conventional Cameras/Multiple Images: In this technique, a set of images is first captured by moving the sensor plane along the optical axis, thereby focusing on different depths in each captured image. This set of images is referred to as focal stack images [60] (Figure 8.1). To obtain the image in which a given object at a certain depth is best focused a focus measure is computed across the focal stack images [61, 62, 50, 45, 44, 63, 46, 10]. This measure gives a quantitative estimate of how focused an image is in any one of the given focal stack images (see Figure 8.1). Thus, the extrema of the focus measure corresponds to the best focused image. Once this
image is known, the pixel intensity corresponding to that object is extracted from that image and pasted on a new image. This procedure is repeated for pixels corresponding to all the objects being imaged and finally a omnifocus image is obtained. Hasinoff et al. [57] have also shown that omnifocus image obtained using focal stack usually have a higher signal-to-noise (SNR) ratio as compared to single shot based techniques. Nagahara et al. [52] combine focal stack images captured using a computational camera and shown that the blur kernel of the integrated images is relatively constant over the set of depths imaged by all the input focal stack images. Thus, a single deconvolution of the integrated image is sufficient to yield an omnifocus image.

![Focal stack images](image)

**Figure 8.1:** (a-f) Focal stack images. The goal is to design a focus measure which detects the best focused image (image (c)).

Traditional focus measures [62, 45, 44, 50] find the best focused image frame for a given pixel across the focal stack by maximizing the gradient present in a window around the pixel location. But, it has been shown in [64, 46] that such methods fail to identify the best focused pixel if the window over which the gradient is being computed lies in the neighborhood of an intensity edge on a locally planar surface (see Figure 8.2). In such a situation, the windows lying in focused image have less/no gradients, whereas the same window in a blurred image contains intensity values which have bled into it due to the defocusing of a sharp edge lying at the border of the window. A synthetic example where gradient maximizes for defocused windows near sharp intensity edges is shown in Figure 8.3. In real images this leads to artifacts in omnifocused images near intensity boundaries. It can be noted that although there intensity edge could be a depth edge also, no focus measure can perform well at such boundaries as the defocus blur kernel is a complex combination of two different blur kernels [60]. As such, the proposed focus measure can not handle depth boundaries and regularization techniques based techniques need to be applied to handle it.

In order to alleviate this problem, we propose a new focus measure by modeling forward focal stack image formation (see Figure 8.4), thus called as generative focus measure. Specifically, a window is first chosen around a pixel location in one of the focal stack images and
Figure 8.2: Synthetic example: (a) focused image consisting of an intensity edge. (b) Zoomed out small window near intensity edge has no gradients. (c) Artificially blurred image of (a). (d) Zoomed out patch (intensity scaled between \([0 \ 255]\) for better visualization) located at the same location as in (a). This window contains significant gradients causing errors in gradient based focus measure.

Figure 8.3: (a) Focused (b) Blurred with isotropic Gaussian blur of \(\sigma = 2.47\). (c-d) Red indicates pixel locations where focus measures based on (c) variance \([50]\), (d) energy of gradient \([45]\) (in a \(5 \times 5\) window) is higher for image (b) compared to the corresponding location in image (a). These regions lie near intensity edges.

it is assumed to be in focus. Then, based on this assumption, the amount of blurring which would be produced in windows at same location but in all other frames is predicted. If the focus assumption were indeed correct, then the predicted blur will be same or close to the actually observed blur in all the focal stack images. Thus, the \(L^2\) norm between the predicted and observed blurred images is defined as the focus measure. This norm minimizes for the best focused image. This modeling is closest to the forward modeling of depth and omnifocus image proposed in \([65]\) and omnifocus image estimation in \([57]\). Compared to both of these techniques, the proposed focus measure based technique is much simpler to compute and is independent of any optimization step which could be prone to local optima if not initialized properly. We show that under noiseless imaging and Gaussian optical blurring \([43]\), the proposed focus measure at each pixel location is unimodal in nature (Section 8.5). Thus a focused image pixel is determined uniquely at each pixel location. Also, our technique is based on capturing all the depths in a desired range to be in focus in at-least a single image. This varies from \([57]\), where this is not a necessary requirement and the omnifocus image is an estimation problem from an optimal set of focal stack images.

Section 8.3 describes our technique in detail. Also, we assume that the various camera parameters, namely focal length, F-number and the location of sensor planes used to cap-
ture focal stack are already known by camera specification or can be estimated via camera calibration [66].

Figure 8.4: Multifocus imaging geometry. Focused (middle) and defocused (left, right).

8.3 Focus Measure Calculation

Since the proposed focus measure depends on formation of focused and defocused images, we first discuss the focal stack geometry in Section 8.3.1. The captured images have the property that each image focuses on different depths. Given this focal stack, Section 8.3.2 explains the proposed technique for generating focus measure vector to obtain an omnifocus image.

8.3.1 Focal Stack Acquisition

A focal stack is acquired by first fixing all the camera parameters while setting the aperture to the maximum, thus allowing for a very small DoF to be focused in each image. The sensor plane is then sequentially moved along the optical axis in discrete amounts $S_t$ ($t$ denotes time) (see Figure 8.4) measured from the center of the lens. At each $S_t$, an image is captured and stored, thus generating a set of input images. The number of acquired images and the corresponding sensor locations can be optimized [67] such that the combined sum of the DoF
of all images is closest to a input larger DoF over which we want to compute the omnifocus image.

From Figure 8.4, given a unit source of light at \( O \), the radius of the point spread function on frames (located at \( S_{t-b}, S_{t+a} \)) neighboring to the focus frame \( f \) (located at \( S_t \)) can be calculated as,

\[
R_{t-b}(x, y) = \frac{D}{2} \left[ 1 - \frac{S_{t-b}}{S_t} \right] ; R_{t+a}(x, y) = \frac{D}{2} \left[ \frac{S_{t+a}}{S_t} - 1 \right]
\]

where, \( D \) is the diameter of the aperture stop. The PSF can be analytically modeled either as a pillbox or a Gaussian distribution. The pillbox model assumes a perfect imaging system devoid of any noise, where the resulting light energy due to defocusing is equally spread across the area of the circular blob. It can be represented as \( h_p \):

\[
h_p(x, y) = \begin{cases} 
\frac{1}{\pi R^2}, & \text{for } x^2 + y^2 \leq R^2 \\
0, & \text{for } x^2 + y^2 > R^2
\end{cases} \tag{8.1}
\]

But, due to imaging imperfections and noise in the imaging system the spread of light energy is not uniform in the circular blob and a Gaussian distribution for the PSF seems more practical.

\[
h_g(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{8.2}
\]

where \( \sigma \approx \frac{R}{c} \) and \( c = \sqrt{2} \) [45]. Thus, given a model of impulse response or PSF \( h = \{h_p, h_g\} \), derived from known camera parameters and an unknown omnifocus image, any defocused image \( g \) captured by the camera can be obtained as the following convolution:

\[
g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - \xi, y - \eta) h(\xi, \eta) d\xi d\eta \tag{8.3}
\]

By applying a forward convolution and assuming each focal stack image as a candidate for being an omnifocused image (locally), we derive a generative focus measure as described below.

### 8.3.2 Generative Focus Measure

This section describes a generative focus measure for omnifocus imaging. It is assumed that various camera parameters namely: intrinsic camera parameters, aperture diameter \( D \) and the sensor plane distances \( S_t \), where \( t \) varies from 1 to \( N \) are known. As the sensor
plane shifts along the optical axis in discrete steps, an image of the scene is captured. The complete set of $N$ images are called focal stack images, as they capture the scene with different amounts of focus and defocus (described in Section 8.3.1).

