A Comparison of Monocular Camera Calibration Techniques

Richard L. Van Hook
Wright State University

Follow this and additional works at: http://corescholar.libraries.wright.edu/etd_all

Part of the Computer Engineering Commons

Repository Citation
A COMPARISON OF MONOCULAR CAMERA CALIBRATION TECHNIQUES

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Computer Engineering

By

RICHARD LOWELL VAN HOOK
B.S., Wright State University, 2008

2014
Wright State University
WRIGHT STATE UNIVERSITY
GRADUATE SCHOOL

16 April 2014

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY
Richard Lowell Van Hook ENTITLED A Comparison of Monocular Camera Calibration
Techniques BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF Master of Science in Computer Engineering.

______________________________
Kuldip Rattan, Ph. D.
Thesis Director

______________________________
Mateen Rizki, Ph. D.
Chair, Department of Computer Science and
Engineering

Committee on
Final Examination

______________________________
Kuldip Rattan, Ph. D.

______________________________
Juan Vasquez, Ph. D.

______________________________
Thomas Wischgoll, Ph. D.

______________________________
Robert E. W. Fyffe, Ph. D.
Vice President for Research and
Dean of the Graduate School
ABSTRACT


Extensive use of visible electro-optical (visEO) cameras for machine vision techniques shows that most camera systems produce distorted imagery. This thesis investigates and compares several of the most common techniques for correcting the distortions based on a pinhole camera model. The methods being examined include a common chessboard pattern based on (Sturm 1999), (Z. Zhang 1999), and (Z. Zhang 2000), as well as two “circleboard” patterns based on (Heikkila 2000). Additionally, camera models from the visual structure from motion (VSFM) software (Wu n.d.) are used. By comparing reprojection error from similar data sets, it can be shown that the asymmetric circleboard performs the best. Finally, a software tool is presented to assist researchers with the procedure for calibration using a well-known fiducial.
# TABLE OF CONTENTS

1  INTRODUCTION ........................................................................................................................ 1
   1.1  Overview .......................................................................................................................... 1
   1.2  Organization of Thesis...................................................................................................... 4
2  BACKGROUND .......................................................................................................................... 5
   2.1  Chapter Overview ............................................................................................................ 5
   2.2  Pinhole Camera Model .................................................................................................... 5
   2.3  Lens Distortion ................................................................................................................. 8
      2.3.1  Radial Distortion ...................................................................................................... 9
      2.3.2  Tangential Distortion ............................................................................................. 10
      2.3.3  Correcting Distortion ............................................................................................. 12
   2.4  Calibration via Calibration Panels .................................................................................. 13
      2.4.1  Chessboard Pattern ................................................................................................ 14
      2.4.2  Symmetric Circleboard Pattern .............................................................................. 16
      2.4.3  Asymmetric Circleboard Pattern ............................................................................ 17
      2.4.4  Required Number of Images .................................................................................. 17
   2.5  Calibration via VSFM ...................................................................................................... 19
      2.5.1  Correspondence problems ..................................................................................... 20
   2.6  Calibration ...................................................................................................................... 26
      2.6.1  Estimation of Camera Model and Extrinsics .......................................................... 26
      2.6.2  Computation of the Distortion Vector ................................................................... 27
      2.6.3  Final Computation of Camera Model and Extrinsics .............................................. 28
3  EXPERIMENTAL METHODOLOGY ........................................................................................... 29
   3.1  Chapter Overview .......................................................................................................... 29
   3.2  Hardware and Software ................................................................................................. 29
   3.3  Calibration Procedure .................................................................................................... 30
   3.4  Data Sets ........................................................................................................................ 31
# 3.5 Performance Metrics

# 3.6 Theoretical Results

# 4 EXPERIMENTAL RESULTS AND ANALYSIS

## 4.1 Average Reprojection Error and Time

## 4.2 Camera Model

## 4.3 Distortion Vector

## 4.4 Best Technique

## 4.5 Application of Camera Model

# 5 CALIBRATION ASSISTANT SOFTWARE

## 5.1 Motivation

## 5.2 Walk-Through

### 5.2.1 Selecting a Calibration Pattern

### 5.2.2 Loading Images and Detecting Features

### 5.2.3 Calibration

# 6 CONCLUSION

## 6.1 Summary of Results

## 6.2 Contributions

## 6.3 Future Work

# APPENDIX A: LIST OF ACRONYMS

# APPENDIX B: CHESSBOARD FEATURES

# APPENDIX C: SYMMETRIC CIRCLEBOARD FEATURES

# APPENDIX D: ASYMMETRIC CIRCLEBOARD FEATURES

# APPENDIX E: VSFM FEATURES

# BIBLIOGRAPHY
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1:</td>
<td>Pinhole camera model geometry</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2:</td>
<td>Pinhole Camera Model with Principal Point</td>
<td>7</td>
</tr>
<tr>
<td>Figure 3:</td>
<td>Lens Configuration</td>
<td>9</td>
</tr>
<tr>
<td>Figure 4:</td>
<td>Illustration of Radial Distortion</td>
<td>10</td>
</tr>
<tr>
<td>Figure 5:</td>
<td>Lens Configuration with Tangential Distortion</td>
<td>11</td>
</tr>
<tr>
<td>Figure 6:</td>
<td>Illustration of Tangential Distortion</td>
<td>11</td>
</tr>
<tr>
<td>Figure 7:</td>
<td>Chessboard Calibration Pattern</td>
<td>15</td>
</tr>
<tr>
<td>Figure 8:</td>
<td>Focus with Harris Corners</td>
<td>15</td>
</tr>
<tr>
<td>Figure 9:</td>
<td>Symmetric Calibration Pattern</td>
<td>16</td>
</tr>
<tr>
<td>Figure 10:</td>
<td>Asymmetric Calibration Pattern</td>
<td>17</td>
</tr>
<tr>
<td>Figure 11:</td>
<td>Correspondence problem - field of view</td>
<td>21</td>
</tr>
<tr>
<td>Figure 12:</td>
<td>Correspondence problem - smooth surfaces</td>
<td>21</td>
</tr>
<tr>
<td>Figure 13:</td>
<td>Correspondence problem - local blur</td>
<td>22</td>
</tr>
<tr>
<td>Figure 14:</td>
<td>Correspondence problem - global blur</td>
<td>22</td>
</tr>
<tr>
<td>Figure 15:</td>
<td>Correspondence problem - non-unique features</td>
<td>23</td>
</tr>
<tr>
<td>Figure 16:</td>
<td>Correspondence Problem - Saturation</td>
<td>24</td>
</tr>
<tr>
<td>Figure 17:</td>
<td>Correspondence problem - reflections</td>
<td>24</td>
</tr>
<tr>
<td>Figure 18:</td>
<td>Correspondence problem – obstructions</td>
<td>25</td>
</tr>
<tr>
<td>Figure 19:</td>
<td>Calibration Process Flowchart</td>
<td>26</td>
</tr>
<tr>
<td>Figure 20:</td>
<td>Data Sets</td>
<td>32</td>
</tr>
<tr>
<td>Figure 21:</td>
<td>Execution Times</td>
<td>38</td>
</tr>
<tr>
<td>Figure 22:</td>
<td>Average Reprojection Error</td>
<td>38</td>
</tr>
<tr>
<td>Figure 23:</td>
<td>(Left) High distortion. (Right) Lines accentuating the distortion</td>
<td>45</td>
</tr>
<tr>
<td>Figure 24:</td>
<td>(Left) Corrected image. (Right) Lines indicating minimal distortion</td>
<td>45</td>
</tr>
<tr>
<td>Figure 25:</td>
<td>Calibration Assistant Initial Screen</td>
<td>48</td>
</tr>
<tr>
<td>Figure 26:</td>
<td>Starting a New Calibration</td>
<td>49</td>
</tr>
<tr>
<td>Figure 27:</td>
<td>Calibration Pattern Selector</td>
<td>49</td>
</tr>
<tr>
<td>Figure 28:</td>
<td>Calibration Software Example Templates</td>
<td>49</td>
</tr>
<tr>
<td>Figure 29:</td>
<td>Finding Features</td>
<td>50</td>
</tr>
<tr>
<td>Figure 30:</td>
<td>Displaying Features</td>
<td>51</td>
</tr>
<tr>
<td>Figure 31:</td>
<td>Ready for Calibration</td>
<td>52</td>
</tr>
<tr>
<td>Figure 32:</td>
<td>After Calibration</td>
<td>52</td>
</tr>
<tr>
<td>Figure 33:</td>
<td>Chessboard #1</td>
<td>59</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1: Timing and Average Reprojection Error</td>
<td>37</td>
</tr>
<tr>
<td>Table 2: Comparison of Camera Models</td>
<td>39</td>
</tr>
<tr>
<td>Table 3: Camera Model Deviations vs. Average Reprojection Error</td>
<td>39</td>
</tr>
<tr>
<td>Table 4: Comparison of focal lengths in millimeters</td>
<td>40</td>
</tr>
<tr>
<td>Table 5: Comparison of principal point in millimeters</td>
<td>41</td>
</tr>
<tr>
<td>Table 6: Comparison of Distortion Vectors</td>
<td>42</td>
</tr>
</tbody>
</table>
Acknowledgements

I would like to thank Dr. Kuldip Rattan, Dr. Juan Vasquez, and Dr. Thomas Wischgoll for their guidance and support throughout this effort. Their expertise and encouragement helped me stay the course throughout the evolution of my thesis topic.

