2004-10-13

Development of a Virtually Calibrated Projection Moire Interferometry Technique Capable of Inaccessible Surface Measurements

Mark Lee Kimber
Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Mechanical Engineering Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/192

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
DEVELOPMENT OF A VIRTUALLY CALIBRATED PROJECTION MOIRÉ INTERFEROMETRY TECHNIQUE CAPABLE OF INACCESSIBLE SURFACE MEASUREMENTS

by

Mark L. Kimber

A thesis submitted to the faculty of

Brigham Young University

Master of Science

Department of Mechanical Engineering

Brigham Young University

December 2004
of a thesis submitted by

Mark L. Kimber

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

__________________________________________
Date Jonathan D. Blotter, Chair

__________________________________________
Date Kenneth W. Chase

__________________________________________
Date Robert H. Todd
As chair of the candidates graduate committee, I have read the thesis of Mark L. Kimber in its final form and have found that (1) its format, citations, and Bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Jonathan D. Blotter
Chair, Graduate Committee

Accepted for the Department

Matthew R. Jones
Graduate Coordinator

Accepted for the College

Douglas M. Chabries
Dean, College of Engineering and Technology
ABSTRACT

DEVELOPMENT OF A VIRTUALLY CALIBRATED PROJECTION MOIRÉ INTERFEROMETRY TECHNIQUE CAPABLE OF INACCESSIBLE SURFACE MEASUREMENTS

Mark L. Kimber
Department of Mechanical Engineering
Masters of Science

Optical-based techniques have found merit in measuring displacement and strain for decades. These techniques are commonly used in numerous applications ranging from large displacements in wind tunnel experiments to displacement measurements on the submicron scale. Projection Moiré Interferometry (PMI) is an out-of-plane displacement measurement technique, and consists of capturing reference and deformed images of a grid pattern projected on the test object. By differencing the reference and deformed images of the projected grid pattern, a fringe pattern is generated from which the displacement field can be extracted. This computation requires calibration procedures that analyze a number of fringe patterns from known displacements to compute the fringe sensitivity constant (FSC) values. This process can be time consuming and for large-scale applications, very costly. In addition, due to the projection-oriented nature of this
technique, measuring displacements in applications with non-viewable, hidden, or inaccessible reference surfaces excludes the use of PMI. In this thesis, a technique is developed which eliminates calibration procedures through implementation of virtual calibration methods, and typical PMI measurement processes are extended to include digital reference images in determining displacements from inaccessible surfaces. Using camera calibration routines and ray tracing techniques, each major component of the PMI arrangement is modeled as virtual components within a computer simulation where the entire calibration process can be performed. A CAD model of the inaccessible surface is then converted to a point cloud and a surface interpolation function is implemented to generate a displacement field, which can be correlated and differenced from the displacement field of the actual object. Many potential applications exist in the automobile, aerospace, and other manufacturing industries. These industries provide numerous large-scale applications where conventional calibration is not cost-effective. In addition, these applications provide instances where differences between the deformed and reference images represent the manufacturing errors due to dimensional variations and assembly processes. An automated, self-calibrating, whole-field projection measuring system would greatly increase inspection efficiency of large production parts and final assemblies. It is in these types of circumstances that the developed techniques would be of most use.
ACKNOWLEDGEMENTS

The research related to this thesis was at times, as all research is, extremely challenging. I would like to thank all those who have assisted me in my efforts and contributed in any way to the completion of this thesis. First, I wish to gratefully acknowledge my advising professor and committee chair, Dr. Jonathan Blotter. He has dedicated many long hours, especially near approaching deadlines, revising and carefully evaluating my research and writing. I also wish to thank the other two members of my committee, Dr. Kenneth Chase, and Dr. Robert Todd, for their helpful insights. In addition, I acknowledge help from my lab mates spending many hours, even days at a time helping me with a difficult bug in my code or understanding a concept, only to find out weeks later the wrong path was taken, and a fresh start would be needed.

These acknowledgments would not be complete if I did not dedicate at least a sentence or two in honor of my wife, Kelli. She has sacrificed and endured as much, or more, than I have. Through years of graduate student pay and many long nights alone with two small children, she has found a way to support me through it all. I would never have attempted such a task without her constant support and encouragement. Lastly, I recognize those moments of inspiration during times of discouragement, and I direct my thanks heavenward.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE PAGE</td>
<td>i</td>
</tr>
<tr>
<td>SIGNATURE PAGES</td>
<td>ii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>iv</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>vi</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xi</td>
</tr>
<tr>
<td>1  INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 PROBLEM STATEMENT</td>
<td>1</td>
</tr>
<tr>
<td>1.2 OBJECTIVE AND GOALS</td>
<td>4</td>
</tr>
<tr>
<td>1.3 HYPOTHESIS</td>
<td>4</td>
</tr>
<tr>
<td>1.4 THESIS OUTLINE</td>
<td>7</td>
</tr>
<tr>
<td>2  BACKGROUND AND LITERATURE REVIEW</td>
<td>9</td>
</tr>
<tr>
<td>2.1 FUNDAMENTALS IN OPTICS</td>
<td>9</td>
</tr>
<tr>
<td>2.1.1 Behavior of Light</td>
<td>10</td>
</tr>
<tr>
<td>2.1.2 Concept of Interference</td>
<td>11</td>
</tr>
<tr>
<td>2.1.3 Interferogram Analysis</td>
<td>12</td>
</tr>
<tr>
<td>2.2 MOIRÉ EFFECT</td>
<td>14</td>
</tr>
<tr>
<td>2.3 MOIRÉ MEASUREMENT METHODS</td>
<td>15</td>
</tr>
<tr>
<td>2.4 PROJECTION MOIRÉ INTERFEROMETRY</td>
<td>17</td>
</tr>
<tr>
<td>2.4.1 PMI Methods at BYU</td>
<td>19</td>
</tr>
<tr>
<td>2.4.2 Limitations Addressed</td>
<td>25</td>
</tr>
<tr>
<td>3  VIRTUAL CALIBRATION PROCEDURES</td>
<td>27</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1-1  Thesis Goals ........................................................................................................ 4
Table 4-1  Obtaining Camera Calibration Results ................................................................. 49
Table 4-2  Final Camera Calibration Results ........................................................................ 49
Table 4-3  Obtaining Projector Calibration Results ............................................................... 51
Table 4-4  Final Projector Calibration Results ...................................................................... 51
Table 4-5  Averages and Errors for Actual and Virtual FSC Values ................................. 61
LIST OF FIGURES

Figure 1-1 Thesis Contribution................................................................. 3
Figure 1-2 Virtual Calibration and Inaccessible Surface Measurement Procedures .... 6
Figure 2-1 Example of Phase Unwrapping.............................................. 14
Figure 2-2 Illustration of the Moiré Effect .............................................. 15
Figure 2-3 Rotation Generated Interferogram ........................................ 16
Figure 2-4 Typical PMI Setup ................................................................. 18
Figure 2-5 Typical PMI Procedures ....................................................... 19
Figure 2-6 Sample Images of Dot Card: a) Original and b) Dewarped .......... 20
Figure 2-7 Sample Images: a) Reference and b) Calibration .................. 21
Figure 2-8 Sample Fringe Patterns: a) Original and b) Filtered ................. 21
Figure 2-9 Phase Stepped Fringe Patterns: a) 0°, b) 90°, c) 180°, and d) 270° .... 23
Figure 2-10 Sample Phase Maps: a) Wrapped and b) Unwrapped ............. 23
Figure 3-1 Virtual PMI Calibration Procedures ....................................... 28
Figure 3-2 Illustration of Intrinsic Parameters ....................................... 30
Figure 3-3 Illustration of Extrinsic Transformation .................................. 31
Figure 3-4 Illustration of Distortion Models: a) Radial, b) Tangential, and c) Combined ................................................................. 34
Figure 3-5 Camera Calibration Using Planar Checkerboard Pattern .......... 36
Figure 3-6 Projector Calibration Using Projected Checkerboard Pattern ....... 38
Figure 3-7 Virtual PMI Setup ............................................................... 39
Figure 3-8 Virtual Camera .................................................................... 41
Figure 3-9 Virtual Projector ................................................................. 42
Figure 3-10  Capturing Virtual Calibration Images....................................................... 43
Figure 4-1  Experimental PMI Setup........................................................................... 46
Figure 4-2  Planar Checkerboard Used for Camera Calibration............................... 47
Figure 4-3  Pixel Error Illustration .............................................................................. 48
Figure 4-4  Bitmap Checkerboard Image Used for Projection.................................... 50
Figure 4-5  Planar Checkerboards in Reference Plane:  a) Actual Undistorted and b) Virtual................................................................................................... 53
Figure 4-6  Judging Camera Correlation:  a) Extracted Corners and b) Differenced Image (Actual – Virtual)................................................................. 54
Figure 4-7  Difference in Camera Extracted Corners:  a) Horizontal and b) Vertical. 54
Figure 4-8  Projected Checkerboards on Reference Plane:  a) Actual Undistorted and b) Virtual ............................................................................................ 55
Figure 4-9  Judging Projector Correlation:  a) Extracted Corners and b) Differenced Image (Actual – Virtual)................................................................. 56
Figure 4-10 Difference in Projector Extracted Corners:  a) Horizontal and b) Vertical.............................................................................................................. 56
Figure 4-11 Virtual Calibration Images:  a) Reference and b) 15° Rotation.............. 58
Figure 4-12 Actual Undistorted Calibration Images:  a) Reference and b) 15° Rotation........................................................................................................... 59
Figure 4-13 Differenced (Actual – Virtual) Calibration Images:  a) Reference and b) 15° Rotation................................................................................................... 59
Figure 4-14 Percent Difference in FSC Coefficients:  a) Quadratic, b) Linear, and c) Constant ......................................................................................................... 61
Figure 4-15 Displacement Difference Introduced by Virtual Calibration.................... 63
Figure 4-16 Average Displacement Difference for all Rows........................................ 64
Figure 4-17 Average Displacement Difference for all Columns.................................... 64
Figure 4-18 Rotated Plate Data..................................................................................... 66
Figure 4-19 Average Displacement Errors with Error bars......................................... 66
Figure 4-20 Sample Displacement with Average Errors and Error bars................. 67
Figure 5-1  Inaccessible Surface Measurement Procedures ........................................ 72
1 INTRODUCTION

This chapter presents a brief synopsis of the background and objectives related to this thesis. Explanation is given of the main limitations of PMI that are addressed as well as the specific goals to this research. The remaining chapters of this thesis are then outlined.

1.1 PROBLEM STATEMENT

For centuries, the classical study of optics has proved to be useful in many scientific advances, particularly metrology. Optical-based measuring techniques are currently used for many types of measurements including flow field,\(^1\),\(^2\) displacement,\(^3\),\(^4\) temperature,\(^5\),\(^6\) and surface roughness,\(^7\),\(^8\) and as well as others. In most instances, optical techniques are preferred over traditional methods because they are typically non-invasive and provide whole field measurement data. Projection Moiré Interferometry (PMI) is a low-cost, non-intrusive, whole-field measuring technique for out-of-plane displacements. In conventional PMI, a grid of equally spaced parallel lines is projected onto the surface of a structure. In a reference state, the projected grid lines have a certain spatial distribution, which changes as the structure is loaded or deformed. When images of the deformed grid lines are subtracted from the reference grid lines, a fringe pattern is generated from which the magnitude of the structural displacements can be extracted. This is done by computing the fringe sensitivity constant (FSC)\(^9\) values, which is usually
performed through calibration procedures. These procedures consist of analyzing fringe patterns generated from known displacement fields. The FSC values are then applied to a fringe pattern generated by differencing the reference image and an unknown displaced image.

As is common with most measurement techniques, PMI has inherent characteristics that allow or limit its use in certain types of applications. This thesis addresses two of the limiting characteristics inherent in PMI. The first limitation of PMI addressed in this work is the tedious and at times cumbersome process of obtaining FSC values from standard calibration procedures. In many instances, the calibration procedure requires the use of a large planar surface and the ability to rotate or translate this surface to known amounts within a certain tolerance range. For large-scale applications, this becomes especially difficult and expensive. The second limitation of PMI addressed in this work is the requirement of at least two images in order to extract displacements. One of these two images corresponds to the reference image of the grid pattern. Obtaining this image becomes difficult when the reference surface is not viewable by the camera. Examples of such applications can be seen when comparing a mold or die to the part produced. In this instance, the mold and actual part represent the reference and deformed surfaces respectively. In many cases, the mold is not viewable by a CCD camera, or is inverted, and a reference image cannot be acquired. This thesis addresses each of those limitations of PMI by developing a virtual calibration procedure for the PMI setup and incorporating a CAD model of the inaccessible reference surface in the displacement measurement process. This contribution is shown in Figure 1-1, which describes the relationship between PMI and the rest of the moiré measurement methods.
There are two types of deformations measured by moiré methods: in-plane and out-of-plane deformations. PMI is described as an out-of-plane measurement method, and like all methods, has strengths and weaknesses. The research performed focuses on the two weaknesses shown in the illustration.

**Figure 1-1 Thesis Contribution**

Overcome the above PMI weaknesses by extending on current techniques and developing the following:
- Virtual calibration of PMI setup
- Method to use CAD model in measurement process.
Many potential applications exist in the automobile, aerospace, and other manufacturing industries. These industries provide numerous large-scale applications where performing calibration is not cost-effective. These applications also provide instances where differences between the deformed and reference images represent the manufacturing errors due to dimensional variations and assembly processes, which can be considered to have inaccessible reference surfaces. An automated, virtual calibrating, whole-field projection measuring system would greatly increase inspection efficiency of large production parts and final assemblies. It is in these types of circumstances that the developed techniques would be of most use.