The set of $N$ focal stack images can be represented as a 3D matrix $MI$, where the index $MI(x, y, k)$ denotes the intensity value at a 2D location $(x, y)$ in the $k^{th}$ focal stack image. Also, let the size of each focal stack image be $X \times Y$. Since the focal stack images are registered, $MI(x, y, k) \forall k \in \{1, \cdots, N\}$ represents the complete set of focused and defocused intensities corresponding to the entire scene in the 3D world. In order to find the optimal index $k^*$ representing the best focus frame for a given scene object, the following focus measure is applied.

- **Step 1:** Repeat the following for all pixel locations $(x, y)$, where $x \in \{1, \cdots, X\}$ and $y \in \{1, \cdots, Y\}$.

- **Step 2:** Select a focal stack image index $k$, where $k \in \{1, \cdots, N\}$.

- **Assume the pixel intensity at $MI(x, y, k)$ is a focused intensity.** If the object appearing at $(x, y)$ was indeed focused in frame $k$ located at a distance of $S_k$ from the lens, then the images formed on all other sensors will be defocused in accordance with the defocus procedure described in Section 8.3.1.

- **Step 3:** Set $a = 1$ and $b = 1$. Since sensor locations $S_k, S_{k-b}$ and $S_{k+b}$ are known, use Equation 8.1 to calculate the blur radius $R_{k-b}$ and $R_{k+a}$, on sensors locations $S_{k-b}$ and $S_{k+a}$.

- **Step 4:** Given the two radius, calculate two Gaussian PSFs $h^a_g$ and $h^b_g$ using Equation 8.2. The choice of Gaussian PSF over pillbox is validated in Section 8.4.

- **Step 5:** Now, assign intensities selected from a $\overline{W} = W_d \times W_d$ sized window around the pixel location $(x, y)$ in the $k^{th}$ focal stack image to $f$. Thus $f$ is of size $\overline{W}$.

- **Step 6:** Apply convolution Equation 8.3 to obtain artificially defocused images as : $g_b = f \ast h^b_g$ and $g_a = f \ast h^a_g$.

- **Step 7:** Go to Step 4, and vary $b$ from 1 to $(k-1)$ to obtain $(k-1)$ artificially defocused images, $G_b = [g_b^1, g_b^2, \ldots, g_b^{k-1}]$. Similarly, vary $a$ from 1 to $(N-k)$ to obtain $(N-k)$ artificially defocused images, $G_a = [g_a^1, g_a^2, \ldots, g_a^{N-k}]$.

- **Step 8:** Calculate focus measure $F(x, y, k)$ for $(x, y)$ being focused in frame $k$ as described below.
If the assumption of intensity $MI(x, y, k)$ being focused made in Step 3 of the algorithm was indeed correct, then the artificially generated images $G_b$ and $G_a$ will be quite similar to the originally observed intensities in a $W$ sized window located around $MI(x, y, k - b)$ and $MI(x, y, k + a)$ where as usual $a \in \{1, \ldots, N - k\}$ and $b \in \{1, \ldots, k - 1\}$. This is in accordance with focal stack imaging theory described in Section 8.3.1. Otherwise, if $k$ were actually defocused, there will be considerable difference between the generated and observed. Based on this intuition, we define our focus measure $F(x, y, k)$ as the $L_2$ norm taken on a $W$ sized window between artificially blurred set $\{G_b, G_a\}$ and the observations $\{MI(x, y, k - b), M(x, y, k + a)\}$. This norm is a measure of how focused an image formed at $(x, y)$ is in the $k^{th}$ focal stack image. Thus we have,

$$F(x, y, k) = F_b + F_a; \text{ where}$$

$$F_b = \sum_{i=1}^{k-1} \sum_{(x,y) \in W} \left[ g_i^b - MI(x, y, k - i) \right]^2$$

$$F_a = \sum_{j=1}^{N-k} \sum_{(x,y) \in W} \left[ g_j^a - MI(x, y, k + j) \right]^2$$

Thus, the focus measure function $F$ is defined such that assuming that the PSF is correctly modeled, it should attain minimum magnitude only for the correctly hypothesized focus frame index. After calculating the focus measure vector, the best focused frame $k^*$ from the set of focal stack images for a pixel location $(x, y)$ can be easily obtained as:

$$k^* = \arg\min_{k \in \{1, \ldots, N\}} F(x, y, k) \quad (8.5)$$

Finally, the intensities in the omnifocus image are obtained as $OF(x, y) = MI(x, y, k^*)$. This process is repeated for all pixel locations to obtain the complete omnifocus image $OF$. Due to the dependence of our focus measure on the existence of an accurate PSF model, the next section is devoted toward an analysis of the effect of choice PSFs on the proposed focus measure.

8.4 Analysis of Focus Measure: Pillbox or Gaussian PSF

The pillbox PSF model is built on the assumption that imaging conditions are ideal, due to which the light energy is uniformly distributed in the region enclosed by the circular defocus blob formed on some sensor plane. On the other hand, a Gaussian PSF model captures the
effects of diffraction and imaging imperfections [43, 62] on the intensity distribution inside the circular blob. One way of analysis each of these would be to calculate proposed focus measure curves using them and then studying the attributes of these curve to infer the characteristics of corresponding PSF models. We proceed by selecting a pixel location $Q = (x_q, y_q)$ from one of the focal stack image frames such that it lies near an intensity edge. The choice is based on the fact that focusing/defocusing effects are more prominent near intensity edges. The focus measure $F(x_q, y_q, l)$ (Equation 8.4) is then evaluated at $Q$ in each of the input focal stack images $k$ by first setting the PSF as $h = h_p$ (pillbox, Equation 8.1) and then $h = h_g$ (Gaussian, Equation 8.2 with $c = \sqrt{2}$) in Step 5 of the algorithm mentioned in Section 8.3.2. Thus, two focus measure curves are obtained. Figure 8.5 shows two such sample curves, with the magnitude of focus measure values plotted against their corresponding focal stack frame index. Henceforth, the pillbox PSF based focus measure curve will be referred to as $P$ and Gaussian PSF based focus curve will be referred to as $G$. Additionally, the top of the Figure 8.5 shows a small region from the corresponding focal stack images in which the focus measure was calculated. It is also observed that both the focus curves attain their minimum values for the same correctly focused image window in spite of using different PSF models. Each of the focus measure plots are analyzed on the basis of the following two features:

- The magnitude of the minimum of focus measure curve attained by the best focused
$$F_{\text{min}} = F(x_q, y_q, k^*)$$
where $$k^* = \arg\min_k F(x_q, y_q, k)$$ (8.6)

The $$F_{\text{min}}$$ for each curve can be thought of as a measure of the accuracy with which a particular PSF model models the defocusing produced by the correctly focused frame on all other images in the focal stack; the lower the value, the better the PSF model.

- The slope of a focus measure curve given as

$$\Delta F = \frac{\partial F(x_q, y_q, k)}{\partial k}$$ (8.7)

The slope captures the information of increase in the magnitude of $$F(x_p, y_q, k)$$, as one moves away from the index of the correctly focused frame $$k^*$$. The larger the increase, larger is the SNR for finding $$k^*$$. Thus, an ideal PSF model for computing the focus measure would be the one, with has lowest $$F_{\text{min}}$$ and has highest $$\Delta F$$ among all the PSF models. For our case, $$P$$ has smaller $$F_{\text{min}}$$, as well as smaller $$\Delta F$$ compared to $$G$$. Since, the goal is to obtain the index of best focused image with less ambiguity, the criteria of highest $$\Delta F$$ is favored over the criteria of least $$F_{\text{min}}$$ for selecting appropriate PSF model. Thus, a Gaussian PSF model corresponding to $$c = \sqrt{2}$$ is used for obtaining the focus measure vector. Not surprisingly, the proposed analysis matches previous works of [62, 45] which have suggested the use of Gaussian PSF.

8.5 Unimodality of the Focus Measure

The search for best focus image frame for a given scene point implies minimizing Equation 8.5. The solution becomes relatively unambiguous and accurate if the focus measure is unimodal in nature and the minimum lies at the correct focal stack image index. In the following, we prove that the proposed focus measure is indeed unimodal in nature under some mild assumptions.