Additionally, I would like to thank the Air Force Research Labs, Sensors Directorate for providing equipment so that I could execute this research. This paper was cleared for public release on 9 April 2014 by the 88th Air Base Wing as public release number 88ABW-2014-1543.
1 INTRODUCTION

1.1 Overview

Cameras have been extremely prevalent within the last decade and are employed in many applications ranging from taking pictures of sporting events to assuring quality control in factories. The latter is in a field referred to as machine vision wherein automated software data mines one or more images and performs some desired analysis. Due to imperfections in the manufacturing and assembly processes and to the type of the lens used, the optical system creates distortions in the imagery so that it does not perfectly reflect reality, which is undesirable for the machine vision field.

In order to account for these effects, a pinhole camera model is used to model the focal length $f_x$ and $f_y$ as well as the principal point $(C_x, C_y)$. This camera model relates a point in world coordinates $(X, Y, Z)$ to a pixel location $(x, y)$. However, this simplistic model does not account for the distortion that the lens imparts upon the image. Lens distortion occurs when a lens magnifies an image unevenly.

The two predominant types of distortion are radial distortion and tangential distortion. Radial distortion occurs when a lens is not perfectly spherical. It has no effect at the center of an image, but evenly magnifies all pixels at the same distance to the center, creating “magnification rings.” The effects of radial distortion can be approximated by the first few terms of a Taylor series expansion that is centered about the center of the image, with coefficients of $K_1$, $k_2$, and $k_3$. Tangential distortion occurs when the optical axis of the lens is not perfectly orthogonal to
the focal plane array in the camera. This causes a non-linear warping of the image. Tangential distortion can be adequately approximated by modeling a thin prism in front of the camera. This adds two new parameters $p_1$ and $p_2$. Together, these parameters form the distortion vector $D = [k_1 \ k_2 \ p_1 \ p_2 \ k_3]^T$.

To solve for the camera model, a mapping between world and pixel space must be established. This is best done by taking images of a well-known fiducial with easily-measurable geometry. Some examples of easily-useable fiducials include a chessboard, a symmetric circleboard, and an asymmetric circleboard. The chessboard’s interior corners are, by definition, Harris corners.

Both circleboard patterns’ features are the center of the dots, which is found by a center-of-mass function. Also of importance is the linearity of the grid in each pattern which is used to measure the effects of distortion. *Visual Structure from Motion* (VSFM) is the fourth technique being compared and does not require a fiducial in the scene for calibration. Instead, it relies on the detection and successful correspondence of numerous SIFT points among multiple images.

A single planar calibration panel cannot provide enough information to solve for all unknowns. An easy solution is to take multiple pictures of the same board where the relative position and pose between the camera and the fiducial vary significantly. However, this change in position and pose creates a separate world coordinate system for each image. These coordinate systems must all be aligned, which can be done by applying a 3D rotation matrix $R$ and 3D translation vector $T$. The relationship
\[ x = K(RX + T), \text{where } K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \]

projects points in world space \( X \) to points in pixel space \( x \). However, this does not account for the imperfections in the lens, and so the distortion vector is applied to come up with a final pixel location.

In order to solve for the unknowns, there are three iterations through the Levenberg-Marquardt optimization technique. The first iteration sets all elements of the distortion vector to zero, and calculate the extrinsics and camera model. The second iteration holds the camera model and extrinsics static while solving for the distortion vector. The third iteration holds the newly-calculated distortion vector static and solves once more for the camera model and extrinsics.

For each iteration, the cost function being minimized is the average reprojection error. That is, it is desirable to minimize the average Euclidean distance between \( x \) and the projection of \( X \).

Three different calibration patterns (chessboard, symmetric circleboard, and asymmetric circleboard) were imaged 10 times with a similar set of positions and poses. For VSFM, a scene was imaged 10 times without any fiducials in the field of view. These four datasets were then calibrated and their average reprojection error, execution time, and proximity to theoretical camera models were examined.

It was determined that the asymmetric circleboard pattern provided the lowest average reprojection error among the techniques examined, with a time that was negligibly different from the other fastest performer. Overall, the chessboard performed the worst with a substantially higher error and roughly four times the execution time of any other method. In
cases where a fiducial cannot be inserted into a scene or where a calibration in a relative environment cannot be done, VSFM provides a suitable calibration.

Lastly, a software tool designed to aid researchers just beginning in the field of camera calibration was developed. It guides the user through selecting appropriate positions and poses of their calibration panels and notifies them when there is enough information to calculate the intrinsics, extrinsics, and distortion vector, providing them those values as well as the average reprojection error.

1.2 Organization of Thesis

Chapter 2 provides the required background material for the calibration processes of each of the three techniques. Chapter 3 describes the metrics that are used to quantify calibration performance and illustrates how the individual experiments were designed and implemented. This is immediately followed by the presentation of results in chapter 4. Chapter 4.5 presents a calibration assistant tool. Finally, chapter 6 summarizes the results and pontificates on areas of future research.
2 BACKGROUND

2.1 Chapter Overview

This chapter provides the background material pertinent to the proposed work. Sections 2.2 and 2.3 delve into the optics of the pinhole camera mode and the optical model for camera lenses. Then, section 2.4 explains the processing pipeline of the chessboard and circleboard patterns whose implementation is based on (Bradski, The OpenCV Library 2000) and (Bradski and Kaehler, Learning OpenCV 2008). Section 2.5 explains how the VSFM technique works. Finally, section 2.6 describes the calibration process.

2.2 Pinhole Camera Model

Modern digital cameras contain a focal plane array (FPA), which is essentially a planar grid of photon-collecting devices referred to as cells. Representation of the path that incoming light rays take when they strike the FPA is referred to as a camera model.
One of the easiest camera models available is the pinhole camera, depicted in Figure 1. This model assumes that all incoming light arrives at a single, small hole (i.e., the “pinhole”) called the \textit{aperture}. Based on this assumption, the \textit{real} image is reflected over both the horizontal and vertical axes onto the \textit{image or projective plane}. The size of the projected image is proportionally smaller than the real image. From similar triangles, this proportion is:

\[
\frac{-x'}{f} = \frac{X}{Z}
\]  

(1)

where \(x'\) is the length of a projected object, \(X\) is the length of the same object in the real world, \(f\) is the focal length of the lens, and \(Z\) is the distance from lens aperture to the object. By reconfiguring the camera model appropriately, the negative sign can be removed due to similar triangles and (1) can be re-written as:
\[ x' = f \frac{X}{Z} \] (2)

The camera model, as described, is still incomplete as it makes several assumptions. The first is the assumption that each cell in the focal plane array is square. This is not always true, particularly for economy-grade cameras. Therefore, (2) (where \( x' \) and \( X \) were generic valuables for any dimension) can be represented:

\[ x' = f_x \frac{x}{Z}, \quad y' = f_y \frac{y}{Z} \] (3)

However, this model is still not complete as it makes the assumption that the center of the lens is located precisely in the center of the focal plane array. While it would be convenient, this is almost never true. Therefore, let there be a new variable \( C \), with components \( C_x \) and \( C_y \), that characterize the principal point (i.e., the offset of the optical center of the lens from the center of the focal plane array). This configuration is depicted in Figure 2. Given that, (3) now becomes:

\[ x' = f_x \frac{x}{Z} + C_x, \quad y' = f_y \frac{y}{Z} + C_y \] (4)

Figure 2: Pinhole Camera Model with Principal Point.
These parameters \( (f_x, f_y, C_x, \text{ and } C_y) \) make up the **intrinsic camera parameters**, often referred to simply as the **intrinsics**. They provide some of the prerequisites needed to map points in world space to points in image space. While there are other camera models available (e.g., the CAHVOR camera model (Gennery 2005)), the pinhole camera model is relatively simple mathematically, so it is of great benefit with respect to computational complexity. The only deficiency the pinhole camera model lacks for most applications is the lack of incorporating lens distortion.

### 2.3 Lens Distortion

While the pinhole camera model sets a solid foundation for modeling rays of light, it is still an incomplete model. The fact of the matter is that the lens attached to a camera is not a single point and does not direct light to a single point on the focal plane array. Therefore, the effects that the lens imparts on the incoming photons must be taken into account. Figure 3 illustrates rays of light going through a lens on the left and then intersecting at the focal plane array on the right. As shown, each ray of light is composed of two parts – a segment of light going through the lens and a second segment going between the lens and the focal plane array. Both parts are actually the same ray of light at different points in time. Also, even though only five rays (1-5) are shown, there are an infinite number of rays that actually pass through the lens, and as such all are subject to the effects of the lens.