1.2 OBJECTIVE AND GOALS

The objectives of this research are to develop techniques for the virtual calibration of typical PMI setups and account for extracting displacements when the reference surface is inaccessible. These objectives will be accomplished through the goals shown in Table 1-1.

<table>
<thead>
<tr>
<th>Table 1-1 Thesis Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Eliminate traditional calibration methods through the development of procedures for virtual calibration of typical PMI setups</td>
</tr>
<tr>
<td>2. Account for inaccessible reference surface displacements by incorporating CAD models in the overall measurement process</td>
</tr>
</tbody>
</table>

1.3 HYPOTHESIS

The two thesis goals will be accomplished through steps outlined in Figure 1-2. This illustration divides the entire process into two separate procedures describing both of
the individual thesis goals. Virtual calibration is performed through four steps. The first step makes use of camera calibration routines\textsuperscript{10-13} to determine the internal characteristic parameters of the CCD camera as well as the precise position and orientation of the test specimen with respect to the camera. The second step uses the results obtained in the first step and similar camera calibration routines to likewise determine the characteristic parameters of the projector, including position and orientation with respect to both the test specimen and camera. In the third step, a virtual PMI setup is created in a computer simulation consisting of two virtual components representing the camera and projector. In this thesis, “virtual” implies a software model as compared to “actual”, which implies the physical or real component. Using the results acquired in steps one and two, these virtual components are created with internal properties which correspond to the components in the actual setup. They are then positioned and orientated to ensure accurate representation of the actual PMI setup. The fourth step uses ray-tracing techniques to capture virtual calibration images. These images are then used in place of traditional calibration images to determine the FSC values, thereby eliminating the task of rotating or translating the physical reference surface known amounts to calculate the FSC values. The system essentially becomes self-calibrating.

The second thesis goal is to incorporate inaccessible reference surfaces in the PMI measurement process. This will be accomplished through three steps as shown in Figure 1-2. The first step is to extract point cloud data from a CAD model of the reference surface. This is done by first converting the file into a common format known as stereolithography, or STL. This format converts the surface into triangle elements from which the element vertices are extracted and used as a cloud of discrete points to describe
the surface. The second step requires adjusting the scale and orientation of the point cloud through a registration process. This is done in order to measure displacements of point cloud data along a direction parallel to that used in the PMI analysis, and requires knowledge of the orientation of the actual object. The third step is to create a displacement field by applying a three dimensional interpolation to the point cloud to create a continuous surface. Results from camera calibration are used in this step to determine the part location corresponding to each camera pixel, thereby generating a displacement field obtained directly from the CAD data, which can then be compared to the displacement field found through a PMI analysis of the actual object. Any differences between these two displacement fields represent the dimensional variations from the CAD model to actual part.

Figure 1-2 Virtual Calibration and Inaccessible Surface Measurement Procedures
1.4 Thesis Outline

The remainder of this thesis describes the developed procedures in detail as well as the validations performed. Chapter 2 presents a literature review and background necessary to understanding PMI-related concepts. Chapter 3 provides a detailed description of the virtual calibration procedures, including relevant theory and background. Chapter 4 describes an experimental validation of virtual calibration procedures, where a flat plate is used as the object under investigation. FSC values are calculated using the virtual calibration procedures, and compared to traditional calibration methods. Chapter 5 presents details related to inaccessible surface measurements and the fundamentals upon which they are based. Chapter 6 provides validation through a specific test case, in which displacements of an airfoil are compared to a theoretical CAD model. In an attempt to measure springback of the airfoil experienced during assembly, displacements of a CAD model representing the inaccessible surface of a “perfect” airfoil is compared to displacements extracted from the assembled airfoil using virtual calibration procedures. Conclusions are made in Chapter 7 as well as recommendations for further research in this area. Chapter 8 list references and is followed by the Appendix, which includes other information relevant to recreating the experimental work presented.
2 BACKGROUND AND LITERATURE REVIEW

This chapter presents an overview of fundamentals common to most optical measuring techniques. Background for moiré measurement methods is presented, and various methods are discussed. Particular attention is drawn to projection moiré interferometry (PMI), and explanation is given of the main limitations of PMI that this work addresses. Topics presented in this chapter are outlined as follows:

- Fundamentals in Optics
- Moiré Effect
- Moiré Measurement Methods
- Projection Moiré Interferometry

2.1 FUNDAMENTALS IN OPTICS

There are numerous optical measuring techniques in practice today, and each is unique in the underlying principles upon which it is based. Describing these principles for every technique in adequate detail would require a document much longer than this thesis. For the present study, it might be useful to describe the fundamentals common to most existing techniques. This will create a foundation of knowledge, which items specific to PMI can be built upon. The behavior of light will first be discussed, since it serves as a starting point for optics in general. The concept of interference is then
presented, which is key to understanding the process of extracting measurements. The analysis techniques commonly employed to accomplish this task are then described.

### 2.1.1 Behavior of Light

Over the course of history, light has been described in various ways. Theories were first proposed of the oscillatory behavior of light by Robert Hooke and Christian Huygens. This line of thought was opposed by Sir Isaac Newton, who was a proponent of the particle nature of light. Eventually, James Maxwell in the nineteenth century suggested that light could be described by the electromagnetic field, which is the accepted description of light today. Light is in fact the visible portion of the electromagnetic spectrum. Therefore, light can be described as a propagating wave.

The motion $\psi(z,t)$ of a two-dimensional propagating harmonic wave can be described by Eq. (2.1), where $z$ is the spatial position, $t$ is time, $U$ is the amplitude, $\lambda$ is the wavelength, and $v$ is the frequency (number of waves per unit time). In addition, the expression within the cosine function is termed the phase where $\delta$ is the phase constant.

$$\psi(z,t) = U \cos \left[ 2\pi \left( \frac{z}{\lambda} - vt \right) + \delta \right] \tag{2.1}$$

To account for the electromagnetic field in three dimensions, Eq. (2.1) is modified as shown in Eq. (2.2), where the field at some point $\vec{r} = (x,y,z)$ is explained for a wave propagating in a direction described by the unit vector $\vec{n}$. The parameter $k$ is referred to as the wave number where $k = 2\pi / \lambda$.

$$\psi(x,y,z,t) = U \cos \left[ k\vec{n} \cdot \vec{r} - 2\pi vt + \delta \right] \tag{2.2}$$
It is a common practice to represent this field in complex form as shown in Eq. (2.3) where \( \phi = k\bar{n} \cdot \bar{r} + \delta \) and is referred to as the spatial phase. It is often understood that the real part of the expression is descriptive of the field, and ‘Re’ will be excluded from all equations from this point. It is readily apparent that the temporal and spatial components can then be separated as shown in Eq. (2.4).

\[
\psi(x, y, z, t) = \text{Re}\{Ue^{i(\phi - 2\pi vt)}\} \tag{2.3}
\]

\[
\psi(x, y, z, t) = Ue^{i(\phi - 2\pi vt)} = Ue^{i\phi}e^{-i2\pi vt} \tag{2.4}
\]

The spatial solution is typically the only component of interest for most optical measuring techniques, and therefore, the solution is expressed in the form shown in Eq. (2.5). This is the basis for the concept of interference, which will be discussed next.

\[
u = Ue^{i\phi} \tag{2.5}
\]

### 2.1.2 Concept of Interference

Interference occurs when two or more unique waves are superimposed, resulting in a constructive/destructive interference pattern. When two waves interfere, electromagnetic wave theory suggests that the resulting waveform is the sum of the original two waves. For illustration purposes, if two waves of the form \( u_1 = U_1 e^{i\phi_1} \) and \( u_2 = U_2 e^{i\phi_2} \) interfere, the sum \( u = u_1 + u_2 \) describes the resulting wave. However, intensity, not amplitude, is the observable quantity, and it helps to write the interference expression as shown in Eq. (2.6), which describes the resulting intensity at a single point.

\[
I = |u|^2 = |u_1 + u_2|^2 = U_1^2 + U_2^2 + 2U_1U_2\cos(\phi_2 - \phi_1) = I_1 + I_2 + 2\sqrt{I_1I_2}\cos\Delta\phi \tag{2.6}
\]
This equation suggests the phase difference ($\Delta \phi$) between the two waves remains constant at a given point over time. In most cases, however, this phase difference varies randomly at any particular point. A parameter known as the degree of mutual coherence $\gamma(\tau)$ is used to account for this random behavior of the phase difference as shown in Eq. (2.7). This parameter is derived from time averaged phase differences at any single point over some time step ($\tau$) and is defined in Eq. (2.8). The degree of mutual coherence varies according to $0 \leq |\gamma(\tau)| \leq 1$, and $\gamma$ is, therefore, a gage in determining how well two wave fields interfere. In reality, this is difficult to estimate, but as will be shown in the next section, methods using multiple images can be incorporated to perform this task.

$$I = I_1 + I_2 + 2\sqrt{I_1I_2}|\gamma(\tau)|\cos \Delta \phi$$

(2.7)

$$\gamma = \frac{1}{\tau} \int_0^\tau e^{i(\phi_1 - \phi_2)} dt$$

(2.8)

### 2.1.3 Interferogram Analysis

Many optical measuring techniques use the concept of interference to generate image patterns called interferograms. These patterns represent the measurement under investigation, and in most cases, the quantitative results of the desired measurement can be extracted from the interferogram. The general techniques used to accomplish this task are now discussed.

Extracting measurement data from an interferogram is based on calculating the phase change for each pixel in the image, which for many techniques, is easily correlated to the measurements under investigation. Using Eq. (2.7), the phase difference could theoretically be calculated from the degree of mutual coherence ($\gamma$) and intensities of both
waves before ($I_1$ and $I_2$) and after ($I$) interference. This must be done for each pixel in the image, and the difficulty in approximating the degree of mutual coherence suggests other solutions should be investigated. The most widely accepted solution is to introduce controlled phase shifts\textsuperscript{20,21} to perform this task. To illustrate this concept, Eq. (2.7) is written in a slightly different form as shown in Eq. (2.9), where $a$ and $b$ replace the corresponding parameters from the previous equation. A controlled phase step ($\beta$) is also introduced, which represents a fixed amount to phase shift the interferogram.

\[ I = a + b \cos(\phi + \beta) \]  

(2.9)

Since the phase step is a controllable parameter and the intensity is determined from pixel values, Eq. (2.9) contains only three unknowns ($a,b,\phi$) at each pixel for any given interferogram. The use of three or more controlled phase steps on the same interferogram enables the phase difference at each point ($\phi$) to be determined directly from the intensities of each phase-stepped image. Gåsvik and Kreis\textsuperscript{22} suggest numerous methods to introduce the phase steps including 3-step and 5-step techniques for known and even unknown phase shifts. The solution is typically of the form: $\phi = \tan^{-1}[C/D]$, and the process is referred to as phase wrapping due to the $\tan^{-1}$ function within the expression. Therefore, the phase values ($\phi$) range from $-\pi/2$ to $\pi/2$. The result is referred to as a wrapped phase map, and includes discontinuities for every extreme change. By unwrapping the phase, these discontinuities are removed and the phase is known for each point on the image. This process is illustrated in Figure 2-1, where the wrapped phase map is shifted up for each discontinuity in order to create the unwrapped phase map. Determining phase across the entire domain is a necessary step when analyzing an
interferogram. A more detailed description of the specific steps used to accomplish this task within the thesis scope is discussed in Section 2.4.1.

![Figure 2-1 Example of Phase Unwrapping](image)

**2.2 MOIRÉ EFFECT**

The moiré effect occurs when two repeating patterns overlap and a set of fringes appear. Figure 2-2 provides an illustration of this effect where elliptical and straight-line patterns are overlapped. The dark and light patterns shown in the overlapped pattern are referred to as moiré fringes. Many examples are also seen by the casual observer, drawing attention due to the quick movement of fringes in relation to the movement of the observer. Such examples include two railings on a staircase seen from a distance (repeating vertical lines), the back and front covers of a household fan (radial repeating patterns), or two window screens (2-D grid pattern). This type of fringe pattern is referred to as manual interference, and can be treated in a similar fashion as other interference patterns described in Section 2.1.3.
As early as 1874, Lord Rayleigh proposed the use of this phenomenon in scientific study, but it did not receive much consideration beyond judging the quality of a grating. It was not until the 1940’s that the moiré effect was applied to the field of displacement and strain measurement. Since that time, applications have grown to include surface and difference contouring, vibration analysis, and investigations in the human body. Much has been done for the advance of moiré metrology and new applications are found frequently for this constantly growing field.

### 2.3 Moiré Measurement Methods

Moiré methods are termed such due to their use of the moiré effect to make measurements. Typically, a grid-like pattern is projected and/or manufactured on the object of interest. Two separate patterns, termed the reference and deformed patterns, are superimposed to create interference. The reference pattern is formed on the object in its initial, or reference state, and as the object is deformed, or given a displacement, the grid
pattern is transformed as well, resulting in the deformed pattern. The resulting interferogram can be analyzed to extract the quantitative displacement data. By way of illustration, if a rotation were the displacement under investigation, reference and deformed grid patterns might look similar to those shown in Figure 2-3. By superimposing the two and generating an interferogram, analysis can be performed to extract the amount the grid has been rotated.

![Figure 2-3 Rotation Generated Interferogram](image)

Within moiré measurement methods, many techniques exist. Most can be described as variations from one of two basic setups, depending on the measurement under investigation. Defined in broad terms, the field consists of two divisions: in-plane and out-of-plane measurements. An example of in-plane displacement measurement is High Sensitivity Moiré Interferometry (HSMI),\(^{31}\) where an extremely fine grating is manufactured onto the specimen. Because of this, deformations experienced by the object are identical to those experienced by the grating. A coherent light source is used to create a reference grating to produce the interference. Out-of-plane measurement methods are typically projection-based and include Shadow and Projection Moiré
methods. Concepts for projection-oriented systems are similar in nature and will be discussed in the following section.