Based on the observability analysis of Favaro et al. [65] which shows that smooth regions and regions with brightness gradient do not get effected by rotationally symmetric blur (in our case Gaussian defocus blur in focal stack images), we present the current analysis for regions in the desired omnifocus image having sufficient textures. For smooth regions, any
of the focal stack images would suffice as they are indistinguishable. We first assume that the sensor planes are shifted along the optical axis in such a manner that all the depths in the scene are captured in focus in at-least one of the focal stack images. Secondly, we assume that the blur can be modeled as having a Gaussian distribution with mean 0 and some variance $\sigma^2$ (as in Section 8.3).

![Orthographic View of Multifocus Imaging](image)

Figure 8.6: The red rays correspond to the formation of a set of focal stack images on sensor plane located along the $x$-axis, with the focused image being formed at $x_f$. The blue rays correspond to hypothesized set of focal stack images assuming $x_k$ is focused and blurring the observed image at $x_k$. But, this blurring in turn generates the green set of rays, as the image at $x_k$ was already defocused.

In order to make our analysis easy, it is assumed that sensor plane distances: $S_x$ and corresponding blur radius formed on that sensor plane due to defocusing: $R_x$ are continuous. Although, the sensor planes are located at discrete distances, yet unimodality of a function in continuous domain automatically holds for the discrete domain as well. Next, a coordinate system is defined with respect to which all distances would be measured. The $y$-axis of this system coincides with the location of the sensor plane closest to the lens of the camera. The $x$-axis coincides with the optical axis as shown in Figure 8.6. Lastly, it is assumed that there is no sensor noise and the image degradation is only due to optical blur.

Now, let us suppose that focal stack is given, such that an object is imaged in best focus on the sensor plane located at a distance of $S_{xf}$ from the origin $O$ (see Figure 8.6). Since, we assume the PSF follows a 2D Gaussian model, the amount of degradation due to blurring is directly proportional to $\sigma^2$, the variance of the Gaussian. Based on the algorithm
proposed in Section 8.3.2, we select an arbitrary sensor location $S_{x_k}, S_{x_k} \neq S_{x_f}$ and assume that it is in focus. Then we calculate the blurring produced on some other sensor located at $S_{x_p}, S_{x_p} \neq S_{x_k}$ as shown in Figure 8.6. If $S_{x_k}$ was indeed focused, then the sensor located at $S_{x_p}$ will be blurred by $\sigma_{k}^{p}$ which can be given as,

$$\sigma_{k}^{p} \approx \frac{D^2}{2\sqrt{2}} \left[ \frac{|S_{x_k} - S_{x_f}|}{S_{x_k}} \right]$$  \hspace{1cm} (8.8)

But, since it is already known that the sensor located at $S_{x_f}$ was in focus, blurring the sensor at location $S_{x_k}$ with $\sigma_{k}^{p}$, would actually produce a different blur at $S_{x_p}$ which can be parameterized by $\sigma_{k}^{p}$ as

$$\bar{\sigma}_{k}^{p} \approx \frac{D^2}{2\sqrt{2}} \sqrt{\left[ \frac{|S_{x_k} - S_{x_f}|}{S_{x_f}} \right]^2 + \left[ \frac{|S_{x_k} - S_{x_p}|}{S_{x_k}} \right]^2}$$  \hspace{1cm} (8.9)

where we have utilized the dependence of $\sigma$ on the geometric blur radius and applied the fact that convolving two 2D Gaussians results in another 2D Gaussian, whose variance is the sum of the variance of the two original Gaussians.

As the amount of blurring is directly proportional to variance, we use the following error measure $\Delta_{S_{x_k}, S_{x_p}}$ to determine the difference in amount of blurring

$$\Delta(S_{x_k}, S_{x_p}) = \sigma_{k}^{p} - \sigma_{k}^{f} = \frac{D^2}{8} \left[ \frac{k - f}{f} \right]^2$$

On partial differentiation of $\Delta(S_{x_k}, S_{x_p})$ w.r.t $S_{x_k}$,

$$\frac{\partial \Delta(S_{x_k}, S_{x_p})}{\partial S_{x_k}} = \frac{2}{\bar{s}_{x_f}^2} \left[ S_{x_k}^2 - S_{x_k} \times S_{x_f} \right]$$  \hspace{1cm} (8.10)

For $S_{x_k} > S_{x_f}$, $\frac{\partial \Delta(S_{x_k}, S_{x_p})}{\partial S_{x_k}} > 0$, which means as the distance of wrongly assumed frame $S_{x_k}$ increases beyond $S_{x_f}$, the difference of blurring PSFs $\Delta(S_{x_k}, S_{x_p})$ increases. In other words, the focus measure defined in Equation 8.4 increases as $S_{x_k}$ increases. Similarly, for $S_{x_k} < S_{x_f}$, $\frac{\partial \Delta(S_{x_k}, S_{x_p})}{\partial S_{x_k}} < 0$, or the error increases again as $S_{x_k}$ moves away from the correctly focused sensor located at $S_{x_f}$ toward the origin. Thus the focus measure increases on both sides of $S_{x_f}$ and has a minimum at $S_{x_f}$ as $\sigma_{k}^{p} = \bar{\sigma}_{k}^{p}$. Thus the focus measure is unimodal about the correctly focused sensor location $S_{x_f}$.
8.6 Optimal Focal Stack Size

In this section, we analytically compute the optimal (minimum) number of image frames to be captured using a non-frontal camera such that all scene points within a radial distance from the first nodal point (intersection of first principal plane and the optic axis) are captured in focus in at least one of the image frames in the focal stack. For the case of focused panoramic imaging using such a camera, a method to compute the optimal number of frames was given by [68] where the non-frontal cameras were rotated such that the focal volume had minimum intersection. In the following, we derive the minimal rotation increment of the non-frontal camera which would allow for grazing intersection of the focal volumes for any two consecutive camera rotations.

8.6.1 Algorithm

Let the sensor angle tilt be $\alpha$, the aperture size $A$, circle of confusion $c$, optical focal length be $f$ and the distance of sensor plane center from the lens center be $\lambda$. For a given image point at a distance of $v$ from the back principal plane, we have the near depth of field (DoF) $z_{near}$ and far DoF range to be $z_{far}$ in terms of the above camera parameters, which we assumed to be known, as

$$z_{near} = \frac{vfA}{vA - fA + fc}$$

$$z_{far} = \frac{vfA}{vA - fA - fc}$$

For all of our analysis, we will assume 1D image sensor capturing a 2D scene. See Figure 8.7. Let the leftmost and rightmost edge of the non-frontal sensor be $N$ and $F$ where the sensor pixel $N$ is closer to the lens as compared to the sensor pixel $F$. Let the DoF corresponding to these pixel locations be $[z_{near}^N, z_{far}^N]$ and $[z_{near}^F, z_{far}^F]$ as obtained from Equations 8.11 and 8.12. The DoF ranges for the two extreme pixels $N$ and $F$ bound an area of 2D space inside which all scene points will appear in focus on the tilted image sensor. We call this area as sharply focused area (SFA). When the sensor is rotated by some angle $\theta$, the SFA also rotates in 2D space bringing a new set of scene points into focus in the new image frame. By repeating this process, all the scene points around the camera can be imaged in focus. An important question here is that what is the minimal number of image frames which will allow us to capture all scene points in focus within a given field of view.
(FoV) as this will lead to reduced capture time and computational efficiency in processing lesser number of image frames to do omnifocus imaging. As, the number of images directly depends on the amount of camera rotation \( \theta \), this problem boils down to finding the minimal (optimal) \( \theta_{\text{opt}} \). Next, we present a technique to analytically compute \( \theta_{\text{opt}} \).