For an ideal optical setup, the total length of any given ray is equal to the total length of any other array that is also passing through the lens. That is, \( A_1 + B_1 = A_2 + B_2 = A_3 + B_3 = \ldots \), where \( A_i \) is
the segment of the light that is passing through the lens and \( B_i \) is the segment of the same light ray traversing the space between the lens and the focal plane array. When this relationship does not hold true, an image experiences lens distortion.

![Figure 3: Lens Configuration.](image)

Though there are many types of distortion, the two dominant forms are radial and tangential distortion. For nearly all applications, these are the only two distortions that are taken into account, and this research follows the same convention.

### 2.3.1 Radial Distortion

Radial distortion occurs when a lens is not a perfect hemisphere, resulting in non-uniform magnification being applied throughout the image. There are predominantly two types of radial distortions: barrel distortion and pincushion distortion.

Barrel distortion occurs when the magnification increases as the distance to the principal point increases. Conversely, pincushion distortion occurs when the magnification decreases the further from the principal point. These effects can be observed in Figure 4.
It is important to note that even though they have opposite effects, both barrel and pincushion distortion can exist simultaneously. Their magnitudes are rarely equal at any given point, and so complex aberrations can occur that are a combination of barrel, pincushion, and various other radial distortions.

### 2.3.2 Tangential Distortion

While radial distortion refers to the *shape* of the lens, tangential distortion is related to the *placement* of the lens. The *lens plane* is the plane that is orthogonal to the lens’s optical axis. Ideally, the lens plane is parallel to the focal plane. However, manufacturing processes are not yet so exact as to manufacture cameras with insignificant dot products between lens axis and focal planes. An example of this can be seen in Figure 5.
The ray in Figure 5 defined by $A_3 + B_3$ is parallel to the optical axis of the lens and is not orthogonal to the focal plane array. Therefore, imagery taken from this setup would exhibit tangential distortion.

Figure 6 demonstrates tangential distortion. Figure 6a is the same as Figure 4b and is repeated for comparison, and Figure 6b is the same image with tangential distortion applied to the vertical axis. Notice how the top of the image appears to be further away than the bottom.
2.3.3 Correcting Distortion

Since the overall effect of lens distortion is that the world does not appear in the image as it truly is, this must be corrected. The method below is a commonly-used technique to correct the distortions and is based heavily on the work of (Fryer and Brown 1986) and (Brown, Close-range camera calibration 1971).

Since radial distortion is zero at the principal point and changes outward from this point, it is best to define a function that has no effect at the principal point but changes the magnification as it radiates outward. To correct this, a Taylor series centered around $\alpha = 0$ (i.e., the center of the image) is quite suitable. A Taylor series centered around $\alpha$ has a sigma notation of

$$
\sum_{n=0}^{\infty} \frac{f^{(n)}(\alpha)}{n!} (\hat{x} - \alpha)^n
$$

where $f^{(n)}(\alpha)$ denotes the $n^{th}$ derivative of the function $f(\alpha)$ and $\hat{x} = [x\ y]$. In practice, the effects of radial distortion are relatively small and can be sufficiently modeled by the first few terms. For this effort, 3 radial distortion parameters ($k_1, \ k_2, \ and \ k_3$) were used, centered around $\alpha = 0$. Substituting $r$ for $\alpha$ to follow the conventionally-used symbol used in optics for radius results in the following pair of equations:

$$
x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{6}
$$

$$
y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{7}
$$

where $x$ and $y$ refer to the location of the original pixels and $x'$ and $y'$ represent the location of the pixels after correcting for radial distortion. Using 3 radial distortion coefficients is sufficient.
for most every lens. In some cases, notably with “fish eye” lenses, more terms are needed to properly compensate. To correct for tangential distortion, the method that Brown proposes in (Brown, Decentering distortion of lenses 1966) is conventionally used and is fundamentally an extension of his work from (Instrument Corporation of Florida Melbourne 1964). In it, he discusses how an appropriately-shaped thin prism can adequately model the effects of tangential distortion. This is now commonly referred to as the plumb bob model and is of the form:

\[ \begin{align*}
    x' &= x + (2p_1 y + p_2 (r^2 + 2x^2)) \\
    y' &= y + (2p_2 x + p_1 (r^2 + 2y^2))
\end{align*} \]

(8) (9)

Where \( p_1 \) and \( p_2 \) are the tangential distortion coefficients that model the thin prism, \( x \) and \( y \) refer to the location of the original pixels and \( x' \) and \( y' \) represent the location of the pixels after correcting for distortion.

Note that whereas the correction for radial distortion was effectively a scalar, the correction for tangential distortion is a non-linear warping function.

2.4 Calibration via Calibration Panels

Given the previous two sections, a total of 9 unknowns must be estimated to provide a calibration solution. These unknowns are the camera model \((f_x, f_y, C_x, and C_y)\) and the distortion vector \((k_1, k_2, p_1, p_2, and k_3)\). The variables represent an unknown transformation in pixel space. This transformation can be determined if both starting and ending states are
known. While not strictly necessary, it is beneficial if there is an artificial object in the scene to makes these states more readily identifiable. These objects are referred to as *calibration panels*.

Some of the easiest, and most commonly used, calibration objects are planar patterns. This is because planar calibration objects have features in two dimensions, and the third dimension can be set arbitrarily (but identical for all points), and for ease, this third dimension is almost always set to be 0. However, non-planar objects are suitable as well if each feature’s location of said object is well-known. 3D calibration objects are typically not used as determining the features’ coordinates in world space is a non-trivial exercise.

### 2.4.1 Chessboard Pattern

A chessboard calibration pattern is one of the simplest patterns. The features of importance are the interior intersections where two edges come together and then split off at opposing right angles. Fundamentally, they are simple Harris corners (Harris 1988). Thus, the intensity gradient for both vertical and horizontal axes can be calculated. High gradients on the horizontal axis are indicative of a vertical line, and high gradients on the vertical axis are indicative of a horizontal line. Large X- and Y-gradients indicate a corner. Since each interior corner actually has two edges coming together, it is an easy matter to detect the intersection. Figure 7 shows features that are in rows 9 wide and columns 6 deep. Therefore, it is a “9x6 chessboard.”
One of the critical aspects to using the chessboard pattern for calibration is camera focus. In order to determine the exact location of the features, excellent focus is required. Figure 8a shows a camera with excellent focus. The transition between black and white patches takes less than 3 pixels in both vertical and horizontal directions. Additionally, the focus is sharp enough that the Bayer pattern\(^1\) of the focal plane array can be seen, especially in the black patches.

Opposite of this, Figure 8b shows a Harris corner with poor focus. The transition between black and white patches takes 6-7 pixels, and the Bayer pattern is not evident anywhere in the image chip.

---

\(^1\) The focal plane array is a 2D array of cells that capture light. The placement of a particular color filter in front of each cell results in that cell being significantly more responsive to the wavelengths of light permitted through the color filter. The arrangement of color filtering is referred to as a Bayer pattern.
It should be noted that the image chips from both figures were taken from the same image where the calibration panel was mostly co-planar with the focal plane of the camera. Though such rotations may be small, they should be thoroughly checked before they are declared negligible.

### 2.4.2 Symmetric Circleboard Pattern

Another calibration pattern that is becoming more and more popular is the symmetric circleboard pattern, an example of which is shown in Figure 9. The center of each dot is equidistantly spaced from vertically and horizontally adjacent dots. Additionally, finding the center of the dots is a straightforward center of mass function. Consequently, this calibration pattern (and all others using solid dots) is robust to focus issues.

![Symmetric Calibration Pattern](image)

**Figure 9: Symmetric Calibration Pattern.**

Despite being robust to focus, symmetric circleboard calibration panels have another issue – that of the dot size. Most algorithms for finding dots use a window of a fixed-size. If the dot exceeds the size of the window, it will not be found. Therefore, care must be taken to choose both dot and window sizes appropriately based on the camera and optical setup.
The symmetric pattern in Figure 9 would be considered an “8x9 Symmetric Circleboard” since the dots are in rows 8 wide and columns 9 tall.

### 2.4.3 Asymmetric Circleboard Pattern

The last calibration pattern to be examined in this research is the asymmetric circleboard pattern. Figure 10 depicts an “11x4 asymmetric circleboard.” At its core, the asymmetric pattern is a full symmetric pattern interwoven with a nearly-whole second symmetric pattern. The additional dots help to minimize any skewing effects during the calibration process.

