January 2015

LOW-COST STRUCTURED-LIGHT 3D CAPTURE SYSTEM DESIGN

Jing Dong
Purdue University

Graduate School Form 30
(Revised 08/14)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Jing Dong

Entitled
Low-Cost Structured-Light 3D Capture System Design

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

JAN P. ALLEBACH

MARY L. COMER

MIREILLE BOUTIN

ZYGMUNT PIZLO

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification/Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): JAN P. ALLEBACH

Approved by: V. Balakrishnan 08/10/2015

Head of the Department Graduate Program Date

LOW-COST STRUCTURED-LIGHT
3D CAPTURE SYSTEM DESIGN

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Jing Dong

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2015

Purdue University

West Lafayette, Indiana

To my husband Nianpei, my son Liam and my parents for their love and support.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor, Professor Jan Allebach, who encourages me to work hard to achieve a set of goals in my research. I want to thank him for his countless hours of reflecting, reading, encouraging, and most of all patience throughout the entire process.

I wish to thank my committee members Professors Zygmunt Pizlo, Mary Comer, and Mireille Boutin who were more than generous with their expertise and precious time.

I also would like to thank Professor Barrett Robinson for the rigid and exquisite fixtures he made for our system.

In addition, I want to thank all my lab mates, especially Yang Lei and Chyuan-Tyng Wu. From everyday discussions with them in the lab, I keep learning outside my field of research.

Last but not least, I want to thank my parents, my husband and my son for supporting me through the whole process.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

ABSTRACT

Dong, Jing Ph.D., Purdue University, December 2015. Low-Cost Structured-Light 3D Capture System Design. Major Professor: Jan P. Allebach.

Currently, three-dimensional measurement is a very important and popular topic in computer vision. Most of the 3D capture products currently in the market are high-end and pricey. They are not targeted for consumers, but rather for research, medical, or industrial usage. Very few aim to provide a solution for home and small business applications. Our goal is to fill in this gap by only using low-cost components to build a 3D capture system that can satisfy the needs of this market segment. In this paper, we present a low-cost 3D capture system based on the structured-light method. The system is built around the HP TopShot LaserJet Pro M275. For our capture device, we use the 8.0 Mpixel camera that is part of the M275. We augment this hardware with two 3M MPro 150 VGA (640×480) pocket projectors. We also describe an analytical approach to predicting the achievable resolution of the reconstructed 3D object based on differentials and small signal theory, and an experimental procedure for validating that the system under test meets the specifications for reconstructed object resolution that are predicted by our analytical model. By comparing our experimental measurements from the camera-projector system with the simulation results based on the model for this system, we conclude that our prototype system has been correctly configured and calibrated and that with the analytical models, we have an effective means for specifying system parameters to achieve a given target resolution for the reconstructed object.

1. INTRODUCTION

To date, three-dimensional measurement is a very important and popular topic in computer vision [1] [2]. Due to its accuracy and time-efficiency, 3D capture is broadly applied in a number of areas, such as object recognition for quality control and inspection [3], dental [4] and facial imaging [5, 6], 3D map building [7], and customization in apparel [8] and footwear [9]. 3D capture techniques may be divided according to contact [10–14] and non-contact methods [15–32]. The problem with contact methods is the slowness and high cost of the probing process. In addition, the probe may cause damage to the surface of the object. Non-contact methods can be classified into passive [18, 23–26, 28] and active methods [15–17, 19, 20, 22, 27, 29–32]. The passive methods completely rely on ambient light, while the active methods emit certain controlled radiation or light and detect its reflection. The most widely used passive 3D imaging system is the stereo vision [25, 28]. And Time-of-Flight (ToF) [29, 30, 33], laser triangulation [15, 17, 19, 20, 27], structured light [16, 22, 31, 32] are typical active 3D imaging systems which employ different radiation sources or measurement methods. For passive method, two or more views are captured by the cameras and the correspondences between the images need to be found to reconstruct the 3D surface. Thus, passive method is limited to reconstruct dense 3D surfaces due to the complexity of finding correspondences. To solve this, structured-light method is more widely used nowadays. It substitutes one of the cameras as a projector, and project a coded pattern. By decoding the captured encoded image, we could easily find the correspondence between the projected image and the captured image; thus, we could reconstruct the 3D point using triangulation.

Structured-light system employs a variety of coding strategies based on the system constraints; for example, number of projected patterns, number of cameras, pixel depth or whether it is a moving scene or a static scene [34]. The existing pattern coding strategies in structured light system can be classified into three categories [35]: time-multiplexing [36], direct codification and spatial neighborhood. Spatial neighborhood is the coding strategy that uses surrounding pixels to create the codeword for the center pixel. It includes non-formal coding [37], De Brujin patterns [38] and M-arrays [39, 40], which is a relative new coding strategy that is time-efficient, accurate and robust against color and occlusion.

Lanman and Taubin [41] provide a tutorial course for beginners to build their own 3D capture system. They explain the mathematics of the triangulation, camera and projector calibration, and develop a classic structured light scanning system and a laser line scanning system. Geng [42] presents a review of the recent advances in 3D surface imaging technologies. He focuses on non-contact surface measurement techniques based on structured light and categorizes and compares different coding strategies. He also discusses the calibration techniques and numerous applications of 3D surface imaging techniques.

Most of the 3D capture products currently in the market are high-end and pricey. They are not targeted for consumers, but rather for research, medical, or industrial usage. Unlike research, medical or industrial object capture, object capture for the home hobbyist or small business does not require very high accuracy. It is a challenge to balance the trade-off between accuracy and system cost. We can sacrifice part of the accuracy to lower the cost of the 3D capture system, on the one hand. On the other hand, we do not want to lose so much accuracy that the object cannot even be recognized. Therefore, our goal is to maintain sufficient accuracy while keeping the system cost within a range that is suitable for home or small business use. A number of factors need to be considered when choosing the system components. The first group of these includes the resolution, light-level output, geometric distortion, and working distance for the projector. The working distance must be chosen such that the whole object station is completely included in the field of view of the camera. Another issue with the projector is the throw ratio. The throw ratio is the ratio between the throw distance and the width of the screen, where the throw distance is the distance from the screen to the projector. Most projectors on the market have a relatively large throw ratio. Therefore, given a desired projected image size, the large throw ratio requires a large throw distance, which causes the whole 3D capture system not to be compact. The second group of factors is related to the camera. These include resolution, sensitivity, geometric distortion, and color fidelity [43]. The latter is important if a second frame is to be captured under normal illumination without the structured light pattern, to provide the surface reflectance information that is to go with the object shape.

In Chapter 1, we first review the background of the structured light based 3D capture system, and discuss the limitation of the one projector system based on our requirement. We introduce the novel design and layout of our dual-projector system in Chapter 2. The camera and projector modeling, calibration procedure and verification of the calibration results are discussed respectively in Chapter 3 and 4. The world coordinates calculation is also presented in Chapter 4. In Chapter 5, we use the small signal theory to analyze the sensitivity of the system by comparing the theoretical analysis results with the experimental results, to establish the resolution relationship between the camera and projector, as well as the design tolerance. Chapter 6 is an application we developed using the 3D depth map we get from our 3D capture system to change the perspective of an image. Chapter 7 is a separate project which aims to correct the lens distortion and perspective distortion of a 2D image. No 3D information is involved and required. We discuss the model and method for the image correction, and the results are presented at the end. In Chapter 8, we present the detailed theory, procedure and result using straight line pattern based 3D capture method and our low-cost dual projector system. In Chapter 9, we analyze the reconstruction accuracy of our system based on the straight line 3D capture method. Last but not least, we present in Chapter 10 an application using the 3D depth map we generated in Chapter 8 to change the perspective of the object, which is similar to Chapter 6.

2. BASIC SYSTEM LAYOUT AND DESIGN

In this section, we introduce our newly designed dual-projector 3D structured light capture system. Structured light method is widely used nowadays due to its time-efficiency and accuracy. In general, the system consists of one projector and one or more cameras. We first project the structured light pattern to the object and use the camera to capture this. By decoding the pattern, we can find the correspondence between the projected pattern and the captured pattern. Then we use the triangulation to find the depth of the object.

Several issues for the components we need to consider will be discussed as following. First is the resolution of the projector. Almost any digital projector can be used in the 3D structured light system. At least a VGA projector (640×480) is recommended. The resolution of the camera should be higher than the projector; otherwise, the reconstruction resolution will be restricted by the cameras resolution. The working distance is chosen such that the whole object station is completely included in the field of view of the camera.

Another issue with the projector is the throw ratio. Throw ratio is a ratio between the throw distance and the width of the screen (Fig. 2.1), where throw distance is the distance from the screen to the projector. Given the dimensions of the Topshot (Fig. 2.2), we need a projector has throw ratio $= \frac{8.5''}{13.5''} \approx 0.63$ and image size could achieve $16.2''$ or project distance could achieve $8.5''$. Most projectors on the market are designed to have relatively large throw ratio. Therefore, given a desired projection image size, the large throw ratio requires a large throw distance, which makes the whole 3D capture system not compact. To solve this issue, we introduce the dual projector system, which uses two relatively short-throw-ratio projectors to cover each half of the platen. Therefore, the projector screen size is reduced by half, and the

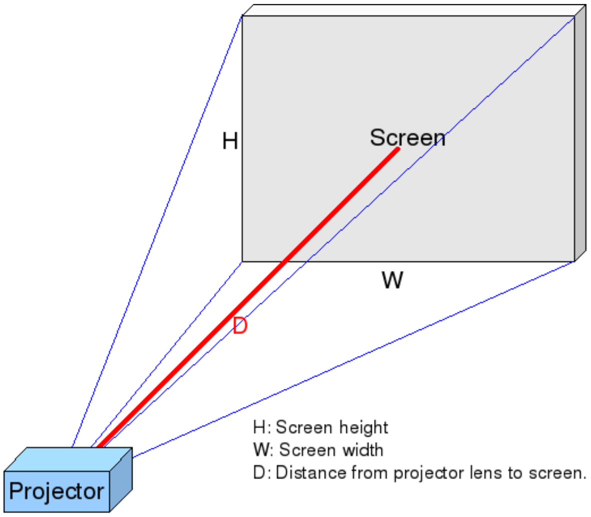


Fig. 2.1. Illustration of the throw ratio and throw distance (image from web)

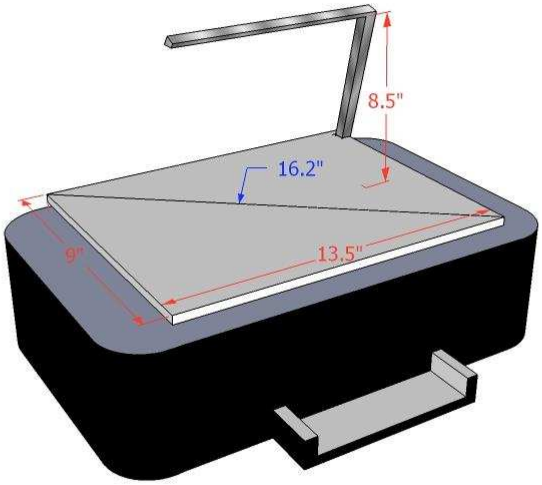


Fig. 2.2. Required throw ratio if use one projector given the dimension of Topshot

required throw distance is reduced as well. The model of this design is plotted using Google Sketchup and showed in Fig.2.3.

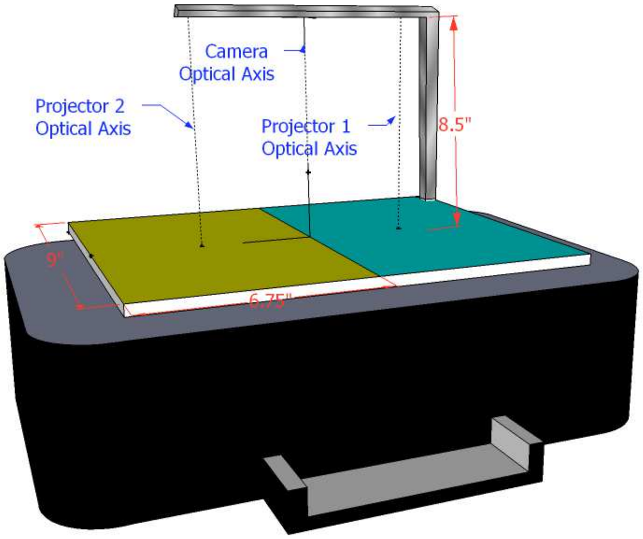


Fig. 2.3. Design of the dual-projector system built around Topshot

Fig. 2.4 shows our newly designed dual-projector 3D capture system. The system is built around the HP TopShot LaserJet Pro M275. For our capture device, we use the 8.0 Mpixel camera that is part of the M275. We augment this hardware with two 3M MPro 150 VGA (640×480) pocket projectors which are the smallest-throw-ratio projectors we find on the market. Both the M275 and the MPro 150 are low-cost products.

By calculating the throw distance using the desired projection image size and throw ratio of the projector, we still need to hang the projectors 5-6 inches above the camera. Therefore, we also design two heavy-duty posts to hold and control the movement of the projectors. Fig. 2.5 is the engineering graph of the post Dr.Barrett Robinson design and build for us. As shown in Fig. 2.4 there are three micrometers to control the projector moving up/down, left/right, and backward/forward accurately. One addition micrometer can control the tilting of the projector and read the degrees as well. The screws on the tower and on the arm of the tower give us extra freedom to adjust the projector to the desired position. More views of the system are shown in Fig. 2.6.

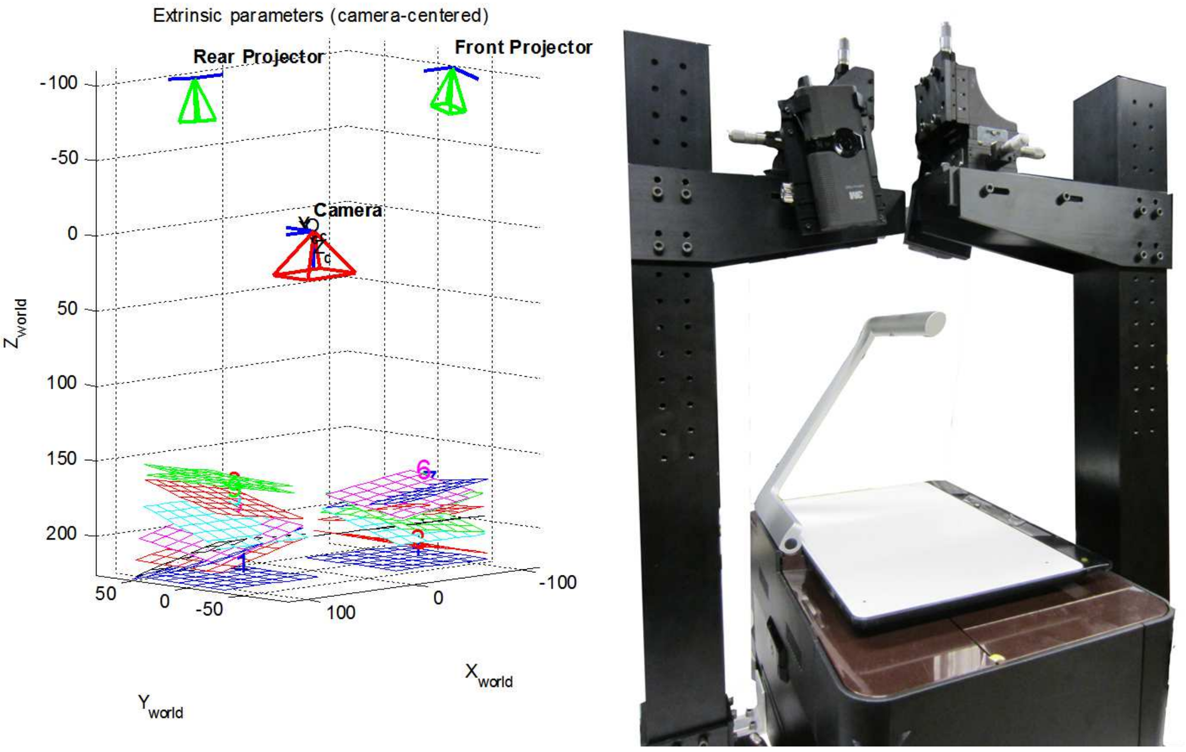
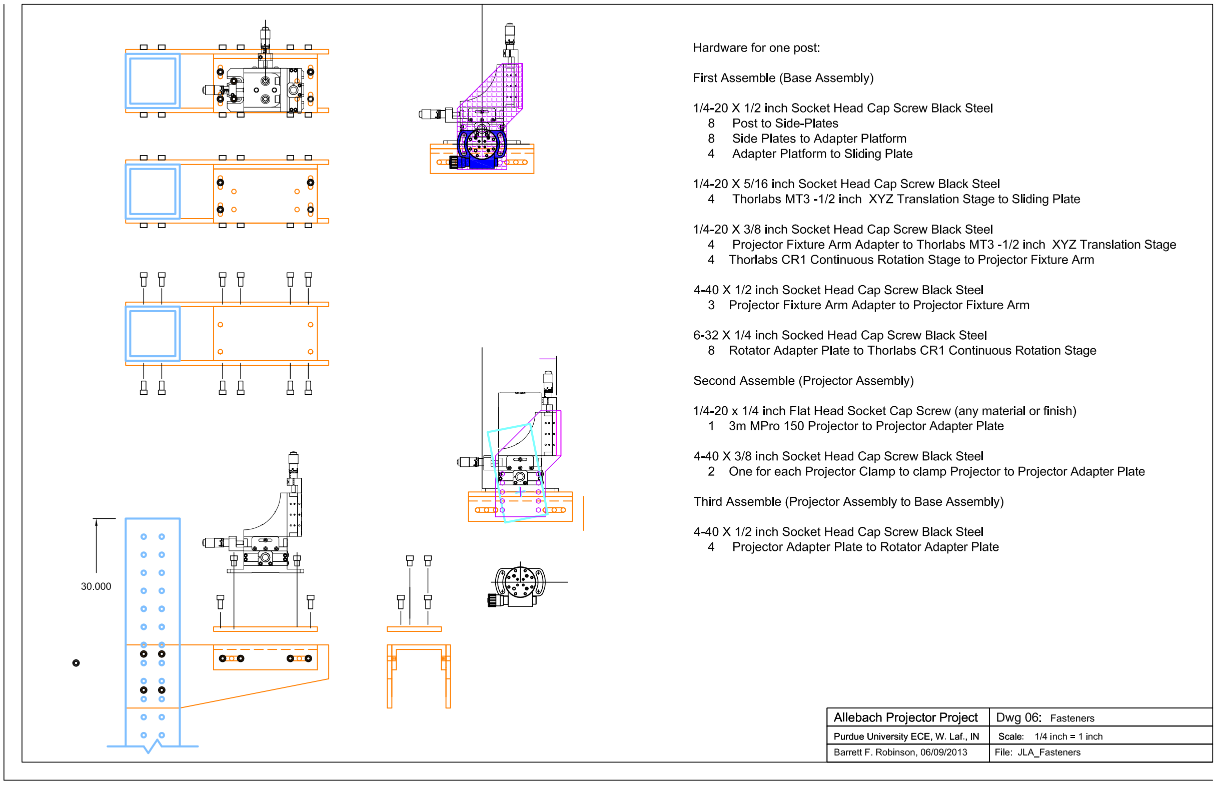


Fig. 2.4. Visualization of the calibration result for the dual-projector 3D capture system (left) and set-up (right)



Hardware for one post:

First Assemble (Base Assembly)

1/4-20 X 1/2 inch Socket Head Cap Screw Black Steel
8 Post to Side-Plates
8 Side Plates to Adapter Platform
4 Adapter Platform to Sliding Plate

1/4-20 X 5/16 inch Socket Head Cap Screw Black Steel
4 Thorlabs MT3 -1/2 inch XYZ Translation Stage to Sliding Plate

1/4-20 X 3/8 inch Socket Head Cap Screw Black Steel
4 Projector Fixture Arm Adapter to Thorlabs MT3 -1/2 inch XYZ Translation Stage
4 Thorlabs CR1 Continuous Rotation Stage to Projector Fixture Arm

4-40 X 1/2 inch Socket Head Cap Screw Black Steel
3 Projector Fixture Arm Adapter to Projector Fixture Arm

6-32 X 1/4 inch Socked Head Cap Screw Black Steel
8 Rotator Adapter Plate to Thorlabs CR1 Continuous Rotation Stage

Second Assemble (Projector Assembly)

1/4-20 x 1/4 inch Flat Head Socket Cap Screw (any material or finish)
1 3m MPro 150 Projector to Projector Adapter Plate

4-40 X 3/8 inch Socket Head Cap Screw Black Steel
2 One for each Projector Clamp to clamp Projector to Projector Adapter Plate

Third Assemble (Projector Assembly to Base Assembly)

4-40 X 1/2 inch Socket Head Cap Screw Black Steel
4 Projector Adapter Plate to Rotator Adapter Plate

Allebach Projector Project	Dwg 06: Fasteners
Purdue University ECE, W. Laf., IN	Scale: 1/4 inch = 1 inch
Barrett F. Robinson, 06/09/2013	File: JLA_Fasteners

Fig. 2.5. Design of heavy-duty posts to hold and control the movement of the projector

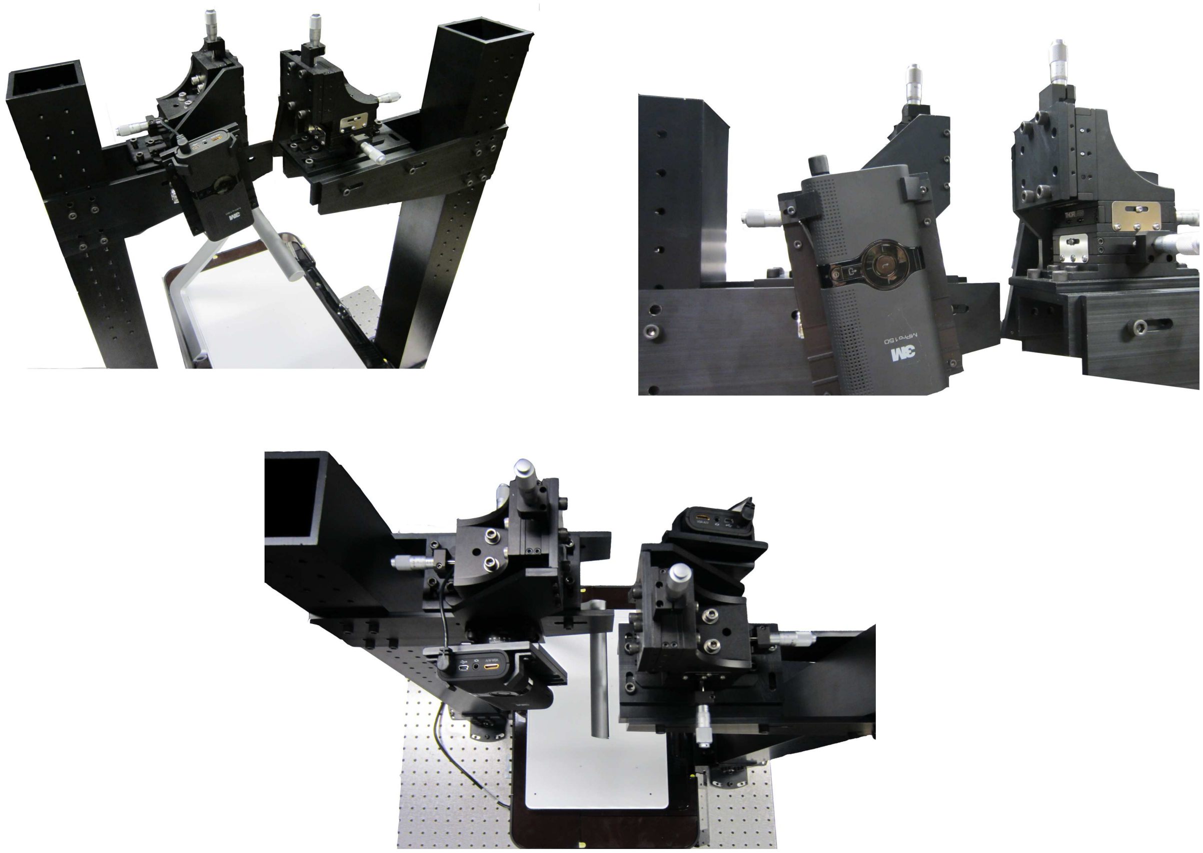


Fig. 2.6. Dual-projector 3D capture system front view, top view and zoom-in view

3. CAMERA MODELING,CALIBRATION AND VERIFICATION

In this section, we introduce the commonly used camera and projector model and calibration method for the 3D capture system. In order to use triangulation to recover the 3D object, we first need to calibrate the camera and projectors. The goal of calibration is to get the intrinsic parameters (focal length, principal point, skew coefficient, distortions) and extrinsic parameters (rotation, translation matrix) of the camera and the projectors.