We assume that we are given the radial scene depth range \( d_{\text{min}} \) and \( d_{\text{max}} \), which we want to be in focus. The values of \( d_{\text{min}} \) and \( d_{\text{max}} \) must satisfy the following criteria (see Figure 8.7 to match the notations).

\[
d_{\text{max}} \leq \sqrt{(X_{\text{near}}^N)^2 + (Z_{\text{near}}^N)^2} \quad (8.13)
\]
\[
d_{\text{min}} \geq \sqrt{(X_{\text{far}}^F)^2 + (Z_{\text{far}}^F)^2} \quad (8.14)
\]

We analyze the validity of these constraints in Section 8.6.3 and assuming that they hold derive optimal \( \theta_{\text{opt}} \).
We first need to compute the intersection of a circle of radius $d_{max}$ with the sharply focused area (SFA). The two points of intersection are denoted as $P_A$ and $P_B$. We have the boundaries of the SFA bounded by image rays corresponding to pixel locations $L$ and $R$ in the image plane, where their location in the lens coordinate system $C_l$ can be given as

$$F = (x_F \cos(\alpha), -(\lambda + x_F \sin(\alpha)))_{v_F}$$

$$N = (x_N \cos(\alpha), -(\lambda + x_N \sin(\alpha)))_{v_N}$$

The near and far DOF corresponding to pixel locations $F$ and $N$ can then be computed as $(Z_{near}^F, Z_{far}^F)$ and $(Z_{near}^N, Z_{far}^N)$ using Equations 8.11 and 8.12, where the image distance $v$ corresponds to the $z$ component of $F$ and $N$ points respectively. The $x$ coordinates of the near and far DOF limits of the corners of the SFA can then be obtained as:

$$X_{near}^F = \left( \frac{x_F \cos(\alpha)}{v_F} \right) Z_{near}^N$$

$$X_{far}^F = \left( \frac{x_F \cos(\alpha)}{v_F} \right) Z_{far}^N$$

$$X_{near}^N = \left( \frac{x_N \cos(\alpha)}{v_N} \right) Z_{near}^N$$

$$X_{far}^N = \left( \frac{x_N \cos(\alpha)}{v_N} \right) Z_{far}^N$$

Now, given the points $P$ and $R$, the line connecting them can be computed and its intersection with the circle $x^2 + z^2 = d_{max}^2$ can be computed to obtain points $P_A$. Similarly, given points $Q$ and $S$ and the desired circle equation, $P_B$ can be computed.

The optimal sensor rotation $\theta$ makes sure that the consecutive focus areas obtained for sensor rotation does not under-lap and the over-lap is minimal. Thus, we define the optimal rotation angle $\theta_{opt}$ as

$$\theta_{opt} = \arccos\left( \frac{P_T A^T P_B}{\|P_A\| \|P_B\|} \right)$$

(8.15)

This is explained in Figure 8.8 for two consecutive sensor rotations by optimal rotation angle. As can be seen there are no gaps in the sharply focused areas resulting in all scene points less than $d_{max}$ to appear in focus in at least one of the input image frames.
8.6.2 Simulation Results

We simulated a non-frontal camera with focal length 8.2 mm, sensor tilt 4.2 degrees and pixel size 0.01 mm and compute the optimal rotation angles using the method in Section 8.6 for different values of maximum radial distances ranging from 2 ft to 8 ft. Using these rotation angles, we plot the SFA for all the camera rotations. We observe that the SFAs are tightly packed as seen in Figure 8.9 and to cover a field of view of 120 degrees, we require at least 47 image frames. In Figure 8.10(a-f), we plot the results for various maximum radial depths ranging from 3 to 8 feet. For each of these depths, we observe that the compute optimal rotation angles tightly pack the SFAs.

8.6.3 Limitations

From Figure 8.7, in our method of computing optimal rotation angles, we made an assumption that

\[ d_{\text{max}} \leq \sqrt{(X_{\text{near}}^N)^2 + (Z_{\text{near}}^N)^2} \]  

(8.16)
Figure 8.9: Sensor tilt is 4.2 degrees. (a) Radial distance $d_{\text{max}} = 2$ ft, optimal sensor rotation $\theta_{\text{opt}} = 2.5981$ degrees compactly packs consecutive sharply focused areas. (b) Detailed view of the compact packing. (c) 47 images are required to capture a field of view of 120 degrees bounded by optical axes for the first and last frame.

Since, the pixel location $N$ on the sensor and other sensor parameters of pixel length, sensor tilt, focal length and sensor distance from the lens are known, the right-hand side of Equation 8.16 is known. Thus, we know the maximum possible value of $d_{\text{max}}$, for which our optimal sensor computation holds. We observe that under the above constraint, increasing $d_{\text{max}}$ leads to decreased values of $\theta_{\text{opt}}$ (see Figure 8.10 (a-f)). If $d_{\text{max}}$ is increased, then $\theta_{\text{opt}}$ results in gaps in consecutive SFAs. See Figure 8.11. This happens because the slope of the line connecting points $S$ and $R$ is more than the slope of the line connecting origin $O$ and $S$. Thus, in order to compactly pact consecutive SFAs, $\theta_{\text{opt}}$ must be reduced. And this reduction will be even higher for increasing $d_{\text{max}}$ which do not satisfy Equation 8.16, thus leading to capture of more image frames to cover a fixed field of view. Thus, for farther off objects, we will end up acquiring larger amount of image frames, which is contrary to the behavior of frontal sensors where for farther depths lesser number of image frames are required. Thus, we restrict the analysis of optimal rotation angles to the above constraint and propose that other sensor parameters, e.g. sensor tilt be leveraged to make sure that large depth ranges, i.e. $d_{\text{max}}$ are mapped inside SFAs.
Figure 8.10: Simulated optimal rotation angles for different depth ranges. (a) $d_{\text{min}} = 1 \text{ ft}$, $d_{\text{max}} = 3 \text{ ft}$, $\theta_{\text{opt}} = 2.565 \text{ degrees}$; (b) $d_{\text{min}} = 2 \text{ ft}$, $d_{\text{max}} = 3 \text{ ft}$, $\theta_{\text{opt}} = 2.565 \text{ degrees}$; (c) $d_{\text{min}} = 1 \text{ ft}$, $d_{\text{max}} = 5 \text{ ft}$, $\theta_{\text{opt}} = 2.539 \text{ degrees}$; (d) $d_{\text{min}} = 3 \text{ ft}$, $d_{\text{max}} = 5 \text{ ft}$, $\theta_{\text{opt}} = 2.539 \text{ degrees}$.

8.7 Depth from Focus

Our depth from focus algorithm takes a focal stack as input. A focal stack is a set of registered images which focus at different scene depths in each image such that each scene point is in focus (imaged sharply) in at least one image frame. Given this focal stack, an all-focused image of the scene can be computed using omnifocus technique presented in Section 8.3.2. We have the image formation model $y = k \ast x$, where $y$ is the observed image, $k$ is the depth-dependent blur kernel and $x$ is the idea sharp image. For our case, $y$ is the focal stack and $x$ is the all-focused image which we have already computed and the goal is to compute scene depth encoded in $k$. Our generative algorithm can be described as:

- Input: camera calibration parameters $U^*$ (Chapter 6), all-focused image $I_f$, focal stack $FS$.

- Repeat the following for all pixel locations $p$ in $I_f$, denoted as $I_f(p)$
  - Back-project pixel $p$ to an image ray $\overrightarrow{p}$ using $U^*$.
  - For discrete depths $d$ along $\overrightarrow{p}$, obtain the hypothesized 3D location of $p$ as $P(x_d, y_d, d)$, where $(x_d, y_d)$ can be obtained from ray geometry of $\overrightarrow{p}$.
  - Using $U^*$, forward project $P(x_d, y_d, d)$ onto all the other images in the $FS$ and compute the blur kernels $k'$ for each image in $FS$. 