![Figure 10: Asymmetric Calibration Pattern.](image)

### 2.4.4 Required Number of Images

Given that there are 9 unknowns that must be estimated for the calibration solution, it would intuitively make sense that a calibration pattern with at least 9 points would be sufficient. However, (Bradski and Kaehler, Learning OpenCV 2008) demonstrates that a planar object is not sufficient. Rather, a single board can only provide 8 unique equations. Therefore, at least two boards must be used. While it is possible to place two (or more) calibration panels within the image scene, a simpler solution is to use multiple images of the same panel. To get a different
position/pose for the calibration panels, either the camera can be moved while the panel stays static, or vise versa.

Regardless of whether the camera or the calibration panel moves, it introduces a problem. The world coordinate system is relative to the position and pose of the calibration panel, which is different for each image. In order to transform the calibration panels into a common coordinate system, let there be a 3D translation vector \( T = [T_X \ T_Y \ T_Z]^T \) and a 3D rotation vector \( R = [R_X \ R_Y \ R_Z]^T \) for each image. Together, \( T \) and \( R \) are known as the **extrinsic camera parameters**, or just the **extrinsics**. It is important to note that there will be one set of extrinsics for every image; that is, every image will have 3 translation and 3 rotation parameters.

The question becomes how many unique views of the calibration panel are needed to solve for all the variables. At this point, parameters from the distortion vector are ignored; they will be determined at a later stage. For \( N \) features in each of \( K \) boards, the following inequality must hold true in order to solve for the unknowns:

\[
2NK \geq 6K + 4
\]

(10)

The left hand side is the number of available constraints across all boards, while the right hand side reflects the extrinsics unique to each board as well as the intrinsics which are common throughout all the images. This can be simplified to:

\[
(N - 3)K \geq 2
\]

(11)

Equation (11) is even simpler since \( N \) is already determined. Per (Bradski and Kaehler, Learning OpenCV 2008), there can only be 4 unique points’ worth of data for each board. Therefore,
Two images would satisfy the requirements, but is a poor choice as the system of equations would be very susceptible to noise. Alternatively, a hundred or even thousands of images could be used to greatly reduce noise. However, the computational requirements of iteratively solving a large system of equations are very high and there are significantly diminishing returns with a high number of images. In practice, using 8-12 images is conventionally used as it provides a good balance between error reduction and processing requirements.

2.5 Calibration via VSFM

Structure from motion (SFM) is a technique for 3D reconstruction that is based upon a series of images from a single camera, and Visual SFM (VSFM) is an implementation of SFM. VSFM does not require calibration objects to be inserted into the scene. Instead, it relies on the scale-invariant feature transform ([Lowe, Object recognition from local scale-invariant features 1999], [Lowe, Distinctive image features from scale-invariant keypoints 2004]) feature detector, commonly referred to as SIFT, to detect “natural” features from each image. The features fulfill the obvious requirement that they are easy to uniquely identify. In addition, they are both invariant to both scale and rotation. Because of this, they are usually easy to identity from different observation points.

Since there are no calibration objects in the scene, multiple images must be used in order to determine the camera’s intrinsic and distortion vectors. As before, each image will have its own arbitrary coordinate system. However, a more pertinent issue is that the set of SIFT features
from each image are not identical. Each image will have features that are common to some or perhaps all of the other images, but will also have features that none of the other images contain. Matches based on the SIFT feature descriptions are used to determine the transformation. Unpaired features and matches whose motion is not similar to the majority of other matches are discarded since they do not provide useful data.

2.5.1 Correspondence problems

The correspondence of feature pairs between images can be fairly challenging. Since the scene content is not controlled as compared to the calibration pattern approach, numerous problems can arise that drastically reduce the number of correct feature matches. Below are some of the most common aspects.

Field of view: The most obvious aspect of feature correspondence is that the images must have the same area of regard; that is, the majority of the scene content between images must be similar. While it is not expected that all images have the exact same area of regard, they must each have a significant area that is common in the other images being used to create the correspondence match. If there is minimal commonality in the fields of regard, as shown below in Figure 11, then there will be a small number of matching SIFT features.
Smooth surfaces: When trying to find SIFT features, areas of the scene that are predominantly homogenous are difficult to match to other pixels. This is because adjacent pixels have very similar signatures to each other and a common set of salient features between image pairs may not exist. Figure 12 shows a picture that prominently shows a computer monitor on the left. On the right is a close-up of a segment of the computer monitor showing individual pixels. Note that they all appear to be identical, even though there are very slight differences in intensity values among the pixels.

Image blur: While smooth surfaces are a result from actual objects in a scene, image blur occurs due to movement of the objects in scene or the camera. There are two types: local blur and

\footnote{The apparent yellow response is just the overlap of the red and green channels}
global blur. Local blur occurs when an object within a scene moves while a camera is still integrating. Figure 13 shows an example of local blur. Notice in the righthand image that the image is blurred near the hand only and the rest of the image is clear.

Additionally, there can be global blur - smearing that affects a whole image. It is possible that the entire scene itself could be moving and cause global blur. Examples of such scenes could include waterfalls, dense traffic, and automated factory lines. However, those scenes are special. The primary cause of global blur is camera motion. If the camera itself is moving while imaging a static scene, all pixels will suffer from smearing to some degree. Figure 14 depicts an instance of global blur where the camera is moving while taking the picture.
Non-unique features: Even when there are relatively few smooth surfaces, there can still be features that appear very similar. A trivial example of this is the interior corners of a chessboard calibration panel. Figure 15 shows a close-up of the two types of interior corners for the chessboard. Note that the two image chips shown on the right are actually identical when a 90° rotation is applied to either one. Therefore, the figure shows 48 very identifiable features, none are distinguishable from the others if the pose is unknown.

Saturation: The cells on a focal plane array gather light during the exposure period. Though the physical process by which camera converts light into a picture are beyond the scope of this research, it is easy to imagine that each cell can only hold so many photons; if excessive photons arrive, they are simply dropped. The result is called saturation, where the camera is only able to represent the brightness of a pixel up to some threshold. All pixels “brighter” than the limit are truncated, and so pixels that should be dissimilar appear alike. Figure 16a shows the effects of significant glare on a portion of the image. Note that even though a quarter of the image is saturated, the majority of the image is still quite usable. Figure 16b shows the effects of global saturation where the saturation occurs over the entirety of the image. Objects in the scene that
are naturally “hot” are washed out. Areas of saturation appear to be homogenous, which causes an undesirable decrease in the expected number of SIFT features.

![Image](image1.jpg)  
**Figure 16:** Correspondence Problem - Saturation.

**Reflections:** Beyond lighting conditions, the type of surfaces in the scene can confuse correspondence algorithms. Reflections can visually duplicate points in the scene, confusing the mapping between features. It is possible that both cameras will see different “translated” reflections for the same surface and can result in a poor distance reading at the reflective surface’s location. Furthermore, the reflections can contain information from areas that are not part of the intended scene (Figure 17, right image).

![Image](image2.jpg)  
**Figure 17:** Correspondence problem - reflections
**Obscurations:** Because the two images were taken from two different locations and (almost always) two different poses, objects in the scene may not appear in both images even if the frustrum cross-sections overlap at the objects. Specifically, given objects $O_A$ and $O_B$ that lie along the optical axis of one camera where $O_A$ is nearer to that camera than $O_B$, said camera may not see $O_B$ since it is hidden by $O_A$. However, the other camera can see both $O_A$ and $O_B$ provided there is no object $O_C$ that blocks either $O_A$ or $O_B$. Since one image can see $O_B$ and the other cannot, there is no known method for finding a correspondence in those areas. In Figure 18, note that the red work light is visible in both images, but that the tripod it sits on is obscured in the right camera by the lawn mower.

![Figure 18: Correspondence problem – obstructions](image)

**Scene Content Change:** With a single camera to take all the images, there will be a temporal displacement between any pair of images. During this time span, it is possible that the scene content itself could have changed. Small changes are acceptable as any feature matches that are outliers will be discarded. However, significant scene content changes can inject significant error into the system.
2.6 Calibration

Everything up to this point has been ground work in preparation for the calibration stage. Figure 19 illustrates the calibration process. This is a well-known and commonly-used process with its fundamentals in (Z. Zhang 2000) and (Brown, Close-range camera calibration 1971).

First, the individual features are extracted from each image according to the calibration method being used; this is described in Section Calibration via Calibration Panels. Then, the camera model and extrinsics are estimated. That estimate is then used to solve for the distortion vector, and a final computation of the camera model and extrinsics is performed.