2.4 PROJECTION MOIRÉ INTERFEROMETRY

Projection Moiré Interferometry (PMI) is an out-of-plane displacement measurement technique, which consists of differencing reference and deformed images of a grid pattern projected on the test object. A typical PMI configuration is shown in Figure 2-4, where a light source is projected through a grid pattern and focusing lens onto the test structure. The out-of-plane displacements under investigation are along the Z-axis of the structure coordinate system. Images are then captured of the projected grid on the object in a reference and deformed state. This is typically accomplished by means of a CCD camera. These images, when differenced, produce an interferogram, from which the out-of-plane displacements can be extracted by computing the FSC values. This is typically done by giving the structure a series of known displacements and analyzing the fringe patterns at each position. This can then be applied for unknown displacements. Through the years, PMI has proved a useful tool in numerous applications, such as wind tunnel tests of large-scale structures\textsuperscript{32,33} and surface contour measurements.\textsuperscript{34} Therefore, it is worthwhile to investigate possible methods to improve the overall process of PMI.
Typical PMI procedures consist of six basic steps as shown in Figure 2-5. The first step is to establish a reference plane from which a normal vector represents the out-of-plane displacements under investigation. The second step consists of removing camera perspective from all images. In the third step a subsequent image, referred to as the “reference image”, is captured of the projected grid on a flat surface placed in the reference plane. The fourth step requires that this flat surface be given a number of known rotations or translations and images captured of the projected grid on the reference plane at each new position. In this work, these images are termed “calibration images,” and are differenced from the reference image to generate fringe patterns. The fifth step consists of analyzing the calibration fringe patterns. With knowledge of the displacement field for each calibration image, the FSC values can then be determined through a
calibration process. The sixth and final step is to capture an image of the projected grid on a deformed object, where the displacement field is unknown. This image is differenced from the reference image to generate a deformed interferogram. By applying the FSC values computed in step five to the deformed fringe pattern obtained, the displacement field of the deformed object can be computed. Details pertaining to methods used at BYU are now explained as well as the limitations imposed by PMI on certain types of applications.

Figure 2-5  Typical PMI Procedures

2.4.1 PMI Methods at BYU

The PMI methods currently available at BYU follow the steps outlined in Figure 2-5. The reference plane is first established, from which the displacement direction and magnitude will be computed. The camera perspective is removed for all images using a dot card placed in the reference plane. The dots on these cards are equally spaced in both the horizontal and vertical directions and the required image transformations are
computed, which alter the image to appear as if the camera is oriented perpendicularly to the reference plane. Images of a sample dot card both before and after applying the image dewarping algorithm are shown in Figures 2-6 a) and b). Only the region of the image covered by the dot card is dewarped. Therefore, the image is also cropped in the process.

Next, both reference and calibration images are captured in order to compute FSC values. To illustrate this process, sample reference and calibration images are shown in Figures 2-7 a) and b), where a rotation about a vertical axis on the left hand side of the image was used for the known displacement of the calibration image. This is performed for multiple rotations, but for illustration purposes, only one will be shown and explained. Rigid body rotations are used in order to force zero displacements at a specified location (center of rotation). Each calibration image is differenced from the reference image to generate a fringe pattern. This image usually contains high frequency noise, which is filtered before further analysis. Sample patterns are shown in Figures 2-8 a) and b) for the original filtered images.
In accordance with analysis methods described in Section 2.1.3, the filtered fringe pattern is phase shifted through four steps ($0^\circ$, $90^\circ$, $180^\circ$, and $270^\circ$), representing no phase shift ($0^\circ$), quarter cycle shift ($90^\circ$), half cycle shift ($180^\circ$), and three quarter cycle shift ($270^\circ$). Using Eq. (2.9), the intensity at a given pixel for each of the four phase steps is described by the notation shown in Eq. (2.10), where the numbered subscripts refer to the consecutive phase steps. These four equations, when solved simultaneously for the phase ($\phi$) yields the results shown in Eq. (2.11). The phase at each pixel is then calculated using the intensities from each of the four phase-stepped fringe patterns. An
interpolation-based phase-shift technique that phase shifts the reference image is used to generate the four fringe patterns. This uses numerous cycles (vertical lines in the reference image) to determine the number of pixels in a cycle, which is then used to calculate the shift in pixels needed for each phase shifted image. When a calibration image is differenced from the reference image at each phase step, the result is a phase-stepped calibration image. Samples of the phase stepped fringe pattern are shown in Figures 2-9 a) – d). The pixel intensities for these four images are used in conjunction with Eq. (2.11) to determine the phase. The \( \tan^{-1} \) function embedded within Eq. (2.11) yields values ranging from \(-\pi/2\) to \(\pi/2\), resulting in a wrapped phase map. This is then unwrapped using a simple linear unwrapping algorithm on each row of the image to provide whole-field phase values. Sample images of wrapped and unwrapped phase maps are shown in Figures 2-10 a) and b).

\[
I_1 = a + b \cos(\phi + \beta_1) \\
I_2 = a + b \cos(\phi + \beta_2) \\
I_3 = a + b \cos(\phi + \beta_3) \\
I_4 = a + b \cos(\phi + \beta_4) \\
\phi = \tan^{-1} \left[ \frac{I_4 - I_2}{I_1 - I_3} \right]
\]  
\text{(2.10)} \quad \text{(2.11)}
The next step is to determine the FSC values, which are described in Eq. (2.12) for a given pixel \((i, j)\) as the ratio of change in phase \((\Delta phase)\) to displacement \((D)\). The
change in phase ($\Delta \text{phase}$) is computed as the difference between the phase at a given pixel and that at a pixel of known displacement. For a calibration image, the known displacement and computed phase at each pixel are used to determine FSC values for every pixel. Methods employed in this research make use of a least squares algorithm to fit a quadratic function to each horizontal row of the image according to Eq. (2.13), where $x$ is the horizontal distance (units of length) from the zero displacement to the pixel of interest.

$$FSC(i, j) = \frac{\Delta \text{phase}(i, j)}{D(i, j)} \quad (2.12)$$

$$FSC(x) = ax^2 + bx + c \quad (2.13)$$

The coefficients $a$, $b$, and $c$ are found from the least squares algorithm, and differ from row-to-row. The FSC data therefore consists of a matrix three columns wide representing the three coefficients to the quadratic polynomial and as many rows as the image. Due to image noise and other uncontrollable variables, these FSC values slightly change from one calibration image to the next, and the average for all calibration images is used as the FSC values for that particular system.

An unknown displacement can then be given to the specimen, and the captured image differenced and phase-stepped in a similar manner as outlined for the calibration images to ultimately generate an unwrapped phase map for the displacement field. The average FSC values for all calibration images can then be applied to the displaced unwrapped phase map to compute full-field displacements.
2.4.2 Limitations Addressed

This thesis addresses two limitations of the technique just described. The first is the requirement to perform calibration, which requires rotating or translating a flat reference surface to known amounts. For large-scale applications, it is not realistic or cost-effective to manufacture a large plate covering the entire measurement domain. In addition, it is extremely difficult to produce a known displacement field within a certain tolerance limit for such a large object.

The second limitation of interest is the exclusion of PMI in applications with inaccessible reference surfaces. Since PMI requires the same object to be used for both the reference and deformed surfaces, the displacements cannot be calculated between an actual object and a digital version of the object. For example, when comparing the surface of a molded object to its theoretical counterpart described by the internal surface of the mold, PMI is currently not an option.

The research performed accounts for each of these limitations through virtual calibration and inaccessible surface measurement procedures. Each of these procedures may be used in conjunction with the other or independently. A number of applications exist in the automobile, aerospace, and other manufacturing industries where the combined procedures would be of great benefit.
3 VIRTUAL CALIBRATION PROCEDURES

This chapter presents in detail the developed procedures for virtual calibration of a typical PMI setup. These methods can be used to replace the conventional methods of rotating an actual flat test object. In the process of explaining each step, the underlying fundamental principles are also presented for the reader unfamiliar with these concepts.

The chapter is outlined as follows:

- Overview
- Camera Calibration
- Projector Calibration
- Virtual PMI Setup
- Virtual Calibration Images

3.1 OVERVIEW

The goal of virtual calibration is to ultimately recreate the entire experimental setup for PMI within a computer simulation. All calculations needed to determine FSC values could then be performed independent of the actual PMI setup. The virtual calibration procedures consist of four basic steps as shown in Figure 3-1. Step 1 makes use of camera calibration techniques\(^{10-13}\) to determine the characteristic intrinsic and extrinsic parameters of the camera. Step 2 uses these results and similar camera calibration routines to perform a projector calibration. Both components are then
modeled as virtual components in step 3 to create a virtual PMI setup. It is within this virtual setup that step 4 is accomplished, namely capturing virtual calibration images. Each of these steps is now discussed in greater detail.

![Diagram showing the virtual PMI calibration procedures](image)

**Figure 3-1 Virtual PMI Calibration Procedures**

### 3.2 Camera Calibration

The first step of the process is to perform camera calibration. The purpose of this step is two-fold: 1) determine internal (intrinsic) parameters of the actual camera and 2) estimate the position and orientation (extrinsic) parameters of the camera relative to the
object in its reference position. Camera models typically account for five intrinsic parameters describing the internal properties, six extrinsic parameters representing the camera’s geometric degrees of freedom relative to an object, and five distortion coefficients describing both radial and tangential distortions. Extraction of the calibration parameters can be performed using many different techniques.\textsuperscript{10-13} Most are built upon the influential work of Tsai\textsuperscript{36} and are based on comparing points on an object to the corresponding points on the image captured by the camera. For the present study, Matlab-based camera calibration routines are used.\textsuperscript{37} A description of each of the calibration parameters follows, as well as details of the extraction methods employed.

### 3.2.1 Intrinsic Parameters

The intrinsic transformation is described by five parameters, or degrees of freedom (DOF). They are the pixel focal lengths in two dimensions ($f_x$ and $f_y$), the image center in two dimensions ($c_x$ and $c_y$) and the pixel skew angle ($\alpha$). An illustration of these five parameters is shown in Figure 3-2. The pixel focal lengths are computed as the ratio of camera focal length ($f$) and pixel dimensions ($s_x$, $s_y$) in both the horizontal and vertical directions. Naturally, the pixel focal lengths are expressed in units of pixels. The image center is a way to describe the intersection point of the optical axis on the image plane, and can be used to quantify the misalignment of the camera. The two values needed to describe this point are also given in pixel units. It is worth noting that a set of intrinsic parameters does not offer a unique solution in that the magnitudes of both the precise distance from the camera lens to the image plane ($f$) and the size of each pixel ($s_x$, $s_y$) are only known in relationship to each other. The ratio describing this relationship is unique, but when determining either focal length or pixel size, one must be fixed in order to
calculate the other. However, it is also important to realize that an image captured using any of the infinite combinations will ultimately yield the same result as long as the ratios are used in the calculations. The intrinsic transformation ($IN$) including all parameters is found in Eq. (3.1) and converts points in the camera’s coordinate systems from units of length to units of pixels.

$$IN = \begin{bmatrix} f_x & \alpha f_y & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$  

(3.1)

![Illustration of Intrinsic Parameters](image)

**Figure 3-2** Illustration of Intrinsic Parameters

### 3.2.2 Extrinsic Parameters

The extrinsic transformation is explained by the standard homogenous transformation matrix,

$$\begin{bmatrix} f_x & \alpha f_y & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

(3.2), and consists of a 3x3 rotational matrix.
submatrix \((R)\) and a 3x1 translational vector \((T)\). The rotational submatrix is composed of three separate 3x3 rotational matrices as shown by Eq. (3.3), where \((\theta_x, \theta_y, \theta_z)\) represent the rotations about each of the three original coordinate axes. This describes the relative orientation from the original to the new coordinate system. The translational vector is shown in Eq. (3.4), and is composed of three translations \((t_x, t_y, t_z)\) along each of the original axes to describe the relative position from the original to the new coordinate system. This is further illustrated in Figure 3-3, where transformation from the original to the new coordinate system represents that from the part to the camera. The pixel array within the camera is represented as a plane parallel to the \(X_2-Y_2\) plane of the camera coordinate system.

Figure 3-3 Illustration of Extrinsic Transformation
With the homogeneous transformation matrix arranged in this way, vectors describing three-dimensional points in one coordinate system can be transformed into vectors describing the same point in space, but with coordinates from a separate system. A vector must first be expanded from three to four elements, which enables matrix multiplication. This is done by inserting a fourth element equal to one. Post-multiplication of a 4x1 vector of the form \( \begin{bmatrix} w_1 \\ x_2 \\ y_2 \\ z_2 \end{bmatrix} \), yields a solution of the form \( \begin{bmatrix} w_1 \\ x_1 \\ y_1 \\ z_1 \end{bmatrix} \) where \( w_1 \) and \( w_2 \) represent vectors in the original and new coordinate systems respectively. Within camera calibration, these six parameters represent the six degrees of freedom from the camera coordinate system to that of the object.