153
Figure 8.11: (a) \(d_{min} = 1 \text{ ft, } d_{max} = 9 \text{ ft, } \theta_{opt} = 2.5219 \text{ degrees} \) computed from our method results in (b) gaps in SFA for consecutive frames. In order to cover these gaps, \(\theta_{opt}\) needs to be reduced, resulting in acquisition of larger number of image frames to cover a fixed field of view.

- Synthetically generate blurred image windows \(y' = k' * I_f\) around \(p\) in all other FS images.
- Compare the sum of squared pixel-wise error between synthesized and observed images.
- Select the depth \(d\) which gives minimum error.

- Output: 3D scene depth.

The computed depth using our depth from focus algorithm uses the calibration algorithm from Chapter 6 which gives an accurate method to back-project pixels to image rays.

8.8 Experiments and Results

8.8.1 Generative Omnifocus Algorithm

The experimental setup consisted of a non-frontal imaging camera [10] with a tilted sensor plane. The aperture was set to wide open and the camera was rotated about the optic center. As shown in [10], such a system allows easy acquisition of focal stack images along with the fact that a wide field of view can be imaged. The obtained images were corrected for magnification by registering the images given the known camera calibration parameters and vignetting was removed using the technique proposed in Castano [69].
Results: Real Datasets

The accuracy of the proposed focus measure was compared to five existing methods: energy of the gradient [45], energy of Laplacian [45], sum modified Laplacian [44], variance in a window [45] and the gray level distribution in a window [46]. All the focus measures were compared using the same window size parameter of $7 \times 7$.

Figure 8.12 shows a focus chart which has many sharp edges. It is placed at a distance of 2 ft from the optic center of the camera and 11 focal stack images are captured. Next, various focus measures [45, 44, 46] are applied and the final omnifocus image is obtained for each of them. A comparison of the obtained results with our generative focus measure is shown in Figure 8.12. The best performer among existing methods is the energy of Laplacian [45] and the gray level distribution based measure [46].

![Figure 8.12: Omnifocus image of a planar scene with comparison on focus frame selection using generative and gradient based techniques. The artifacts due to gradient methods are indicated in zoomed out windows in black arrow, where (a) energy of gradient [45], (b) gray level distribution [46], (c) sum modified Laplacian [44] and (d) variance.](image)

In Figure 8.13, we apply gradient based focus measure and our proposed focus measure on a scene with non-planar objects. The artifacts due to traditional methods is clearly visible in the Figure 8.13 (middle). The computed initial depth map by just using the focus measure is also shown in Figure 8.13 (right). This depth map is refined by applying volumetric graph cuts [70].

In Figure 8.14, we compare the performance of traditional focus measures with generative focus measure for a pixel located near an intensity edge. As hypothesized earlier, traditional
focus measures tend to fail near such locations. As can be seen in the plot, traditional focus measures [45, 44, 46] peak at the frame index 1, which corresponds to a defocused frame. Whereas the proposed measure and that of [46] correctly finds the focused image index to be 3.

In Figure 8.15, where the region for calculating focus measure contains considerable gradients, all the measures find the best focused frame correctly.

Finally, in Figure 8.16, we give a quantitative estimate of the accuracy of our focus measure in finding the best focused frame. A region of the scene is selected whose focus frame index is already known. The index can be correctly predicted based on the fact object depth and the sensor plane distances are known. Then, all the existing focus measures are applied at each pixel location in this patch and the best focused frame index as predicted by them is obtained. Thereafter, the absolute difference of focus measure prediction and the ground truth knowledge of focus index is calculated. The obtained differences are scaled between [0, 255] and shown in Figure 8.16. In the images shown, a 0 intensity indicates a perfect match with the ground truth observation, whereas any other intensity indicates false matches. As is evident, most of the focus measures find the focus frame index correctly at the boundary if intensity edge, but fail at the windows which do not contain the boundary. Compared to that, our generative measure detects all the focus frame indexes correctly. The bottom row indicates the percentage of erroneous predictions by each measure. Our measure has
Figure 8.14: Normalized focus measure around an intensity edge using various techniques. Only the proposed technique and [46] succeed in finding the best focused frame 3 (encircled with solid lines). All focus measures were calculated in a $7 \times 7$ window around a pixel.

3% error compared to more than 50% error in all other measures.

**Computational Time**  Given $N$ focal stack images, the computational complexity of our focus measure is $O(N^2)$ as compared to $O(N)$ of traditional focus measures.

Focus Measure Analysis

Although, the method of [46] is specifically aimed at solving the same problem as ours, yet the initial application of that focus measure yields artifacts in smooth regions as is shown in Fig 8.12(d), which is also common with other conventional measures. This is because the focus measure calculations at a pixel location are based on computation of gradients within each frame, which are subject to image noise. Thus, image noise adds randomness to selection of the best focus frame. Thus, neighboring window patches in smooth regions attain different focus frame indexes. Compared to this, an estimate of focus measure for one frame by using information from multiple frames, like our generative focus measure, suppresses noise. Thus, our focus measure is locally smooth. Consequently, the omnifocus image obtained by our method are smooth in regions where there is no texture at all. Thus, the requirement of image smoothing methods like graph cuts [46] are not necessary in our case for obtaining best omnifocus image.
Figure 8.15: Normalized focus measure for various focus measure in a window containing sharp intensity edge. All the measures peak for the correct focused frame.

8.8.2 Depth from Focus Algorithm

In this section, we show the results of depth estimation on different datasets. We show the setup, the corresponding omnifocus image and the computed 2D depth maps and 3D mesh of the objects. We experiment on three different datasets, namely teddy, rocks and cylinder with differing complexities in the shape and texture of the objects in the scene. The depth resolution for each pixel location is set to five mm to avoid computational costs. Some more optimization is done, by implementing Gaussian blurring as two separable one-dimensional kernels.

Teddy Dataset

The Teddy dataset has a textured teddy bear along with some geometric objects in the scene. The bow tie of the bear is smooth which leads to errors in depth computation.

Rocks

This dataset consists of a planar surface, a prism shaped object and some natural rocks. The geometric objects have been covered with paper and artificial texture is drawn on them.
Figure 8.16: The leftmost image is the region whose ground truth focus frame index is known. The other images represent the absolute difference between the focus frame index by different focus measures and the ground truth focus frame index. The numbers in the bottom represent the percentage of erroneous predictions. The proposed generative focus measure has the best performance.

to create some artificial blurring information.

Cylinder

This dataset has a cylinder shaped object in the scene. The cylinder geometry allows for accurate visualization of the accuracy of the depth estimation code.

For each of the datasets, we capture a focal stack and compute its omnifocus image. The results of depth estimation on each of these data sets is shown in Figures 8.17, 8.18 and 8.19.