3.1 Camera Model

The camera model with lens distortion [44] is shown in Fig. 3.1, and we will discuss the model in detail as follows:

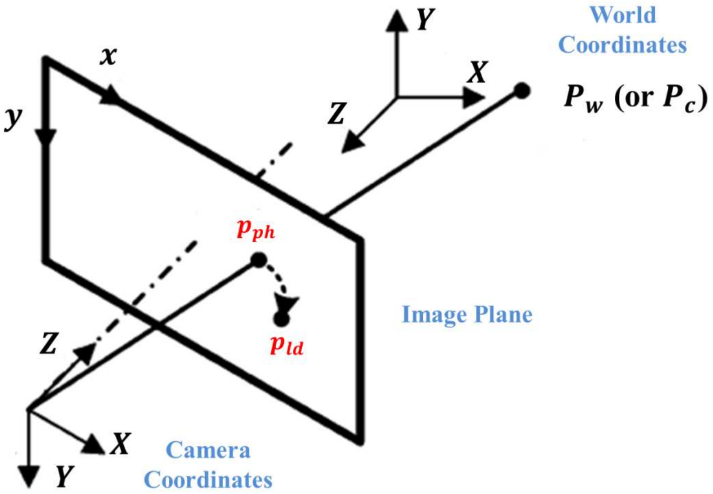


Fig. 3.1. Pinhole camera model with lens distortion

Given a 3D object point in the world coordinate system $P_w = \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$, we first convert it to the camera coordinate system $P_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$ using Eqn. 3.1,

$$P_c = \mathbf{R}_c \times P_w + \mathbf{T}_c \tag{3.1}$$

where $\mathbf{R}_c$ is the 3×3 rotation matrix and $\mathbf{T}_c$ is the 3×1 translation matrix which relates the world coordinate system to the camera coordinate system. Then the 2D point on the camera's image plane using pinhole camera prediction is $p^c_{ph} = \begin{bmatrix} x^c_{ph} \\ y^c_{ph} \end{bmatrix}$, and the 2D point on the image plane considering lens distortion is $p^c_{ld} = \begin{bmatrix} x^c_{ld} \\ y^c_{ld} \end{bmatrix}$

Consider if we project the object point $P_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$ to the normalized image plane using Pinhole camera model, we get the 2D image coordinate $p^c_{phn} = \begin{bmatrix} x^c_{phn} \\ y^c_{phn} \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}$. By adding the radial and tangential lens distortion to p^c_{phn}, we get the distorted projection on the camera's normalized image plane is:

$$p^c_{ldn} = \begin{bmatrix} x^c_{ldn} \\ y^c_{ldn} \end{bmatrix} = \begin{bmatrix} \varphi_1(x^c_{phn}, y^c_{phn}) \\ \varphi_2(x^c_{phn}, y^c_{phn}) \end{bmatrix} \tag{3.2}$$

where

$$\begin{aligned} \begin{bmatrix} \varphi_1(x^c_{phn}, y^c_{phn}) \\ \varphi_2(x^c_{phn}, y^c_{phn}) \end{bmatrix} = & \begin{bmatrix} x^c_{phn} \\ y^c_{phn} \end{bmatrix} + (k^c_1 r^2 + k^c_2 r^4 + k^c_5 r^6) \begin{bmatrix} x^c_{phn} \\ y^c_{phn} \end{bmatrix} \\ & + \begin{bmatrix} 2k_3 x^c_{phn} y^c_{phn} + k_4(r^2 + 2x^{c2}_{phn}) \\ k_3(r^2 + 2y^{c2}_{phn}) + 2k_4 x^c_{phn} y^c_{phn} \end{bmatrix} \end{aligned} \tag{3.3}$$

and $r^2 = x_{phn}^{c2} + y_{phn}^{c2}$, $\mathbf{K}^c = \left[\begin{array}{c} k_1^c, k_2^c, k_3^c, k_4^c, k_5^c \end{array}\right]$ is the camera's lens distortion coefficient.

Then the coordinates on the camera's image plane considering pinhole camera model with lens distortion are:

$$\begin{bmatrix} x_{ld}^c \\ y_{ld}^c \\ 1 \end{bmatrix} = KK^c \begin{bmatrix} x_{ldn}^c \\ y_{ldn}^c \\ 1 \end{bmatrix} \tag{3.4}$$

where

$$KK^c = \begin{bmatrix} f_x^c & f_{s\theta}^c & u_o^c \\ 0 & f_y^c & v_o^c \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

KK^c is called the camera intrinsic parameter matrix. Here f_x^c and f_y^c are the camera's focal length in the x and y direction; u_o^c and v_o^c are the coordinates of the principal point on the camera's image plane; and $f_{s\theta}^c$ is the camera's skew coefficient.

3.2 Camera Calibration Procedure

The camera calibration procedure we use comes from Zhang's method [45]. We first print a checkerboard pattern composed of 20mm×20mm squares and attach it to the platen board, which is a white board on the top of the Topshot. Then we vary the positions of the platen and capture a sequence of 12 views of it. We next apply edge detection to extract the grids of the checkerboard pattern. Calibration is done in two steps. In the initialization step, we compute a closed-form solution for the calibration parameters regardless of any lens distortion. Next we use iterative gradient descent method to minimize the total reprojection error over all the calibration parameters. We use an open-source camera calibration toolbox to implement the camera calibration [46].

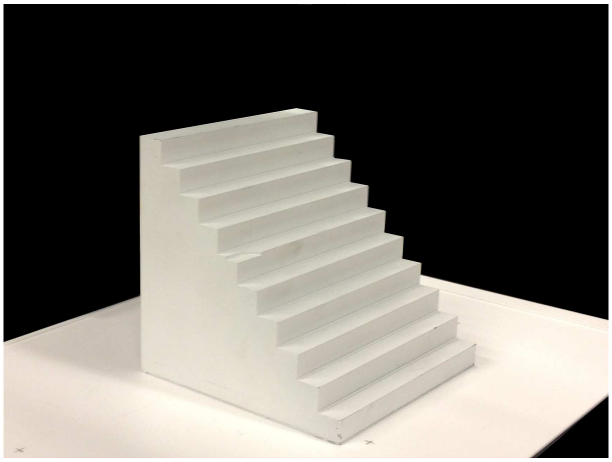


Fig. 3.2. Centimeter-step block

3.3 Verification of the Camera Calibration Results

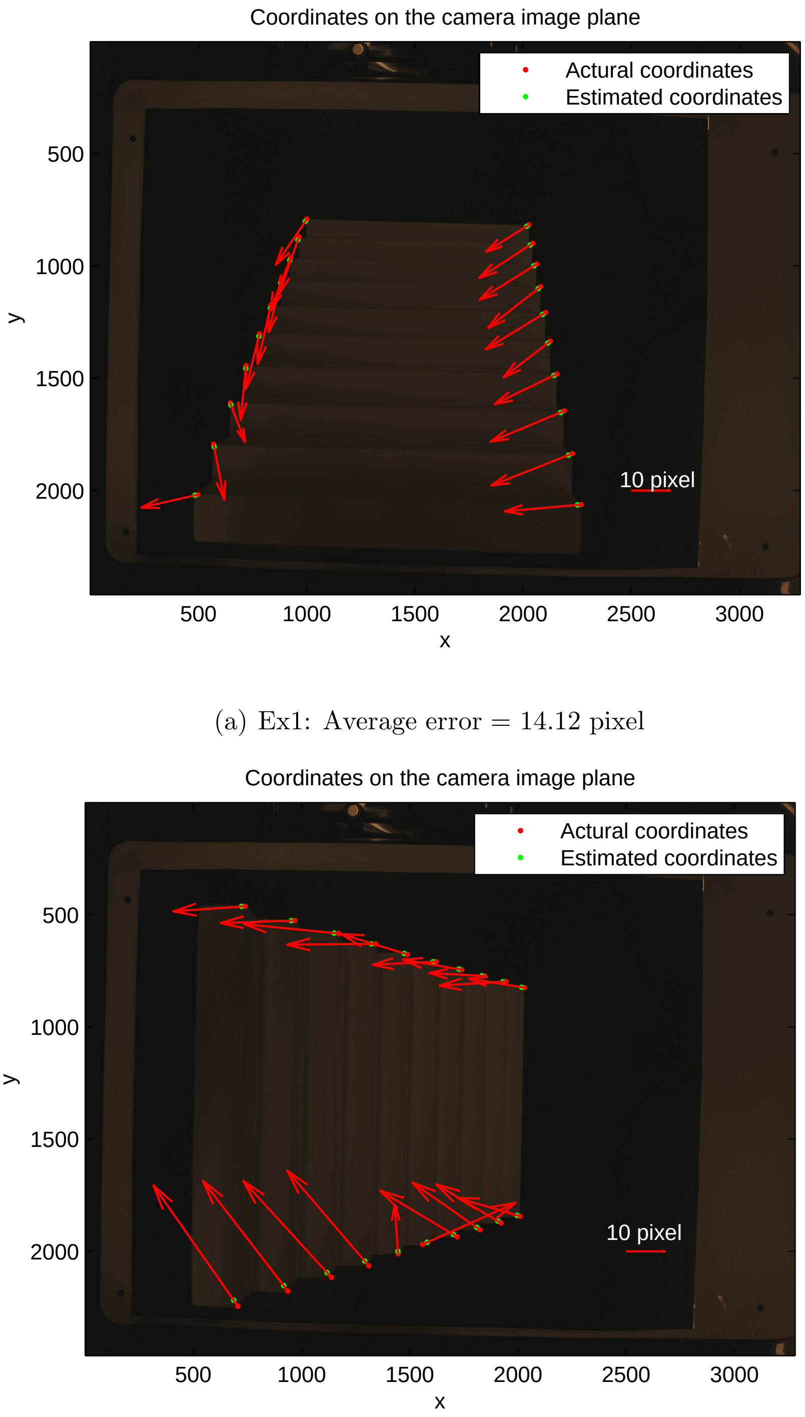
The object we use for the verification is a centimeter-step block (Fig. 3.2), which is 10cm ×10cm×10cm, each step size is 1cm. The verification procedure is designed as following:

1. Place the centimeter-step block on the platen, and use the camera embedded in Topshot to capture the image.

2. Measure the world coordinates of one corner point, and calculate the rest feature points' world coordinates.

3. Open the image in Matlab and record the camera image coordinates of the feature points.

4. Estimate the model-predicted image coordinates, and compare with the actual image coordinates. The results for two configurations of the block are shown in Fig. 3.3.



(a) Ex1: Average error = 14.12 pixel

(b) Ex2: Average error = 19.54 pixel

Fig. 3.3. Two examples of the verification results

4. PROJECTOR MODELING, CALIBRATION AND VERIFICATION

In this section, we introduce the commonly used camera and projector model and calibration method for the 3D capture system. In order to use triangulation to recover the 3D object, we first need to calibrate the camera and projectors. The goal of calibration is to get the intrinsic parameters (focal length, principal point, skew coefficient, distortions) and extrinsic parameters (rotation, translation matrix) of the camera and the projectors.

4.1 Projector Model

Projector can be modeled as an inverse camera. Therefore, we will have a similar relationship between the 3D point coordinates and the 2D point coordinates on the image plane.

Consider a point on the projector's 3D coordinates $P_p = \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix}$. P_p can be calculated from

$$P_p = \mathbf{R}_p \times P_c + \mathbf{T}_p \tag{4.1}$$

where $\mathbf{R}_p$ is the 3×3 rotation matrix and $\mathbf{T}_p$ is the 3×1 translation matrix which relates the camera coordinate system to the projector coordinate system. Then the 2D image point on the projector's normalized image plane is $p_{phn}^{p} = \begin{bmatrix} x_{phn}^{p} \\ y_{phn}^{p} \end{bmatrix} =$

$\begin{bmatrix} X_p/Z_p \\ Y_p/Z_p \end{bmatrix}$. By adding the radial and tangential lens distortion to p^p_{phn}, we get the distorted projection

$$p^p_{ldn} = \begin{bmatrix} x^p_{ldn} \\ y^p_{ldn} \end{bmatrix} = \begin{bmatrix} \varphi_1(x^p_{phn}, y^p_{phn}) \\ \varphi_2(x^p_{phn}, y^p_{phn}) \end{bmatrix} \tag{4.2}$$

where φ_1, φ_2 are defined in Eqn. 3.3

Then the coordinates on the projector's image plane considering pinhole camera model with lens distortion are:

$$\begin{bmatrix} x^p_{ld} \\ y^p_{ld} \\ 1 \end{bmatrix} = KK^p \begin{bmatrix} x^p_{ldn} \\ y^p_{ldn} \\ 1 \end{bmatrix} \tag{4.3}$$

where

$$KK^p = \begin{bmatrix} f^p_x & f^p_{s\theta} & u^p_o \\ 0 & f^p_y & v^p_o \\ 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

KK^p is called the projector's intrinsic parameter matrix. Here f^p_x and f^p_y are the projector's focal length in the x and y direction; u^p_o and v^p_o are the coordinates of the principal point on the projector's image plane; and $f^p_{s\theta}$ is the projector's skew coefficient.

4.2 Projector Calibration Procedure

The basic idea to calibrate a projector is to consider the projector as an inverse camera. As we mentioned in the Sec. 3.1, the function of camera is to capture the object points in 3D world coordinate and convert it to the 2D image plane; while the function of the projector is to project points on the image plane to the 3D world coordinates. Thus, after obtaining the corresponding 3D and 2D points, the calibration procedures are similar for the camera and projector. The detailed procedure is stated as follows [41]:

1. Calibrate the camera using the procedure we described in Section 2.1 and get the intrinsic and extrinsic parameters.

2. Attach a small portions of the checkerboard pattern to the platen.

3. Recover calibration plane in camera coordinate system

4. Project a checkerboard to the calibration board and detect corners

5. Apply ray-plane intersection to recover 3D positions for each projected corner

6. Calibrate the projector using the correspondences between the 2D points of the image that is projected and the 3D projected points.

After we finish the projector calibration, we will also get the intrinsic and extrinsic parameters for each projector. Use the calibration results we get, along with the correspondence between the projected image and the captured image from the coded pattern, we could recover the depth information of our object.

4.3 Verification of the Projector Calibration Results

We use the same centimeter-step block(Fig. 3.2) for the projector calibration verification as we use for the camera calibration verification. The procedure is designed as following:

1. Place the centimeter-step block on the platen, project a designed dotted pattern (Fig. 4.1) to it (Fig. 4.2(a)).

2. Measure the world coordinates of one dot on the block, and calculate the rest dots' world coordinates.

3. Estimate the model-predicted dots' coordinates on the projector's image plane, and compare with the actual digital image (Fig. 4.2(c)).

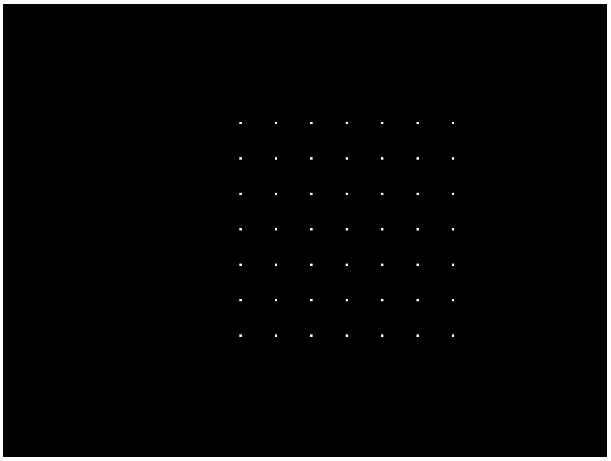


Fig. 4.1. Designed dot pattern for projection

4. Remove the centimeter-step block, and project the pattern directly on the platen (Fig. 4.2(b)). Repeat the above procedure. The results are shown in (Fig. 4.2(d)).

4.4 World Coordinates

4.4.1 World Coordinate Calculation

To calculate the world coordinates of a 3D point, we need to known the correspondence between the 2D point coordinate on camera and projector image plane. Suppose we know the coordinate on camera image plane is $\begin{bmatrix} x_{ld}^c \\ y_{ld}^c \end{bmatrix}$, use the camera calibration results and Eqn. 3.4, we can get $\begin{bmatrix} x_{ldn}^c \\ y_{ldn}^c \end{bmatrix}$. And by solving Eqn. 3.2, we can get the result for $\begin{bmatrix} x_{phn}^c \\ y_{phn}^c \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}$. Similarly, suppose the correspondence image point of $\begin{bmatrix} x^c_{ld} \\ y^c_{ld} \end{bmatrix}$ is $\begin{bmatrix} x^p_{ld} \\ y^p_{ld} \end{bmatrix}$, we can use the inverse of Eqn. 4.2 and 4.3 to solve $\begin{bmatrix} x^p_{phn} \\ y^p_{phn} \end{bmatrix} = \begin{bmatrix} X_p/Z_p \\ Y_p/Z_p \end{bmatrix}$

Last, we use the rotation and translation relationship (Eqn. 4.1) to get the world coordinates:

$$\begin{bmatrix} Z_p \\ Z_c \end{bmatrix} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{T}_p \tag{4.5}$$

where first column of $\mathbf{A}$ is $\begin{bmatrix} X_p/Z_p \\ Y_p/Z_p \\ 1 \end{bmatrix}$ and second column of $\mathbf{A}$ is $-\mathbf{R}_p \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ 1 \end{bmatrix}$.

Therefore, use the value of $\begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}$, Z_c and Eqn. 3.1, we get the result for $\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$.

4.4.2 Experiment and Results

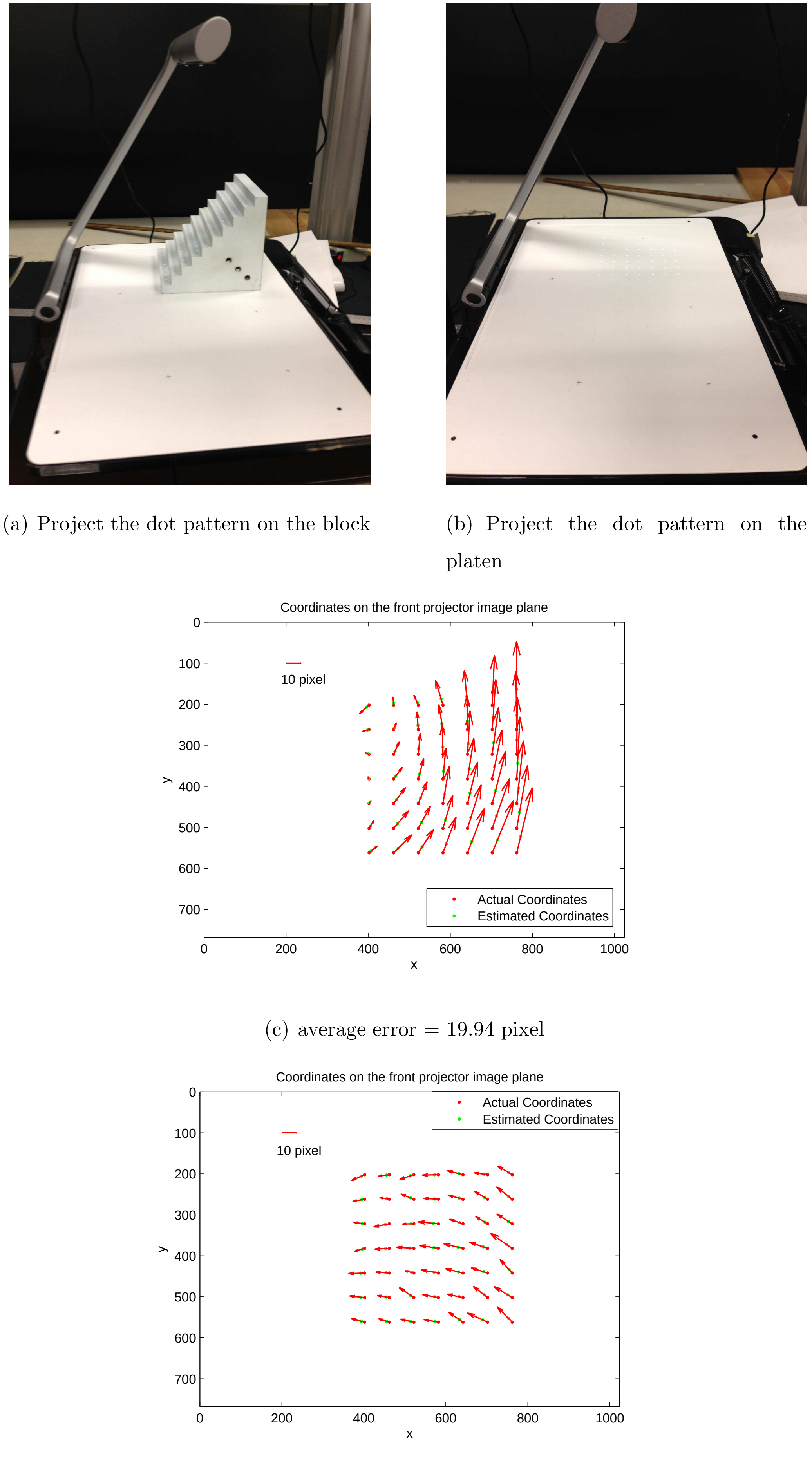
The experiment for world coordinates calculation and comparison is designed as following:

1. Place the centimeter-step block on the platen, project the same dot pattern (Fig. 4.1) as for projector verification to it (Fig. 4.2(a)).

2. Capture the block with the dot pattern on it using the camera embedded in Topshot (Fig. 4.3), and record the dots' coordinates on camera image plane (Fig. 4.4). Since we design the project dot pattern, the dots' coordinates on projector image plane is known (Fig. 4.5).

3. Measure the world coordinates of one dot on the block, and calculate the rest dots' world coordinates.

4. Estimate the model-predicted dots' world coordinates using the method we discussed in Sec. 4.4.1,along with the known correspondence of the dots on camera and projector image plane (Fig. 4.4 and Fig. 4.5). Compare the predicted value with the actual recorded world coordinates (Fig. 4.6). The differences in X_w, Y_w, Z_w and the overall Euclidean distance difference are plot in Fig. 4.7.