### 3.2.3 Distortion Coefficients

To account for distortion, a point on the object described by the vector \( \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}^T \) is pre-multiplied by the extrinsic transformation and a conversion matrix from four to three degrees of freedom according to Eq. (3.5). This is then normalized \((z_n = 1)\) yielding the image projection point \((P_n)\), which describes in units of length the ideal location for a ray traveling from the object to the image plane with an assumed focal
length of unity. Distortion changes the image plane-ray intercept point from an ideal location to a distorted location. The magnitude of this change depends on the relative location of the intercept point compared to that of the image center. Both radial and tangential distortions are accounted for, which assumes a distortion profile developed by Brown.\footnote{39} This is often referred to as the “plumb-bob” model. A five element distortion vector \( D \) is applied to the normalized projection point according to Eq. (3.6) where \( r^2 = x_n^2 + y_n^2 \). The distorted image point \( (P_d) \) is then converted to a final distorted pixel coordinate \( (P_p) \) using the intrinsic transformation as shown in Eq. (3.7).

\[
P_n = \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}
\]

(3.5)

\[
P_d = \left(1 + D_1 r^2 + D_2 r^4 + D_3 r^6\right) \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} 2D_4 x_n y_n + D_5 (r^2 + 2x_n^2) \\ D_6 (r^2 + 2y_n^2) + 2D_3 x_n y_n \end{bmatrix}
\]

(3.6)

\[
P_p = \begin{bmatrix} f_x & \alpha \cdot f_y & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_d \\ 1 \end{bmatrix}
\]

(3.7)

Distortion can be a difficult concept to visualize. Common occurrences of extreme radial distortion, such as a fish-eye lens or peephole, assist the unfamiliar reader in recognizing those effects. Tangential distortion, however, is slightly more difficult to envision. This occurs due to a de-centering of optical components within the system of lenses. Examples of each type of distortion, and the combined distortion effects are shown in Figures 3-4 a) – c), describing radial, tangential, and combined distortion.
models respectively. The pixel shift due to distortion is represented by an arrow in each case, and for illustration purposes, the contours shown are given in pixel units as well.

Figure 3-4 Illustration of Distortion Models: a) Radial, b) Tangential, and c) Combined

3.2.4 Extraction of Calibration Parameters

Equation (3.8) illustrates the general relationship between a three-dimensional point \((X,Y,Z)\) on an object in its own coordinate system to a two-dimensional pixel coordinate \((U,V)\) on the image plane of the camera. Most camera calibration techniques
map points on an object to corresponding points in the image and use various optimization routines to solve for the calibration parameters.

\[
\begin{bmatrix}
U \\
V \\
1
\end{bmatrix} =
\begin{bmatrix}
f_x & \alpha \cdot f_y & c_x \\
0 & f_y & c_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
R_{[3x3]} \\
T_{[3x1]}
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

It is worth noting that the intrinsic parameters and distortion coefficients are independent properties of the camera and remain constant for any captured image. However, the intrinsic calculations assume the distance from lens to image plane remains fixed, which is not the case if the focus is adjusted or a different zoom configuration is used. Therefore, care must be taken when calibrating the camera, to ensure that this distance remains fixed. The extrinsic properties change for each unique image captured by the camera, but the use of multiple images assists in their estimation. In the camera calibration routines used for this work, multiple images are captured of a planar checkerboard pattern. Every checker corner represents a point on the object that can be used in conjunction with the corresponding corner found in the image. This concept is illustrated in Figure 3-5 where a point on the checkerboard plane is mapped to the corresponding point in the image. Since the checker dimensions are known, the corner location of each checker in the object coordinate system is also known. In addition, for each image captured, the corresponding pixel coordinates for those corners are located using a corner extracting algorithm on the image. The task then becomes estimating the intrinsic, extrinsic, and distortion parameters. Optimization routines based on an iterative gradient descent with an explicit computation of the Jacobian matrix are used to
perform this task. The two purposes of camera calibration, as it relates to virtual calibration of the PMI setup, can then be accomplished. The first purpose is to determine the intrinsic parameters, including distortion coefficients for the camera and the second is to estimate the extrinsic parameters of the camera relative to the reference plane established in the PMI setup. These results are used to both calibrate the projector and to model a virtual camera used in a simulation of the PMI setup.

![Planar Checkerboard Pattern](image)

**Figure 3-5** Camera Calibration Using Planar Checkerboard Pattern

### 3.3 PROJECTOR CALIBRATION

The second step in the virtual calibration process is to determine the characteristic parameters of the projector. The same model used for the camera can be adapted to model the projector. The projector’s five intrinsic parameters, five distortion coefficients, and six extrinsic parameters can be estimated using the techniques described in the
previous section. In this step, however, a projected checkerboard pattern is used in place of the planar checkerboard used in camera calibration. With the camera and reference plane remaining in fixed positions, the projector is then placed in numerous positions and images are captured of the projected checkerboard on the PMI reference plane. The projector calibration process is illustrated in Figure 3-6, where the projected checker corners on the reference plane are first extracted as pixel coordinates \((U_1, V_1)\) in the camera coordinate system. The pixel locations are transformed into three dimensional points on the object \((X, Y, Z)\) using the intrinsic, distortion, and extrinsic parameters of the camera determined during the camera calibration step. The object coordinates are then used with the corresponding projector pixel coordinates \((U_2, V_2)\) of the image used for projection. The projector pixel coordinates, similar to the object coordinates for camera calibration, are known values. The projector parameters can then be estimated with the same optimization routines used for camera calibration. It should be noted that it is imperative for the camera to first be properly calibrated in order to perform projector calibration. This is needed to transform the camera pixel coordinates of the projected checkerboard to object coordinates in the reference plane.
3.4 VIRTUAL PMI SETUP

After calibration of the actual camera and projector is complete, the next step is to create a virtual PMI setup. This step is accomplished by creating two virtual components representing the camera and projector. Using the intrinsic parameters for both the camera and projector, and the extrinsic parameters relating each component to the reference plane, this can be done in a computer simulation. In this work, the ray tracing and modeling software Zemax\(^{42}\) was chosen to accomplish the tasks associated with this step. It is recognized that various solutions and software packages exist which could perform similar tasks and assist in creating a simulated PMI setup. Zemax was ultimately chosen for this research due to its reliable ray tracing engine, availability of several key user defined functions, and relatively inexpensive cost. A complete tutorial on performing
these tasks using this software is available in the Appendix. The computer simulation of a PMI system is shown in Figure 3-7, where the rays from a virtual projector, represented by a point light source, travel through a slide containing the projected grid or checkerboard pattern. The rays then strike the test structure, scatter to a single point representing the camera location, and are finally collected by the image plane of the virtual camera. The two vital components here are the virtual camera and projector, and methods used for modeling are described in the next sections.

![Virtual PMI Setup](image)

**Figure 3-7 Virtual PMI Setup**

### 3.4.1 Virtual Camera Modeling

The virtual camera is modeled as a pinhole camera, where all incoming light travels through an infinitesimally small aperture and then intersects the image plane. Realistically speaking, a pinhole camera design is not a viable option for manufacturing actual cameras. An extremely small aperture would never let enough light through to
produce distinguishable features on the image plane. In fact, an infinitesimally small aperture theoretically rejects all incoming light rays. In an actual camera, incoming light strikes the surface of the lens at many angles and positions. A good lens design refracts these various light rays to points on the image plane extremely close to those described by the pinhole model. Since all rays travel through the same point in a pinhole model, there is no need to refract the ray any further at the camera entrance. Therefore, the virtual camera is free of any distortion, and is designed to match all the intrinsic parameters of the actual camera, minus the distortion effects calculated in camera calibration.

The virtual camera is shown in Figure 3-8, where all incoming light is focused to a single point at the camera entrance and collected by the detecting plate (pixel array). In order to force all rays to travel through the same point, a user defined scatter function was written,\textsuperscript{43} which controls the scatter profile at the surface of the test structure. The ray-tracing engine is capable of using dynamic link libraries (DLLs) to direct a scattered ray. When a ray intercepts a point on the test structure, the local coordinates of that point are used to calculate a vector towards the camera location. The scatter function then outputs this vector as the direction of the scattered ray. This scatter code in its entirety can be found in the Appendix.
In creating the virtual camera, focal distance, size of the image plane, and the number of pixels are all variables defined by the user. One small limitation is that only rectangular pixels are allowed ($\alpha = 0^\circ$), but for most cameras, this is a safe assumption. Therefore, only four of the five intrinsic parameters can be modeled. The user defined variables must be chosen in conjunction with the pixel focal lengths and image center parameters given as intrinsic parameters of the camera. The extrinsic parameters are used to position and orient the virtual camera in reference to the reference plane.
3.4.2 Virtual Projector Modeling

The virtual projector is represented by a point light source, and like the virtual camera, contains no distortion. This is shown in Figure 3-9 where light rays travel through a slide containing the grid or checkerboard pattern toward the object. The slide is simply a bitmap image where size and aspect ratio are user defined variables. The optimized intrinsic parameters of the actual projector can then be used to assign values to the virtual projector to ensure equality between the two. The focal length and pixel size are treated similarly as was done with the virtual camera model. For the projected checkerboard, one value (focal length or pixel size) is chosen, thereby fixing the other at a value determined from the pixel focal length parameters.
3.5 **Virtual Calibration Images**

After modeling of virtual projector and camera is complete, they are placed in positions calculated based on the extrinsic parameters. The final task is to capture virtual calibration images. This process is shown in Figure 3-10 where the grid pattern is projected to the reference plane, and scattered toward the virtual camera. The reference plane is given a series of known displacement fields with the software, which in this case are rigid body rotations. Images are captured in the virtual PMI setup for each known displacement and differenced with the reference image to generate fringe patterns. These fringe patterns may then be analyzed using typical PMI methods.

![Figure 3-10 Capturing Virtual Calibration Images](image-url)
It is important to note that the projector calibration parameters were extracted using a checkerboard slide, whereas the calibration images require a grid pattern slide. Therefore, some knowledge regarding the relative sizes of the two slides is necessary. In the present study, an LCD projector is used to project the images. The size (units of length) of each of these images is chosen independent of projector location or size of the other slide. Care must be taken to ensure the length ratio of the two slides is kept constant from actual to virtual PMI setups. With this accomplished, the virtual projected pattern can be refracted toward the virtual camera and captured on the image plane. One way to judge the effectiveness of the entire setup is to project a pattern on the reference plane in both actual and virtual PMI setups, capture images, and then compare the two images. These two images, when differenced, should show no sign of interference. A phase difference might be present depending on the location of the slide center, but this discrepancy is accounted for in the PMI analysis.

In conclusion, with a properly calibrated camera and projector, the PMI calibration images can be captured solely within a computer simulation of the PMI setup. The checkerboard used for camera and projector calibration assume a planar surface, but the need to rotate or translate this large planar surface to known amounts is eliminated.
4 EXPERIMENTAL VALIDATION OF VIRTUAL CALIBRATION

In order to validate the methods developed for virtual calibration and obtain a better understanding of the sources and magnitudes of error, a flat plate experiment was performed. In this chapter, results from each step of virtual calibration are presented followed by a brief discussion and interpretation of results. Using techniques currently available at Brigham Young University, the FSC values are calculated both in the actual and virtual setups. Investigations are made to determine differences and the overall displacement error introduced from the process.

4.1 EXPERIMENTAL SETUP

The experimental PMI setup is shown in Figure 4-1. This consists of an LCD digital projector (Epson model 7700p) with a native display format of 1024 x 768 pixels for the light source and focusing lens. The grid pattern used was a bitmap image (640 x 480 pixels) with 0.25 lines/pixel. The CCD camera (Hitachi, model KP-M1U) with 640 x 480 pixel resolution was controlled by a computer, which had a frame grabber to digitize the pixel-sensed voltage to an 8-bit intensity pattern. A flat aluminum plate (15 cm x 15 cm) was used as the reference plane to measure displacements and was mounted on a rotary turntable with 1/60° resolution.
4.2 

**Camera Calibration**

Camera calibration was performed using a 25 x 17 checkerboard pattern of 1 cm square checkers as shown in Figure 4-2. The camera calibration software calculates the optimal calibration parameters based on numerous images. These parameters explain the extracted corners with the smallest possible error. The error is computed for each corner by differencing the pixel locations for the extracted corners and those determined through re-projection, or assuming the calibration parameters to be perfect. This re-projection error serves as a gauge to assist the user in identifying any outlying data, or poorly extracted corners. To illustrate this process, a sample plot of the pixel error is shown in Figure 4-3, where possible irregular data is labeled. Notice the vast majority of re-
projection error lies near zero in both the horizontal and vertical directions. Because data from a single image is color-coded, the blue points in the upper left corner and the black points on the left side suggest that only a few corners for each of those two images were poorly extracted. However, the groups of green and magenta data points suggest most corners from both images were poorly extracted. For the previous case, another corner extraction with a larger window enclosing each corner might decrease those errors, while the latter case implies that image removal during calculation would decrease the overall pixel error. It is also important to note that the corner extracting algorithm is accurate to about 0.1 pixels, and this introduces errors as well.

Figure 4-2 Planar Checkerboard Used for Camera Calibration
Initially, 30 images were captured and used to perform camera calibration. For reference, these images are included in the Appendix. The steps taken to obtain results during the camera calibration process are shown in Table 4-1. First, two images were found to have irregular data and were excluded. After another optimization, two separate images were found to have similar irregular data. Next, four of the five optimal distortion coefficients and their corresponding uncertainties were extremely small, so they were set to zero and only one coefficient was estimated during optimization. Then, the window size used as part of the corner extracting algorithm was increased in order to use more data to determine the pixel locations of the corners. After this, irregular data was again noticed in a single image, and it was excluded. The final steps consist of increasing the corner extracting window until the pixel error reached minimum. The results were again optimized to obtain the intrinsic camera parameters and distortion
coefficients shown in Table 4-2. The skew angle ($\alpha$) was set to zero to be consistent with the assumption of using square pixels. To investigate the soundness of this assumption, a separate optimization was run which included estimation of the skew angle ($\alpha$). This yielded a result of 0.0003 rad. (0.017°), and the assumption was considered validated. After removing the distortion effects from the image of the checkerboard in the reference plane, the intrinsic parameters, minus distortion effects, were then used to find the six extrinsic parameters also shown in Table 4-2 describing transformation from the coordinate system of the reference plane to that of the camera. It is worthwhile to note that the percent change in optimal parameters during the process described in Table 4-1 was extremely small (< 0.4%). This suggests the results obtained are very satisfactory.