2.6.1 Estimation of Camera Model and Extrinsic

The first step is to assume that there is no distortion (i.e., \( k_1 = k_2 = p_1 = p_2 = k_3 = 0 \)). This is done to temporarily constraint the size of the problem space. For every feature, the location
in both pixel space ($x$) and some, potentially arbitrary, world space ($X$) are known. These are related through a projection given by:

$$
x = sKWX
$$

where $K = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix}$ and $W = [R \ T]$ \hspace{1cm} (12)

The term $s$ is an arbitrary scale factor whose purpose is to explicitly denote that the homography is valid up to a particular scale. Given enough appropriate features (see Section 2.4.4), the system of equations becomes over-solved, and it is solved using the Levenberg-Marquart optimization technique (Levenberg 1944)(Marquardt 1963) to minimize the reprojection error. That is, it minimizes the average Euclidean distance between ($x_{proj}^{'}$,$y_{proj}^{'}$) and ($x$, $y$) with the relation:

$$
erreproj = \frac{\sum (x_{proj}^{'} - x)^2 + (y_{proj}^{'} - y)^2}{n}
$$

where $n$ is the total number of features among all the images and ($x_{proj}^{'}$,$y_{proj}^{'}$) is the projected pixel location of the world coordinate $X$.

### 2.6.2 Computation of the Distortion Vector

Once the intrinsic and extrinsics are determined, the next step is to solve for the distortion vector coefficients. This is done by projecting each point from world space into pixel space, denoted by $x_{proj}$. This projected point is then corrected for distortion using (14) as

$$
x_{proj}^{'} = (1 + k_1r^2 + k_2r^4 + k_3r^6)x_{proj} + (2p_1x_{proj}y_{proj} + p_2(r^2 + 2x_{proj}^2))
$$

\hspace{1cm} (14)
\[ y'_{\text{proj}} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) y_{\text{proj}} + (2 p_2 x_{\text{proj}} y_{\text{proj}} + p_1 (r^2 + 2 y^2_{\text{proj}})) \]

It is pertinent to note that \((x_{\text{proj}}, y_{\text{proj}})\) is just the projected coordinate \(x\) from (12). Since \(x'_{\text{proj}}\) and \(y'_{\text{proj}}\) are not known \textit{a priori} and the distortion vector cannot be directly solved for. Therefore, the Levenberg-Marquardt optimization is used once again.

### 2.6.3 Final Computation of Camera Model and Extrinsics

At this stage, there is a rough estimation of the camera model and extrinsics and a close estimation of the distortion vector. The last step in the calibration process is to re-evaluate the camera model and extrinsics. There is a final iteration through the Levenberg-Marquardt optimization, except this time the distortion vector is kept static while it solves for the camera model and the extrinsics.

At this point, the calibration is complete. For every image, the camera model and distortion vector is applied in order to undistort the image.
3 EXPERIMENTAL METHODOLOGY

3.1 Chapter Overview

This chapter describes how the experiments were conducted. It describes the hardware that was used, illustrates the optical setup, describes theoretical results, and presents the performance metrics.

3.2 Hardware and Software

The camera used in this research is an Allied Vision Technologies’ GE4900C gigabit Ethernet camera capable of producing 8-megapixel imagery at 3 frames per second. The GE4900C was combined with a LiNOS 80mm F-mount lens.

A Hewlett-Packard 8560W mobile workstation was used to process the data. With a quad-core i7-2860QM CPU running at 2.5 GHz and 16GB of DDR3 RAM, the laptop has an Nvidia Quadro 1000M graphics card with 2 GB GDDR3 and core, shader, and memory clocks of 700 MHz, 1400 MHz, and 1800 MHz, respectively.

With respect to software, the OpenCV library (version 2.4.5 (OpenCV 2014)) was used to detect features with all techniques using a calibration panel. Additionally, that same library was used to solve for the intrinsics, extrinsics, and distortion vectors. For the VSM approach, Visual Structure from Motion (version 0.5 (Wu n.d.)) was used to detect the SIFT features, and Yasutuka
Furukawa’s PVMS/CMVS implementation was used to create a 3D model, thus solving for the camera model.

### 3.3 Calibration Procedure

The first step in calibration is to ensure that the camera is in focus at the desired distance. To do this, the camera was aimed directly at the chessboard calibration pattern with zero rotation about the optical axis. By doing this, the horizontal and vertical lines of the board also appear horizontal and vertical, respectively, in the image itself. This makes it easy to see the pixel transition between the black and white squares, thus acting as a way to quantify camera focus.

The lens was focused at the pattern 10 feet away as that provided a good balance between field of view and the resolution of the target at that distance. Once good focus was achieved, the focusing ring on the lens was locked down in order to ensure the focal length was constant throughout the experiments.

At this point, an image of the chessboard pattern was captured from the camera. Immediately following, the chessboard pattern was replaced with each circleboard pattern, in turn, in the same position and pose as the chessboard pattern. Nine more sets of images were taken with each calibration panel in the same way. Keeping the positions and poses of the boards as identical as possible minimizes any variance. Conversely, the 10 images for VSFM used a static scene with a moving camera. This was necessary for VSFM to have sufficiently rich scene content that changed between images.
The chessboard and symmetric circle patterns had a total of 48 points, while the asymmetric circle pattern had 49. The position of the calibration pattern in each image was deliberately chosen such that the points spanned the entirety of the image. This is a necessary condition to correct the lens distortion. VSFM finds a large quantity of SIFT features as control points, but it is not practical to limit the number of features based on location in the image. Therefore, no restriction was placed on the number of features, nor their location, for VSFM. Each of the 10-image sets was processed using the method described in Section Calibration.

3.4 Data Sets

Figure 20 shows the chessboard, symmetric circleboard, asymmetric circleboard, and VSFM images in (a)-(d), respectively. Each row shows a comparable set of images for the three calibration pattern techniques. Notice how each triplet are nearly identical to each other sans the calibration panel itself. The position and pose of the panel, the position and pose of the camera, the lighting of the scene, etc. were all kept static for a quantitative comparison.

Since VSFM performs much better on a “natural” scene where a high number of SIFT features can be found, images for this technique did not include a calibration panel. The only requirements were that the position and pose of the camera for each image weren’t drastically changed in order to help ensure that VSFM found valid correspondences between SIFT features from different images. Note that this method of calibration differs from the others in the frame of reference. The other techniques kept the camera static and moved the features while the features for VSFM were kept static as the camera was relocated.
Figure 20: Data Sets.

(a) Chessboard
(b) Symmetric Circleboard
(c) Asymmetric Circleboard
(d) VSFM
3.5 Performance Metrics

Fundamentally, there are two metrics of interest for these experiments: the average reprojection error described in (13) and the execution time.

In the ideal case, the camera model and distortion vector is able to model the optical configuration precisely and correct the image perfectly, resulting in 0 pixels of reprojection error. However, the ideal camera calibration is not seen in practice. Conventionally, a calibration is considered to be “correct” if the average reprojection projection error is below 1 pixel, and good calibrations are below 0.75 pixels. A calibration below 0.5 pixels, though excellent, is quite rare and nearly-always requires an extremely methodical and precise calibration process.

Aside from the quality of the calibration is the execution time it takes to calculate the camera model. This includes the time it takes to load the images, perform feature detection, and calibrate the cameras to determine the intrinsics, extrinsics, and the distortion vector. Although each technique processes these stages, the actual work performed in each stage may differ.

Obviously, it is desirable for a technique to run as fast as possible as a calibration technique that requires too long a time may be undesirable. This is measured as the execution time, which is a combination of the time it takes to load the images from the hard drive, the time to detection the features in the imagery, and the time to calculate the intrinsics, extrinsics, and distortion vector. It is important to note that this research is a comparison of camera calibration techniques rather than the optimization of these techniques. There was no modification of software or design of experimentation to minimize the total execution time. This research just
reports the execution time of each algorithm’s publically-available implementation and compares them.

### 3.6 Theoretical Results

Though reprojection error is the single metric by which the quality of the camera calibration is being measured, the values of the individual variables within the camera model can be used to quantify the calibration performance. The principal point is perhaps the easiest to determine as it is simply the image center. For the GE4900C with a resolution of 4872x3248, the theoretical principal point is half of those dimensions; thus, \((C_x, C_y) = (2436, 1624)\).

If a lens was focused at infinity, a good estimate for the focal length would be:

\[
\begin{align*}
    f_x &= FS_x \\
    f_y &= FS_y
\end{align*}
\]  

where \(f_x\) and \(f_y\) are the focal length in pixel coordinates, \(F\) is the focal length of the lens in world coordinates (typically in millimeters), and \(S_x\) and \(S_y\) are the pixel density of the camera imager in world coordinates (typically in micrometers). However, as previously stated, the camera was focused at 10 feet, which is well short of infinity for this lens. Therefore, the focal length is actually estimated by:

\[
\begin{align*}
    f_x &= qS_x \\
    f_y &= qS_y
\end{align*}
\]

The value for \(q\) is given by:

\[
\frac{1}{p} + \frac{1}{q} = \frac{1}{F}
\]  

34
where $p$ is the distance from the focal plane to the target and $F$ is the focal length at infinity. The 80mm lens means that $F = 80$ mm. With that and $p = 3048$ mm (10 ft.) as the distance to the target, it follows that $q = 82.156$ mm. The pixel density is the inverse of the cell size. For the GE4900C, the pixel pitch in the imager is 7.4 $\mu$m. Thus, the pixel density is 135.135 pix/mm.