(a) Project the dot pattern on the block

(b) Project the dot pattern on the platen

(c) average error = 19.94 pixel

(d) average error = 9.54 pixel

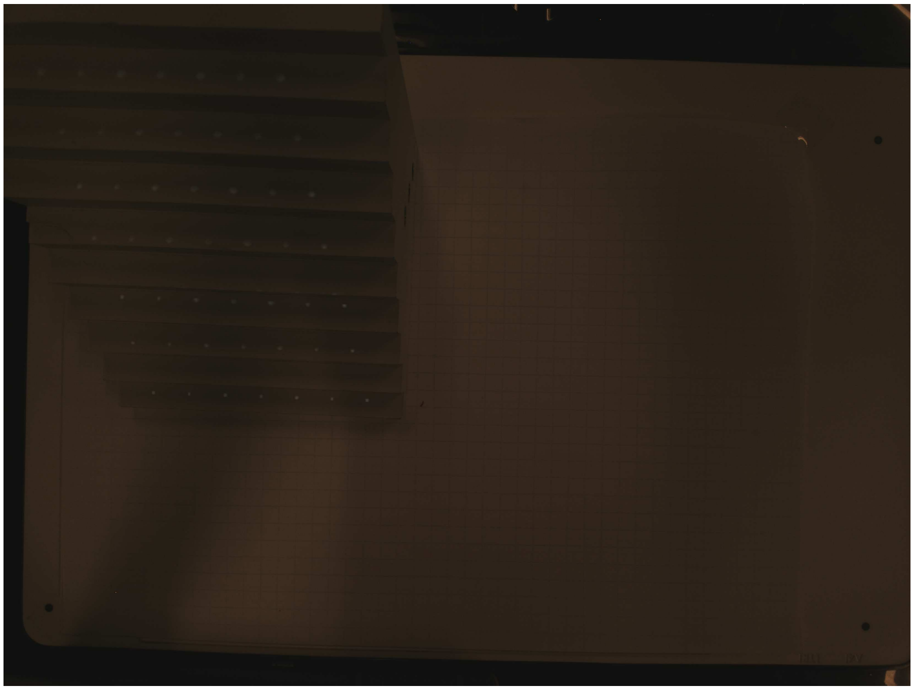


Fig. 4.3. Captured block with dot pattern projected on it

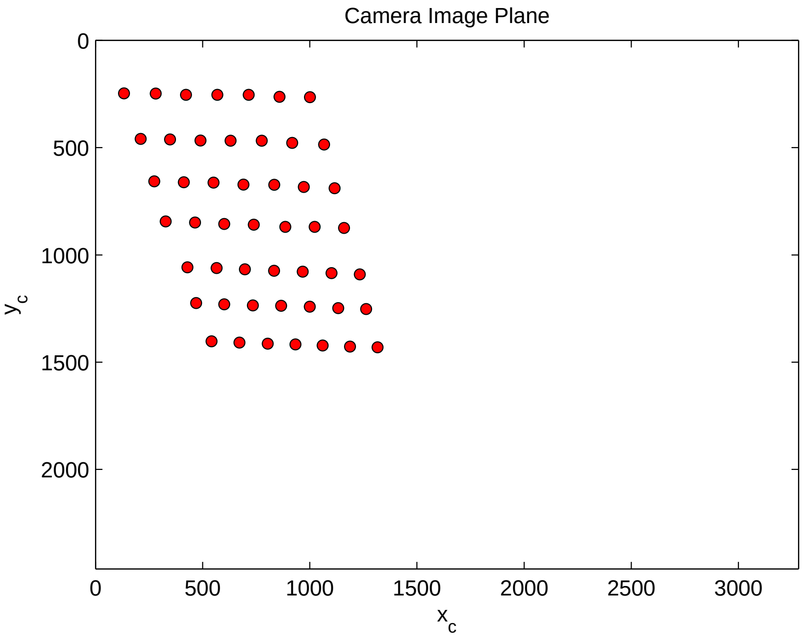


Fig. 4.4. Recorded dots' coordinates on camera image plane

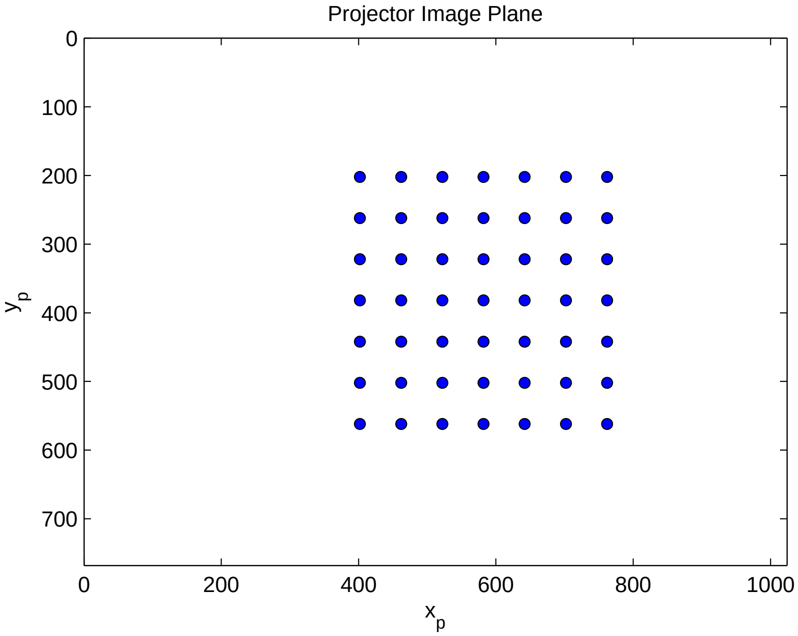


Fig. 4.5. Dots' coordinates on designed pattern is known

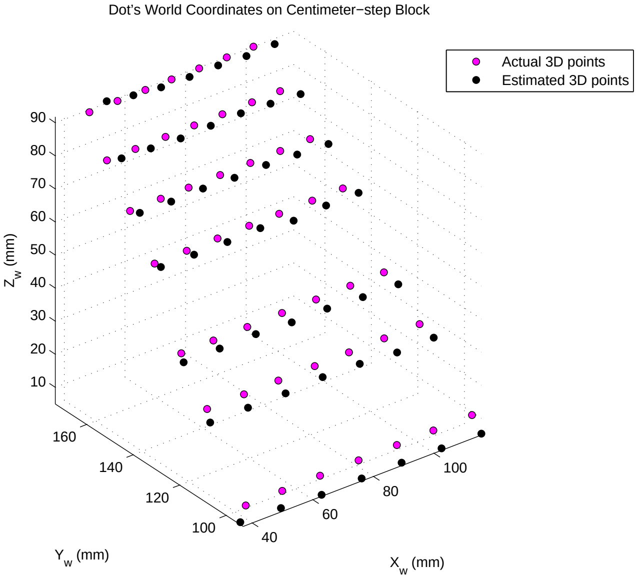


Fig. 4.6. Estimated and actual dots' world coordinates

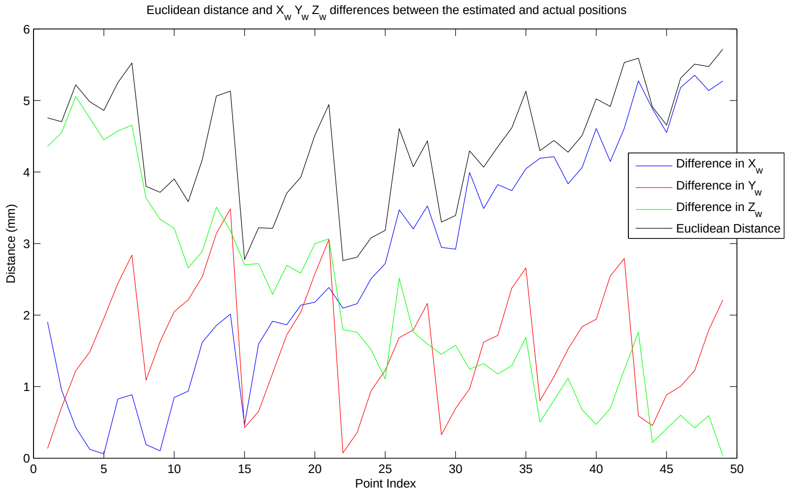


Fig. 4.7. Plot of difference in X_w, Y_w, Z_w

5. SENSITIVITY ANALYSIS FOR ESTABLISHING DESIGN TOLERANCES

The sensitivity analysis is to evaluate the amount of shifting on the camera image plane given a certain shifting on the projector image plane. In this section, we first give out the analytical sensitivity analysis, following by the experimental sensitivity measurement of the system, and last the comparison of the simulation and the experiment results.

5.1 Theoretical Sensitivity Analysis

Based on Eqn. 3.3 and 4.2, we can write the 2D point coordinates on the projector's normalized image plane considering lens distortion before shifting is:

$$p^p_{ldn} = \begin{bmatrix} x^p_{ldn} \\ y^p_{ldn} \end{bmatrix} = \begin{bmatrix} x^p_{phn} \\ y^p_{phn} \end{bmatrix} + T_d(X_p, Y_p, Z_p; \mathbf{K}^p) \tag{5.1}$$

where $T_d(X_p, Y_p, Z_p; \mathbf{K}^p)$ is the lens distortion before shifting and $\mathbf{K}^p = \begin{bmatrix} k^p_1, k^p_2, k^p_3, k^p_4, k^p_5 \end{bmatrix}$ is the projector's lens distortion coefficient vector.

Considering a small shift in the 3D coordinates, say $\begin{bmatrix} \Delta X_w \\ \Delta Y_w \\ \Delta Z_w \end{bmatrix}$ in the world's coordinate system. Then the shifting in other coordinates can be easily calculated using:

$$\begin{bmatrix} \Delta X_c \\ \Delta Y_c \\ \Delta Z_c \end{bmatrix} = \mathbf{R}_c \begin{bmatrix} \Delta X_w \\ \Delta Y_w \\ \Delta Z_w \end{bmatrix} \tag{5.2}$$

$$\begin{bmatrix} \Delta X_p \\ \Delta Y_p \\ \Delta Z_p \end{bmatrix} = \mathbf{R}_p \begin{bmatrix} \Delta X_c \\ \Delta Y_c \\ \Delta Z_c \end{bmatrix} \tag{5.3}$$

Therefore, the 2D coordinates on the projector's normalized image plane considering lens distortion after shifting is:

$$p^{p\prime}_{ldn} = \begin{bmatrix} x^{p\prime}_{ldn} \\ y^{p\prime}_{ldn} \end{bmatrix} = \begin{bmatrix} \varphi_1(x^{p\prime}_{phn}, y^{p\prime}_{phn}) \\ \varphi_2(x^{p\prime}_{phn}, y^{p\prime}_{phn}) \end{bmatrix} \tag{5.4}$$

where φ_1, φ_2 are defined in Eqn. 3.3 and $p^{p\prime}_{phn} = \begin{bmatrix} x^{p\prime}_{phn} \\ y^{p\prime}_{phn} \end{bmatrix} = \begin{bmatrix} \frac{X_p+\Delta X_p}{Z_p+\Delta Z_p} \\ \frac{Y_p+\Delta Y_p}{Z_p+\Delta Z_p} \end{bmatrix}$

Then we can write

$$p^{p\prime}_{ldn} = \begin{bmatrix} x^{p\prime}_{ldn} \\ y^{p\prime}_{ldn} \end{bmatrix} = \begin{bmatrix} x^{p\prime}_{phn} \\ y^{p\prime}_{phn} \end{bmatrix} + T^{\prime}_d(X_p, Y_p, Z_p; \Delta X_p, \Delta Y_p, \Delta Z_p; \mathbf{K}^p) \tag{5.5}$$

where $T^{\prime}_d(X_p, Y_p, Z_p; \Delta X_p, \Delta Y_p, \Delta Z_p; \mathbf{K}^p)$ is the lens distortion after shifting.

Ignore the change of lens distortion caused by shifting:

$$T^{\prime}_d(X_p, Y_p, Z_p; \Delta X_p, \Delta Y_p, \Delta Z_p; \mathbf{K}^p) \approx T_d(X_p, Y_p, Z_p; \mathbf{K}^p) \tag{5.6}$$

we get

$$p^{p\prime}_{ldn} \approx p^{p\prime}_{phn} + T_d(X_p, Y_p, Z_p; \mathbf{K}^p) \tag{5.7}$$

Then the shifting on the projector's image plane is:

$$\begin{aligned} \begin{bmatrix} \Delta x_p \\ \Delta y_p \end{bmatrix} &= \begin{bmatrix} f^p_x & f^p_{s\theta} \\ 0 & f^p_y \end{bmatrix} (p^{p\prime}_{ldn} - p^{p}_{ldn}) = \begin{bmatrix} f^p_x & f^p_{s\theta} \\ 0 & f^p_y \end{bmatrix} (p^{p\prime}_{phn} - p^{p}_{phn}) \\ &= \begin{bmatrix} f^p_x & f^p_{s\theta} \\ 0 & f^p_y \end{bmatrix} \left(\begin{bmatrix} \frac{X_p+\Delta X_p}{Z_p+\Delta Z_p} \\ \frac{Y_p+\Delta Y_p}{Z_p+\Delta Z_p} \end{bmatrix} - \begin{bmatrix} \frac{X_p}{Z_p} \\ \frac{Y_p}{Z_p} \end{bmatrix} \right) \end{aligned} \tag{5.8}$$

which can be write as:

$$f^p_x Z_p \Delta X_p + f^p_{s\theta} Z_p \Delta Y_p + (-f^p_x X_p + f^p_{s\theta} Y_p - Z_p \Delta x_p)\, \Delta Z_p = Z_p^2 \Delta x_p \tag{5.9}$$

$$f^p_y Z_p \Delta Y_p + \left(-f^p_y Y_p - Z_p \Delta y_p\right) \Delta Z_p = Z_p^2 \Delta y_p \tag{5.10}$$

Since the sensitivity analysis is not spatial invariant, assume we know position in 3D world coordinate system, i.e. $\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$ is given and we only consider the projection on the calibration board, i.e. $\Delta Z_w = 0$.

Then we have:

$$\begin{bmatrix} \Delta X_p \\ \Delta Y_p \\ \Delta Z_p \end{bmatrix} = \mathbf{R}_p\mathbf{R}_c \begin{bmatrix} \Delta X_w \\ \Delta Y_w \\ 0 \end{bmatrix} \tag{5.11}$$

which can be write as:

$$(\mathbf{R}_p\mathbf{R}_c)^{-1}_{31}\Delta X_p + (\mathbf{R}_p\mathbf{R}_c)^{-1}_{32}\Delta Y_p + (\mathbf{R}_p\mathbf{R}_c)^{-1}_{33}\Delta Z_p = 0 \tag{5.12}$$

where $(\mathbf{R}_p\mathbf{R}_c)^{-1}_{ij}$ is the i^{th}row j^{th}column element of the inverse matrix of $\mathbf{R}_p\mathbf{R}_c$.

Write Eqn. 5.9, 5.10 and 5.12 in the matrix form to solve $\begin{bmatrix} \Delta X_p \\ \Delta Y_p \\ \Delta Z_p \end{bmatrix}$, we get:

$$\begin{bmatrix} \Delta X_p \\ \Delta Y_p \\ \Delta Z_p \end{bmatrix} = \begin{bmatrix} f^p_x Z_p & f^p_{s\theta} Z_p & -f^p_x X_p + f^p_{s\theta} Y_p - Z_p \Delta x_p \\ 0 & f^p_y Z_p & -f^p_y Y_p - Z_p \Delta y_p \\ (\mathbf{R}_p\mathbf{R}_c)^{-1}_{31} & (\mathbf{R}_p\mathbf{R}_c)^{-1}_{32} & (\mathbf{R}_p\mathbf{R}_c)^{-1}_{33} \end{bmatrix}^{-1} \begin{bmatrix} Z_p^2 \Delta x_p \\ Z_p^2 \Delta y_p \\ 0 \end{bmatrix} \tag{5.13}$$

Thus, the shifting on the camera's image plane is:

$$\begin{aligned} \begin{bmatrix} \Delta x_c \\ \Delta y_c \end{bmatrix} &= \begin{bmatrix} f^c_x & f^c_{s\theta} \\ 0 & f^c_y \end{bmatrix} (p^{c\prime}_{ldn} - p^c_{ldn}) = \begin{bmatrix} f^c_x & f^c_{s\theta} \\ 0 & f^c_y \end{bmatrix} (p^{c\prime}_{phn} - p^c_{phn}) \\ &= \begin{bmatrix} f^c_x & f^c_{s\theta} \\ 0 & f^c_y \end{bmatrix} \left(\begin{bmatrix} \frac{X_c+\Delta X_c}{Z_c+\Delta Z_c} \\ \frac{Y_c+\Delta Y_c}{Z_c+\Delta Z_c} \end{bmatrix} - \begin{bmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \end{bmatrix} \right) \end{aligned} \tag{5.14}$$

where

$$\begin{aligned}
\begin{bmatrix} \Delta X_c \\ \Delta Y_c \\ \Delta Z_c \end{bmatrix} &= \mathbf{R}_p^{-1} \begin{bmatrix} \Delta X_p \\ \Delta Y_p \\ \Delta Z_p \end{bmatrix} \\
&= \mathbf{R}_p^{-1} \begin{bmatrix} f_x^p Z_p & f_{s\theta}^p Z_p & -f_x^p X_p + f_{s\theta}^p Z_p - Z_p \Delta x_p \\ 0 & f_y^p Z_p & -f_y^p Y_p - Z_p \Delta y_p \\ (\mathbf{R}_p \mathbf{R}_c)_{31}^{-1} & (\mathbf{R}_p \mathbf{R}_c)_{32}^{-1} & (\mathbf{R}_p \mathbf{R}_c)_{33}^{-1} \end{bmatrix}^{-1} \begin{bmatrix} Z_p^2 \Delta x_p \\ Z_p^2 \Delta y_p \\ 0 \end{bmatrix}
\end{aligned} \tag{5.15}$$

and $\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$ can be calculated from Eqn. 3.1.

5.2 Experimental Sensitivity Analysis

Our goal of the experimental sensitivity analysis is to evaluate the shifting on camera image plane $\Delta x_c, \Delta y_c$ caused by 1 pixel shifting on projector image plane($\Delta x_p = 1, \Delta y_p = 1$). The experimental procedure is designed as follows:

1. Design a checkerboard pattern. Our checkerboard pattern has size 64 pixel ×64 pixel for each square.

2. Project the designed checkerboard pattern to the platen using one projector and using the camera to capture the projected pattern.

3. Detect the corners of the checkerboard on the captured image using Harris corner detection and calculate the distance of each pair of adjacent corners ΔD_{64}. Fig. 5.1 shows the detected corners when projecting using the front projector.

4. Since each adjacent corners are 64 pixels apart on the projector image plane, simply divide ΔD_{64} by 64, we will get the shifting on camera's image plane caused by 1 pixel shifting on projector image plane.

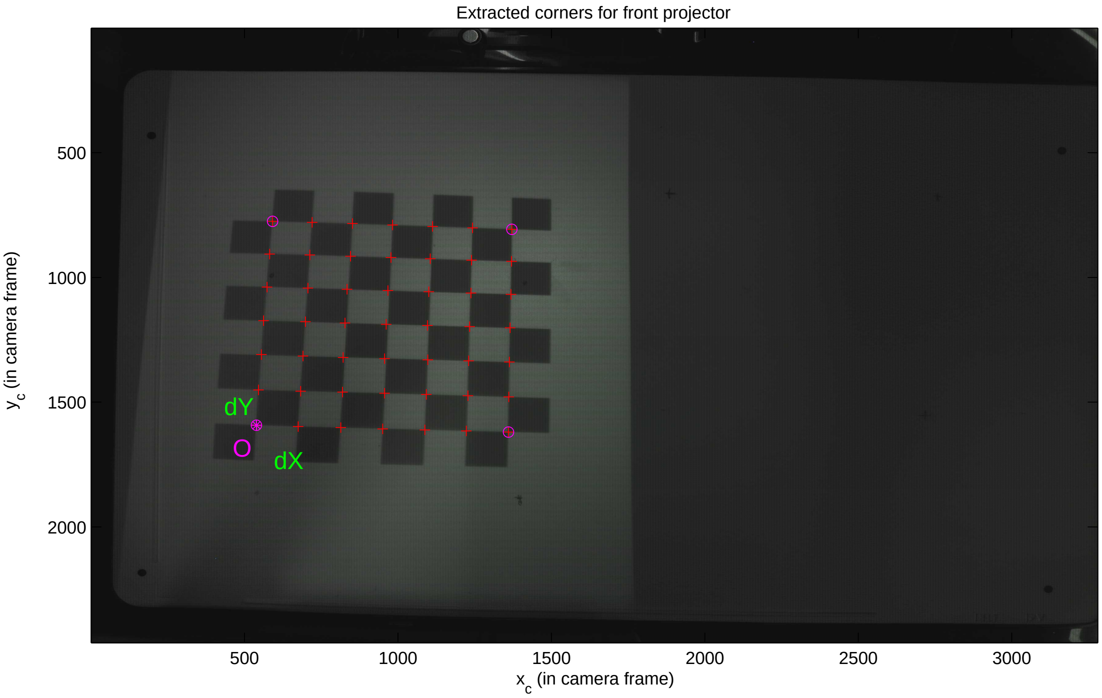
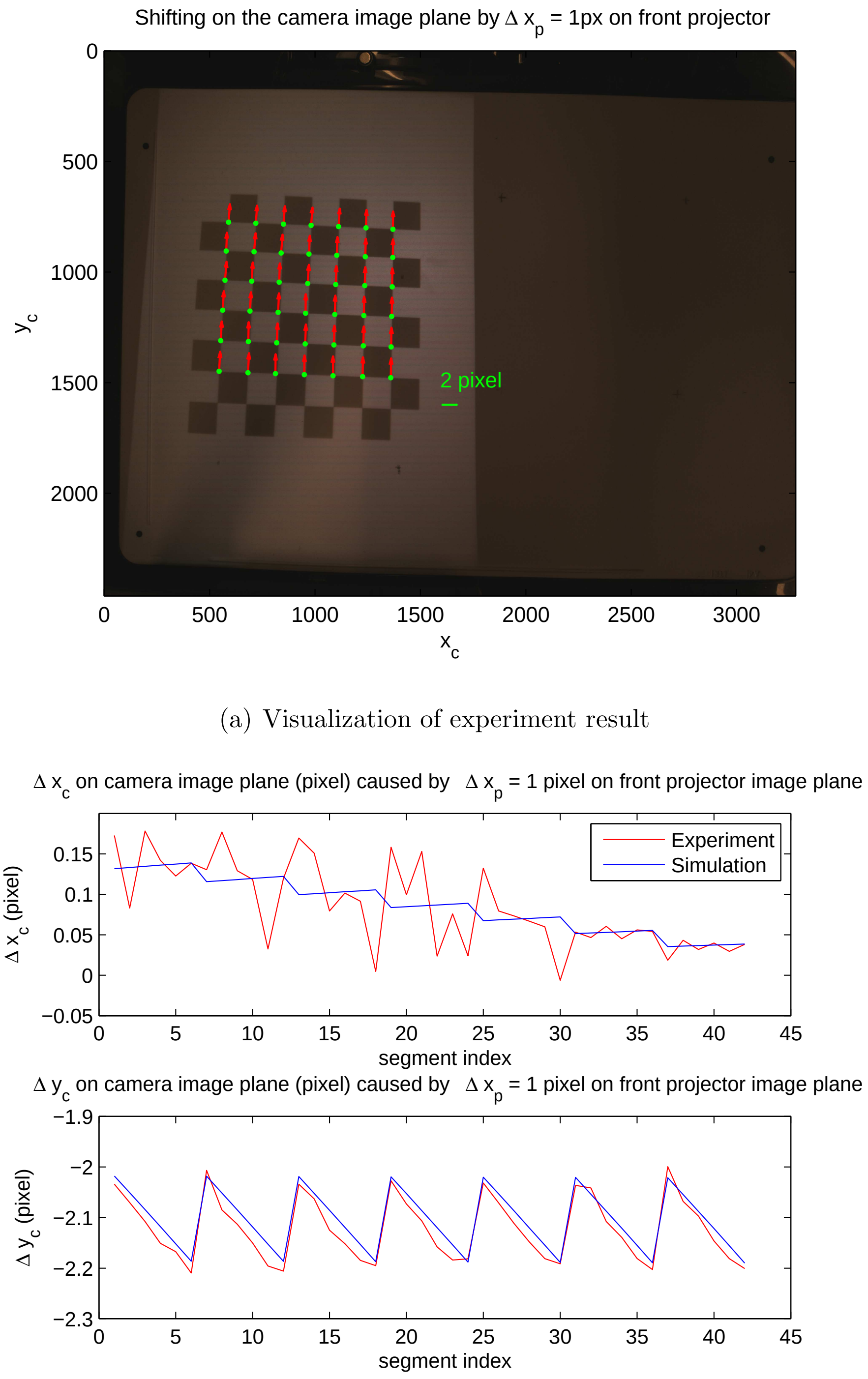


Fig. 5.1. Harris corner detected checkerboard corners used for sensitivity analysis