### Table 4-1 Obtaining Camera Calibration Results

<table>
<thead>
<tr>
<th>Step Number and Description</th>
<th>Pixel Error (x)</th>
<th>Pixel Error (y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 All 30 images</td>
<td>0.19744</td>
<td>0.11072</td>
</tr>
<tr>
<td>2 Exclude images 15 and 30</td>
<td>0.19002</td>
<td>0.10296</td>
</tr>
<tr>
<td>3 Exclude images 5 and 11</td>
<td>0.18215</td>
<td>0.10228</td>
</tr>
<tr>
<td>4 Estimate only 1 distortion coefficient</td>
<td>0.18251</td>
<td>0.10211</td>
</tr>
<tr>
<td>5 Change window size from 5 to 6 pixels</td>
<td>0.18195</td>
<td>0.09931</td>
</tr>
<tr>
<td>6 Exclude image 4</td>
<td>0.17861</td>
<td>0.09919</td>
</tr>
<tr>
<td>7 Change window size to 7 pixels</td>
<td>0.17825</td>
<td>0.09774</td>
</tr>
<tr>
<td>8 Change window size to 8 pixels</td>
<td>0.17777</td>
<td>0.09703</td>
</tr>
<tr>
<td>9 Change window size to 9 pixels</td>
<td>0.17763</td>
<td>0.09657</td>
</tr>
</tbody>
</table>

### Table 4-2 Final Camera Calibration Results

<table>
<thead>
<tr>
<th>Intrinsic Parameters</th>
<th>Distortion Coefficients</th>
<th>Extrinsic Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_x$</td>
<td>1977.318 ± 3.606</td>
<td>$\theta_x$</td>
</tr>
<tr>
<td>$f_y$</td>
<td>1963.994 ± 3.563</td>
<td>$\theta_y$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0°</td>
<td>$\theta_z$</td>
</tr>
<tr>
<td>$c_x$</td>
<td>316.713 ± 3.692</td>
<td>$t_x$</td>
</tr>
<tr>
<td>$c_y$</td>
<td>233.492 ± 2.780</td>
<td>$t_y$</td>
</tr>
<tr>
<td>$D_1$</td>
<td>-0.3507 ± .0050</td>
<td>$t_z$</td>
</tr>
<tr>
<td>$D_2$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$D_3$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$D_4$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$D_5$</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

* Not estimated due to square pixels of camera
4.3 PROJECTOR CALIBRATION

Projector calibration was performed using the bitmap checkerboard image shown in Figure 4-4 with an image resolution of 850 x 850. It consisted of 17 checkers in both the horizontal and vertical directions of 50 x 50 pixels.

![Bitmap Checkerboard Image Used for Projection](image)

This pattern was projected on the PMI reference plane from various projector locations, and thirty images were captured of the projected checkerboard on the reference plane as seen by the camera. During this process, the camera and reference plane were kept in fixed positions described by the extrinsic parameters computed during camera calibration. The camera calibration results were then used to determine locations on the reference plane of the projected checker corners. This was used in conjunction with the
prior knowledge that each checker before projection is 50 x 50 pixels, and results were
optimized to find the intrinsic parameters and distortion coefficients. A similar process
employed for camera calibration was used for projector calibration as well, and each step
of the process is shown in Table 4-3. Initially, all images were used to optimize results.
Next, four of the five distortion coefficients were set to zero due to extremely small
optimal values and uncertainties. After optimization, the final distortion coefficient was
found to show similar results, and was likewise set to zero. The aspect ratio was then
forced to be unity. In other words, the pixel focal lengths were forced to be equal, which
was done because of knowledge regarding the actual image used for projection. Lastly,
two images were discarded due to irregular data. The optimal intrinsic parameters of the
projector as well as the extrinsic degrees of freedom from part to projector are shown in
Table 4-4.

Table 4-3 Obtaining Projector Calibration Results

<table>
<thead>
<tr>
<th>Step Number and Description</th>
<th>Pixel Error (x)</th>
<th>Pixel Error (y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1   All 30 images</td>
<td>0.14481</td>
<td>0.14760</td>
</tr>
<tr>
<td>2   Estimate only 1 distortion coefficient</td>
<td>0.14638</td>
<td>0.14763</td>
</tr>
<tr>
<td>3   Estimate no distortion coefficients</td>
<td>0.14795</td>
<td>0.14870</td>
</tr>
<tr>
<td>4   Force aspect ratio = 1</td>
<td>0.14804</td>
<td>0.14870</td>
</tr>
<tr>
<td>5   Exclude images 19 and 20</td>
<td>0.13712</td>
<td>0.13834</td>
</tr>
</tbody>
</table>

Table 4-4 Final Projector Calibration Results

<table>
<thead>
<tr>
<th>Intrinsic Parameters</th>
<th>Distortion Coefficients</th>
<th>Extrinsic Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_x$</td>
<td>2843.982 ± 25.520</td>
<td>$\theta_x$ 178.784°</td>
</tr>
<tr>
<td>$f_y$</td>
<td>2843.982 ± 24.520</td>
<td>$\theta_y$ 22.967°</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0°</td>
<td>$\theta_z$ 0.605°</td>
</tr>
<tr>
<td>$c_x$</td>
<td>439.286 ± 8.533</td>
<td>$t_x$ -215.935 mm</td>
</tr>
<tr>
<td>$c_y$</td>
<td>897.870 ± 3.855</td>
<td>$t_y$ -17.869 mm</td>
</tr>
<tr>
<td>$c_z$</td>
<td></td>
<td>$t_z$ 654.192 mm</td>
</tr>
</tbody>
</table>

* Not estimated due to square pixels of image.
4.4 Simulated PMI Setup

Once the camera and projector were successfully calibrated, the virtual PMI setup was created. This requires modeling the camera and projector within a simulation. Results from both these components are presented, with investigations made to determine the degree of correlation from virtual to actual components.

4.4.1 Virtual Camera Model

The intrinsic parameters of the virtual camera were developed to match the optimized parameters of the actual camera. The focal length was chosen to be a fixed parameter (25 mm) and then used in conjunction with the pixel focal lengths of the actual camera to determine pixel size for the image plane of the virtual camera. The pixel size was calculated to be 12.64 µm x 12.73 µm to yield an image plane of 8.092 mm x 6.110 mm. The pixel length and height was then used to determine the image center in units of length. Using the top left corner of the image plane as a reference, the calculated values for the image center are 4.004 mm and 2.972 mm in the horizontal and vertical directions. Next, through matching the extrinsic values for the reference plane in both the actual and virtual PMI setups, a comparison was made between images of identical checkerboards placed in the reference plane. This serves as an estimate of how well the two setups correlate. The actual and virtual reference checkerboards are shown in Figures 4-5 a) and b). In Figures 4-6 a) and b), the extracted corners for each image are shown with a differenced image found by subtracting pixel values from the two images, which serves as an estimate of how well the two cameras correlate. The average absolute difference between the two images is 0.1362 pixels in the horizontal and 0.1126 pixels in the vertical directions. Further investigations were made to explore any apparent trends in
extracted corners that might suggest inadequacies in one or both camera models. For instance, if a parabolic trend was discovered in the pixel difference along the horizontal or vertical directions, it could be assumed that the distortion model of the actual camera needs improvement. If only a random pattern is observed, both models can be considered adequate. These plots are shown in Figures 4-7 a) and b) for the horizontal and vertical directions respectively. The horizontal data was computed for each row using column indices (1-13) describing the horizontal location in units of corner number. The vertical data was computed for each column in a similar fashion to the horizontal data, except the pixel difference for each column is plotted against the row indices (1-10). Both plots show only random trends, and the correlation was considered adequate. It should be noted that because the virtual camera does not include any distortion effects, they must first be removed from images captured with the actual camera in order to be compared to one captured by the virtual camera.

Figure 4-5   Planar Checkerboards in Reference Plane: a) Actual Undistorted and b) Virtual
4.4.2 Virtual Projector Model

The virtual projector model was likewise created using the intrinsic parameters of the actual projector. Again, the focal length was chosen to be a fixed parameter (100 mm) and used to determine the pixel size. This was calculated to be 35.16 µm x 35.16 µm yielding a total length of 29.888 mm x 29.888 mm for the checkerboard slide. The
image center was converted from pixel units to units of length using the pixel length calculated above. These results were 15.446 mm and 31.571 mm in the horizontal and vertical directions respectively measured from the top left corner of the image. After matching the extrinsic parameters of the virtual projector to the actual projector, images of the projected checkerboard pattern were captured on the reference plane as seen by the camera in both setups. These two images are shown in Figures 4-8 a) and b). The extracted corners for both images and the differenced image are shown in Figures 4-9 a) and b). The average absolute pixel error was calculated to be 0.1362 pixels in the horizontal and 0.1126 pixels in the vertical directions. As was done with the camera modeling, further investigations were made to determine trends within the pixel differences. The plots depicting these trends across each row and along each column are shown in Figures 4-10 a) and b). Only a random scatter of data is observed, which suggests the model for both the actual and virtual projectors is satisfactory.

Figure 4-8  Projected Checkerboards on Reference Plane: a) Actual Undistorted and b) Virtual
While capturing images of projected patterns within the experiment, the two bitmap images used for the grid and checkerboard pattern slides were different sizes (lengths). The checkerboard slide was made smaller than the grid slide to enable more checker corners to be projected on the reference plane as seen by the camera. This does not change the location of the projector, but it does mean that the two images
representing each slide cannot simply be interchanged within the simulation. The location of the slide remains fixed, but the size must be altered. Because the checkerboard was used for calibration, the length calculated (29.888 mm) serves as a parameter to determine the size needed for any other slide. The size of each actual slide before projection was 27.94 cm for the grid pattern and 17.78 cm for the checkerboard pattern. The length ratio of these slides must be constant from the actual to virtual setups. For this experiment, the length of the virtual checkerboard slide was multiplied by the length ratio (27.94/17.78) to find the size needed for the virtual grid slide within the simulation. This assumes the focal length will be kept constant within the simulation. The calculated length for the virtual grid slide was 46.967 mm.

4.5 Virtual Calibration Images

To capture calibration images, the reference plane was given a series of known displacements in the virtual PMI setup. Each displacement was a rigid body rotation about the local y-axis of the reference plane. These were completed at 1° increments from 11° to 20°, and an image was captured at each calibration position. These rotations were selected because they represent displacements near those expected for the test case presented in Chapter 6, and it was predetermined that if possible, the experimental validations of virtual calibration would be used for the test case as well.

For visual comparison, the virtual reference and 15° rotated images are shown in Figures 4-11 a) and b). Images of the same displacement fields captured in the actual setup are shown in Figures 4-12 a) and b). Interference should not be noticeable when comparing actual and virtual calibration images. The differences (actual – virtual) for the reference and 15° rotated images are presented in Figures 4-13 a) and b). It is apparent
that the reference image shows no visible signs of interference. The $15^\circ$ rotated image, at first glance, appears to have a fraction of an interference fringe, but careful inspection proved this was attributed to the change in pixel intensity of the projected grid across the actual image. This was due to the projector becoming unfocused at the far end of the rotated plane, which is evident when comparing the actual reference and rotated images. The results here seem to look promising, but the actual test will be to analyze fringe patterns and generate FSC values in both setups and compare.

![Figure 4-11 Virtual Calibration Images: a) Reference and b) $15^\circ$ Rotation](image)

**Figure 4-11** Virtual Calibration Images: a) Reference and b) $15^\circ$ Rotation
4.6 RESULTS

Each calibration image, whether actual or virtual, was differenced from the reference image of the respective setup to generate fringe patterns. They were then used with standard PMI techniques discussed in Section 2.4.1 to determine the FSC values, and virtual results were compared to actual results. This assumes the FSC values found
from actual calibration images contain no error. This is not the case because they are determined experimentally, and will not generate a perfect displacement field. However, for purposes of validation, since identical procedures are used to determine both virtual and actual FSC values, this will provide further indication of the level of accuracy between virtual and actual setups.