Additionally, since the cell is square, it is the case that $S_x = S_y$ and $f_x = f_y$. Therefore, let there be two new variables $f$ and $S$ such that $f = f_x = f_y = q$ and $S = S_x = S_y$. Therefore,

$$f = FS = (0.082156)(135.135) = 11,102.21$$

pixels.

Unlike the focal length and principal point, theoretical values for the distortion vectors cannot feasibly be calculated. While an ideal lens can have a distortion vector that is known beforehand, camera manufactures do not routinely do this for their lenses as it requires passing a high-resolution laser through the lens and sensing the laser beam’s location on a distant surface. Even if this were done for samples of a particular lens model, the imperfections in each individual lens can be significantly different and can change the distortion vector.

It is important to point out that while these metrics can provide additional insight into the quality of a particular camera calibration, they are never actually used in practice. This is because the equipment needed to empirically determine the needed measurements is expensive and typically not available in most situations. Where these metrics become very pertinent, however, is in synthetic data where the values are known a priori without any measurement.
It is equally as important to note that these are theoretical values and very likely are not the parameters in reality. Therefore, a measured value that is identical the theoretical value may actually be incorrect.
4 EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Average Reprojection Error and Time

Table 1 shows the timing and average reprojection error for each calibration technique. The values are an average of five independent runs of the software. It is expected that the load times for all techniques should be nearly identical. This is certainly the case with the chessboard, symmetric circleboard, and asymmetric circleboard. VSFM only reports timing results as an integer, and so it is very likely the time was just below 1 second. The difference in load times is estimated to be a tenth of a second and is likely caused by a difference of implementation to read the image files. As such, this aspect of the performance is not a significant item for comparison.

<table>
<thead>
<tr>
<th>Calibration Method</th>
<th>Chessboard</th>
<th>Symmetric Circleboard</th>
<th>Asymmetric Circleboard</th>
<th>VSFM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Image Time (seconds)</td>
<td>1.05</td>
<td>1.06</td>
<td>1.06</td>
<td>0.00</td>
</tr>
<tr>
<td>Feature Detection Time (seconds)</td>
<td>76.98</td>
<td>19.44</td>
<td>21.64</td>
<td>7.00</td>
</tr>
<tr>
<td>Calibration Time (seconds)</td>
<td>1.72</td>
<td>1.75</td>
<td>1.84</td>
<td>10.00</td>
</tr>
<tr>
<td>Total Execution Time (seconds)</td>
<td>79.75</td>
<td>22.24</td>
<td>24.54</td>
<td>17.00</td>
</tr>
<tr>
<td>Average Reprojection Error (pixels)</td>
<td>0.94</td>
<td>0.58</td>
<td>0.55</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Additionally, it is expected that the chessboard, symmetric circleboard, and asymmetric circleboard all have similar times for the calibration phase as the size of the system of equations is nearly the same for each.
The general trend for the results, shown in Figure 21 and Figure 22, is that a calibration technique that had a greater execution time resulted in a better calibration as noted by the average reprojection error. The exception to this is that the chessboard simultaneously took the longest to complete while having the worst average reprojection error, having spent nearly four times as long searching for its features as the other two calibration pattern techniques.

It should also be noted that VSFM has components that have been optimized to run on a graphical processing unit (GPU). None of the other techniques take advantage of GPUs or any other parallelization. It is very likely that each calibration pattern technique could be significantly improved by doing so. Of particular interest would be the high probably that both
circleboard pattern approaches could execute faster than VSFM while still retaining their lower reprojection error.

### 4.2 Camera Model

Even though there is no truth data for the particulars of the camera model, it is nonetheless interesting to compare the results of each calibration technique (see Table 2).

**Table 2: Comparison of Camera Models**

<table>
<thead>
<tr>
<th></th>
<th>Theoretical</th>
<th>Chessboard</th>
<th>Symmetric Circleboard</th>
<th>Asymmetric Circleboard</th>
<th>VSFM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_x )</td>
<td>11,102.21</td>
<td>11,006.21</td>
<td>11,230.67</td>
<td>10,785.06</td>
<td>11,391.34</td>
</tr>
<tr>
<td>( f_y )</td>
<td>11,102.21</td>
<td>11,036.06</td>
<td>11,314.85</td>
<td>10,824.88</td>
<td>11,391.34</td>
</tr>
<tr>
<td>( C_x )</td>
<td>2,436.00</td>
<td>2,835.50</td>
<td>1,989.24</td>
<td>2,407.73</td>
<td>N/A</td>
</tr>
<tr>
<td>( C_y )</td>
<td>1,624.00</td>
<td>1,485.51</td>
<td>2,037.46</td>
<td>1,754.67</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 3 shows a comparison between the difference of the theoretical camera model to the empirical camera model, as well as the average reprojection error. The green highlights show a low deviation, the yellow a significant deviation, and the red a large deviation. Though the categorization of the values is subjective, it does reveal that each camera model has portions that relate well to the theoretical model, as well as portions that deviate significantly.

**Table 3: Camera Model Deviations vs. Average Reprojection Error**

<table>
<thead>
<tr>
<th></th>
<th>Chessboard</th>
<th>Symmetric Circleboard</th>
<th>Asymmetric Circleboard</th>
<th>VSFM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta f_x )</td>
<td>+93.00</td>
<td>-128.46</td>
<td>+317.15</td>
<td>-289.13</td>
</tr>
<tr>
<td>( \Delta f_y )</td>
<td>+66.15</td>
<td>-212.64</td>
<td>+277.33</td>
<td>-289.13</td>
</tr>
<tr>
<td>( \Delta C_x )</td>
<td>+399.50</td>
<td>-446.76</td>
<td>-28.27</td>
<td>N/A</td>
</tr>
<tr>
<td>( \Delta C_y )</td>
<td>-138.49</td>
<td>+413.46</td>
<td>+130.67</td>
<td>N/A</td>
</tr>
<tr>
<td>Error</td>
<td>0.94</td>
<td>0.58</td>
<td>0.55</td>
<td>0.77</td>
</tr>
</tbody>
</table>
Looking closer at the focal length in particular, Table 4 back projects the pixel measurements into world coordinates via equations (16) and (17), taking values for $f$ from Table 3 and calculating $F$. Specifically, it calculates what the focal lengths would be in the standard optics unit - millimeters. The theoretical model is an ideal case, so minor deviations are expected, with one acute exception. The focal length of a lens focused at infinity is less than a focal length of the lens focused at a finite distance. Since the lens in this research was focused at 10 ft. rather than at infinity, the 80mm lens should have an empirical value greater than 80mm. Clearly, the camera model for the asymmetric circleboard does not conform to this expectation.

Table 4: Comparison of focal lengths in millimeters

<table>
<thead>
<tr>
<th></th>
<th>Theoretical</th>
<th>Chessboard</th>
<th>Symmetric Circleboard</th>
<th>Asymmetric Circleboard</th>
<th>VSFM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_x$</td>
<td>82.16mm</td>
<td>81.45mm</td>
<td>83.11mm</td>
<td>79.81mm</td>
<td>84.30mm</td>
</tr>
<tr>
<td>$F_y$</td>
<td>82.16mm</td>
<td>81.37mm</td>
<td>83.73mm</td>
<td>80.10mm</td>
<td>84.30mm</td>
</tr>
</tbody>
</table>

There are several reasonable explanations for this. The lens itself is advertised as an 80mm lens, but manufacturing imperfections can result in lenses that deviate slightly. However, machine vision lenses (which the LINOS lens is) are manufactured with strict quality control measures in place; deviations are typically on the order of 0.25mm or less. Another potential contributor to this aspect is the distance to which the camera is focused. However, the distance was accurate to within half an inch, which would change the focal length less than a hundredth of a millimeter. The conclusion therefore is that the circleboard camera model is using an inaccurate focal length. This is likely the result of the Levenberg-Marquardt optimization being stuck in a
local minimum. Regardless of this discrepancy, the asymmetric circleboard still provides the lowest average reprojection error.

To give an idea of the deviation of the principal points for each camera model, Table 5 shows the delta from the theoretical in both pixel space and world space. VSFM’s principal point is not included since its acceptance of the theoretical principal point as truth makes a comparison moot.

<table>
<thead>
<tr>
<th></th>
<th>Chessboard</th>
<th>Symmetric Circleboard</th>
<th>Asymmetric Circleboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta C_x$</td>
<td>+399.50 pix</td>
<td>-446.76 pix</td>
<td>-28.27 pix</td>
</tr>
<tr>
<td>$\Delta C_y$</td>
<td>-138.49 pix</td>
<td>+413.46 pix</td>
<td>+130.67 pix</td>
</tr>
</tbody>
</table>

As can be seen, the principal points for each model deviate on the order of millimeters. In the case of the asymmetric circleboard, it is under a millimeter of difference compared to the theoretical model. The focal plane array for the GE4900C is 36mm wide and 24mm tall. This means that the chessboard calibration’s error is 6% of the size of the focal plane array, the symmetric circleboard is off just shy of 11%, and the asymmetric circleboard is off only 2.3%.