5. Repeat for the other projector.

5.3 Comparison of Simulation and Experiment Results

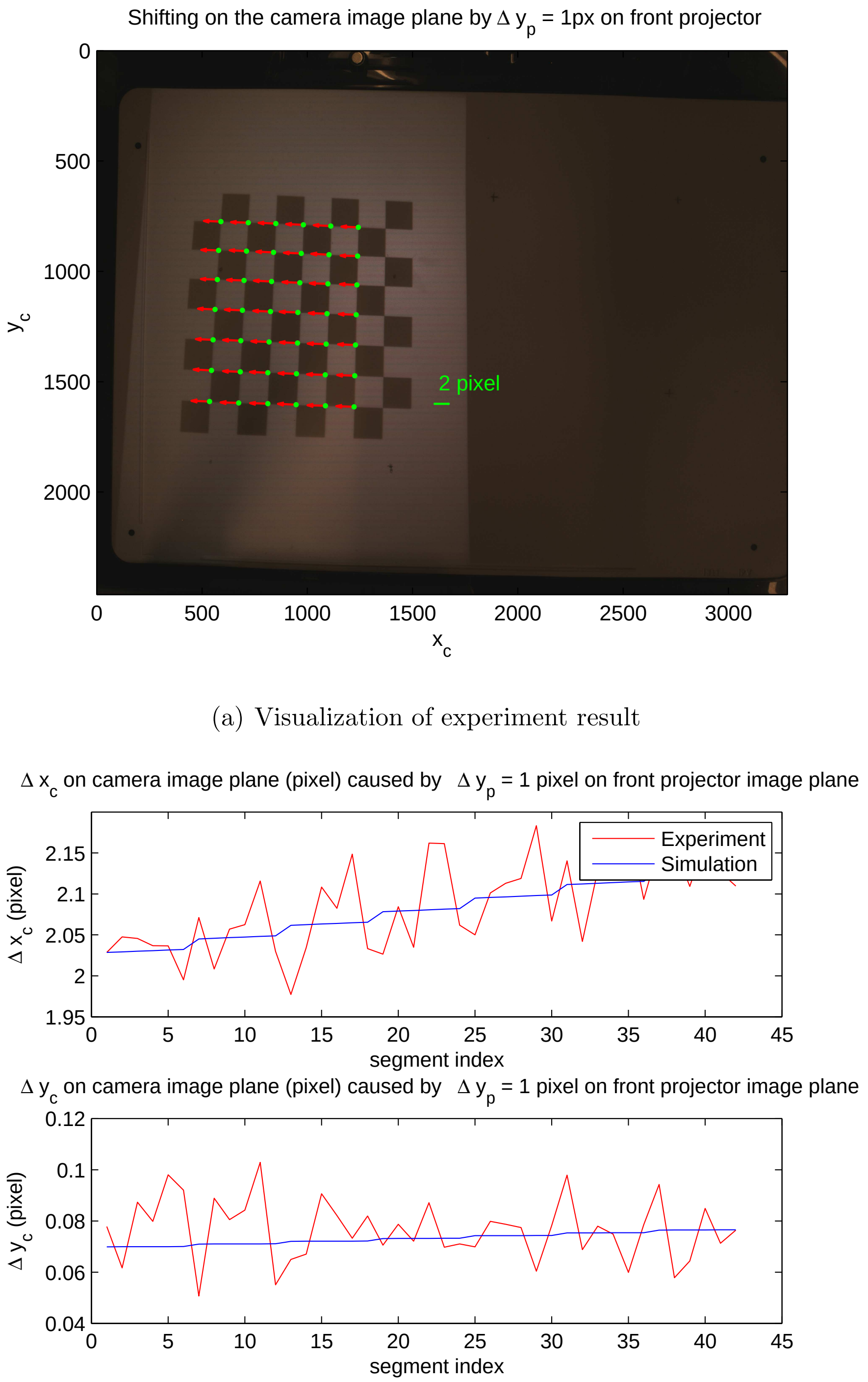
Our simulation result is based on differentials and small signal theory which we discuss in Sec. 5.1. We give out an analytical representation of $\Delta x_c, \Delta y_c$ in terms of $\Delta x_p, \Delta y_p$ in Eqn. 5.14. For the experiment result, we follow the procedure we design in Sec. 5.2. And Fig. 5.2 and Fig. 5.3 show the visualization of the experiment result on camera image plane and a comparison of the simulation and experiment result.



(a) Visualization of experiment result

(b) Comparison of simulation and experiment result

Fig. 5.2. Sensitivity analysis comparison for front projector, $\Delta x_p = 1$ pixel



(a) Visualization of experiment result

(b) Comparison of simulation and experiment result

Fig. 5.3. Sensitivity analysis comparison for front projector, $\Delta y_p = 1$ pixel

5.4 Design Tolerance

During the design process of the 3D capture system, not only do we want to know the resolution we can get from the camera image in terms of projector's resolution, we also want to know the system tolerance with respect to the movement of the camera and projector.

5.4.1 Camera Tolerance

Camera tolerance is to determine the amount of shifting a camera can tolerant without shifting too much on the image plane, for a fixed point in space. Suppose in the camera coordinate system, the camera's center is at the origin $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$. Considering a fixed 3D point in space, the coordinates are $\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$. If we shift the camera center by $\begin{bmatrix} \Delta X_c \\ \Delta Y_c \\ \Delta Z_c \end{bmatrix}$, then the 3D point's coordinates become $\begin{bmatrix} X_c - \Delta X_c \\ Y_c - \Delta Y_c \\ Z_c - \Delta Z_c \end{bmatrix}$. Since the arm of the Topshot, where the camera is embedded, can only be folded up or down, we assume $\Delta X_c = 0$.

Using the Eqn. 5.14, and considering the tolerance on the camera's image plane is within 1 pixel, we have:

$$\begin{bmatrix} \Delta x_c \\ \Delta y_c \end{bmatrix} = \begin{bmatrix} f_x^c & f_{s\theta}^c \\ 0 & f_y^c \end{bmatrix} \begin{bmatrix} \frac{X_c - \Delta X_c}{Z_c - \Delta Z_c} - \frac{X_c}{Z_c} \\ \frac{Y_c - \Delta Y_c}{Z_c - \Delta Z_c} - \frac{Y_c}{Z_c} \end{bmatrix} < \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \tag{5.16}$$

Solve Eqn. 5.16, we get the camera position tolerance is:

$$\Delta X_c = 0$$

$$\begin{bmatrix} \Delta Y_c \\ \Delta Z_c \end{bmatrix} < \begin{bmatrix} -f^c_{s\theta} Z_c & f^c_x X_c + f^c_{s\theta} Y_c + \frac{1}{\sqrt{2}} Z_c \\ -f^c_y Z_c & f^c_y Y_c + \frac{1}{\sqrt{2}} Z_c \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{\sqrt{2}} Z^2_c \\ \frac{1}{\sqrt{2}} Z^2_c \end{bmatrix} \tag{5.17}$$

5.4.2 Projector Tolerance

Projector tolerance is similar to camera tolerance, which is to determine the amount of shifting a projector can tolerant without shifting too much on the image plane, for a fixed point in space. Suppose in the projector coordinate system, the projector's center is at the origin $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$. Considering a fixed 3D point in space, the coordinates are $\begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix}$. If we shift the projector center by $\begin{bmatrix} \Delta X_p \\ \Delta Y_p \\ \Delta Z_p \end{bmatrix}$, then the 3D point's coordinates become $\begin{bmatrix} X_p - \Delta X_p \\ Y_p - \Delta Y_p \\ Z_p - \Delta Z_p \end{bmatrix}$. Considering our fixture which hold the project can only rotate in the $Y_c - Z_c$ plane, we assume $\Delta X_p = 0$.

Using the Eqn. 5.8, and considering the tolerance on the projector's image plane is within 1 pixel, we have:

$$\begin{bmatrix} \Delta x_p \\ \Delta y_p \end{bmatrix} = \begin{bmatrix} f^p_x & f^p_{s\theta} \\ 0 & f^p_y \end{bmatrix} \begin{bmatrix} \frac{X_p - \Delta X_p}{Z_p - \Delta Z_p} - \frac{X_p}{Z_p} \\ \frac{Y_p - \Delta Y_p}{Z_p - \Delta Z_p} - \frac{Y_p}{Z_p} \end{bmatrix} < \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \tag{5.18}$$

Solve Eqn. 5.18, we get the projector tolerance is:

$$\Delta X_p = 0$$

$$\begin{bmatrix} \Delta Y_p \\ \Delta Z_p \end{bmatrix} < \begin{bmatrix} -f^p_{s\theta} Z_p & f^p_x X_p + f^p_{s\theta} Y_p + \frac{1}{\sqrt{2}} Z_p \\ -f^p_y Z_p & f^p_y Y_p + \frac{1}{\sqrt{2}} Z_p \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{\sqrt{2}} Z^2_p \\ \frac{1}{\sqrt{2}} Z^2_p \end{bmatrix} \tag{5.19}$$

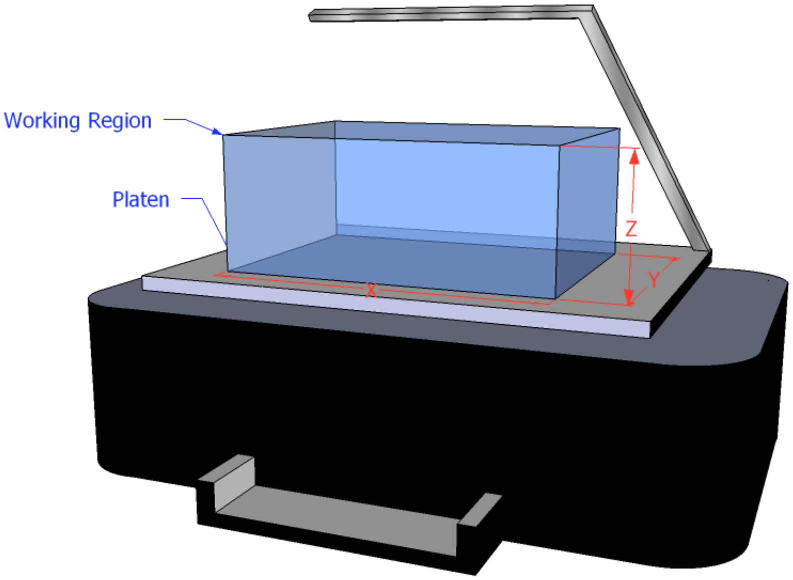


Fig. 5.4. The working region of the dual-projector system, $X = 240mm$, $Y = 160mm$, $Z = 60mm$

5.4.3 Simulation Results for Camera and Projector Tolerance

Considering the working region of the dual-projector system is a rectangular parallelepiped. The dimension is $X = 240mm$, $Y = 160mm$, $Z = 60mm$, as shown in Fig. 5.4.

To simulation the camera tolerance, considering inside the working region, sample one point in every 25mm in X,Y,Z direction. Then use Eqn. 5.17 to calculate the tolerated shifting camera could bear for each sample point. The shifting is converted from camera coordinates to world coordinates using Eqn.5.2 and plotted in Fig. 5.5. As we can see from the plot, camera tolerance is larger for the points close to camera optic axis, and decreases for points away from camera optic axis.

Similarly, to simulation the front projector tolerance, we only consider half of the working region,since one projector only covers half of the platen. Then, we sample one

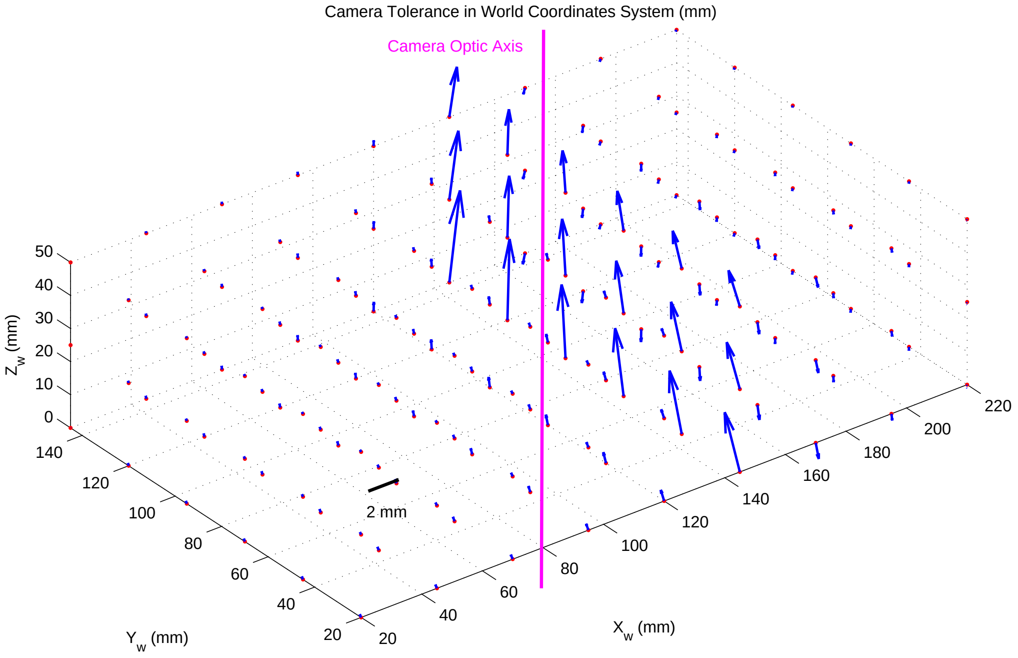


Fig. 5.5. Simulation of the camera tolerance results. The blue arrow indicates the amount and direction the camera could move, so that the shift on camera image plane for that specific 3D point is less than 1 pixel

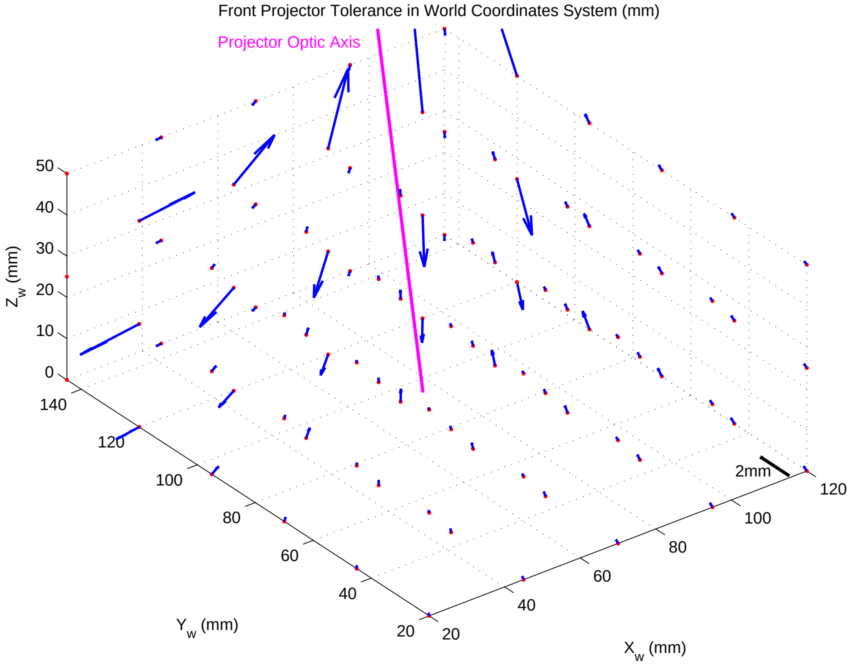


Fig. 5.6. Simulation of the projector tolerance results. The blue arrow indicates the amount and direction the projector could move, so that the shift on projector image plane for that specific 3D point is less than 1 pixel

point in every 25mm in X,Y,Z direction and use Eqn. 5.19 to calculate the tolerated shifting projector could bear for each sample point. The shifting is converted from projector coordinates to world coordinates using Eqn.5.2, 5.3 and plotted in Fig. 5.6. As we can see from the plot, projector tolerance is larger for the points close to projector optic axis, and decreases for points away from projector optic axis.

6. CHANGE PERSPECTIVE FOR AN IMAGE USING 3D DEPTH MAP

In this chapter, we discuss the change of perspective for an image using 3D depth map our structured light system generates.

6.1 Perspective Change

Perspective is caused by different object-lens distance. It includes wide-angle distortion and telephoto distortion. The former one is also called the extension distortion, which is caused by the closeness to object and result in the extension of distance. That means closer object appears abnormally large, and distant object appears abnormally small. The latter one is also called compression distortion, which is due to the largeness of the object-lens distance and result in the compressing of distance. In this situation, closer object appears abnormally small, and distant object appears abnormally large. There is a common misunderstanding that the perspective of an object is changed by focal length of the camera. Fig. 6.1 shows an example of using different focal lengths, same object-lens distance, we can crop the image to have exactly the same view, with different depth of views. And Fig. 6.2 illustrates that it is the object-lens distance changes the perspective of an object , not focal length.

Perspective can be controlled by perspective control lens, or corrected in post-processing. It requires 3D information to change the perspective.

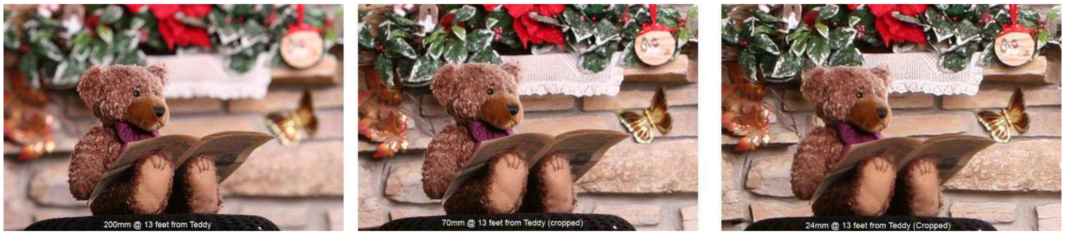

200mm @ 13 feet from Teddy
70mm @ 13 feet from Teddy (cropped)
24mm @ 13 feet from Teddy (Cropped)

Fig. 6.1. An example of different focal lengths, same object-lens distance maintains the same view (image from web)

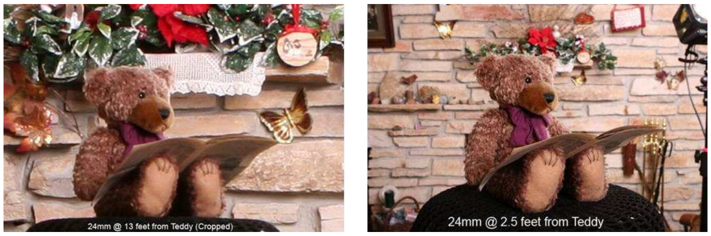

24mm @ 13 feet from Teddy (Cropped)
24mm @ 2.5 feet from Teddy

Fig. 6.2. An example of same focal length, different object-lens distances change the views

6.2 The Mathematics of Lens

The lens equation establishes the relationship between the focal length f_c, the object-lens distance d_o and the image-lens distance d_i, which is expressed as:

$$\frac{1}{f_c} = \frac{1}{d_o} + \frac{1}{d_i} \tag{6.1}$$

As we discussed in Sec. 6.1, in order to change the perspective of an object, we need to change the object-lens distance d_o. As we increase d_o, since f_c is fixed, d_i decreases. This means the object becomes smaller and flatter, which would change the perspective.

Fig. 6.3 illustrates this principle. The red star on the left-hand side of the lens represents the original object. The red star on the right-hand side of the lens is the image of the object. If we increase d_o to d'_o,the image of the object appears smaller and flatter on the image plane, which is shown on the lower-right corner, where x_i, y_i are the original image coordinates and x'_i, y'_i are the reconstructed image coordinates of one particular point.

6.3 Perspective Change

The flowchart of perspective change is showed in Fig. 6.4. First, we need to capture our object use focal length f_c and object-lens distance d_o. d_o varies by positions on our object, which is called a depth map. The 3D depth map is generated by our 3D capture system prototype. The depth map has 36×27 points. We need to interpolate these points to all the pixels of the image, which are 1600×1200 points using the Shepard Interpolation [47]. And we also increase d_o to d'_o by a certain ratio. Therefore, by solving the equations

$$\frac{1}{f_c} = \frac{1}{d_o} + \frac{1}{d_i} \tag{6.2}$$

$$\frac{1}{f_c} = \frac{1}{d'_o} + \frac{1}{d'_i} \tag{6.3}$$

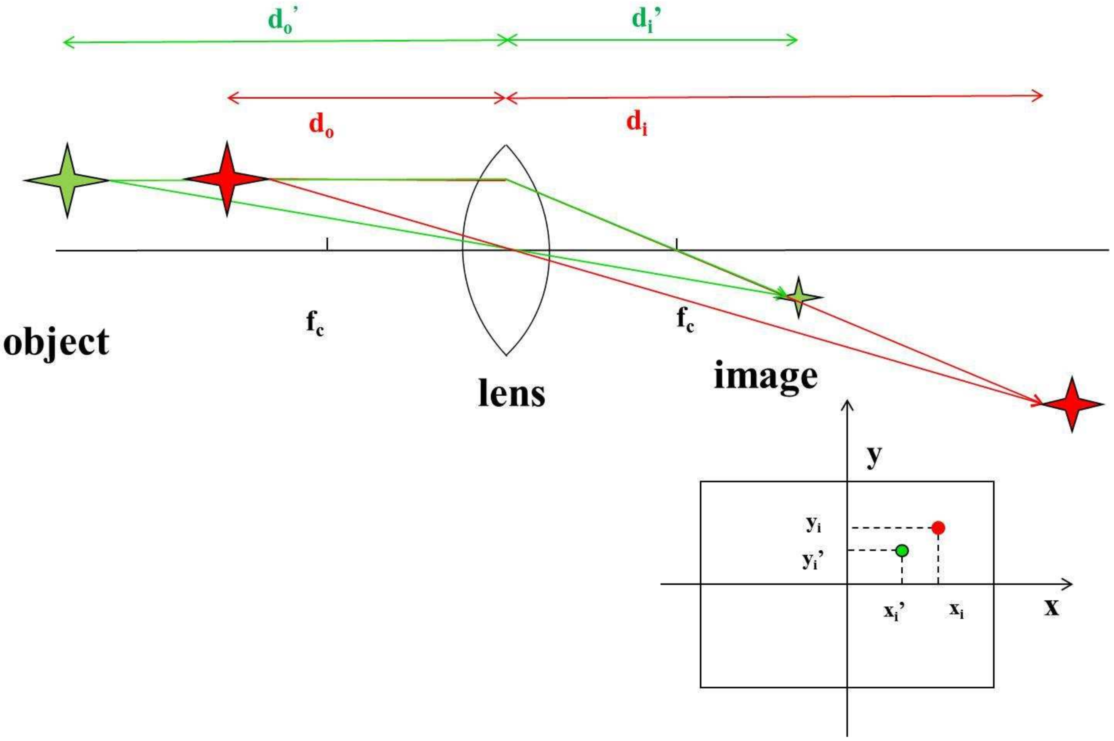


Fig. 6.3. Lens optics

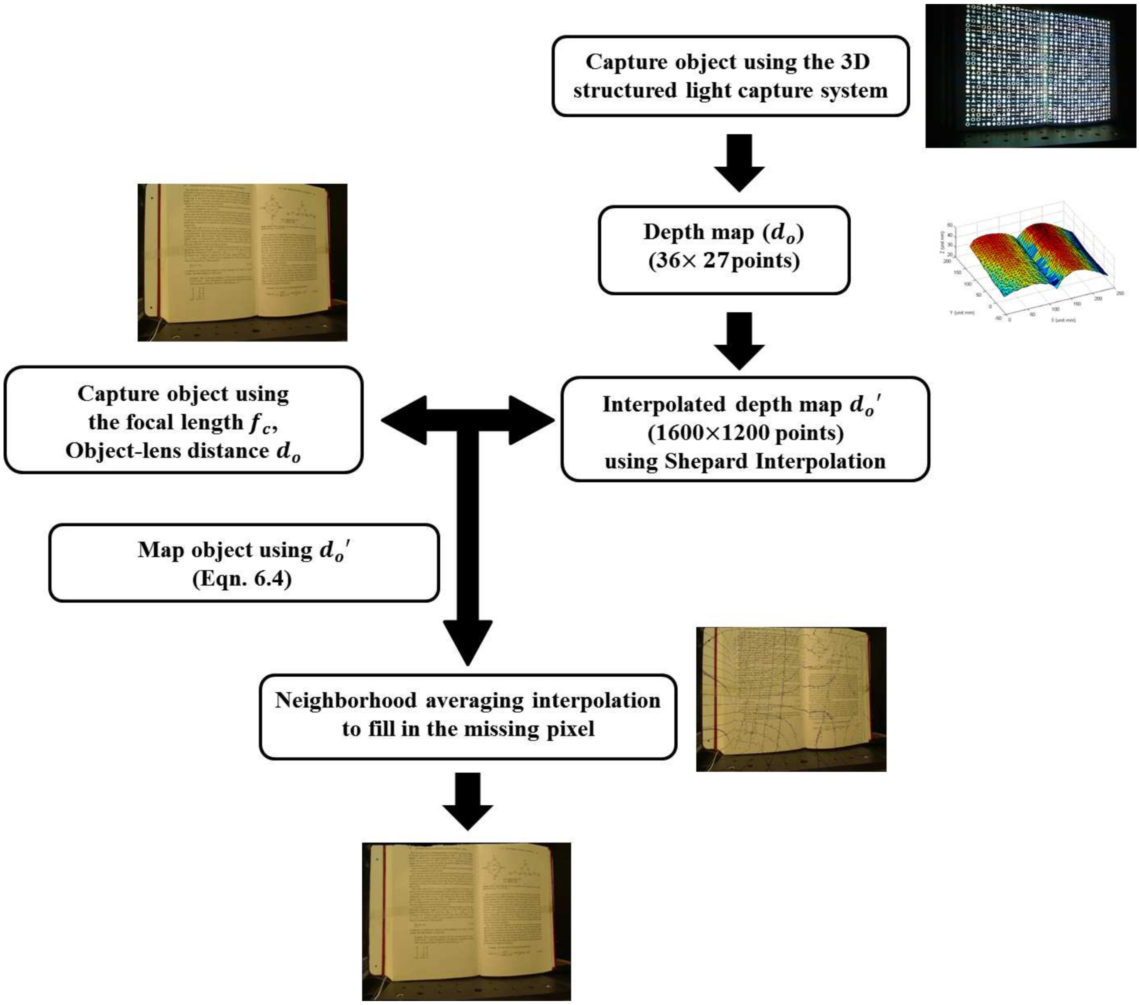

Capture object using the 3D structured light capture system
Depth map (d_o) (36× 27points)
Interpolated depth map $d_o{}'$ (1600×1200 points) using Shepard Interpolation
Capture object using the focal length f_c, Object-lens distance d_o
Map object using $d_o{}'$ (Eqn. 6.4)
Neighborhood averaging interpolation to fill in the missing pixel

Fig. 6.4. Flowchart for perspective change using 3D depth map

we get

$$\frac{d_i'}{d_i} = \frac{y_i'}{y_i} = \frac{x_i'}{x_i} = \frac{d_o'(d_o - f_c)}{d_o(d_o' - f_c)}$$

(6.4)

Now, we can calculate the output pixel locations using Eqn. 6.4 for a given pixel on the input image. This is an input-driven mapping, which will cause missing pixel on the output image. We use local neighborhood averaging to fill in the missing pixels and generate the final output image, in which the perspective has been changed.