The methods available at BYU make use of a four-step phase-shifting algorithm to create a wrapped phase map of each fringe pattern. This is then unwrapped yielding continuous phase information over the entire image. FSC values are determined for each pixel, given the computed phase and known displacement field. A quadratic polynomial is then fit to each row of the image relating horizontal position to FSC values, as shown in Eq. (4.1), where the coefficients \(a\), \(b\), and \(c\) are determined using a least squares approach and \(x\) is the distance in units of length along the reference plane. An array is then generated three columns wide with as many rows as the image, representing the three coefficients of the quadratic polynomial for each row. The FSC arrays generated from only virtual calibration images were then compared to those found using only actual calibration images. The coefficients were analyzed for each row of the image and discrepancies introduced from virtual calibration were calculated as a percent difference from those of the actual calibration. The average FSC values from both virtual and actual setups are presented in Table 4-5. In addition, the percent difference of the quadratic, linear, and constant term for each row are shown in Figures 4-14 a) – c). It is apparent from the plots that standard deviations increase dramatically from the quadratic to constant terms. These results suggest the quadratic coefficient is the most difficult to
match during the virtual calibration process, showing the largest average and standard deviation of error.

\[ FSC(x) = ax^2 + bx + c \]  

(4.1)

Table 4-5  Averages and Errors for Actual and Virtual FSC Values

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Actual FSC (ave.)</th>
<th>Virtual FSC (ave.)</th>
<th>Percent Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadratic ((a))</td>
<td>-0.0097</td>
<td>-0.0103</td>
<td>-6.37 %</td>
</tr>
<tr>
<td>Linear ((b))</td>
<td>0.2537</td>
<td>0.2569</td>
<td>-1.25 %</td>
</tr>
<tr>
<td>Constant ((c))</td>
<td>1.9618</td>
<td>1.8967</td>
<td>3.32 %</td>
</tr>
</tbody>
</table>

Figure 4-14  Percent Difference in FSC Coefficients: a) Quadratic, b) Linear, and c) Constant
For any unwrapped phase map, displacements at each pixel are generated according to Eq. (4.2) where $D(x)$ is the displacement, $\Delta\text{phase}(x)$ is the change in phase from any pixel to the pixel of zero displacement, and $FSC(x)$ is the FSC value at that pixel calculated from Eq. (4.1). Since the ultimate goal of the entire measurement method is to find displacements, investigations were made to estimate the effect of discrepancies in FSC values from actual to virtual calibrations.

$$D(x) = \frac{\Delta\text{phase}(x)}{FSC(x)}$$  \hspace{1cm} (4.2)

Using the same unwrapped phase map, the whole-field percent difference in displacement using the two sets of coefficients was calculated according to Eq. (4.3) where the subscripts $\text{act}$ and $\text{vir}$ represent data from actual and virtual PMI setups. Since the phase change for a pixel of the unwrapped phase map remains fixed, this simply becomes a percent difference of the inverse FSC values. This was calculated for every pixel with results shown in Figure 4-15. The average percent difference was -2.68% with a standard deviation of 0.65%. From the plot, it is apparent that the largest errors occur near the edge of zero displacement (far left side of image). Since displacements are small in this region, it becomes more difficult to estimate the FSC values and the results tend to be inflated.

$$\% \text{ Error} = \frac{D(x)_{\text{act}} - D(x)_{\text{vir}}}{D(x)_{\text{act}}} = \frac{\Delta\text{phase}(x)}{FSC(x)_{\text{act}}} - \frac{\Delta\text{phase}(x)}{FSC(x)_{\text{vir}}}$$  \hspace{1cm} (4.3)
The error was further analyzed for each row and column and the averages are shown in Figure 4-16 and Figure 4-17. The errors in the rows show no apparent trend, which suggests that displacement errors in the vertical direction vary randomly. However, a definite trend is evident for errors in the columns, and appears to be quadratic in nature. This is mainly attributed to the large percent error in the quadratic coefficient between actual and virtual FSC values. The quadratic coefficient proved to be the worst estimated of all three coefficients, and therefore had the largest effect on the error.
Figure 4-16  Average Displacement Difference for all Rows

Figure 4-17  Average Displacement Difference for all Columns
Further analysis was performed on experimental displacements of the flat plate, given a known displacement field. This served as an additional valuable link between actual and virtual PMI setups to estimate and characterize the errors introduced in the overall process. The virtual FSC values were used in conjunction with phase maps generated from images captured in the actual PMI setup to compute displacements. These were then compared to the true displacements, and results from five separate rigid body rotations (1° increments from 11° to 15°) are shown in Figure 4-18. The experimental plots shown were computed by averaging each column from the whole-field displacement in each case. The experimental data seems to correlate quite well with the actual data. The displacement errors (in cm) for each of the five rotations are shown in Figure 4-19, as well as the average error and the associated error bars. These were calculated using a sample size of five (Each rotation was treated as a separate data set) to calculate a 95% confidence interval for the average error for each column along the length of the beam. On average, the error is described with a magnitude of 0.0032 cm ± 0.0464 cm (based on a standard deviation of 0.0374 cm). In order to better visualize this error, it was applied to displacements from a 13° rigid body rotation. The result is shown in Figure 4-20 with error bars at discrete data points along the length of the beam.

In conclusion, the virtual calibration techniques developed have been shown to yield whole-field displacement results within 3% of their traditional technique counterparts. In addition, experimental displacements using virtual FSC values with actual unwrapped phase maps show a high degree of correlation to the actual displacements given (± 0.05 cm).
Figure 4-18  Rotated Plate Data

Figure 4-19  Average Displacement Errors with Error bars
Figure 4-20  Sample Displacement with Average Errors and Error bars
5 INACCESSIBLE SURFACE MEASUREMENT PROCEDURES

In order to extract displacement measurements from PMI, at least two images must be taken. One of these corresponds to a reference image of the grid pattern. Obtaining this image becomes difficult when the reference object is inaccessible for the camera and cannot be acquired through traditional PMI methods. One example of this occurs when a part is produced using a mold or die. The mold and the actual workpiece represent the reference and deformed surfaces respectively, but the dimensional inconsistencies cannot be measured using conventional projection measuring techniques because the surface of the mold cavity is inverted when compared to that of the part. The differences in this case could be attributed to the material properties of the workpiece, or residual stresses produced in the workpiece during the manufacturing process.

Many potential applications exist in the automobile, aerospace, and other manufacturing industries. Differences between the deformed and reference surfaces for these applications represent the manufacturing errors due to dimensional variations and assembly processes. The reference surface corresponds to a non-existent ideal surface, which cannot be manufactured, and therefore cannot be used with traditional PMI techniques. A CAD model of the non-existent or inaccessible surface can be used as a reference surface from which relative displacements are measured.
In this chapter, procedures are explained which incorporate inaccessible surfaces represented by CAD models and the information is outlined as follows:

- Overview
- Point Cloud Extraction
- Point Cloud Registration
- Displacement Generation

5.1 Overview

It is appropriate at this point to clarify terminology used in this chapter and throughout the remaining portion of the thesis. Displacements are traditionally defined as measurements describing the movement of a single object. This suggests the same object be considered when performing this type of measurement. An inaccessible surface and the actual part are indeed different objects, but in this thesis, the relative difference between the two surfaces is treated as a displacement, and is referred to as such. In addition, often times it is advantageous to express the contours from a curved surface, such as an airfoil or car door, relative to some datum plane. In essence, the nature of this type of measurement is quite similar to that describing the difference between a CAD model and actual part, namely expressing the measurement as a difference between two separate objects. For this research, these measurements are also considered displacements.

Once the FSC values are extracted from either standard or virtual calibration procedures, the displacement field can then be generated for any surface relative to the reference plane. In many instances, it would be extremely advantageous to directly compare this displacement field to that of an inaccessible surface. These surfaces could
be inverted, hidden, or simply only exist as a CAD model. Recent efforts\textsuperscript{44} have shown inaccessible reference surface measurements can be performed by inserting the CAD model into a simulated PMI setup and using it as the reference state for determining displacements through traditional PMI methods. Through properly calibrating actual and virtual cameras, a direct pixel comparison of displacements generated from the actual object can be compared to those generated from the CAD model within the virtual PMI setup. In a way, the explained process suggests experimentally determining the displacement field of a CAD model. However, the displacements can also be extracted directly from the CAD model thereby eliminating the need to gather experimental data. This procedure consists of three steps as outlined in Figure 5-1. The first step is to extract a point cloud from the CAD model, which is first converted into a common format known as STL. This point cloud represents discrete surface points of the CAD model. The second step requires changing the orientation of the point cloud in order to measure displacements along the same direction as the displacements in the PMI analysis. However, methods must first be employed to determine this direction and a plane from which the magnitudes will be measured. For the present study, these parameters are described by the reference plane established during the PMI analysis from which the displacement direction is perpendicular. The third step is to create a displacement field by applying a surface interpolation function\textsuperscript{45} to the point cloud to create a continuous surface of points. A prerequisite to this step is determining the geometric points on the reference plane represented by each pixel of the camera. In this research, results from camera calibration are used in this step, thereby generating a displacement field obtained directly from the CAD data, which can then be compared to
the displacement field found through a PMI analysis of the object. Each of these three steps will now be discussed in more detail.

Figure 5-1 Inaccessible Surface Measurement Procedures

5.2 **Point Cloud Extraction**

The first step in creating a displacement field from a CAD model is to convert it to a point cloud. For this research, the CAD model is first converted into stereolithography (STL) format. This format requires the object to be a three dimensional solid model and exports the surface as a collection of triangles, or facets. Each facet shares its vertices with surrounding facets to create a mesh covering the surface. To illustrate this, a unit triangle solid and the STL faceted mesh counterpart are shown in Figures 5-2 a) and b). Here the solid is represented as a mesh consisting of eight small triangles. For curved surfaces, the faceted model is an approximation.
A portion of a sample STL file is shown in Figure 5-3, which illustrates the format. Notice that the file begins by specifying a solid and finishes by closing the solid. Between these two callouts is a description of each triangle, or facet. Included in this description are three vertices and a unit normal vector. For the facet listed, the three vertices are (0 1 0), (0 0 0), and (0 1 1), with a unit normal vector in the direction of 
\[ \vec{n} = (-1\vec{x} + 0\vec{y} + 0\vec{z}) \]. Matlab code available in the Appendix was written to extract the vertices for each facet of the solid. The result is a three-dimensional point cloud describing the surface of the CAD model. The density of these points is automatically generated when converting to STL format and increases or decreases over the entire surface depending on the change in profile. This effect is illustrated in Figures 5-4 a) – c) where an A-arm is shown as a solid model, a triangle mesh, and a point cloud. Notice that the extracted point cloud is denser wherever more facets are needed to describe the surface.
Figure 5-3  Portion of Sample STL File

```plaintext
solid
  . . .
facet normal -1 0 0
  outer loop
    vertex 0 1 0
    vertex 0 0 0
    vertex 0 1 1
  endloop
endfacet
  . . .
endsolid
```

Figure 5-4  A-Arm:  a) Solid Model, b) Triangle Mesh, and c) Point Cloud
5.3 POINT CLOUD REGISTRATION

Point cloud registration requires first obtaining the direction and relative magnitude for displacements. In other words, a datum plane must be established in space that describes both the direction and magnitude of displacements. For the present study, a dewarping algorithm is used to remove camera perspective, and uses the reference plane established in the PMI analysis to determine displacement direction and magnitude. This dewarping process essentially transforms all images to appear as if the camera were lined up perpendicular to the reference plane. If camera perspective is not removed, the displacements will be measured along the optical axis of the camera from the reference plane.

The next step is to properly register the point cloud so that displacements will be measured from a common reference plane. When performing a PMI analysis on an object, a reference plane must first be established. A vector normal to this reference plane describes the direction of displacement measurement, so care must be taken to ensure the point cloud represents displacements from that same plane. Through rotating and/or translating the point cloud, this can easily be accomplished. The homogeneous transformation matrix \((HT)\) explained in section 3.2.2 is used to perform this task, and for clarification purposes is presented in Eqs. (5.1) – (5.3). This 4 x 4 matrix describes transformation from any one coordinate system to another. When post multiplied by a vector of the form \(w_2 = [x_2 \ y_2 \ z_2 \ 1]^T\), yields a solution of the form \(w_1 = [x_1 \ y_1 \ z_1 \ 1]^T\) where \(w_1\) and \(w_2\) represent vectors in the original and new coordinate systems respectively. The homogeneous transformation is determined from
the actual setup to the point cloud and multiplied by all point cloud data to convert it to a coordinate system identical to that of the actual setup.

\[
HT = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{T}_{3 \times 1} \\
0_{3 \times 3} & 1 \end{bmatrix}
\]

(5.1)

\[
\mathbf{R} = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \theta_x & -\sin \theta_x \\
0 & \sin \theta_x & \cos \theta_x
\end{bmatrix} \begin{bmatrix}
\cos \theta_y & 0 & \sin \theta_y \\
0 & 1 & 0 \\
-\sin \theta_y & 0 & \cos \theta_y
\end{bmatrix} = \begin{bmatrix}
\cos \theta_x \cos \theta_y & -\sin \theta_x & 0 \\
\sin \theta_x \cos \theta_y & \cos \theta_x & 0 \\
-\sin \theta_y & 0 & 1
\end{bmatrix}
\]

(5.2)

\[
\mathbf{T} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T
\]

(5.3)

### 5.4 Displacement Generation

To generate displacements, first the geometric location on the actual object represented with each camera pixel must be determined. Since camera perspective can be removed from all images, only the pixels per unit length and height ratios need to be determined, which can be done with a variety of techniques. Without dewarping techniques, these ratios vary across the image, but this task may also be accomplished through camera calibration procedures discussed in section 3.2. Because all properties of the camera have been optimized, each pixel coordinate of the entire pixel array can be used to determine the corresponding location on the object. This creates a mesh of data points where displacements may be generated, and correspond to camera pixels. Since the pixels do not correspond to the point cloud, a surface approximation can be applied to the point cloud to create a surface of points with identical spacing on the point cloud as the pixel array has on the actual object. This surface of points is then directly compared...
to the displacement field generated from an object using conventional PMI methods and the differences can be determined between an inaccessible surface and an actual object.

In conclusion, applications where PMI techniques have not traditionally been used due to inaccessible reference surfaces could now benefit from a whole-field projection measuring system. This would greatly increase inspection efficiency of large production parts and final assemblies.
6 TEST CASE – AIRFOIL

A specific application, where both virtual calibration and inaccessible surface measurement procedures would be advantageous, is found in the airframe industry. The leading edge of a Boeing airfoil donated to BYU was used as a test case. For this application, it would be beneficial to determine the magnitude of the dimensional and surface variations relative to a non-existent “perfect assembly.” Results from virtual calibration and each step of the inaccessible surface measurement procedure are presented followed by a discussion and interpretation of results.