8.9 Discussion

In this chapter, we have presented a new generative focus measure and have shown its advantages above existing focus measures via a number of analytical and empirical ways. The generative focus measure correctly handles windows near image boundaries which results in artifacts free omnifocus image. We also propose a method to compute the optimal size of focal stack so that all scene objects within a specified depth range would be imaged sharply in at-least one or more of the image frames. We then present results of applying our non-frontal calibration results from Chapter 2 and the omnifocus imaging method to compute depth of objects using focus cue.
Figure 8.17: Teddy dataset: (top left) shows an image of the objects in the scene, the camera location and a scale to empirically measure the distance of various objects in the scene; (top middle) shows the omnifocus image of the teddy bear and other objects behind the bear; (top right) shows the medial filtered ($11 \times 11$) 2D depth estimate of the scene; (bottom row) shows two 3D views of the teddy bear.
Figure 8.18: Rocks dataset: (left) shows the setting of various geometric objects in the scene. The objects consist of a flat planar board, a prism object and more generic shaped objects in the form of rocks; (middle top) shows the omnifocus image of the scene; (right top) shows the medial filtered (11 × 11) 2D depth map; (middle and right bottom) shows two 3D views of this dataset.
Figure 8.19: Cylinder dataset: (left) shows a cylindrical container and a flat board as the geometric objects in the scene; (middle top) shows the omnifocus image; (top right) shows the median filtered \((11 \times 11)\) 2D depth map of the cylinder; (middle and right bottom) shows two 3D view of the cylinder.
Part II

Train Monitoring System
CHAPTER 9
MOTION-BASED BACKGROUND SUBTRACTION
AND PANORAMIC MOSAICING FOR FREIGHT
TRAIN ANALYSIS

9.1 Introduction

We propose a new motion-based background removal technique which along with panoramic
mosaicing forms the core of a vision system we have developed for analyzing the loading
efficiency of intermodal freight trains. This analysis is critical for estimating the aerody-
namic drag caused by air gaps present between loads in freight trains. The novelty of our
background removal technique lies in using conventional motion estimates to design a cost
function which can handle challenging texture-less background regions, e.g. clear blue sky.
Supplemented with domain knowledge, we have built a system which has outperformed some
recent background removal methods applied to our problem. We also build an orthographic
mosaic of the freight train allowing identification of load types and gap lengths between
them. The complete system has been installed near Sibley, Missouri, USA and processes
about 20-30 (5-10 GB/train video data depending on train length) trains per day with high
accuracy.

Intermodal freight trains are the most common and economical mode of transporting
goods across long distances in the North American Freight Railroads network. These trains
are composed of different kinds of loads mounted and placed securely on rail cars. Each
load of an intermodal train is placed on a long iron platform with wheels called as a rail
car as shown in Figure 9.1(a) and a series of such rail cars of different lengths are attached
together to form a train. Loads of different sizes and types, as shown in Figure 9.1(b-f), can
be placed on each of the rail cars.

We define the arrangement of these loads across the length of an intermodal train as the
loading pattern for that train. These trains operate at speeds of 75−80 miles per hour (mph).
At such high speeds, the air drag between the gaps of loads creates considerable amount
of air resistance resulting in increased fuel consumption and operating costs. It was shown
in [71] that an analysis of loading efficiency and gaps in an intermodal freight train can help
railroad companies evaluate their loading techniques at intermodal facilities and save fuel
Figure 9.1: Different load types: (a) rail car, (b-f) different kinds of loads, including (b) double stack with upper and lower stacks of same length, (c, d) double stack with upper and lower stack of different lengths, (e) single stack and (f) trailer.

A machine vision system which can compute length of all gaps present in an intermodal freight train by analyzing a video of the train was proposed in [72, 73]. The background in such videos consists of trees, sky etc. visible through the gaps and above the train (Figure 9.2(a)). The foreground consists of the fast moving train (Figure 9.2(b)). An accurate background subtraction in these videos allows us to identify gaps and boundaries of loads on the train. In the current setting of intermodal train videos, the problem of background subtraction is made challenging due to the following constraints and requirements:

- **Accuracy**: foreground needs to be detected accurately for correct gap estimation between successive loads.

- **Image distortion**: Radial distortion and perspective projection cause similar scene points to image differently in consecutive frames of the video. This change in shape of foreground objects is more prevalent when the object is close to the camera, e.g. loads in intermodal train videos.

- **Illumination variations**: Long-term and short-term changes in weather conditions and sunlight direction can modify the captured intensity of foreground and background in a single video.

- **Camouflage**: The foreground and the background can be of similar color making it difficult to distinguish them.
Computational speed: Since the goal is to develop a computer vision application for real-world use, the computational speed of background subtraction is critical. Typically 30-40 trains need to be captured and processed per day, where each train is captured at 30 FPS and requires approximately 5 – 10 GB of storage. Thus the background removal has to be fast while also being accurate.

Image noise: Photon noise and sensor noise are inherent in acquired videos as the aperture is kept open for short periods to avoid motion blur due to fast moving trains.

In this chapter, we focus on developing a background subtraction method which can handle aforementioned issues. Traditionally intensity modeling [74, 75, 76] have been used for background subtraction. We employ a motion estimation based technique for background subtraction as they can handle persistent dynamic nature of background and foreground [77] e.g. train in our videos. But motion estimation is known to fail at texture-less regions [78] e.g background consisting of clear sky. Thus, we need to define a new cost function which can handle such situations. In addition, we also show a technique to create panoramic mosaic of the complete background removed train. Our contributions are:
1. Designing a motion based cost function (Section 9.3) which is robust than naive motion estimates to distinguish static background from dynamic foreground. Specifically, it can handle texture-less regions which are known to be challenging for motion estimation [78]. This simple method has just a single parameter $\tau$ (Section 9.3.2) which needs to be set manually as compared to other sophisticated methods which typically have multiple parameters. The upside of this is that we can handle many videos with varying illumination conditions while still obtaining accurate background subtraction results (Section 9.8).

2. Generating an orthographic panoramic mosaic (Section 9.4) of the train using motion estimates and background removed images. This is useful for gap detection, visualization and classification of loads on the train [72].

9.2 Previous Work

Background subtraction is a popular and well studied problem in computer vision. We present the prior work with respect to generic background subtraction and domain dependent background removal pertaining to intermodal freight train analysis.

- **Generic background subtraction**: Many techniques have been developed for generic background subtraction [79, 80]. The most common technique is to model pixel intensities as a time series and fit a dynamic unimodal or multi modal Gaussian distribution to them [81, 74, 82]. Elgammal [83] proposed a non-parametric modeling of background distribution based on kernel-density estimation. All of these techniques appear to fail and become parameter sensitive if the foreground and background are similar in intensity. This affects the applicability of these techniques on videos captured under wide range of illumination conditions. The background subtraction problem can also be modeled as foreground extraction by employing motion based features to distinguish fast moving foreground and static/quasi-static background [77, 84].

- **Intermodal background subtraction**: For intermodal freight train analysis, Kumar [72] did background subtraction using simple edge detection techniques. But this technique required appropriate values for a number of parameters making it unsuitable for handling wide range of videos. This was followed by a statistical learning based approach in [73], which employed domain knowledge to learn background removal parameters but still the sensitivity of this algorithm to its parameters made it difficult to generalize to different background and illumination conditions throughout an year.
9.3 Motion-Based Background Subtraction

In this section, we describe a hypothesis and validation based technique for background removal using motion based features for intermodal freight train videos. We define this feature at each pixel location in an image frame as the amount by which this pixel shifts horizontally across consecutive frames. The vertical motion is assumed to be negligible. This is imposed by calibrating the pose of the camera such that there is only horizontal motion of the train. We also assume that there is at least a single frame of complete background visible before the train appears in the video. This provides us with a model for the background.

A video with $N$ image frames is denoted as $V$. Thus $V = \{I_1, \ldots, I_t, \ldots, I_N\}$ where $I_t$ is the image frame at time index $t$. Let us consider $I_t$ from which we want to remove background. The hypothesis and validation steps are as follows.

- **Hypothesis:** As the camera is static and the rails of the track are fixed, we know the location of the moving railcar and wheels of the train in a image of the train. Thus, we know the location of some regions of the foreground. We select two image patches A and B at the known foreground location in $I_t$ and the next image frame $I_{t+1}$ respectively. The height of patch A and patch B is same but patch B is wider than patch A. Patch A is then correlated with shifted versions of patch B and the shift which results in maximum correlation is computed. This shift is thus the initial estimate of the velocity of the foreground, i.e. the train. We denote this shift as $v$ and this is our hypothesized train velocity in pixel shifts/frame. The correlation is computed by applying normalized cross correlation (NCC) [48]. This technique is invariant to linear changes in illumination. A patch based correlation (and not a pixel) also ensures robustness to assumed Gaussian image noise.