Fig. 6.5 is a plot of Eqn. 6.4. We can conclude from the plot that, first, for a given focal length, increase the lens-object distance changes the perspective , especially from d_o to $2d_o$. Second, as focal length increases, the change mentioned above becomes more dramatically. For this reason, when we remapping the image, we use a larger f_c as well, since the original $f_c \approx 3$mm, which is too small that even large change in d_o fails to show change in perspective views. Our results are shown in Fig. 6.6.

6.4 Shepard Interpolation

In this section, we introduce the Shepard Interpolation [47], which is also known as inverse distance weighted interpolation we used in Sec. 6.3.

$$f(x, y) = \sum_{i=1}^{N} w_i f(x_i, y_i) \tag{6.5}$$

$$w_i = \begin{cases} \frac{D_i^{-2}}{\sum_{j=1}^{N} D_j^{-2}} & \text{if } D_i \neq 0 \\ 1 & \text{if } D_i = 0 \end{cases} \tag{6.6}$$

where $f(x, y)$ is the value to be interpolated, $f(x_i, y_i)$ is the value at known data point (x_i, y_i), and $D_i^{-2} = \frac{1}{|(x-x_i)^2+(y-y_i)^2|}$

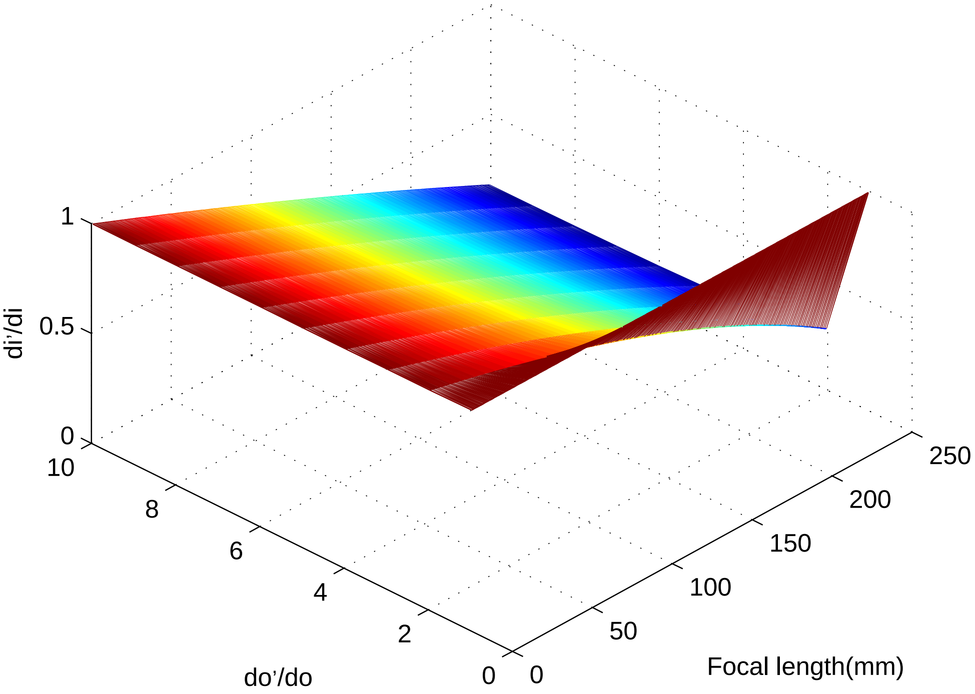
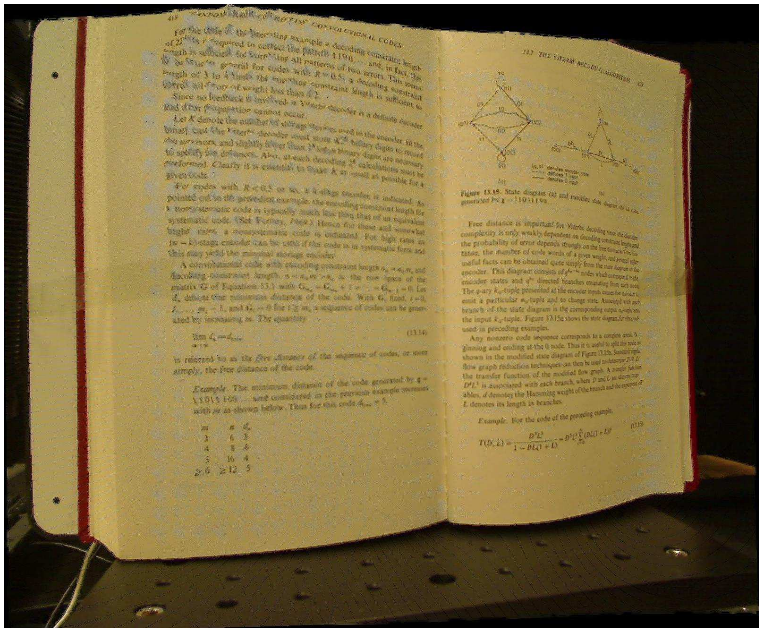
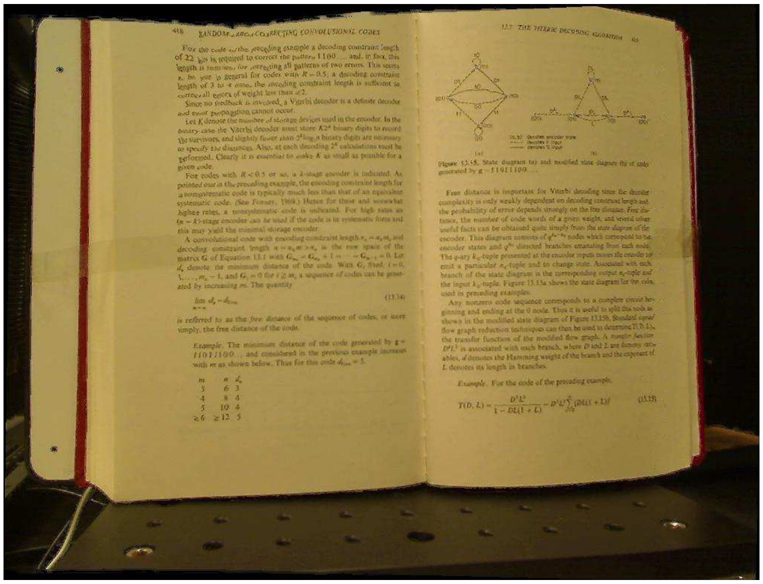


Fig. 6.5. Plot of Eqn. 6.4, $d_o \approx 300$mm



(a) Original image $f_c = 100$mm, object-lens distance d_o



(b) Corrected image $f_c = 100$mm, change object-lens distance to $d_o' = 2d_o$

Fig. 6.6. An example of perspective change

The shortcoming of Shepard Interpolation is when the number of data points is large, the calculation of $f(x, y)$ becomes proportionately longer.

7. LENS DISTORTION AND PERSPECTIVE DISTORTION CORRECTION FOR IMAGE

The goal of this project is to correct the lens distortion and the perspective distortion that an iPhone5 camera may introduce. The corrected grid pattern image will be used as an orthophoto to measure the size of the coffee grinds.

7.1 Blue Grid Pattern

The blue grid pattern (Fig. 7.1) is designed to be a calibration target both for the calibration process and the distortion correction process. We choose the color blue because we can easily extract it from the brownish coffee grind particles. Instead of using solid blue squares, which may experience more glaring when imaging, only use squares with blue outlines.

7.2 Lens Distortion Correction

The captured image has noticeable lens distortion (mainly radial distortion, may include Barrel, Pincushion, or a mixture of these two). Straight lines are not captured as straight lines. This is caused by the failure of a lens to be rectilinear. See Fig. 7.2. First we calibrate camera using our blue grid pattern (Fig. 7.1) and Zhang's camera calibration method [45]. The calibration result will include the lens distortion parameters k_1 and k_2. Barrel distortion typically will have a positive term for k_1 and pincushion distortion will have a negative value.

Let $(x_{correct}, y_{correct})$ be the correct location without radial distortion:

$$x_{distorted} = x_{correct} + (x_{correct} - u_0)(k_1 r^2 + k_2 r^4) \tag{7.1}$$

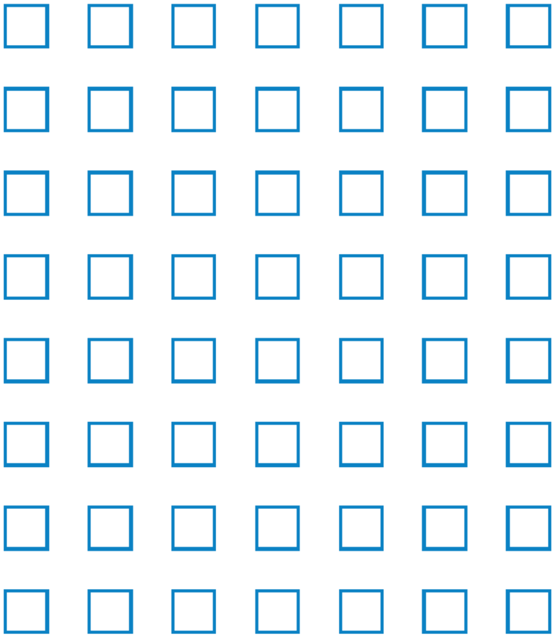


Fig. 7.1. Blue grid pattern used for calibration and orthophoto generation

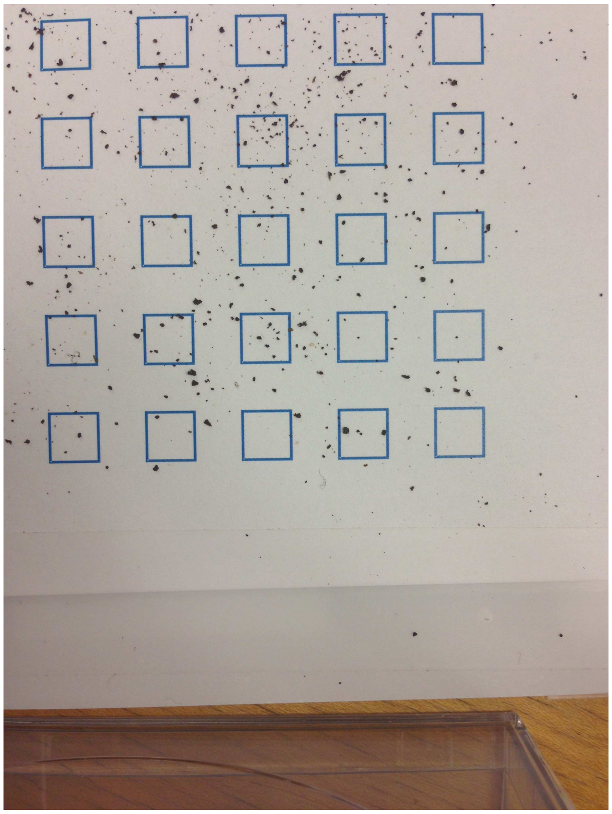


Fig. 7.2. Example of an original image suffers from lens distortion and perspective distortion

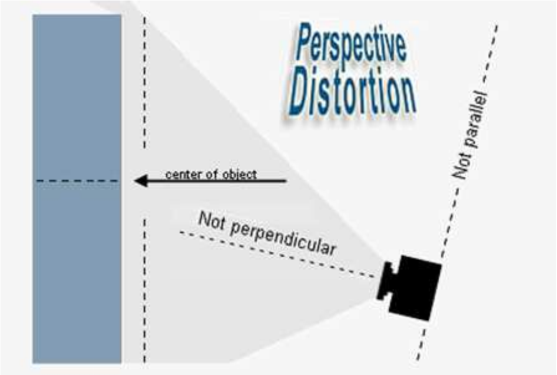


Fig. 7.3. Illustration of perspective distortion (image from web)

$$y_{distorted} = y_{correct} + (y_{correct} - v_0)(k_1 r^2 + k_2 r^4) \tag{7.2}$$

Where $r^2 = (x_{correct} - u_0)^2 + (y_{correct} - v_0)^2$, u_0 and v_0 are principal points (intersection of camera optical axis and image plane).

Then we can correct the radial distortion by warping the captured image with a reverse distortion. For each pixel in the correct image, map its corresponding location in the distorted image using Eqn.7.1 and 7.2, and apply bilinear interpolation in the distorted image to get the RGB value.

7.3 Perspective Distortion Correction

Perspective distortion occurs when the camera optical axis is not perpendicular to the center of the object (Fig. 7.3). The parallel lines in the scene are not parallel in the image (Fig. 7.2).

When capture the image of particles, use blue grid pattern as background reference. By using the following method, we can use four corner points of a rectangle in the scene to find the homography to correct the perspective distortion, i.e. make all parallel lines in the scene also parallel in the image, all lines orthogonal in the scene are orthogonal in the image, squares in scene have unit aspect ratio in the image, and circles in the scene are circular in the image.

Suppose we have a point in world coordinates $\begin{bmatrix} x^{world} \\ y^{world} \end{bmatrix}$ and we write this in homogeneous coordinates as $\begin{bmatrix} kx^{world} \\ ky^{world} \\ k \end{bmatrix}$.

Similary, the corresponding point in image coordinates is $\begin{bmatrix} x^{img} \\ y^{img} \end{bmatrix}$, and we write this in homogeneous coordinates as $\begin{bmatrix} k'x^{img} \\ k'y^{img} \\ k' \end{bmatrix}$.

The relationship between these two can be expressed in the following equation:

$$\begin{bmatrix} k'x^{img} \\ k'y^{img} \\ k' \end{bmatrix} = \mathbf{H} \begin{bmatrix} kx^{world} \\ ky^{world} \\ k \end{bmatrix}$$

where $\mathbf{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & 1 \end{bmatrix}$ is the Homography we want to solve.

Expanding both sides of the equation using matrix multiplication, we get:

$$k'x^{img} = k(H_{11}x^{world} + H_{12}y^{world} + H_{13})$$

$$k'y^{img} = k(H_{21}x^{world} + H_{22}y^{world} + H_{23})$$

$$k' = k(H_{31}x^{world} + H_{32}y^{world} + 1)$$

Plug in the third equation to the first two, we get two equations from this pair of points:

$$x^{img} = H_{11}x^{world} + H_{12}y^{world} + H_{13} - H_{31}x^{img}x^{world} - H_{32}x^{img}y^{world}$$
$$y^{img} = H_{21}x^{world} + H_{22}y^{world} + H_{23} - H_{31}y^{img}x^{world} - H_{32}y^{img}y^{world}$$

Since there are eight unknowns in **H**, we need 4 pairs of points to solve **H**. We write the eight equations in matrix form:

$$\begin{bmatrix} x_0^{img} \\ y_0^{img} \\ x_1^{img} \\ y_1^{img} \\ x_2^{img} \\ y_2^{img} \\ x_3^{img} \\ y_3^{img} \end{bmatrix} = \begin{bmatrix} x_0^{world} & y_0^{world} & 1 & 0 & 0 & 0 & -x_0^{img}x_0^{world} & -x_0^{img}y_0^{world} \\ 0 & 0 & 0 & x_0^{world} & y_0^{world} & 1 & -y_0^{img}x_0^{world} & -y_0^{img}y_0^{world} \\ x_1^{world} & y_1^{world} & 1 & 0 & 0 & 0 & -x_1^{img}x_1^{world} & -x_1^{img}y_1^{world} \\ 0 & 0 & 0 & x_1^{world} & y_1^{world} & 1 & -y_1^{img}x_1^{world} & -y_1^{img}y_1^{world} \\ x_2^{world} & y_2^{world} & 1 & 0 & 0 & 0 & -x_2^{img}x_2^{world} & -x_2^{img}y_2^{world} \\ 0 & 0 & 0 & x_2^{world} & y_2^{world} & 1 & -y_2^{img}x_2^{world} & -y_2^{img}y_2^{world} \\ x_3^{world} & y_3^{world} & 1 & 0 & 0 & 0 & -x_3^{img}x_3^{world} & -x_3^{img}y_3^{world} \\ 0 & 0 & 0 & x_3^{world} & y_3^{world} & 1 & -y_3^{img}x_3^{world} & -y_3^{img}y_3^{world} \end{bmatrix} \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \end{bmatrix}$$

Thus, for each image suffers from the projective distortion, we pick four points in this image, and given the world coordinates of these four points, we are able to solve **H**.

7.4 Corner Detection Algorithm

Corner detection plays an important role during both the camera calibration process and perspective distortion correction process. In the camera calibration process, we need to extract all the corner points, and use their world coordinates and image coordinates to get the intrinsic parameters and extrinsic parameters of the camera. And in the perspective distortion correction process, we need at least four corners, the upper left, upper right, lower left, lower right corners to find the homography. The idea of this algorithm is to transect each side of each square to find the outer boundary points, and fit straight lines to these points to get the outer boundaries.

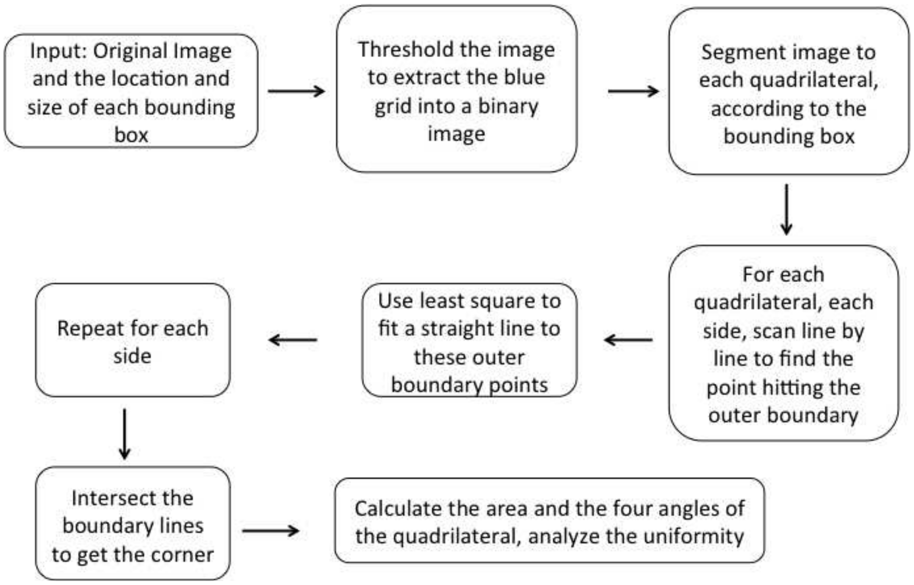

Input: Original Image and the location and size of each bounding box
Threshold the image to extract the blue grid into a binary image
Segment image to each quadrilateral, according to the bounding box
For each quadrilateral, each side, scan line by line to find the point hitting the outer boundary
Use least square to fit a straight line to these outer boundary points
Repeat for each side
Intersect the boundary lines to get the corner
Calculate the area and the four angles of the quadrilateral, analyze the uniformity

Fig. 7.4. Flowchart for corner detection algorithm

And then intersect these boundaries to get the corners. The detailed flow chart is presented in Fig.7.4. Fig.7.5 shows how to find one outer boundary point by searching a single transect of the left boundary. Fig. 7.6 illustrates the line fitting of the outer boundaries and the red crosses represent the four corners.

7.5 Method and Flowchart

1. Acquire and set up capture device and system. (Fig.7.8) Use a box as a camera stand to keep the distance between the pattern and the camera lens constant.

2. First Stage: model based correction (Scientific Approach)

    (a) Correct lens distortion

        i. Capture a set of blue grid pattern images for camera calibration.

        ii. Extract the all the corner points of the squares in the grid pattern

        iii. Use OpenCV library to do camera calibration, get the lens distortion parameters for the camera.