6.1 EXPERIMENTAL SETUP

The test object was the leading edge of the Boeing airfoil, donated to the ADCATS research group at Brigham Young University. The unassembled components include a pre-bent piece of sheet metal and two ribs as shown in Figures 6-1 a) and b). During the assembly process, the airfoil is pressed into a fixture and riveted to an elastic frame. The fixture and assembled airfoil are shown in Figures 6-2 a) and b). When the assembled airfoil is removed from the fixture, it experiences some springback and does not maintain the exact shape of the fixture. Therefore, discrepancies exist between the internal surface of the fixture and the external surface of the assembled airfoil. The goal of this work was to determine the shape differences between the airfoil fixture and the assembled airfoil. These differences were measured by comparing the displacement
fields of the assembled airfoil and a CAD model representing the internal surface of the fixture. Virtual calibration techniques were used on the assembled airfoil and inaccessible surface measurements were made on the CAD model. This experimental displacement field was then compared to displacements obtained using a coordinate measuring machine (CMM) with an accuracy of ± 5 µm. Scans were performed by the ADCATS research group with the CMM on both the assembled airfoil and internal surface of the fixture, and then differenced to acquire the displacement field under investigation.

![Figure 6-1 Unassembled Airfoil: a) Pre-bent Sheet Metal and b) Ribs](image)

**Figure 6-1** Unassembled Airfoil: a) Pre-bent Sheet Metal and b) Ribs
The airfoil was approximately 30.33 cm long and 13.00 cm wide and the images used for testing represented a small portion of the airfoil (approximately 10 cm by 9 cm) as shown in Figure 6-3. A summary of the experimental setup with equipment specifications is found in Section 4.1. For this experiment, the airfoil was inserted into the experimental setup as shown in Figure 6-4 where parallelism was forced between the flat bottom edge of the airfoil and the reference plane. The rotational center of the turntable was used as the origin with displacements measured perpendicular to the reference plane, along the z-axis shown in the illustration. The offset shown was added to the entire displacement field in order to represent a surface topology of the assembled airfoil. Virtual calibration results as well as those from each step of the inaccessible surface measurement procedure are presented in the following section.
Figure 6-3  Measurement Window of Airfoil

Figure 6-4  Airfoil Insertion Point in PMI Setup
6.2 VIRTUAL CALIBRATION

The results (FSC values) presented in Chapter 4 from the virtual calibration procedures were used for displacement extraction in this test case as well. These results were shown to introduce an average displacement error of less than 3.0% for a given unwrapped phase map when compared to FSC values generated using conventional methods (see page 63). The virtual FSC values were eventually applied to the fringe pattern generated from differencing the projected grid patterns on the reference plane and assembled airfoil. This was done in order to prove that virtual calibration procedures could be used in conjunction with inaccessible surface measurement procedures in applications where they are found to be complimentary.

6.3 PMI DATA GATHERING

To extract displacements, the projected grid images from the reference plane and assembled airfoil were differenced to generate a fringe pattern. The original images captured are shown in Figures 6-5 a) and b). A region of interest algorithm was applied to the airfoil image to detect the pixels within the measurement domain and reject all others by forcing the value to be zero (pure black). These images were then dewarped using the planar dot card shown in Figure 6-6 to remove camera perspective, and cropped to a size of 554 x 417 pixels as show in Figures 6-7 a) and b). A fringe pattern was generated from differencing these two images and then phase-stepped using a four-step phase-shifting technique on the reference image. Fringe patterns were also generated at each step (0°, 90°, 180°, and 270°) and then filtered as shown in Figures 6-8 a) – c). The pixel intensities for the four images were used in conjunction with Eq. (6.1) to generate a wrapped phase map of the entire measurement domain. This is shown in Figure 6-9
where values cycle from \(-\pi/2\) to \(\pi/2\) due to the \(\tan^{-1}\) function within the equation. This image was then unwrapped to provide continuous phase information across the region of interest as shown in Figure 6-10.

\[
\phi = \tan^{-1}\left( \frac{I_{270} - I_{90}}{I_0 - I_{180}} \right) 
\]  

(6.1)

Figure 6-5  Projected Grid Images: a) Reference Plane and b) Airfoil

Figure 6-6  Planar Dot Card Image Used for Dewarping
Figure 6-7  Dewarped Cropped Images (554 x 417 pixels):  a) Reference Plane and b) Airfoil

Figure 6-8  Filtered Phase Shifted Fringe Patterns:  a) 0°, b) 90°, c) 180°, and d) 270°
Figure 6-9  Wrapped Phase Map of Airfoil

Figure 6-10  Unwrapped Phase Map of Airfoil
For the region of interest, the unwrapped phase map was then used with the virtual FSC values to obtain whole-field displacement measurements $D(i, j)$ according to Eq. (6.2), where $i$ and $j$ represent pixel locations in the horizontal and vertical directions, and $j_{zero}$ is the column number of the zero displacement pixel in the region of interest for any particular row. A two and three-dimensional view of the experimental displacement field is shown in Figure 6-11 and Figure 6-12. The general shape of the airfoil is also shown in order to better visualize displacements. The maximum displacement of 5.440 cm is equal to the offset that has been added to each value. The maximum displacement measured by the PMI analysis, then is the range shown on the image (max – min), or 1.669 cm.

$$D(i, j) = \frac{\text{phase}(i, j) - \text{phase}(i, j_{\text{zero}})}{\text{FSC}(i, j)}$$  \hspace{1cm} (6.2)

![Figure 6-11 2D View of PMI Displacement Field for Assembled Airfoil](image)
These results were compared to the scans performed by the coordinate measuring machine (CMM) on the assembled airfoil. A displacement field for the same region of interest was generated strictly from CMM data and differenced from the PMI displacement field. The percent difference is shown in Figure 6-13, which varies throughout the image from -0.0455 cm to 0.0448 cm. The band at the far left side of the image represents a section where scans were not performed by the CMM, and cannot be compared to the PMI displacements. To better gauge the accuracy, percent difference was computed using the displacements, and is shown in Figure 6-14. It is important to note that displacements from the PMI analysis equal zero on the far left side of the image and reach a maximum at the far right side of the image. Percent differences near the edge of zero displacement were not calculated because even for small differences, the percent difference tends to be extremely large. Values in this region did not seem to reflect the
true accuracy measured, and were masked out of the image. The percent differences shown in Figure 6-14 vary from -11.22 % to 5.00 %, and appear to be somewhat constant for any particular column. The average percent difference for each column is shown in Figure 6-15. The large differences near the edge of zero displacement can easily be recognized. Inside of that region, it suggests that the PMI analysis overestimated the displacements for some distance until finally leveling off to a certain degree.

Figure 6-13  Experimental Displacement Field vs CMM Data: Difference
Figure 6-14  Experimental Displacement Field vs CMM Data: Percent Difference

Figure 6-15  Average Percent Difference (PMI vs. CMM) Along Columns
6.4 POINT CLOUD EXTRACTION

Having applied the virtual calibration procedures to generate a displacement field, a similar displacement field for the CAD model of the internal surface of the assembly fixture was created. The solid CAD model and the extracted point cloud are shown in Figures 6-16 a) and b). The extracted point cloud was generated by converting the solid CAD model into STL format, which represents the surface as triangle shaped facets. The vertices were read from the file and plotted in space. For this object, 46 triangles were needed to describe the surface using 138 vertices. The vertices are shared between triangles, and therefore not all are unique geometric locations. The total number of unique points in the point cloud data is 48. However, since the form of the fixture (and airfoil for that matter) allows two different z-values for any x-y coordinate, the point cloud must be trimmed to prevent problems during surface fitting. The CAD model was trimmed to slightly larger than the viewing window in the PMI analysis. This is seen in Figures 6-17 a) and b), where the surface is trimmed at 15.00 cm and 11.15 along the two directions shown.

![Figure 6-16 Assembly Fixture: a) Solid Model and b) Point Cloud](image)

Figure 6-16  Assembly Fixture:  a) Solid Model and b) Point Cloud
6.5 **Point Cloud Registration**

The coordinate system of the point cloud was transformed to match that of the assembled airfoil. Visualization of the needed transformation is shown in Figure 6-18, which suggests that a simple rotation about the z-axis describes this transformation. The homogeneous transformation matrix from object to point cloud consisted of a single rotation about the z-axis ($\theta_z = -90^\circ$) with no other rotations or translations needed. This was then used to transform every coordinate of the point cloud according to Eq.(6.3), where $(X,Y,Z)$ represents point cloud data and $HT$ is the homogeneous transformation matrix. With this registration complete, the point cloud data is ready to be used in generating displacements.

$$
\begin{bmatrix}
\cos(\theta_z) & -\sin(\theta_z) & 0 & 0 \\
\sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1 \\
\end{bmatrix}
$$

(6.3)
6.6 DISPLACEMENT GENERATION

In order to generate displacements, the pixel representation on the object must first be determined. With current PMI techniques, the camera perspective is removed from each image so that the images appear as if the camera was lined up perpendicular to the reference plane. The result is constant pixel per unit length and height ratios for each row and column in the image. To determine this, the planar checkerboard used for camera calibration was placed in the reference plane, and an image was captured and dewarped with the same dot card used in the PMI analysis. The original (640 x 480 pixels) and dewarped (560 x 431 pixels) images are shown in Figures 6-19 a) and b). The dewarped image was analyzed to determine the pixel per length and height ratios by using a corner extracting algorithm to find the pixel locations of each checker corner in the image. Using this data with the geometric size of each checker, the average calculated ratios were 0.2413 mm/pixel and 0.2416 mm/pixel in the horizontal and
vertical directions respectively. These values were then used as inputs in the surface interpolation program, specifying points in two directions (along the x-axis and y-axis) in which to create displacements in the third direction (along z-axis). The resulting displacement field is shown in Figure 6-20, where each data point represents a single pixel. Since the CAD model was perfectly flat along the vertical direction, the resulting flatness of the interpolated data was analyzed to gage the accuracy of the algorithms. The standard deviation of the displacements along each column is shown in Figure 6-21 with an average for all columns of $0.35 \times 10^{-3}$ cm. This suggests that displacements have been generated within reasonable limits and the interpolation functions employed yield acceptable results.

![Planar Checkerboard Images in Reference Plane: a) Original (640 x 480 pixels) and b) Dewarped (554 x 417 pixels)](image-url)
Figure 6-20  CAD Displacement Field From Surface Interpolation

Figure 6-21  Interpolated CAD Surface Standard Deviation of Columns
6.7 **COMBINED RESULTS**

The displacement field generated from inaccessible surface measurements on a CAD model was compared to that generated from virtual calibration procedures on the assembled airfoil. Because of the registration process, the two displacement fields can simply be differenced at each pixel location. The result is shown in Figure 6-22 and the differences introduced by the entire process were quantified by comparing the displacements to those found using the coordinate measuring machine (CMM). The CMM displacement field (assembly scan – fixture scan) is shown in Figure 6-23. For comparison, the color scale was forced to equal to that found in Figure 6-22. Percent differences between the two displacement fields were calculated using the CMM data as the standard for comparison. These results are shown in Figure 6-24, where the percent difference ranges from -14.39 % to 14.31 %. As observed for the PMI analysis in Section 6.3, this data seems to be somewhat constant along any particular column. The average percent difference for each column is shown in Figure 6-25. Since the CAD model of the fixture and the CMM scan of the fixture are highly comparable, the errors shown are mostly due to the inadequacies associated with the PMI analysis. To illustrate this fact, a single cross-sectional slice of the data was investigated. A fixture scan, an assembly scan, and PMI data were each used for the analysis. This data is shown in Figure 6-26, where the error bars shown on the PMI data were computed following procedures described in Section 4.6, and illustrated in Figure 4-18 through Figure 4-20. The data contained in the cross-sectional slice was used to compute two separate percent differences. The first was the percent difference in the PMI data compared to the assembly scan. The second was the percent difference between PMI data and the fixture
scan compared to the difference between the assembly and fixture scans. The result is shown in Figure 6-27 and reveals the same trends compared to the column averages computed in each case (see Figure 6-15 and Figure 6-25).

Figure 6-22  Combined Displacement Field (PMI – CAD)
Figure 6-23  CMM Displacement Field (Fixture Scan – Assembly Scan)

Figure 6-24  Percent Difference in Combined Measurement Process
Figure 6-25  Average Percent Error in Combined Measurement Process Along Columns

Figure 6-26  Cross-Sectional Slice of Data
Figure 6-27  Percent Difference for Cross-Sectional Slice
7 CONCLUSIONS

This chapter summarizes the results from the validation and test case. Recommendations are made for future work, and a review is conducted of previous and future publications relating to this research.

7.1 SUMMARY

A method has been developed for determining out-of-plane deformations, which makes use of virtual calibration techniques and incorporates CAD models in the measurement process. The test case performed has shown errors within 15% of the displacements under investigation, which suggests that techniques employed could serve as an extremely valuable tool in these types of applications. The error is attributed to three recognized sources. One source of error is introduced when attempting to use a common insertion point between the planar checkerboard in the reference plane and the flat plate. Small differences here contribute to the overall error. Methods for lessening or eliminating this effect could possibly be developed. The second source of error is attributed to the camera resolution (640 x 480 pixels). A higher resolution camera will assist in this effort, but it is also worthwhile to investigate any image analysis algorithms that might enhance image quality without requiring a higher resolution. The third source of error is introduced by the PMI measuring techniques. Projection-oriented measuring systems typically are not as precise as other methods, but for some applications, the
relatively uncomplicated layout and cost-effectiveness outweigh the negative effects of accuracy. For this reason, they are ideal for large-scale applications. Typically, the PMI errors lessen as the FSC values increase, producing a more sensitive setup. However, as this is done, more fringes will appear for a given displacement. Too many fringes with respect to the camera resolution produce an unrecognizable fringe pattern. In a way, the PMI errors are limited to the FSC values, which are dependent on the camera resolution. In addition, the displacements measured during the PMI analysis of the assembled airfoil (difference between reference plane and assembly) were roughly twenty times as large as those measured during the combined process (difference between assembly and fixture). This creates somewhat of a dilemma in choosing suitable FSC values. For an application where both sets of displacements are nearly equal, results will produce less error. In conclusion, the developed measurement technique overcomes two limitations of traditional PMI methods, and further research could provide results that are even more accurate.