- **Validation:** Given an estimate $v$ of the train velocity in terms of pixel shifts/frame, the next step is to validate other parts of the image and test if they conform to this motion. The regions which pass this validation test should correspond to foreground, while the remaining regions will correspond to background. But such a validation test will fail for texture-less (zero image gradient) background regions, as two patches separated by some pixel-shifts/frame will match for any hypothesized velocity including $v$. To avoid this we incorporate the idea of validation to design a new cost function which can handle texture-less background regions.
9.3.1 Generic Motion Estimation

In this section, we implement the validation step for each pixel and compute few quantitative values, which are later useful in designing our proposed cost function in Section 9.3.2. We consider four image frames: $I_{t-1}$, $I_t$, $I_{t+1}$ and $I_{bg}$ (Figure 9.3). Here, $I_{bg}$ is the latest estimate of the background image. The first instance of this image corresponds to the image frame captured just prior to the appearance of the train in the video. This is done by applying the Gaussian Mixture Model (GMM) based technique [74] to the image frames in the beginning of the video. As the background is assumed to be visible at the beginning of the train video, this technique can model the background quite efficiently and detect the train as a foreground object as soon as it appears in the first image frame.

Now, let us consider a pixel $p$ with coordinates $(x,y)$ in $I_t$ (Figure 9.3). Its velocity is unknown to us. If $p$ belonged to foreground it should be observable at location $(x-v, y)$ in $I_{t-1}$ and at location $(x+v, y)$ in $I_{t+1}$. This is illustrated in Figure 9.3 by the pixel surrounded by the dashed square window. We assume that a local patch around $p$ also moves with velocity $v$ and select square patches $W_t$, $W_{t-1}$, $W_{t+1}$ and $W_{bg}$ centered at location $(x, y)$, $(x-v, y)$, $(x+v, y)$ and $(x, y)$ in image frames $I_t$, $I_{t-1}$, $I_{t+1}$ and $I_{bg}$ respectively.

![Figure 9.3: Validating the hypothesized velocity $v$ at two image patches: red (dashed boundary) belonging to foreground and orange (solid boundary) belonging to background.](image)

Given $W_{t-1}, W_t, W_{t+1}$ and $W_{bg}$, we compute the NCC [48] values among these image patches as shown in Equations 9.1, 9.2 and 9.3. The NCC values lie between $-1$ and
where high value indicates matching candidate patches, while smaller values indicate unmatched patches. It can be observed that Equation 9.1 and Equation 9.2 encode the validation method (Section 9.3) as they check for the correctness of hypothesized velocity $v$ using $\text{NCC}_{cp}$ and $\text{NCC}_{cn}$. If they are close to 1, then $W_t$ is a foreground image patch. We denote these equations as validation equations. But, such a criterion alone will not detect texture-less background correctly as shown in Section 9.3.2. The inclusion of $\text{NCC}_{cbg}$ in the validation analysis will be critical to solve this problem. This forms the basis of our proposed cost function $C_p$ in Equation 9.4.

\[
\begin{align*}
\text{NCC}_{cp}(x, y) &= \text{NCC}[W_t(x, y), W_{t-1}(x - v, y)] \\
\text{NCC}_{cn}(x, y) &= \text{NCC}[W_t(x, y), W_{t+1}(x + v, y)] \\
\text{NCC}_{cbg}(x, y) &= \text{NCC}[W_t(x, y), W_{bg}(x, y)]
\end{align*}
\]

Before going further, we note that due to image distortion and perspective projection different parts of the train move with slightly perturbed values of $v$. Thus, we increase the set of hypothesized velocities to $v' = [v - \delta, v + \delta]$ and then compute Equations 9.1-9.3 for each $v'$. We select the candidate with maximum value. We empirically set $\delta = 3$.

### 9.3.2 Proposed Cost Function

The problem with using simple validation based techniques (Section 9.3) and corresponding equations (Equations 9.1 and 9.2) to classify $W_t(x, y)$ as foreground/background can be explained as:

**Case 1. If $W_t(x, y) \in \text{foreground}$:** $\text{NCC}_{cp} \approx 1$ and $\text{NCC}_{cn} \approx 1$ as we cross-correlate similar patches.

**Case 2. If $W_t(x, y) \in \text{background}$:** If background is textured, then $\text{NCC}_{cp} \approx -1$ and $\text{NCC}_{cn} \approx -1$, but if background is texture-less then $\text{NCC}_{cp} \approx 1$ and $\text{NCC}_{cn} \approx 1$ as the background patches at $(x, y), (x - v, y)$ and $(x + v, y)$ are similar. This observation satisfies Case 1 above and classifies $W_t(x, y)$ as foreground.

**Case 3. If $W_t(x, y) \in \text{foreground} + \text{background}$:** If pixel $p$ is located at the foreground and background boundary, then $W_t(x, y)$ will include both foreground and background regions. It is known that motion estimation in such regions is challenging [78]. In our case, we post-process these regions after our background subtraction to get refined foreground boundaries.

Based on these observations, we propose the following cost function using the NCC infor-
Figure 9.4: (a) Input image. (b) foreground cost \( C_p \). (c) Extracted foreground.

\[
C_p = \frac{[NCC_{cp} + NCC_{cn} - 2 \times NCC_{cbg}]}{4}
\]  

(9.4)

It can be observed that if \( W_t(x, y) \in \text{foreground} \), we have \( C_p(x, y) \approx 1 \) and if \( W_t(x, y) \in \text{background} \), then \( C_p(x, y) \leq 0 \) for textured as well as texture-less regions. The application of this cost function is demonstrated in Figure 9.4 where background needs to be subtracted in a block (yellow) at the center of the image. The background consists of both textured (trees) as well as texture-less (blue sky, clouds) regions. The computed \( C_p \) at all pixel locations inside the block is shown in Figure 9.4(b). As can be seen most of the background have \( C_p \leq 0 \) (see color bar). To extract the foreground inside the block, a threshold \( \tau \) is set and each column of the cost \( C_p \) is compared to find the index where the threshold is reached. This gives the top edge of the container. From earlier analysis of \( C_p \), ideally \( \tau = 0 \) should be able to differentiate background and foreground. In practice, we found that \( \tau = 0.2 \) gives better performance. This is because due to image noise there are no ideal texture-less regions and a slightly higher value of \( \tau \) is preferable. Once the top boundary of the container is found, all the pixels below it are classified as foreground (Figure 9.4(c)) to construct panoramic mosaics as follows.

9.4 Panoramic Mosaic Generation

Let a background subtracted image corresponding to \( I_t \) be denoted as \( f_{gt} \). Since we also know the pixel velocities for each pixel in the foreground, they can be averaged to obtain a
global velocity $v_{fg}$ of the foreground. This implies that a new image patch of width $v_{fg}$ will be seen in frame $I_{t+1}$. Thus, one can select image patches of width $v_{fg}$ from the center of each background subtracted image (where image distortion is least) and concatenate such patches to create an orthographic mosaic of a background subtracted intermodal train video. Such a technique guarantees that the shape of the train is not elongated or shortened due to overlap or underlap of image patches in the mosaic. A mosaic is useful for visualization of the complete train and can be used for classification of individual load types [72]. A sample mosaic for a train is shown in Figure 9.5(a, b).
9.5 Gap Detection

Given the mosaic of the complete train, the vertical edges of the containers can be detected by thresholding the one-dimensional profile of the projection of the two-dimensional panoramic mosaic. Once the vertical edges are known, the top edge of the containers can also be computed by finding the boundary of the mask of the container.

9.6 Load Classification

Once the edges of the loads are detected, the next step is to classify the loads into one of the following three categories: single stack, double stack and trailer. The accuracy of this classification is important because based on a load being classified as a double stack, we look for the edge of the smaller stack in an unequal-sized double-stack configuration. The algorithm for load classification is described in the following subsections.