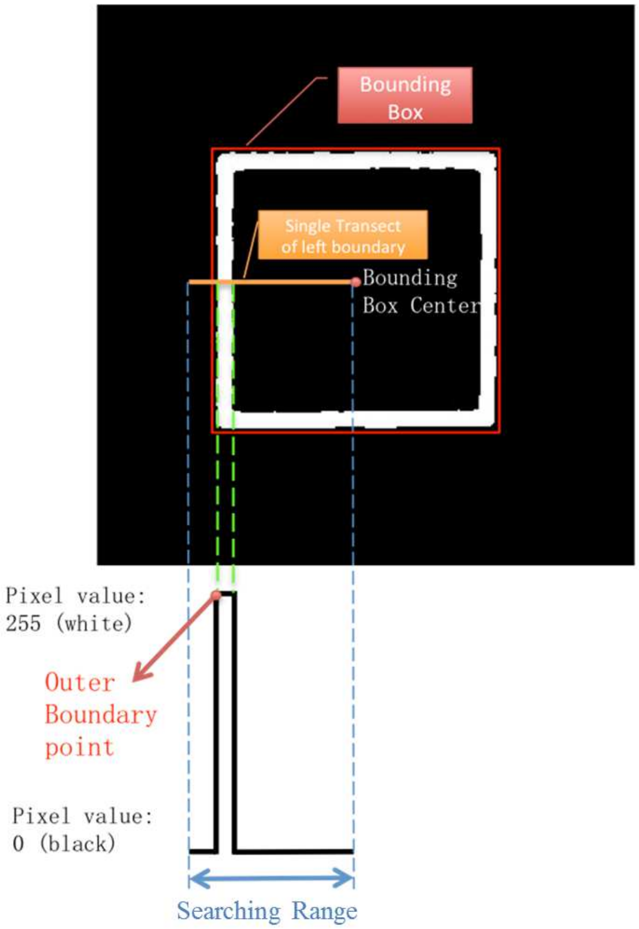


Fig. 7.5. Illustration of finding outer boundary point for a single transect of the left boundary

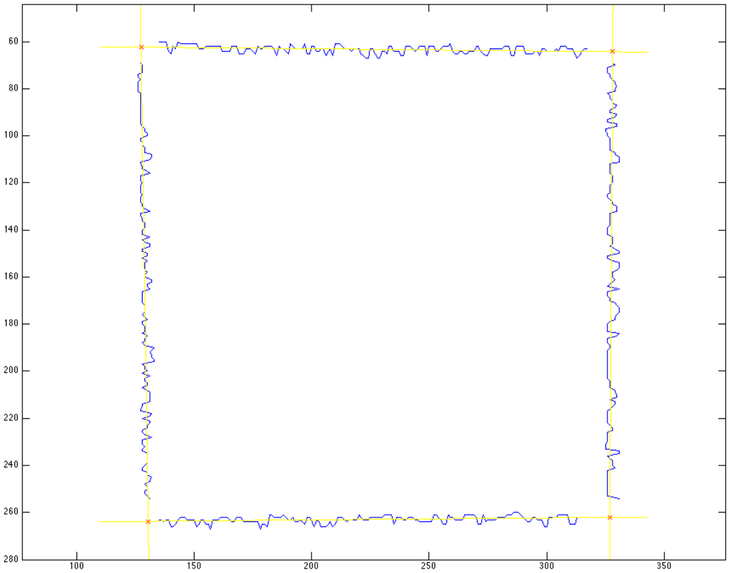


Fig. 7.6. Fitting lines to find the boundaries. The intersections of lines are corners

iv. Use OpenCV library and the lens distortion parameters to correct the distorted image.

(b) Correct projective distortion

i. Sprinkle some coffee grind particles on the pattern, and capture the image.

ii. Use the corner detection algorithm to detect the upper left, upper right, bottom left, bottom right corners of the grid.

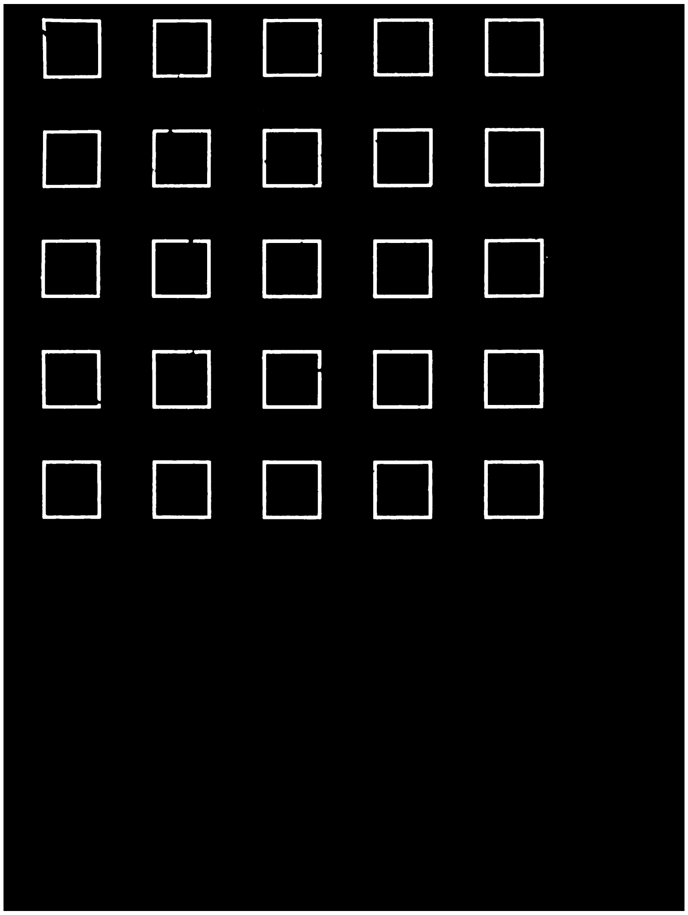
Use the method discussed in previous section to solve the homography H between the four pairs of points in world coordinate system and image coordinate system. Correct the image using the homography H to get the perspective-distortion-free image.

3. Second Stage: measurement based correction (Engineering Approach) For every blue square, generate a re-mapping factor $\alpha = \frac{\text{size of the corrected square}}{\text{size of the distorted square}}$, use Shepard Interpolation [47] to generate a scalar map for every pixel in the image. Given a particle position, easily apply the corresponding scalar to the size of the distorted particle to get the correct size for the particle.

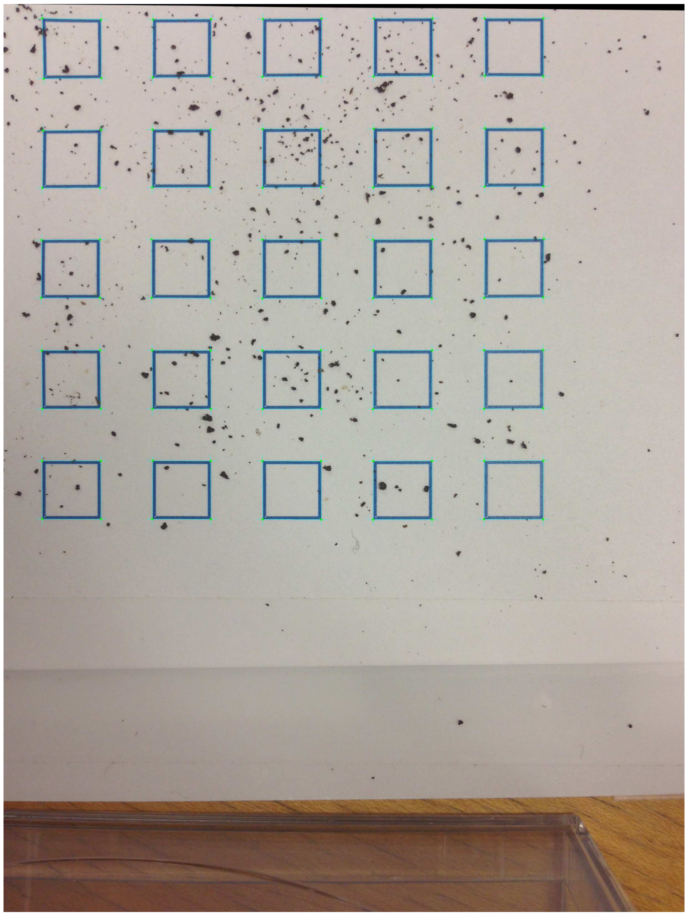
Fig. 7.9 shows the flowchart of the algorithm.

7.6 Results

Fig. 7.10 shows the image after lens distortion correction based on the original image Fig. 7.2, and Fig. 7.11 shows the bounding box corners for preliminary perspective distortion correction. Fig. 7.12 is the final corrected image with detected corner points on it.



(a) Blue grid extracted binary image



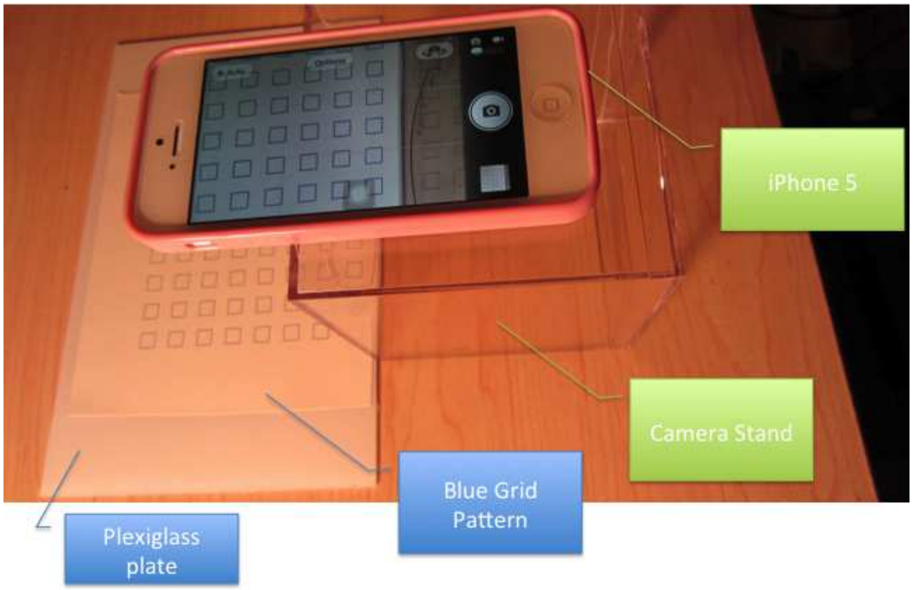


Fig. 7.8. System set-up

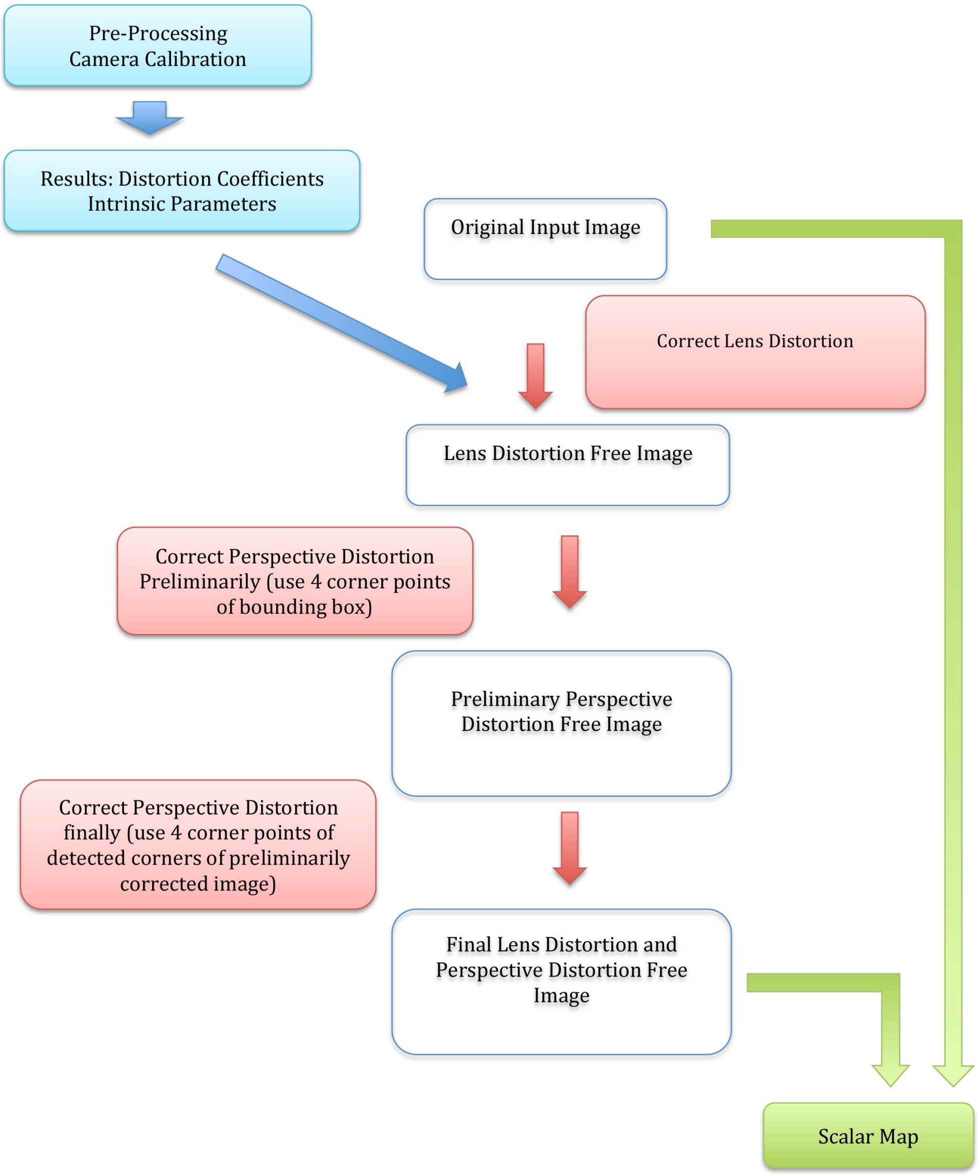

Pre-Processing
Camera Calibration
Results: Distortion Coefficients
Intrinsic Parameters
Original Input Image
Correct Lens Distortion
Lens Distortion Free Image
Correct Perspective Distortion Preliminarily (use 4 corner points of bounding box)
Preliminary Perspective Distortion Free Image
Correct Perspective Distortion finally (use 4 corner points of detected corners of preliminarily corrected image)
Final Lens Distortion and Perspective Distortion Free Image
Scalar Map

Fig. 7.9. Flowchart of the algorithm

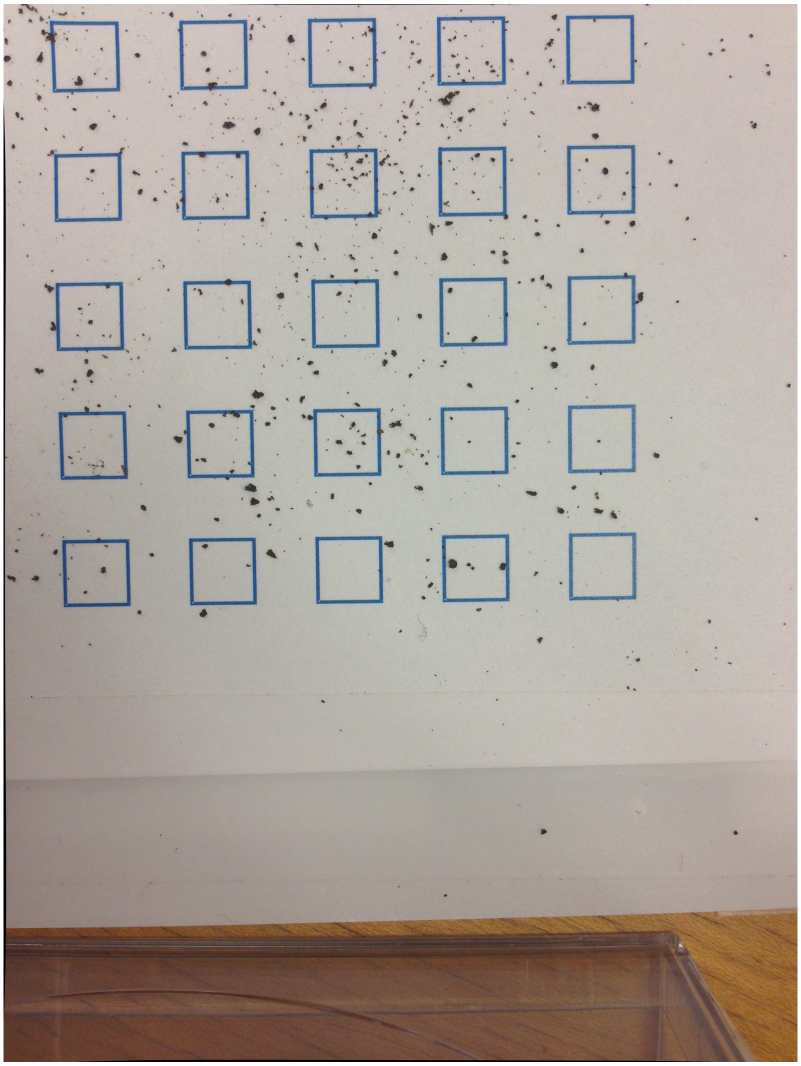


Fig. 7.10. Lens distortion corrected image

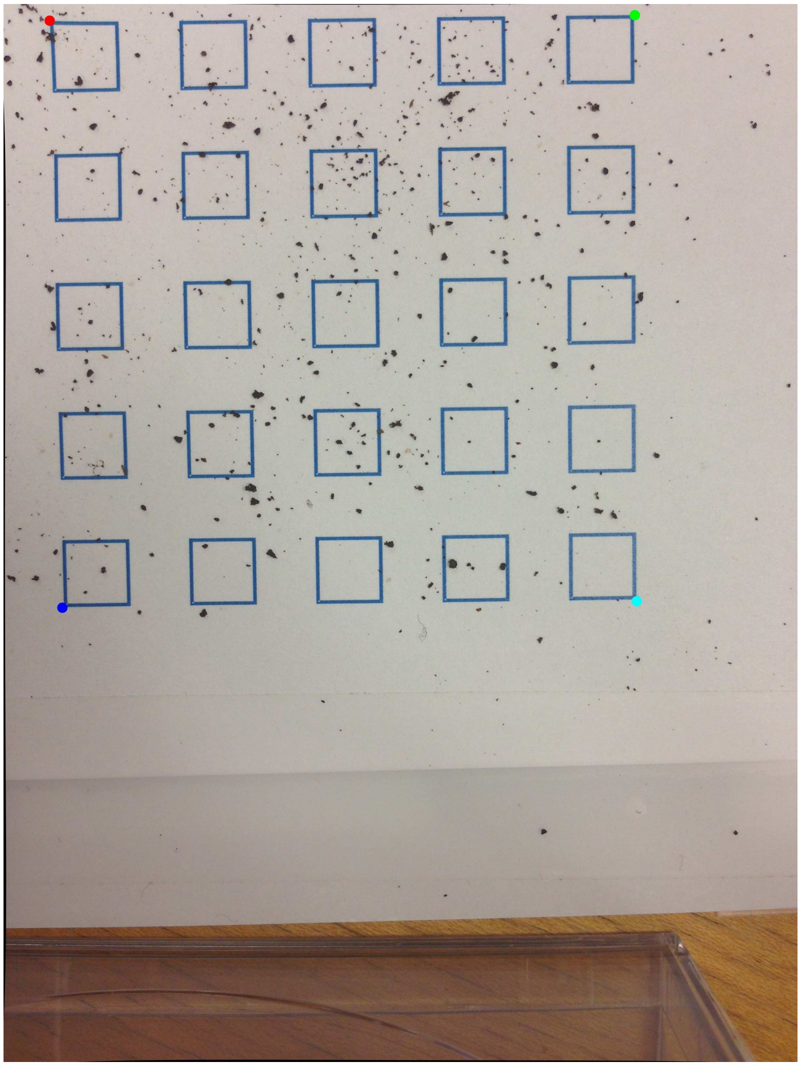


Fig. 7.11. Bounding box corners for preliminary perspective distortion correction

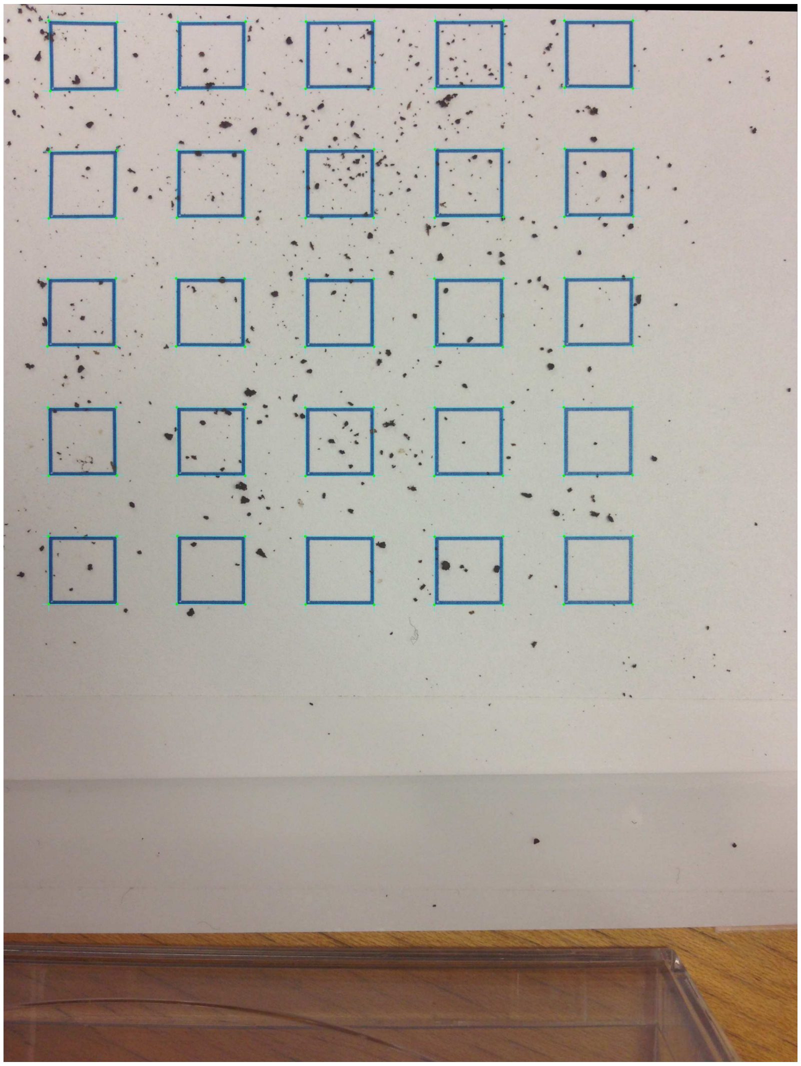


Fig. 7.12. Final corrected image with detected corners

8. 3D RECONSTRUCTION USING STRAIGHT LINE PATTERNS

In this chapter, we present one possible structured light coding strategy using our dual-projector system to capture 3D data. We first introduce the depth map acquisition theory, followed by the experiment set-up and finally the experimental result.

8.1 3D Depth Map Acquisition Theory

Use the pinhole camera model with lens distortion we discussed in Sec. 3.1, we form the forward mapping of a 3D point from world coordinate system to camera image plane and projector image plane respectively. Take the camera as example in Fig. 8.1, we first map the point in world coordinate system to camera coordinate system using camera extrinsic parameters, followed by normalization and lens distortion. Last but not least, we multiply the camera intrinsic matrix to get the 2D coordinates of the 3D point on camera image plane. The projector has a similar forward mapping, except we need to convert the world coordinates to camera coordinates first and then to projector coordinate system.

This whole process is invertible expect the normalization. Therefore, given a point on camera image plane $\begin{bmatrix} x_{ld}^{c} \\ y_{ld}^{c} \end{bmatrix}$ and the corresponding point on projector image plane $\begin{bmatrix} x_{ld}^{p} \\ y_{ld}^{p} \end{bmatrix}$, we can calculate the reverse mapping up to normalization, where we get $\begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}$ for camera and $\begin{bmatrix} X_p/Z_p \\ Y_p/Z_p \end{bmatrix}$ for projector.