7.2 RECOMMENDATIONS

Future research in PMI techniques could prove valuable in decreasing the error introduced during analysis. Perhaps a new method could be developed to calculate FSC values and create a more accurate PMI setup. Research performed in this thesis made use of phase-stepped reference images to compute a wrapped phase map. Methods that are more accurate could possibly be developed, thereby directly affecting the measurement capabilities of PMI techniques in a general sense, which would also have a positive impact on techniques developed in this research. In addition, camera calibration techniques could be explored to investigate better accuracy and require less image
capturing than those techniques used in the current research. They were adequate to prove that the concepts work and to generate acceptable results, but in reality, errors are introduced in this process as well. Methods to alleviate this negative effect could also be researched in greater depth. Finally, methods could be explored which incorporate two sets of FSC values. This would overcome the dilemma mentioned of measuring two significantly different displacement fields with the same FSC values. One set could be used to measure the contours of the actual object, while the other could be used to determine the relative displacements between the actual object and CAD model.

7.3 PUBLICATIONS

Work describing the virtual camera creation has been published as part of the Society of Experimental Mechanics 10th International Congress & Exposition, which took place June 7-10, 2004. Plans exist to publish two separate journal articles describing each of the two procedures developed. The virtual calibration procedures will be submitted for publication in *Applied Optics* and the inaccessible surface measurement procedures will be submitted for publication in *Optics and Lasers in Engineering.*
8 REFERENCES


37 Matlab camera calibration toolbox developed by Jean-Yves Bouguet, PhD found on http://www.vision.caltech.edu/bouguetj/calib_doc/.


40 In Matlab camera calibration toolbox documentation under “First calibration example - Corner extraction, calibration, additional tools” on http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html


42 Software developed by Zemax Corporation – 4901 Morena Blvd. Suite 207, San Diego, CA 92117-7320 USA

43 Zemax user manual version November 12, 2003 page 334 (User defined scattering)


45 Matlab function *griddata* written by Clay M. Thompson, August 21, 1995.

46 Association for the Development of Computer-Aided Tolerancing Systems (ADCATS) research group at Brigham Young University.
This Appendix contains information regarding specific procedures needed to re-create some of the experimental work presented in the body of the thesis. A tutorial for creating a virtual PMI setup within the software program Zemax is presented, followed by a collection of images used for camera and projector calibration. Methods used to extract point cloud data from a CAD model are then explained.

**Zemax Tutorial for Creation of Virtual PMI Setup**

The learning curve associated with this task caused it to be the most time consuming step of the research. This tutorial is intended to shorten that learning curve for the interested reader. It is divided into five sections: 1) introduction to Zemax, 2) virtual camera modeling, 3) virtual projector modeling, and 4) user defined scatter functions. It is important to note that Zemax was created mainly for those concerned with designing and manufacturing optical components. The user manual (version November 12, 2003) is 619 pages, but no information will be found on accomplishing the tasks associated with this project. The user manual can provide the reader with information regarding the basic layout and functionality of Zemax, so the goal of this section is to provide a complete tutorial for those things which are absent from the user manual and which customer support is not available.

**Zemax Introduction**

Ray tracing within Zemax can be done using two different methods: sequential and non-sequential. The tasks of creating a virtual PMI setup are all completed using non-sequential ray tracing methods. This is done under the File menu as shown in Figure
A-1, and the non-sequential component editor should then appear as shown in Figure A-2. Each line in this editor represents an object, whether a light source, a rectangle, or any of the numerous built in object types. There are a number of common parameters used to describe each object, as well as additional object specific parameters. For instance, all objects are defined by material (optional), and six degrees of freedom describing the geometric location and orientation from the world coordinate frame to that of the object. These six transformations are performed in accordance with the homogeneous transformation matrix and supposing the object were a rectangle, additional parameters would include the x and y half widths. Details for each object type can be found in the user manual. By default, the six degrees of freedoms for an object use the world coordinate system as a base coordinate system. A particularly useful feature is the ability to change this under the column heading “Ref Object,” which takes an object number (row) as an input, and uses that object for the base coordinate system. The origin of the plate in the actual setup can be used as the world frame, or the camera and projector can simply be described relative to this frame, regardless of its location.
When inserting a new object, the default is a Null Object. This is used as the world origin, which is typed in the Comment column for reference as shown in Figure A-3. It might help to specify to draw the local axis of this object on the layout. This is
done by selecting the option highlighted in Figure A-3. The layout can be viewed by selecting from the main menu options shown in Figure A-4. A window titled “NSC 3D Layout” should then appear.

![Non-Sequential Component Editor: Component Group on Surface 1](image)

**Figure A-3** Creating a World Origin
The reference plane in the PMI setup is represented by a Rectangle object in the virtual setup. First, insert a new object by pressing “Ctrl+Insert” on the keyboard, and then double clicking on the new object created. This will open an “Object 2 Properties” window as shown in Figure A-5, where a “Rectangle” may be selected as the object type. Next parameters must be defined to describe the position, orientation, and size of this object. It is important to note that the geometric center of an object is used as the local origin, so in this case the reference plane must be translated in the x and y directions amounts equal to the half widths in each direction defined for the rectangle. This ensures the corner of the reference plane lies on the same point as the world origin. Procedures to create each of the remaining components (camera and projector) are now explained.
The virtual camera is relatively uncomplicated, and consists of two objects: an “Ellipse” and a “Detector Rect.” The ellipse serves as a marker to represent the camera location, and rays pass through the center unaltered. The object number for the world origin is inserted into the “Ref Object” column of the ellipse as shown in Figure A-6. The six degrees of freedom for the ellipse will then represent the transformation from the world origin to the camera, and the values determined from camera calibration should be
inserted here. Next, the rectangular shaped detector is inserted using the ellipse object number in the “Ref Object” column. The “X Position” and “Y Position” columns are used to describe the image center. Keep in mind that values of zero for both parameters represent a perfectly centered camera. The “Z Position” column represents the camera focal length, and is a negative value. The size of the detector and number of pixels in two directions are then used with the remaining intrinsic parameters of the camera to create a virtual camera with all the same properties as the actual camera.

![Non-Sequential Component Editor: Component Group on Surface 1](image)

**Figure A-6** Determining Camera Location
Virtual Projector

The virtual projector is represented by “Source Point” and “Slide” objects. The point source is first inserted using the world origin object number in the “Ref Object” column. The six geometric degrees of freedom determined from projector calibration are used to describe the transformation from the world origin to the projector. There are three other parameters needed to describe the point light source. The first is “# Layout Rays,” which specifies the number of rays shown in the “NSC 3D Layout” window. The
second is “# Analysis Rays,” which denotes the number of rays to use during the ray tracing process. The third parameter is “Cone Angle,” which describes the cone angle of the projection. The “Slide” object is then inserted and the desired image to use for projection is then selected. This image must first be saved in the “ZEMAX\Images” directory. The three translations and the geometric size of the image are used in conjunction with the intrinsic parameters determined from projector calibration. The setup including all components is shown in Figure A-9. Now the task becomes scattering the rays at the surface of the reference plane, which is described next.

Figure A-8  Selecting Image for Slide Object
User Defined Scatter Function

In order to scatter the rays to a single point representing the location of the camera, a user defined scatter function was written in Visual C++ as a dynamic link library (DLL). The code in its entirety is presented in Section 0. Once the DLL has been generated, it must be copied to the “ZEMAX\Objects\DLL\SurfaceScatter” directory. It can then be selected for the scatter profile of the reference plane. This is done by double...
clicking on the reference plane object and selecting the “Coating/Scattering” tab. Select the “User Defined” option for scattering and the appropriate DLL from the drop-down menu as shown in Figure A-10. Once this is complete the user inputs fields appear, which are specified by the DLL. For this research, the “PMI_CAL.DLL” was used for the scatter profile. It requires user inputs for the three translations from the local coordinate system of the reference plane to that of the camera. In addition, the rotation angle used for a particular calibration image and the reference plane half width are also used in the calculations. These parameters are shown highlighted in Figure A-11. Also highlighted are the scatter fraction and number of rays options. The scatter fraction is a value from 0 to 1 describing the percentage of rays to scatter at the surface, and the number of rays specifies the quantity of rays in which to split each intercepting ray before scattering. The entire setup depicting scattered rays is shown in Figure A-12, and the final task is to run the ray tracing. This is done by selecting the options shown in Figure A-13. A “Detector Control Surface 1” window will appear with a number of options. The “Scatter Rays” option must be selected as shown in Figure A-14, and then the “Trace” option can be selected to start the ray trace routines. The detector then collects the light rays and the result can be viewed by selecting the “Detector Viewer” option shown in Figure A-13.
Figure A-10  Selecting Scatter Profile for Reference Surface

Figure A-11  Input Parameters for PMI_CAL.DLL
Figure A-12  Setup Including All Components and Scattered Rays

Figure A-13  Beginning the Ray Trace Routines
Scatter code

/***************************************************/
/*        */
/* This program generates a scatter profile */
/* that represents a paraxial, or pinhole */
/* camera. It can also be used in PMI */
/* calibration procedures. */
/*        */
/* Adapted by Mark Kimber on September 17, */
/* 2003 from code originally written by */
/* Kenneth E. Moore on July 3, 2003. */
/*        */
/***************************************************/
#include <windows.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
int __declspec(dllexport) APIENTRY UserScatterDefinition(double *data);
int __declspec(dllexport) APIENTRY UserParamNames(char *data);
void Normalize(double *x, double *y, double *z);

BOOL WINAPI DllMain (HANDLE hInst, ULONG ul_reason_for_call, LPVOID lpReserved)
{
    return TRUE;
}

/* the data is stored as follows:
data[ 0] = the total number of values in the passed data array
data[ 1] = x position of specular ray
data[ 2] = y position of specular ray
data[ 3] = z position of specular ray
data[ 4] = x cosine of specular ray, on output it is the scattered ray
data[ 5] = y cosine of specular ray, on output it is the scattered ray
data[ 6] = z cosine of specular ray, on output it is the scattered ray
data[ 7]  = x normal
data[ 8]  = y normal
data[ 9]  = z normal

data[10] = 0 initially
if the DLL scatters the ray return 1 in data[10].
if the DLL returns full polarization data return 2 in data[10].

data[11] = millimeters per unit length (1.0 for mm, 25.4 for inches, 10.0 for cm and 1000.0 for meters)
data[12] = relative energy (to be computed by the dll and returned)
data[16] = a random value to use as a seed
data[17] = wavelength in microns

data[20] = incident Ex real
data[21] = incident Ex imaginary
data[22] = incident By real
data[23] = incident By imaginary
data[24] = incident Ez real
data[25] = incident Ez imaginary

data 40-45 need to be computed if the DLL sets data[10] = 2

data[40] = output Ex real
data[41] = output Ex imaginary
data[42] = output By real
data[43] = output By imaginary
data[44] = output Ez real
data[45] = output Ez imaginary

data[51] = input parameter 1 from user
data[52] = input parameter 2 from user
eetc... up to data[maxdata] where maxdata = int(data[0])

Return 0 if it works; else return -1.
*/

/* The code starts here -- The objective is to force all rays to refract to a single point representing the pinhole for the camera. */

int __declspec(dllexport) APIENTRY UserScatterDefinition(double *data)
{
    double dx, dy, dz;
    double ang, xhalf;
    double posx, posy, posz;

    /* return 100% transmission */
data[12] = 1.0;

    /* parameters input by user -- 3 translations from the part coordinate frame to the camera coordinate frame. In addition, for PMI calibration purposes, the rotation angle (deg) and the x half width of the plate are inputs */
    posx = data[51];
    ang = data[52]*3.141593/180;  //convert to radians
    posy = data[53];
    xhalf = data[54];
    posz = data[55];

    /* Create vector from surface point to center of lens and normalize. This accounts for a rotated plate for PMI calibration procedures */
    dx = cos(ang)*posx - sin(ang)*posz + xhalf*(cos(ang)-1)-data[1];
    dy = posy - data[2];
    dz = sin(ang)*posx + cos(ang)*posz + xhalf*sin(ang) - data[3];
/* A function to normalize a vector */
void Normalize(double *x, double *y, double *z)
{
    double temp;
    temp = (*x)*(*x) +(*y)*(*y)+(*z)*(*z);
    temp = sqrt(temp);
    if (temp == 0) return;
    temp = 1.0/temp;
    *x *= temp;
    *y *= temp;
    *z *= temp;
}

CALIBRATION IMAGES

The images used for camera and projector calibration are shown to illustrate the
genral angles and positions that produced calibration results presented in Sections 4.2
and 4.3. For this research, sample calibration images were quite helpful in determining
approximate locations and orientations needed to produce certain results. For this intent,
these images are included.
Camera Calibration
Figure A-15  Camera Calibration Images
Projector Calibration
The Matlab code for extracting point cloud data from an STL file is presented for the interested reader.

```matlab
% This file generates a point cloud data (i x 3) array from the facet
% vertices of an STL file. Inputs are the STL file and the number of
% facets. Output is unique 3-D points representing the surface.

function points = stl2pointcloud(file,num_facets);

% Open STL file and data file to write to
fid = fopen(file,'r');
fid2 = fopen('data.txt','w');

% get rid of 'solid' line
a = fgets(fid);

% loop through facet vertices
for i = 1:num_facets
    a = fgets(fid);
    a = fgets(fid);
    for j = 1:3
        a = fgets(fid,12);
        a = fgets(fid);
        a2 = str2num(a);
        if isempty(a2) == 0
            fprintf(fid2, '%f ',a2);
        end
        fprintf(fid2, '
');
    end
    a = fgets(fid);
    a = fgets(fid);
end

fclose(fid);
fclose(fid2);

% Read point cloud data
a3 = load('data.txt');

% Find number of unique points
points = unique(a3,'rows');
```