### 4.3 Distortion Vector

The distortion vectors for each technique are shown in Table 6. For a perfect lens with no distortion, the distortion coefficients would all be 0; the further from 0 a coefficient is, the more it is correcting for distortion. Overall, there appears to be relatively little lens distortion.
whether radial or tangential. Though the chessboard and asymmetric circleboard patterns have $|k_3| > 8$, that is the third term in the Taylor series expansion and has relatively little effect.

In this case, the relatively small distortion is not surprising; the lens used was an 80mm LINOS lens. This is a high-quality machine vision lens. If a lower-quality lens had been used that showed significant “fish-eye” distortions, the coefficients for the radial distortion would be very high.

### Table 6: Comparison of Distortion Vectors

<table>
<thead>
<tr>
<th></th>
<th>Chessboard</th>
<th>Symmetric Circleboard</th>
<th>Asymmetric Circleboard</th>
<th>VSFM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>-0.066852</td>
<td>+0.067328</td>
<td>-0.101249</td>
<td>-0.001823</td>
</tr>
<tr>
<td>$k_2$</td>
<td>+1.078905</td>
<td>-0.424226</td>
<td>+1.342383</td>
<td>N/A</td>
</tr>
<tr>
<td>$k_3$</td>
<td>-8.705922</td>
<td>-0.093603</td>
<td>-8.353241</td>
<td>N/A</td>
</tr>
<tr>
<td>$p_1$</td>
<td>-0.003410</td>
<td>+0.005285</td>
<td>-0.006491</td>
<td>N/A</td>
</tr>
<tr>
<td>$p_2$</td>
<td>+0.007852</td>
<td>-0.010953</td>
<td>-0.001983</td>
<td>N/A</td>
</tr>
</tbody>
</table>

VSFM assumes there is no tangential distortion, and it only uses a single term for the radial distortion. The logical conclusion is that VSFM assumes that input images come from high-quality optical systems. This is not a bad assumption given the actual lens used and the empirical distortion values. It could well be the case that the distortion vector for all of these could include a single radial coefficient and no tangential coefficients, and still produce a relatively low reprojection error.

Interestingly enough, the chessboard and asymmetric circleboard methods have nearly identical distortion vectors, which would indicate that the quality of their reprojection errors is also likely to be close. However, their average reprojection errors are significantly different as the chessboard has by far the highest error and the asymmetric circleboard has the lowest average.
reprojection error. Since the distortion vector is determined after the extrinsics have been
initially solved, it can be assumed that the divergence came in the final stage of calibration
where the camera model and extrinsics are computed the second time. This is also supported by
the fact that both techniques initially assume an ideal camera model and yet in the end have
significantly different camera models.

4.4 Best Technique

Given the results, the obvious question is which technique is the best for single-camera
calibration. Based on the reprojection error, the asymmetric circleboard is the best choice. The
slightly longer execution time between it and the symmetric circleboard is a matter of seconds
and is negligible when compared to the time to set up each calibration panel.

However, there are situations where using an asymmetric calibration pattern (or, indeed, any
calibration panel) is not possible. The most pertinent is when the camera model is different
between calibration and subsequent uses. An excellent example of this is aerial imagery. The
camera’s environment on the ground and in the air is very different. Thermal expansion of the
physical camera and lens significantly affects camera models, and so any calibration performed
on the ground would almost certainly be invalid at altitude.

One way to perform the calibration from the air would be to arrange to have a very large
calibration pattern on the ground that is visible from the sky. While doable, it is typically not
very practical due to the size, manufacturing cost, and maintenance of such a fiducial. However,
based on the results of this research, VSFM would be an ideal candidate. Since it is based on SIFT
features rather than a known calibration pattern, it generally works well as long as the image scene is not featureless and valid image pairs are available. Recall the discussion from Section 2.5.1 which identified the issues affecting good correspondence matching.

4.5 Application of Camera Model

The 80mm lens used above is a high-quality machine-vision lens with a relatively long focal length. Therefore, it comes as no surprise that the distortion vector is nearly negligible. Because of this, the “before” and “after” pictures look nearly identical to the human eye. Since they typically can only be differentiated at the pixel level, it would not have been beneficial to show results from the 80mm lens.

While the mathematics behind the camera calibration process are sound, it is desirable to provide an example that is both more appealing and more substantial for visual inspection. A Prosilica GE1660C camera was paired with a MegaPixel CCTV 8mm lens. A calibration was performed on this setup and the results are in Figure 23. The left is an unaltered image, and the right is the same image with red lines superimposed to bring out the curvature caused by the distortion. Notice in the right image the substantial bowing of the left side of the near doorway as well as the smaller but still significant bowing of the right side of the far doorway.
Figure 23: (Left) High distortion. (Right) Lines accentuating the distortion.

Figure 24 shows the image from Figure 23 (left) that has had a calibrated camera model applied to it. Note that all doorway edges are now straight thus indicating they are free of any significant distortion. In this calibration, \((C_x, C_y) = (818, 585), f_x = 1485.7, f_y = 1501.1\), and 
\((k_1, k_2, p_1, p_2, k_3) = (-0.43, 0.16, 0.000012, -0.000033, 0.27)\). The reprojection error from this calibration is 0.81 pixels.

Figure 24: (Left) Corrected image. (Right) Lines indicating minimal distortion

This example verifies that the camera calibration procedure does indeed work, even for wide field-of-view lenses.
5 CALIBRATION ASSISTANT SOFTWARE

5.1 Motivation

As new researchers enter the field of optics, image processing, and related fields and they begin to learn about camera calibration, the question that invariably gets asked is “how many images do I need?” Current literature typically says that 8-12 images are sufficient, but the caveat is consistently that it really depends on how rich your data set is, where rich refers to the uniqueness of the positions and poses for each pattern (or for the camera itself if the scene is kept static).

A setup where there is minimal difference between the images will result in a calibration that was very susceptible to noise. This is because similar views provide mostly redundant information. Therefore, if a particular view by itself is not good for calibration, it is highly probable that few (if any) of the images will be sufficient for the calibration. However, if the camera views are sufficiently distinct, they can provide unique data to the system of the equations that the other views could not provide.

An aspect of camera calibration that is not often discussed is the distortion vector. Regardless of how many images are used in the calibration, there is the need to have feature points that span the entirety of the camera’s field of view. This is not to say that each image must accomplish this individually. Rather, the points from each image combined must sufficiently cover the field of view. When the distortion vector is calculated, it is based upon the location of feature points. If those feature points are only from a subsection of the field of view, then the calibration process
will apply a global operation to correct for distortion based on a local sampling of points that may not be representative of the rest of the field of view.

As part of this research, a tool was created to assist new users through the calibration process – the Calibration Assistant. It is a Java-based graphical user interface (GUI) that uses the java bindings for OpenCV to perform the calibration. The software uses a “Simon says” approach that guides the user in the placement of their calibration pattern. This qualitative approach has consistently resulted in good camera models. The following sections describe how to use the Calibration Assistant.

5.2 Walk-Through

The GUI is designed to be easy to use, even for novices. Figure 25 shows the initial screen when the software is loaded. The left two-thirds of the screen are reserved for displaying the user’s image. The upper-right corner shows the template pose that the user should mimic in their imagery. The middle-right shows information about the current calibration pattern being used. Lastly, the bottom right is reserved for displaying the outputs of a successful calibration.
5.2.1 Selecting a Calibration Pattern

To begin a new calibration, click on File in the menu and select New Calibration (see Figure 26). This will bring up a new window shown in Figure 27.
From the drop down menu, select a calibration pattern. Then, select the dimensions of the features in the board and click on the *Update* button. The calibration pattern that was selected will be displayed. This should match the physical calibration that will be used. If it does not, the settings may be altered until the software calibration pattern is correct. If there is an issue with the parameters, the error message is displayed at the bottom. Once a proper calibration pattern is selected, click *OK* (see bottom-right corner of Figure 28).

### 5.2.2 Loading Images and Detecting Features
At this point, images can be loaded into the software so that feature points can be extracted. This is done by clicking on the Load Image button and selecting the image to be used. After loading the image into memory, the software will automatically search for the feature points. Once found, they will be overlaid on the image as shown in Figure 29.