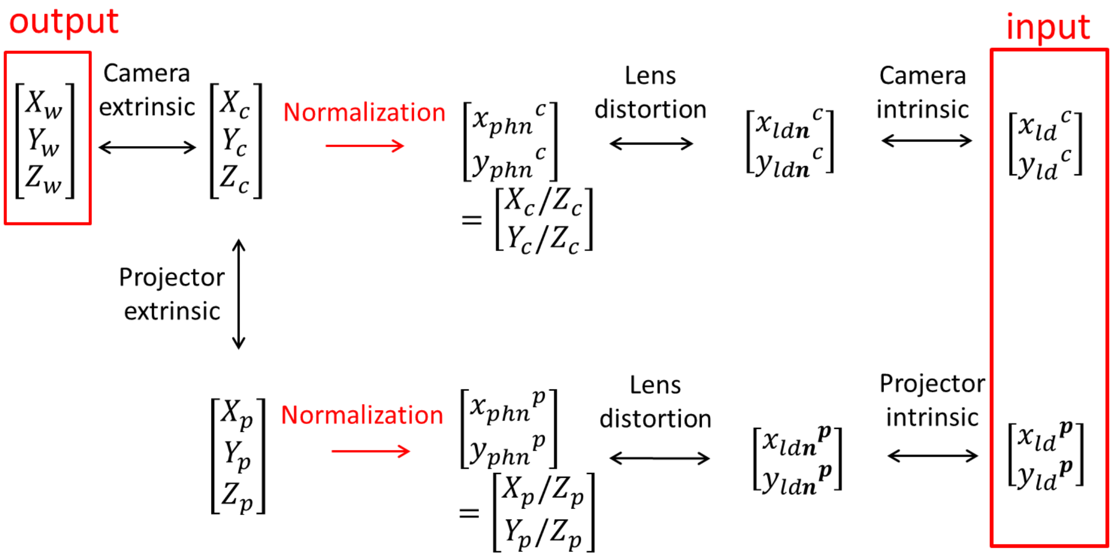

output
input
$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$
Camera extrinsic
$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$
Normalization
$\begin{bmatrix} x_{phn}{}^c \\ y_{phn}{}^c \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}$
Lens distortion
$\begin{bmatrix} x_{ldn}{}^c \\ y_{ldn}{}^c \end{bmatrix}$
Camera intrinsic
$\begin{bmatrix} x_{ld}{}^c \\ y_{ld}{}^c \end{bmatrix}$
Projector extrinsic
$\begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix}$
Normalization
$\begin{bmatrix} x_{phn}{}^p \\ y_{phn}{}^p \end{bmatrix} = \begin{bmatrix} X_p/Z_p \\ Y_p/Z_p \end{bmatrix}$
Lens distortion
$\begin{bmatrix} x_{ldn}{}^p \\ y_{ldn}{}^p \end{bmatrix}$
Projector intrinsic
$\begin{bmatrix} x_{ld}{}^p \\ y_{ld}{}^p \end{bmatrix}$

Fig. 8.1. Mapping relationship between 3D world coordinates, camera image plane and projector image plane. This whole process is invertible expect the normalization.

Next, we use the rotation and translation relationship to get the world coordinates. First by using Eqn. 4.1, we have

$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} = \mathbf{R}_p \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} + \mathbf{T}_p \tag{8.1}$$

which is equivalent to

$$Z_p \begin{bmatrix} \frac{X_p}{Z_p} \\ \frac{Y_p}{Z_p} \\ 1 \end{bmatrix} = \mathbf{R}_p Z_c \begin{bmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ 1 \end{bmatrix} + \mathbf{T}_p \tag{8.2}$$

By solving Eqn. 8.2, which is an overdetermined system with two unknowns, we get Z_c and Z_p simultaneously. Use $\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \mathbf{R}_c{}^{-1} \left(\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} - \mathbf{T}_c \right)$, we get the corresponding world coordinates $\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$ for $\begin{bmatrix} x_{ld}^c \\ y_{ld}^c \end{bmatrix}$ and $\begin{bmatrix} x_{ld}^p \\ y_{ld}^p \end{bmatrix}$.

8.2 Reconstruction Using Straight Line Patterns

In this section, we introduce a 3D capture procedure by projecting a series of vertical and horizontal line patterns, and detecting the captured distorted lines. By decoding the positions of the intersections of horizontal and vertical lines, we are able to reconstruct the depth information. We will introduce the straight line pattern design first and discuss the procedure of experiment.

8.2.1 Pattern Design

We first design 30 images that each contains a black background and a horizontal line, and 40 images that each contains a black background and a vertical line. The

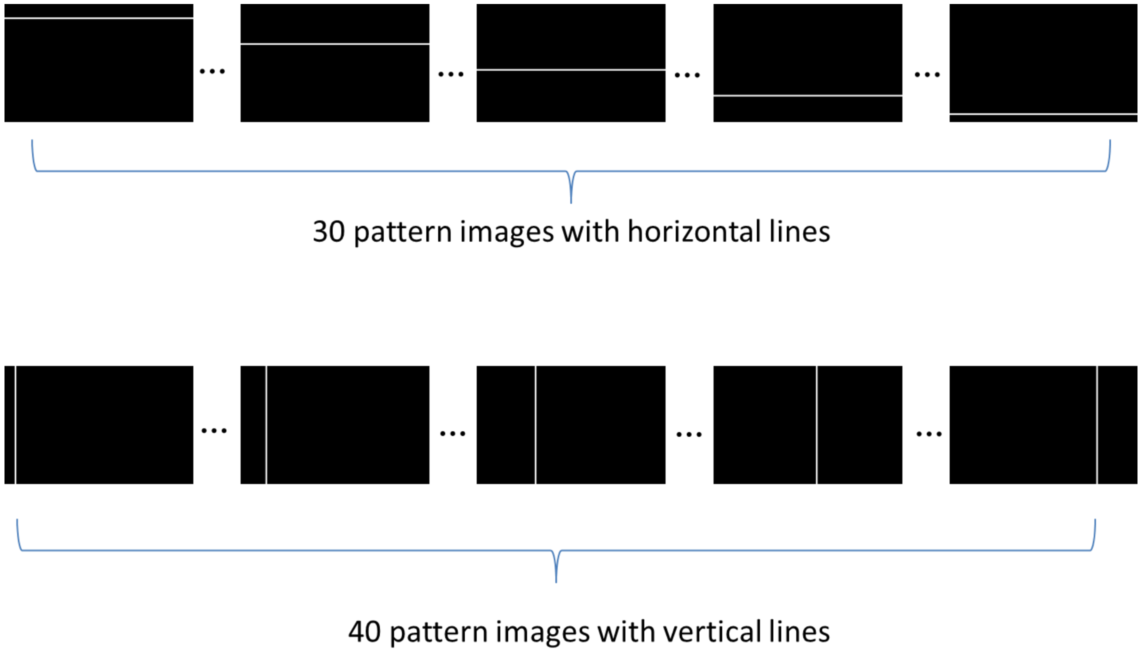

...
30 pattern images with horizontal lines
40 pattern images with vertical lines

Fig. 8.2. The straight line patterns includes 30 images that each contains a black background and a horizontal line, and 40 images that each contains a black background and a vertical line. The lines are in different positions in different images. Horizontal lines are shifted 3 pixels for each consecutive image, and vertical lines are shifted 5 pixels. The width for each line is 10 pixel.

lines are in different positions in different images. Horizontal lines are shifted 3 pixels for each consecutive image, and vertical lines are shifted 5 pixels. The width for each line is 10 pixel. The pattern series are shown in Fig. 8.2.

8.2.2 Experiment Procedure

The experiment procedure is shown in Fig. 8.3. We first project all the line patterns sequentially, and capture the series of distorted lines on the object using the camera. By extracting the position of all detectable distorted lines, we obtain the intersection of one horizontal line and one vertical line as one data point on camera image plane. Since the data point's location are given in the projector image plane

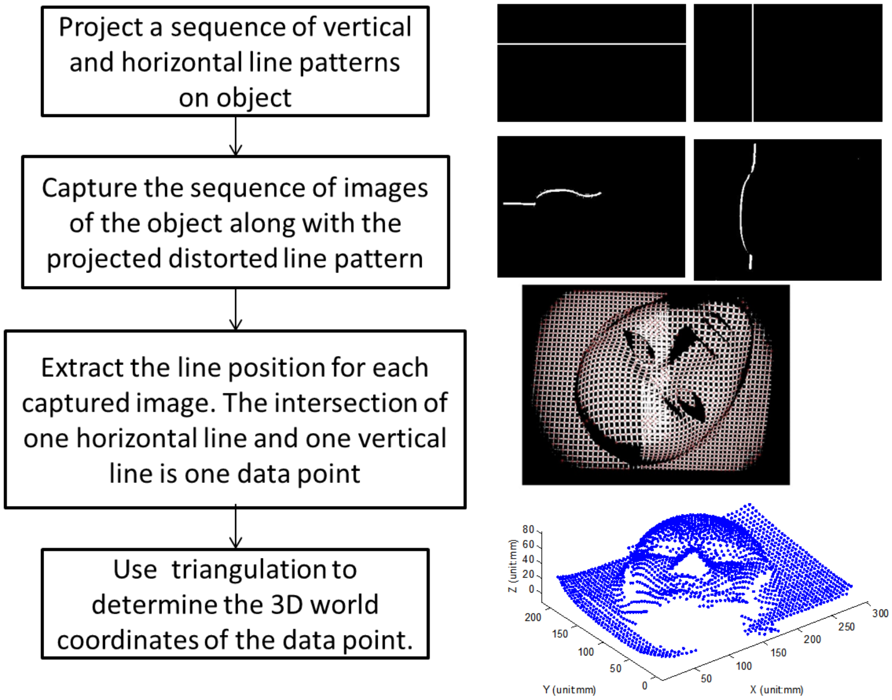

Project a sequence of vertical and horizontal line patterns on object
Capture the sequence of images of the object along with the projected distorted line pattern
Extract the line position for each captured image. The intersection of one horizontal line and one vertical line is one data point
Use triangulation to determine the 3D world coordinates of the data point.
Z (unit:mm)
80
60
40
20
0
200
150
100
50
0
Y (unit:mm)
50
100
150
200
250
300
X (unit:mm)

Fig. 8.3. 3D reconstruction procedure using straight line patterns.

using the line design we mentioned above, we have the corresponding data point in both camera and projector image plane, hence we can extract the 3D information for this point using the method from Sec. 8.1.

8.3 Data Fusing Procedure and 3D Reconstruction Result

Use the above approach, we are able to generate the 3D model for our object. However, due to shadow and obliqueness, we are not capable of getting a complete model use single capture angle. To solve this, we introduced a simple data fusing procedure shown in Fig. 8.4.

First, we tape the object on a piece of 8.5″ × 8.5″ cardboard, and place the cardboard on the platen for 3D capture. Rotate the cardboard every 90° around the

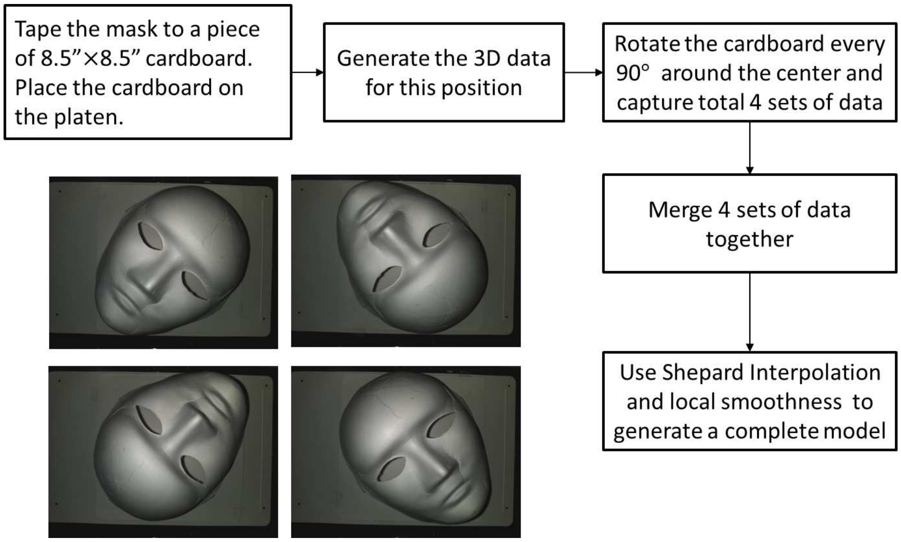

Tape the mask to a piece of 8.5”×8.5” cardboard. Place the cardboard on the platen.
Generate the 3D data for this position
Rotate the cardboard every 90° around the center and capture total 4 sets of data
Merge 4 sets of data together
Use Shepard Interpolation and local smoothness to generate a complete model

Fig. 8.4. Rotate object 90° and fuse the 4 sets of data together

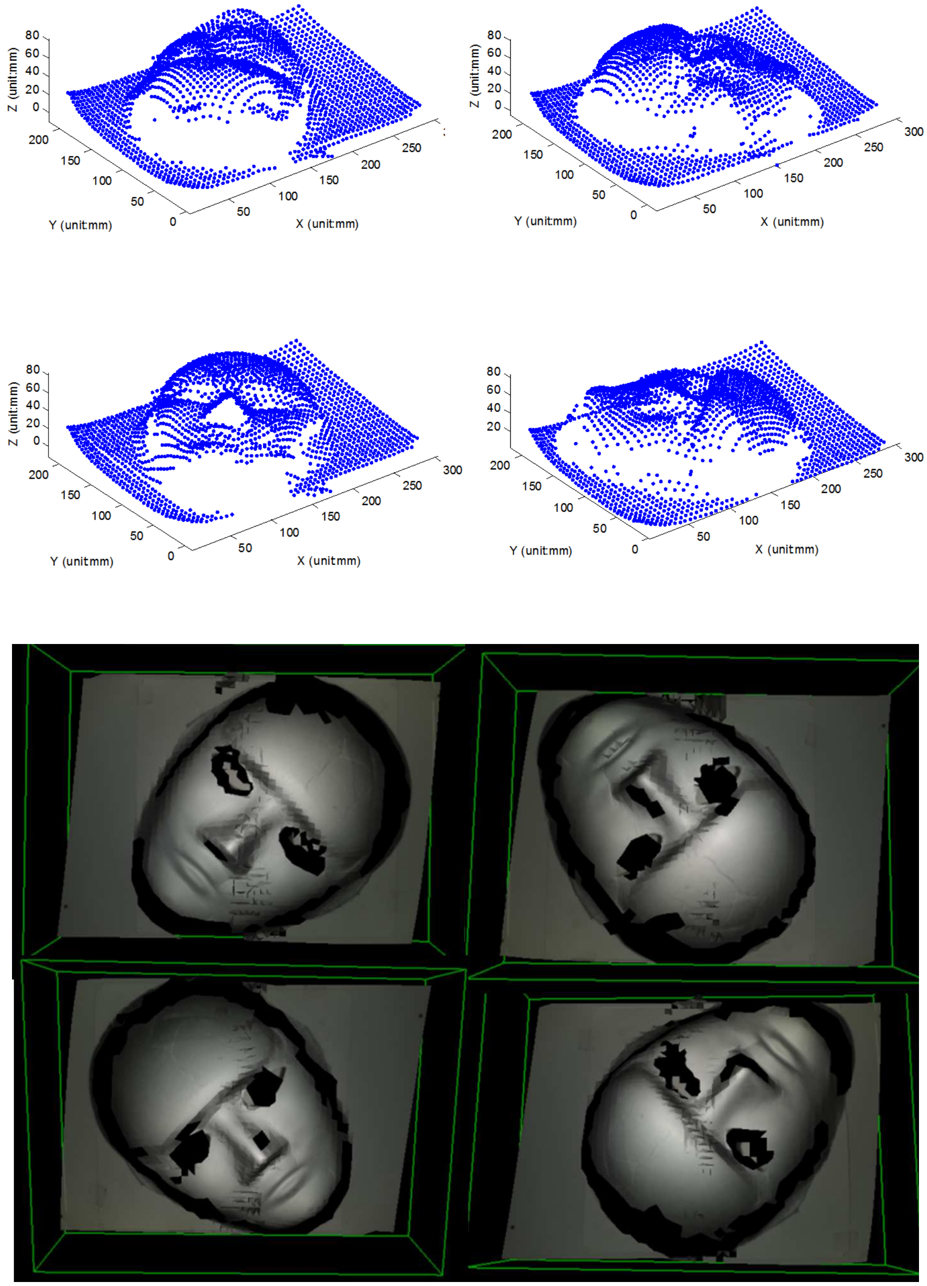


Fig. 8.5. Four sets of generated 3D data points and corresponding 3D reconstruction results before fusing.

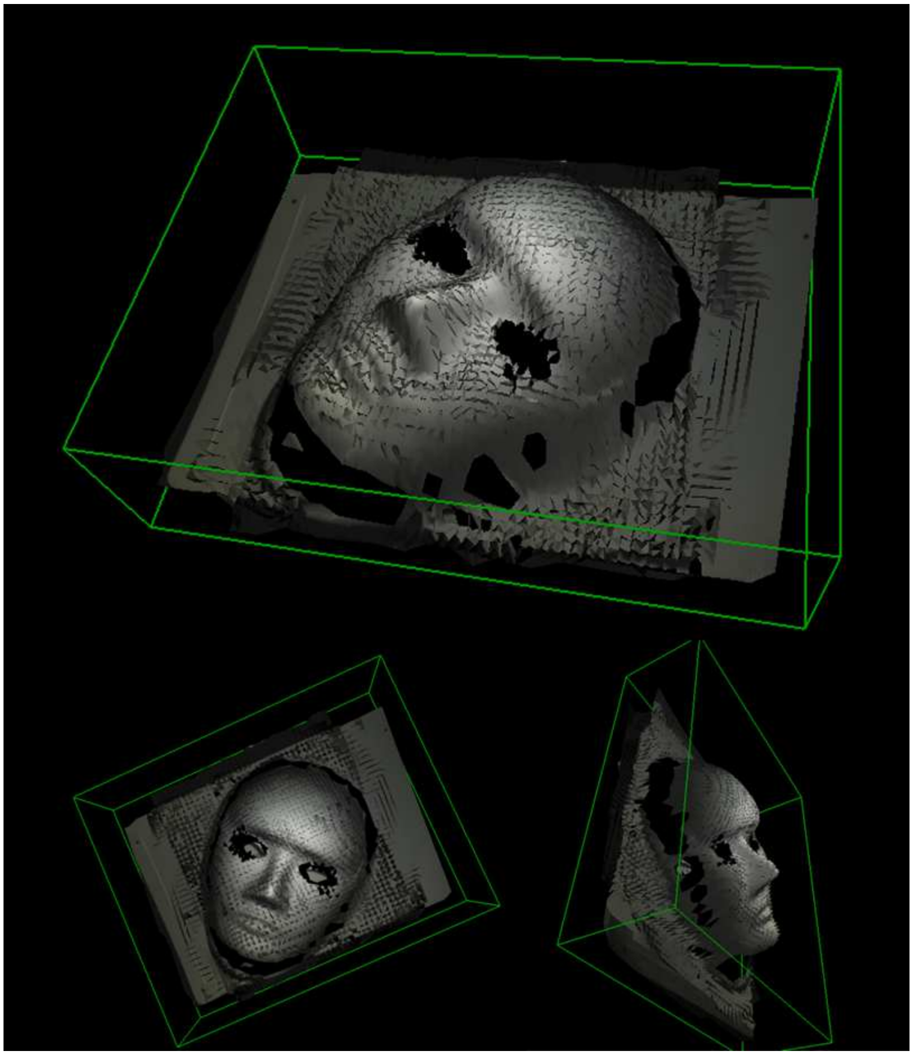


Fig. 8.6. 3D reconstruction result by fusing four views together

center of the cardboard and capture total 4 sets of data. The generated four sets of 3D data points and the corresponding 3D reconstruction results are shown in Fig. 8.5.

By rotating 3 sets of data to match the remaining set of data using the following 2D rotation transformation, we can merge them all together:

$$\begin{bmatrix} {X_w}^{'} \\ {Y_w}^{'} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \end{bmatrix} \tag{8.3}$$

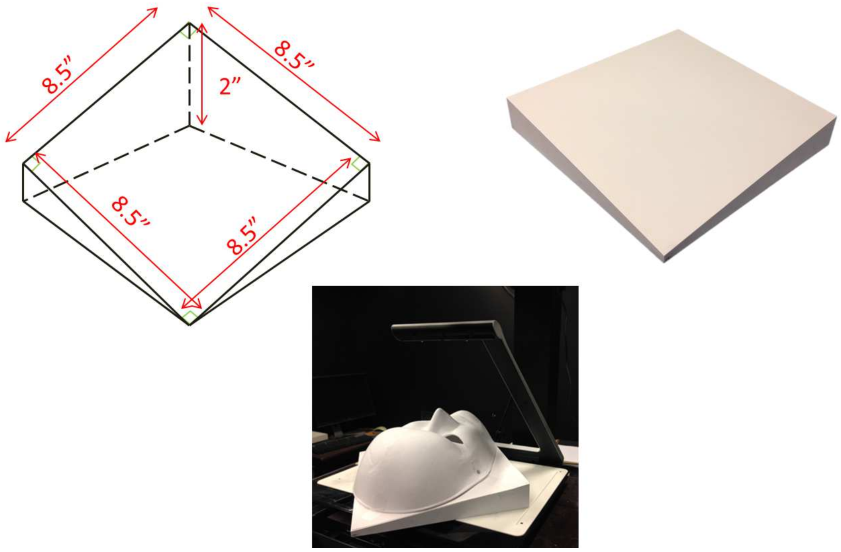


Fig. 8.7. The dimension of the wedge and how we use it to capture the oblique edge around the object.

Figure. 8.7 shows the design of a wedge we use to increase the degree of freedom of the rotation, in order to capture the oblique edge around the object that we are not able to get using the above procedure. Therefore, we use the wedge to capture the forehead and chin of the mask separately. The results are shown in Fig. 8.8.

To merge the data captured using the wedge with the original four sets of data, we need to apply the following 3D rotation transformation:

$$\begin{bmatrix} {X_w}' \\ {Y_w}' \\ {Z_w}' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{8.4}$$

The complete data point after fusing all six sets of data together is shown in Fig. 8.9. To fill in any missing area remaining, we use Shepard interpolation and local averaging to smooth the surface. Figure. 8.10 shows the generated surface result after applying Shepard interpolation to original data points and final surface result after

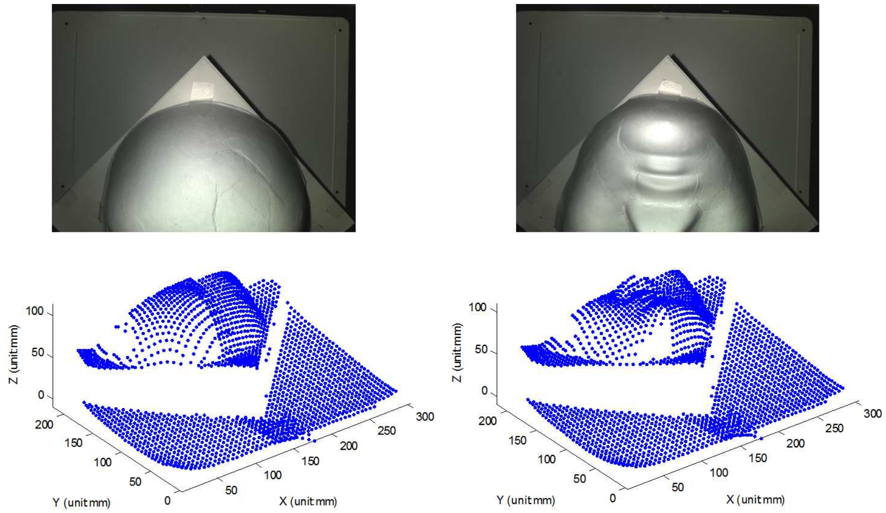


Fig. 8.8. Captured 3D data for forehead and chin using wedge.