9.6.1 Single Stack Detection

The single stacks differ from the other types of loads in that their height is small, roughly around three to four feet. From the load specifications on the height of a single stack and using camera position and the height of the rail car, we can calculate the maximum possible height $h_{ss}$ of a single stack in pixel values in an image. We also have the height of the top of a load as $h_l$. Thus if $h_l \leq h_{ss}$, we classify that load as a single stack. Since double stacks and trailers could be of same size, we cannot use similar techniques for identifying them.

9.6.2 Trailer Detection

The trailers are characterized by their container shaped body but having wheels and an axle at the bottom. Due to the existence of a gap at the bottom of a trailer, the camera is able to view the base of the trailer. The base is characterized by low intensity values in the range of 0-10 (maximum intensity 255), as there is no direct natural light falling on it. To detect the trailer we look for a region of pixels near the base of the trailer, which falls in this low intensity range. If we are able to find such a region of pixels, we classify that load as a trailer.
9.6.3 Double Stack Detection

All the loads, which are not single stack or trailer, are assumed to be double stacks. The double stacks are characterized by two stacks of equal or unequal lengths kept on top of one another such that there is always a thin gap between the two stacks. The position of the camera is such that this gap is detected as a thin strip (two or three pixels wide) of black line of intensity close to zero. To detect the presence of this gap, we take a window of fixed size around the center of the double-stack configuration. The intensity values in this window are projected horizontally along the $x$-direction by summing them up to give rise to a one-dimensional array. The location of the minimum intensity value in this one-dimensional array corresponds to the location of the *midline*, which is defined as the boundary line between the upper and the lower stacks. To detect if the lower and upper stacks are of the same size or different sizes, we choose two windows near the left boundary of the double stack, one of these is above the middle line and the other is below the middle line. We look for the presence of the edge of the smaller stack in these windows, by projecting the foreground mask profile along the $y$-direction in that region and finding the location of steep change in projected profile which will correspond to the edge of the load. We repeat this process for the right boundary of the double stack. Thus we detect double-stack containers along with the widths of the upper and the lower stacks. See Figure 9.6.

9.7 Gap Length Estimation

Given the load classification in Section 9.6, there are various kinds of gaps which can be detected by our system. These gaps depend on the type of neighboring containers around
them and are broadly classified into upper and lower level gaps as shown in Figure 9.7 by blue and red lines respectively. Based on the gap lengths, our system outputs a histogram of the distribution of various gap types in a mosaic of the intermodal freight train as shown in Figure 9.8.

Figure 9.7: Different types of gaps: (a) different length double stack and single stack pair and a pair of same length double stacks, (b) same length double stack and trailer pair.

Figure 9.8: Histogram of gap type distribution for a given intermodal freight train.
9.8 Results

9.8.1 Data Acquisition

A video acquisition system is installed at Sibley, MO, USA. A camera of focal length 8 mm is used to capture the video of the train at 30 fps. An auto-exposure routine is run before capturing a train and the camera parameters are adjusted for current lighting conditions. These are then kept fixed for the entire video. The size of each image frame is 640 × 480. Each train consumes approximately 5 – 10 GB of space depending on its length.

9.8.2 Background Subtraction

The proposed technique has been tested on a wide variety of videos captured over a period of 12 months under varying illumination conditions. Our results of background subtraction are shown in Figure 9.9 on three different kinds of videos. The results are compared with three state-of-the-art methods: GMM [74] (our implementation), SOBS [75] and VIBE [76] (author’s implementation). We use all the implementation with default values and keep them same for all the experiments. Figure 9.9(a) top row shows the results for a video with clear blue sky, where all the methods perform well. Although, the foreground is classified as background inside the containers as the texture of trees and container is similar. Figure 9.9(a) middle row shows results for a cloudy sky. Here also the performance of our technique is at par with other methods. Although, the VIBE method is not able to remove all the background. In Figure 9.9(a) bottom row, we have a video captured during the evening when the illumination levels are really low. It can be seen that all the methods except for our technique fail to detect the foreground near the bottom of the image (rail car and wheels).

9.8.3 Gap Length Accuracy

Figure 9.9(b) shows the orthographic mosaic generated from background subtracted images. The boundaries of the container have been post-processed to take care of error resulting from using patch-based technique for computing NCC values. This mosaic can be used to compute the length of all gaps in pixel lengths. We manually computed the accuracy of background subtraction, by visually inspecting 1200 gaps in such mosaics and comparing the foreground boundary from the video and the one computed in the mosaic. The gap
Figure 9.9: (a) Background subtraction mask on different illumination conditions. The background mask in proposed technique is missing at the boundaries as those regions were either not present or are lost in previous and next image frames respectively. (b) Orthographically projected panoramic mosaics. They can be concatenated from left to right to obtain the complete mosaic.

detection accuracy is shown in Figure 9.10.

9.8.4 Computational Speed

On a 2.67 GHz, Intel Core i7 CPU with 64-bit windows, the background subtraction and mosaic generation is done at the rate of 16.2 fps while the video acquisition rate is 30 fps.

9.9 Discussion

This chapter proposes a new cost function which can handle texture-less background regions, while applying motion-based background subtraction. It has been implemented as part of a machine vision system for analyzing gap lengths in intermodal freight trains. The system has been functional at an outdoor location.
Figure 9.10: Histogram of different types of erroneous and accurate gap detection estimates after background removal.
CHAPTER 10

CONCLUSION AND FUTURE WORK

In this dissertation, we have looked at a number of problems and proposed techniques both computational and analytical to overcome them. We have taken a more optics based approach to model the problem of camera calibration. By doing so, we have shown better results than state-of-the-art results in this field. In this analysis, we have also proposed four new different calibration methods depending on different imaging constraints and the requirements of the application. For example, a pupil-centric imaging model with non-frontal sensor and radial distortion can be used for computing image sensor tilt, depth from defocus and undistortion. The focal stack approach uses the blur cue in addition to geometric cues to do calibration. The equivalence of entrance-pupil and radial distortion is more of an investigative work where we try to understand why radial distortion is modeled as an infinite series and if there is a more optics based understanding of this distortion. This equivalence relationship can be used for accurately back projecting image rays in the scene. We have also proposed a new focus measure for omnifocus imaging and together with calibration have used it for 3D depth estimation. We have also developed a new background subtraction method as a part of the machine vision system to analyze train videos and compute the length of gaps between the train containers.

There are many aspects of this work which can be extended. For example, the non-frontal sensor model can be extended to the problem of self-calibration. Also, there has been some work on efficient analytical solvers for polynomial equations for many multiview geometry problems [85]. We plan to explore whether our analytical solutions can be formulated in this framework. The problem of depth estimation has many challenges pertaining to texture-less regions, where cues will be necessary to get good depth estimates.
REFERENCES

[1] A. Kumar and N. Ahuja, “Generalized pupil-centric imaging and analytical calibra-
tion for a non-frontal camera,” in IEEE Conference on Computer Vision and Pattern
Recognition, June 2014, pp. 3970–3977.

[2] A. Kumar and N. Ahuja, “Generalized radial alignment constraint for camera calibra-
tion,” in Proceedings of the International Conference on Pattern Recognition, August
2014, pp. 184–189.

202–207.


imaging,” in IEEE International Conference on Computational Photography, April 2013,
pp. 1–8.

[6] A. Kumar, J. M. Hart, and N. Ahuja, “Motion-based background subtraction and
panoramic mosaicing for freight train analysis,” in International Conference on Image
Processing, September 2013, pp. 4564–4568.

C. Barkan, “Aerodynamic analysis of intermodal freight trains using machine vision,”
in Proceedings of the 9th World Congress on Railway Research, Lille, France, May 2011.

“Machine vision analysis of intermodal loading efficiency on heavy-haul railroads,” in
Proceedings of International Heavy Haul Association Conference, Calgary, Canada, June
2011.

“Monitoring the aerodynamic efficiency of intermodal train loading using machine vi-
sion,” in Proceedings of Transportation Research Board Annual Conference, Washington