![Figure 29: Finding Features.](image)

Once the points are displayed, visually investigate the points to verify that OpenCV located the features correctly. For the chessboard pattern, this means that the features should be located at the intersection of four squares. For circleboard patterns, the features should be at the center of each dot. If the features were not detected in the appropriate position, you may need to reposition the board slightly and try again. Also, be sure to verify that the board is in roughly the same pose as indicated in the upper-right corner. It is not necessary to mirror it exactly, but there should not be any significant deviations from the template. If there are any issues, you may click on Load Image to replace the current image with another. Once satisfied that the features are good and the pose is correct, click on Accept Image. This will save the feature points and advance the template to the next pose. A new image is then loaded in.
There is a button labeled *Show Points*. When pressed, this will toggle the overlay of the current list of points from all images onto the currently displayed image. By displaying all accepted features, it becomes intuitively obvious where there is a lack of data for the distortion vector calculations and thus where further images should be positioned in the camera’s field of view to provide data in the appropriate region. This is shown in Figure 30.

![Figure 30: Displaying Features.](image)

### 5.2.3 Calibration

After six calibration patterns, the software will allow a calibration, but shades the *Calibrate* button yellow to indicate that it may not be a good calibration. After 10 images, the button will be shaded green (see Figure 31) to indicate that a good calibration is highly likely. Again, this is dependent on the precision of the located points, the degree to which the pose of the calibration pattern follows the template, and the coverage of the points in the camera’s field of view. Additional images beyond 10 can be used, but the template will not provide poses at that point.
When all desired images have been loaded and all desired features accepted, the final step is to calibrate. Click on the *Calibrate* button and the calibration assistant will perform single camera calibration. The camera model, distortion vector, and average reprojection error are displayed in the bottom right as shown in Figure 32.

If the average reprojection error is over 1 pixel, the calibration should not be considered a success and should be re-attempted after identifying and correcting for possible errors.

Alternatively, an average reprojection error under 1 pixel should be taken to be a success. The values should be preserved by the user for later use with the user’s own applications.
6 CONCLUSION

6.1 Summary of Results

This research has focused on the calibration of a single camera using four techniques: a chessboard calibration pattern, a symmetric circleboard calibration pattern, an asymmetric circleboard calibration pattern, and Visual Structure from Motion. For the calibration patterns, 10 images were taken with similar positions and poses of their feature points; VSFM uses SIFT features naturally present in the scene, so no calibrations patterns were used. All four techniques were compared based on similarity to the ideal camera model and average reprojection error.

Of the four techniques, the asymmetric circleboard performed the best, both having the lowest reprojection error as well as being the closest to the theoretical camera model. It was within several seconds of being the fastest algorithm as well, which is negligible from a human perspective. This makes it the most desirable method when a calibration pattern can be used. Additionally, none of the three calibration pattern techniques have been optimized to the extent of VSFM, so it is entirely possible that at least the circleboard patterns could perform much faster than VSFM with some optimization of OpenCV’s implementation.

In situations when using a fiducial is not possible, VSFM is a capable replacement. Although it had a higher reprojection error, it was still well within acceptable limits, and its independence from placing artificial objects in the scene makes it ideal for ad-hoc and on-the-fly camera calibrations.
Lastly, the calibration assistant software is a basic tool that leads new researchers through the calibration process. Providing step-by-step guidance from start to finish, it consistently provides good camera models and distortion vectors.

### 6.2 Contributions

Several contributions have been made in this research. The first is a quantifiable comparison of four methods of calibrating a single camera, including an analysis of which technique performs best in various situations.

The second contribution has been a thesis paper describing the fundamentals of camera calibration that fully explains the calibration process. While there are numerous papers in the literature that describe the process, most gloss over the issues of why a calibration may not be successful. Examples of this include the importance of the camera being in focus, locking down the focusing element, the size of the dots for both circleboard patterns, etc. This paper identifies these aspects and explains what they are and why they are important to the calibration process.

A final contribution is the development of GUI-based software as a tool to calibrate a camera with a chessboard, symmetric circleboard, or asymmetric circleboard pattern. It provides a simple yet straight-forward interface that allows a user to calibrate a camera, providing feedback throughout the process to allow the user to correct potential issues before they are accepted and inject error into the system.
6.3 Future Work

There are several areas of this research that merit further investigation. Probably the most important among them is the need for truth data. Since it is not trivial to actually measure the true camera model, it is nearly always the case that there are only empirical values. If it were, in fact, easy to measure the true camera model, the calibration techniques used in this research would be irrelevant and possibly would not have been developed.

In order to get truth data, the obvious solution is to use synthetic data where the parameters are completely known a priori. In this way, any deviations from the true values can be quantitatively measured and further experiments where particular variables can be finely adjusted could be conducted. The only real caveat to using synthetic data is making sure that it is representative of the real-world. Properties such as ambient light, reflections, focus issues, integration time, and blurring must be modeled appropriately to ensure that results from synthetic data will translate well into the real world.

Another area of research that could be easily conducted with synthetic data would be a study of the range of poses. While the calibration assistant software guides the user with poses from a known good calibration, it would be beneficial to know the most effective combination of poses that minimizes the number of images needed to provide a sufficiently rich data set. First, the range of the independent rotations should be investigated. Then, combinations of different rotations can be combined. Once the extremes are known, a sampling of the interior of the problem space should provide the ideal combination for the given conditions.
Although OpenCV has not adopted anything beyond the three calibration patterns described in this research, there are, in fact, many other calibration patterns that have been independently explored. A trivial example of this would be using rings instead of solid dots. A more comprehensive study would include a calibration pattern that had a mixture of feature types, with the goal of extracting the best of all pattern types.

The calibration process should be examined to determine if an additional iteration of the Levenberg-Marquart optimization would be significantly beneficial. The chessboard and asymmetric circleboard patterns both started with an ideal camera model and have very similar distortion vectors, but their final camera model and average reprojection error are very different. This deviation could only have occurred at the last stage of calibration since the distortion vector is fixed. An additional iteration through the process, tagged on to the end of the currently-accepted process, would increase time but may significantly improve the camera model.

Finally, an investigation into using the direct linear transformation to solve for the initial camera intrinsics should be conducted. Let there be a matrix $P = sKW$, which simplifies (12) to $\hat{x} = P\hat{X}$. In this instance, $\hat{x}$ and $\hat{X}$ are matrices containing all the pixel and world points, respectively. $P$ is therefore the projection matrix that projects a point in world space into pixel coordinates. By definition of the direct linear transformation,

$$P = \hat{x}\hat{X}^T(\hat{X}\hat{X}^T)^{-1}$$

(18)
This would provide a linear method to solve the initial camera matrix rather than the non-linear Levenberg-Marquardt technique. As such, the quality and speed of the direct linear transformation should be compared to the Levenberg-Marquardt technique.
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFRL</td>
<td>Air Force Research Labs</td>
</tr>
<tr>
<td>FPA</td>
<td>Focal Plane Array</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphical Processing Unit</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>SFM</td>
<td>Structure from Motion</td>
</tr>
<tr>
<td>USAF</td>
<td>United States Air Force</td>
</tr>
<tr>
<td>VSFM</td>
<td>Visual Structure from Motion</td>
</tr>
<tr>
<td>WSU</td>
<td>Wright State University</td>
</tr>
</tbody>
</table>
APPENDIX B: CHESSBOARD FEATURES

The following figures show the chessboard calibration images with the feature locations overlaid. Each color represents a row of features, with each dot in the row marking the exact location of an individual feature. The images have been cropped to better show the features’ locations.

![Chessboard #1](image.png)

Figure 33: Chessboard #1
APPENDIX C: SYMMETRIC CIRCLEBOARD FEATURES

The following figures show the symmetric circleboard calibration images with the feature locations overlaid. Each color represents a row of features, with each dot in the row marking the exact location of an individual feature. The images have been cropped to better show the features’ locations.

Figure 43: Symmetric Circleboard #1
Figure 44: Symmetric Circleboard #2

Figure 45: Symmetric Circleboard #3
Figure 48: Symmetric Circleboard #6

Figure 49: Symmetric Circleboard #7
Figure 50: Symmetric Circleboard #8

Figure 51: Symmetric Circleboard #9
APPENDIX D: ASYMMETRIC CIRCLEBOARD FEATURES

The following figures show the asymmetric circleboard calibration images with the feature locations overlaid. Each color represents a row of features, with each dot in the row marking the exact location of an individual feature. The images have been cropped to better show the features’ locations.

Figure 53: Asymmetric Circleboard #1
Figure 58: Asymmetric Circleboard #6

Figure 59: Asymmetric Circleboard #7
Figure 60: Asymmetric Circleboard #8

Figure 61: Asymmetric Circleboard #9
Figure 62: Asymmetric Circleboard #10
APPENDIX E: VSFM FEATURES

The following figures show the VSFM calibration images with a graphical description of the features overlaid onto the image. The center of each circle represents the pixel location of the center of the feature while the size of the circle denotes the scale of the feature. The radius of each circle symbolizes the orientation of the feature.

Figure 63: VSFM #1
BIBLIOGRAPHY