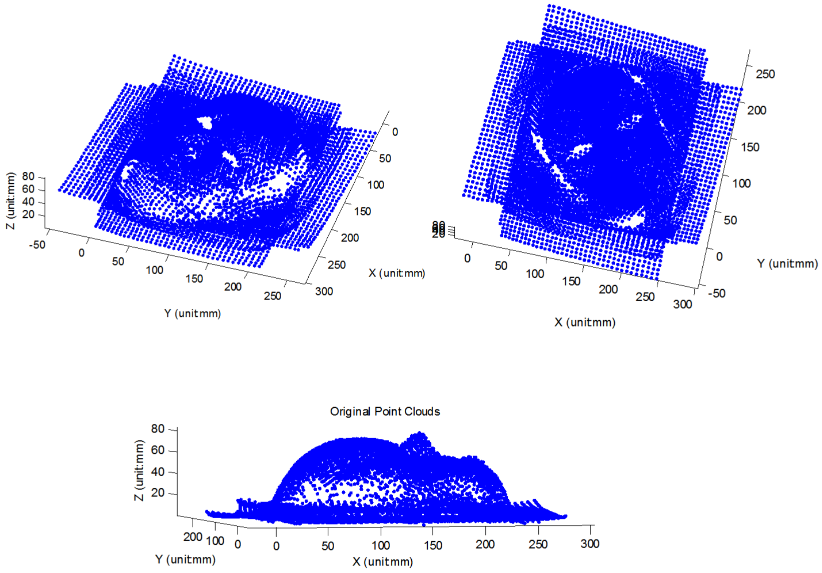
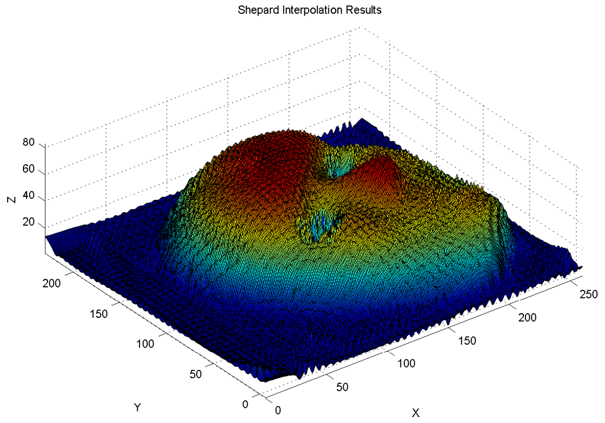
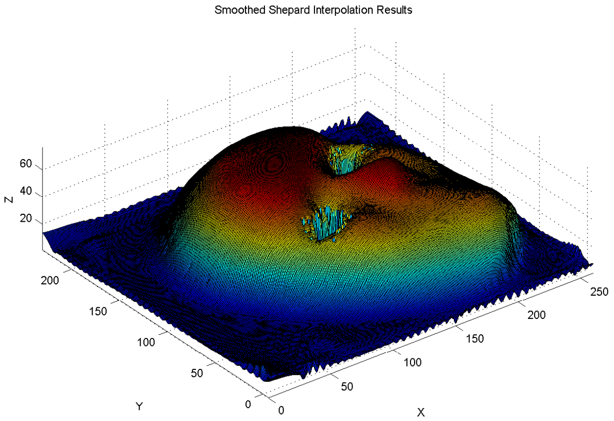


Fig. 8.9. Complete data points after fusing all six sets of data together

applying local smoothing to Shepard interpolated result. This way, we get a nearly complete object model shown in Fig. 8.11.



(a) Generated surface result after applying Shepard interpolation to original data points



(b) Final surface result after applying local smoothing to Shepard interpolated result

Fig. 8.10. Surface reconstructed result after Shepard interpolation and local smoothing

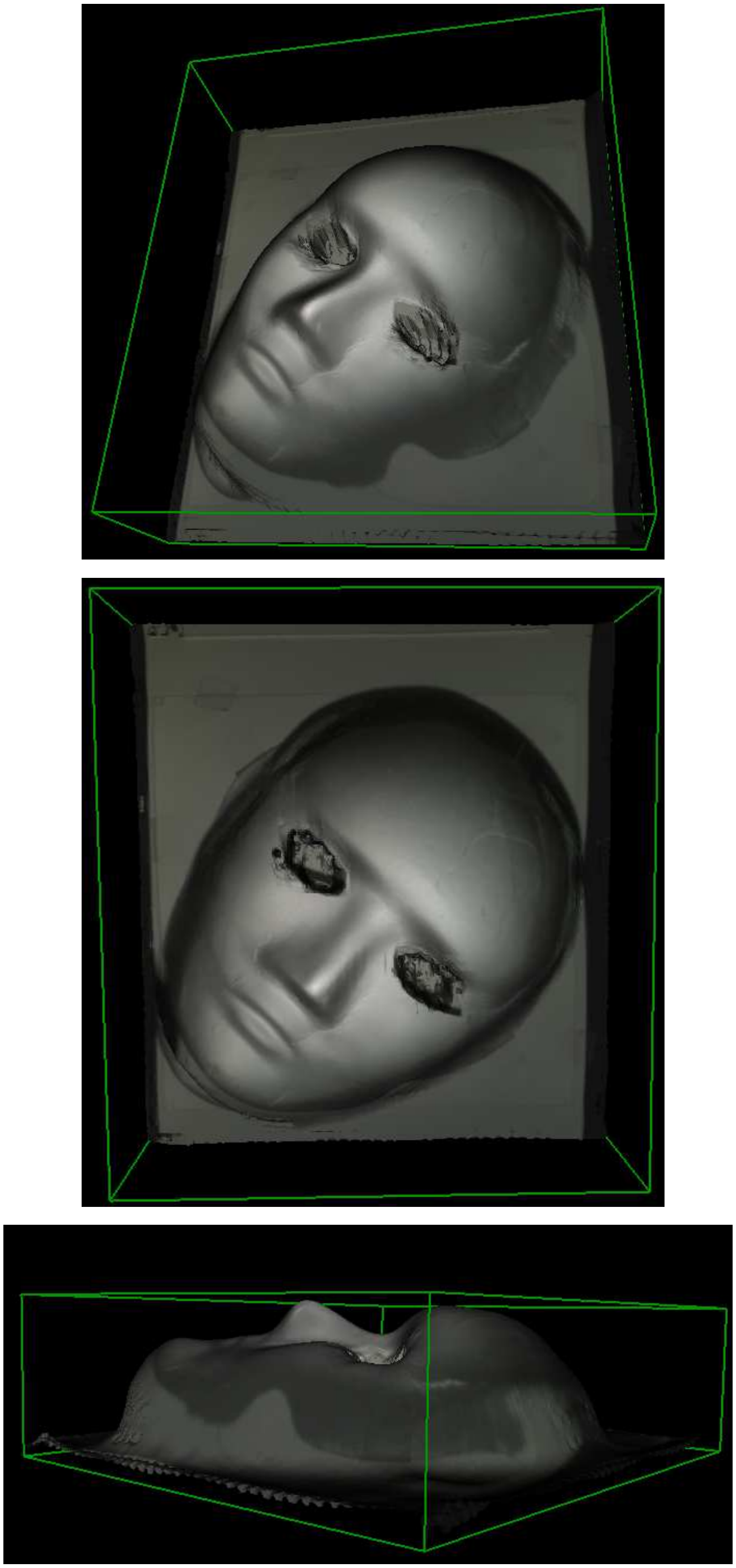


Fig. 8.11. Final 3D reconstructed result

9. SYSTEM RECONSTRUCTION ACCURACY ANALYSIS

In this chapter, we present the system accuracy analysis [31] using a new staircase by comparing the a set of points' manually measured coordinates on the staircase and the reconstructed coordinates using the straight line pattern 3D reconstruction procedure.

9.1 Experiment Set-up

Figure 9.1 shows our newly designed staircase specifically for the accuracy analysis. The dimension of the staircase is $24cm(L) \times 16cm(W) \times 6cm(H)$. There are total six steps, and each step is $16cm(L) \times 4cm(W) \times 1cm(H)$. Figure 9.2 is the flowchart for the accuracy experiment. We first place the staircase on the platen, and project a grid pattern(Fig. 9.3) on the staircase and mark the position of the intersections on the staircase. Figure 9.4 shows the marker positions on the staircase. The grid pattern is actually the superposition of a subsample of the vertical and horizontal lines we introduced in Sec. 8.2.1. This way, it guarantees that we get 3D information of all the markers. Next, we run the 3D capture procedure using straight line patterns described in Sec. 8.2.2, and get the markers' reconstructed 3D coordinates X_r, Y_r, Z_r. Last but not least, we use a caliper to manually measure all the markers' actual 3D coordinates X_o, Y_o, Z_o and compare with the reconstructed ones.

9.2 System Accuracy Result and Analysis

In this section, we first present the result for the accuracy analysis. Then we analyze the possible causes for these errors and validate our hypothesis.

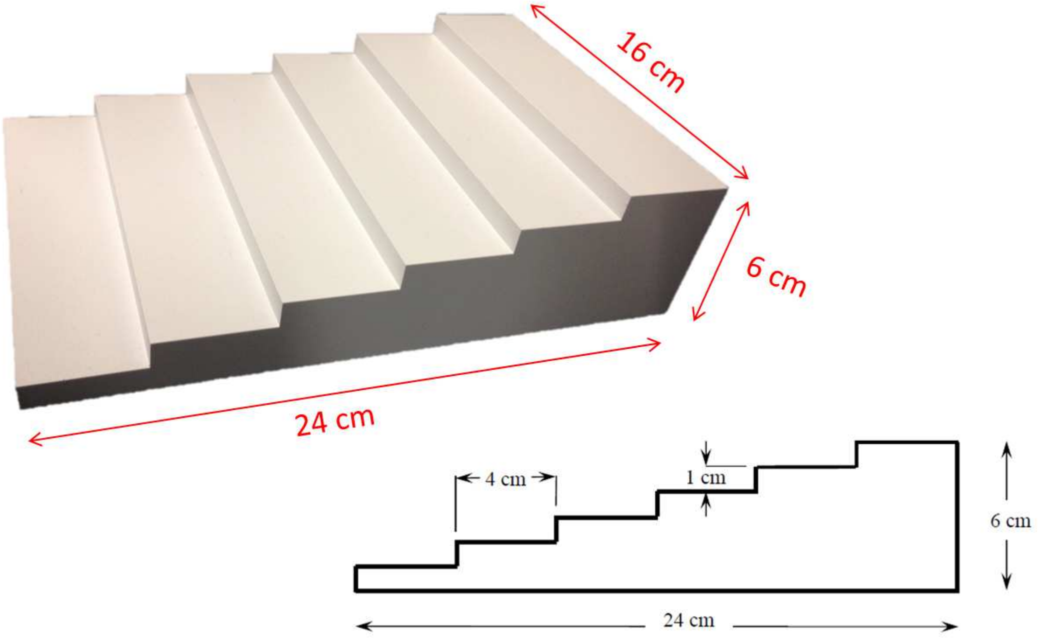


Fig. 9.1. Dimensions of the staircase used for accuracy analysis

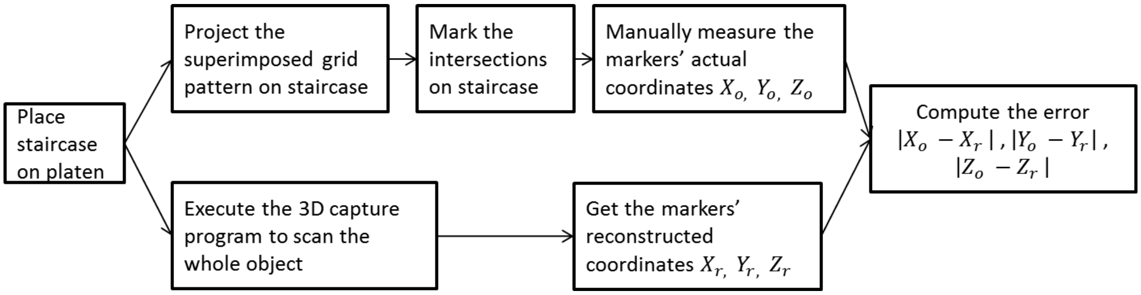

Place staircase on platen
Project the superimposed grid pattern on staircase
Mark the intersections on staircase
Manually measure the markers' actual coordinates $X_o,\ Y_o,\ Z_o$
Execute the 3D capture program to scan the whole object
Get the markers' reconstructed coordinates $X_r,\ Y_r,\ Z_r$
Compute the error $|X_o - X_r|$, $|Y_o - Y_r|$, $|Z_o - Z_r|$

Fig. 9.2. Flowchart for the accuracy analysis

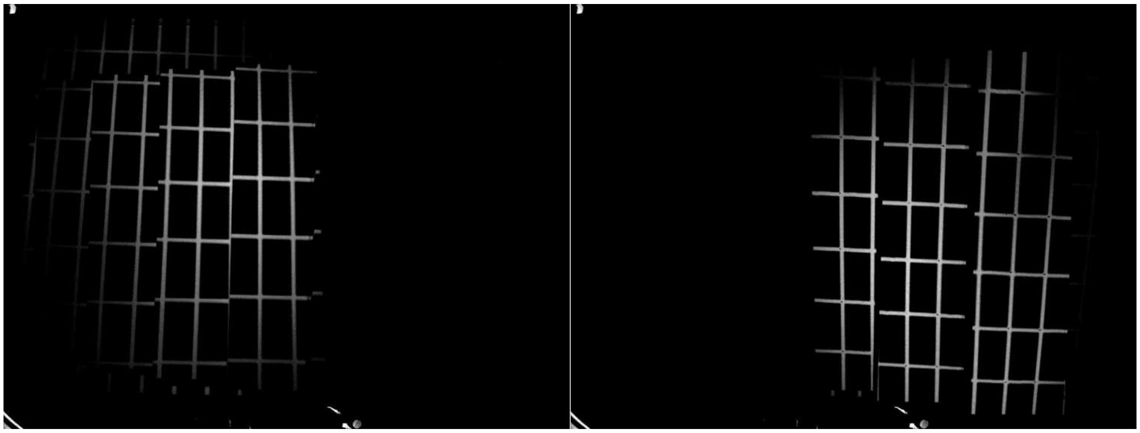


Fig. 9.3. Superimposed grid pattern projected by rear (left) and front (right) projector to determine the position of the markers used for accuracy analysis

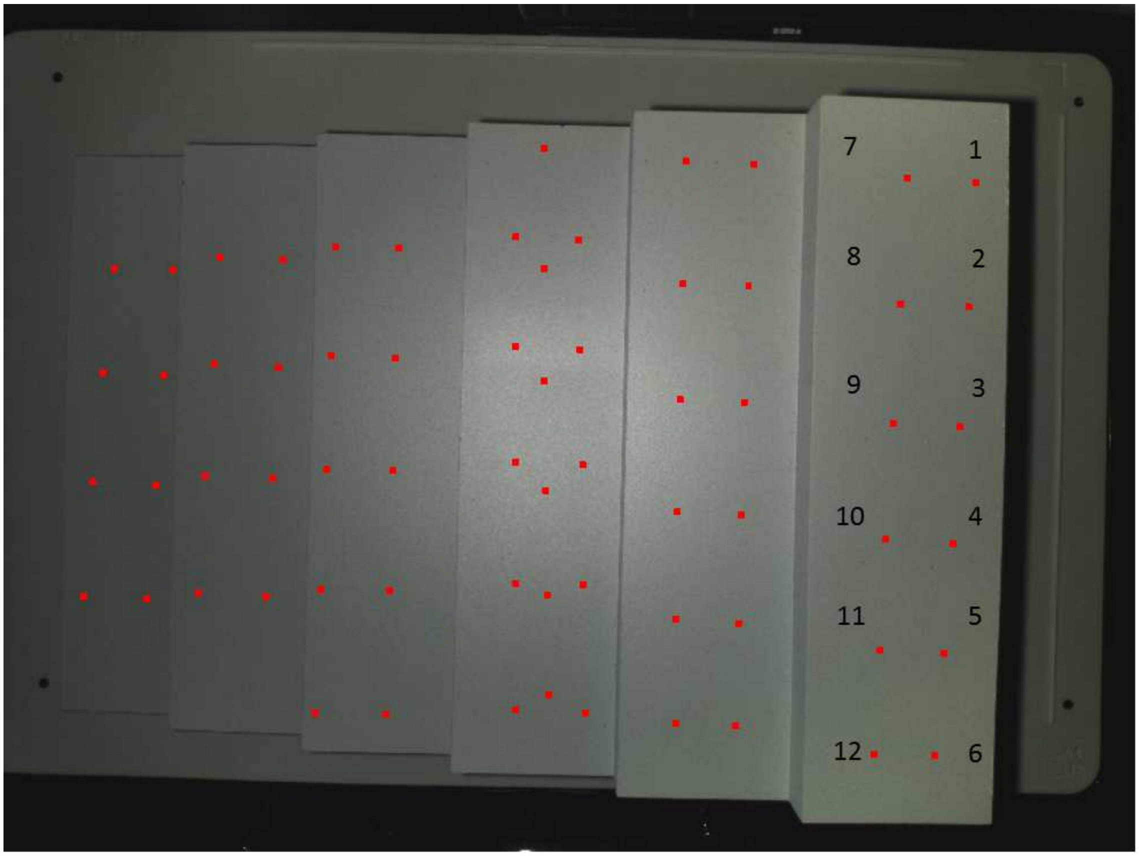


Fig. 9.4. Markers on staircase that used for measuring accuracy of the system and the order of indexing

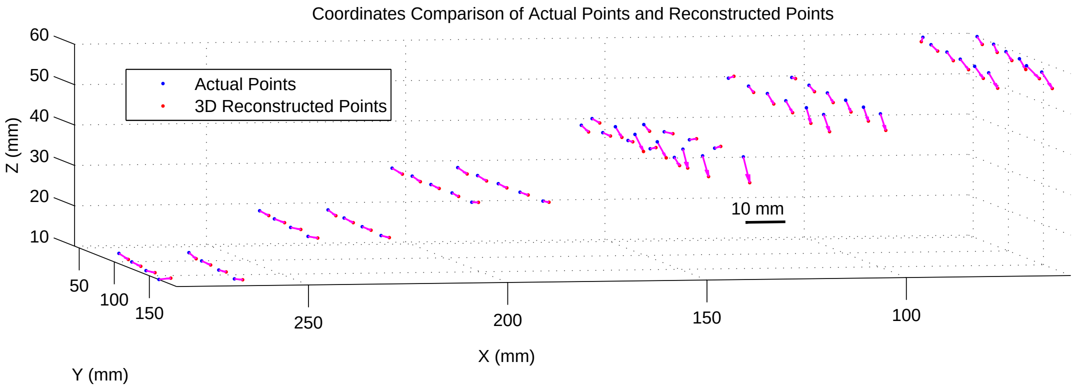


Fig. 9.5. Comparison of the measured and reconstructed coordinates of the markers on the staircase. The blue dots represent the actual 3D coordinates X_o, Y_o, Z_o we measured manually using a caliper and the red dots are the reconstructed 3D coordinates X_r, Y_r, Z_r using our 3D capture system. The magenta arrows represent the error vectors.

9.2.1 System Accuracy Result

Figure. 9.5 is the comparison of the measured and reconstructed coordinates of the markers on the staircase. The blue dots represent the actual 3D coordinates X_o, Y_o, Z_o we measured manually using a caliper and the red dots are the reconstructed 3D coordinates X_r, Y_r, Z_r using our 3D capture system. The magenta arrow represent the error vector.

Figure9.6 compares the absolute differences between the measured and reconstructed coordinates of the markers in X Y and Z axis. The red line represents the error in X axis ($\overline{|X_o - X_r|} = 2.17 \pm 0.61$). The blue line shows the error in Y

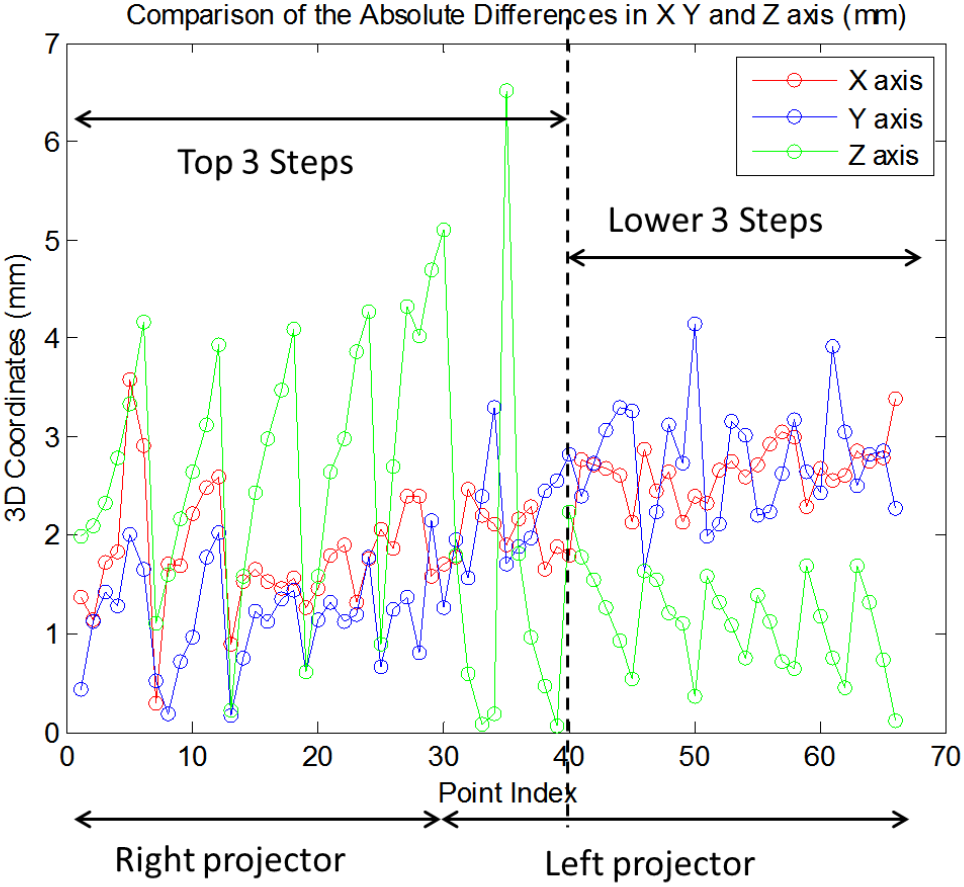


Fig. 9.6. Comparison of the absolute differences between the measured and reconstructed coordinates of the markers in X Y and Z axis, where $\overline{|X_o - X_r|} = 2.17 \pm 0.61$, $\overline{|Y_o - Y_r|} = 1.95 \pm 0.92$ and $\overline{|Z_o - Z_r|} = 1.92 \pm 1.4$.

axis ($\overline{|Y_o - Y_r|} = 1.95 \pm 0.92$) and the green line is the error in Z axis ($\overline{|Z_o - Z_r|} = 1.92 \pm 1.4$). The X axis is the point index. All the points are indexed in raster order from upper right corner as shown in Fig. 9.4. Points 1 to 30 are the data from right projector and points 31 to 66 are the data from left projector.

9.2.2 Z Direction Error Analysis

As shown in Fig. 9.6, points 1 to 40 are the points on the top 3 steps of the staircase, and points 41 to 66 are the points on the lower 3 steps. Based on the sensitivity analysis in Chap. 5, displacement on projector image plane $\Delta x_p = 1$ pixel and $\Delta y_p = 1$ pixel cause approximately $\Delta x_c = 2$ pixel and $\Delta y_c = 2$ pixel on camera image plane on Z = 0 plane. Since the intersection size of vertical line and horizontal line on projector image plane is 10 pixel × 10 pixel by design. Then the intersection size on camera image plane is 20 pixel × 20 pixel according to the sensitivity analysis. If we consider an error on camera image plane $\Delta x_c = 5$ pixel and $\Delta y_c = 5$ pixel in the detection process, this causes an average error in Z direction $\overline{\Delta Z} = 1.36mm$. Since the intersection size on camera image increases on higher floors, the error in Z direction $\Delta Z = |Z_o - Z_r|$ increases as well. Therefore, we conclude that the reconstruction performs better on the lower levels than the higher levels.

9.2.3 X, Y Direction Error Analysis

As shown in Fig. 9.6, both X and Y direction error is increasing as the point index increases. Our hypothesis for this is when we measure the points' X and Y coordinates X_o and Y_o, the measurement is locally, instead of globally. The reference point we use is point No.1, and the coordinates of the reference point is measured globally. Next we measure the rest of the points' coordinates with regard to the reference point and combine with the reference's coordinates to get all points' coordinates. Based on our measuring process, the points closer to the reference point have smaller errors, while the points further away from the reference point have larger errors, which is consistent with the experiment result.

To further validate this hypothesis, we choose the last point No. 66 as our reference point and measure all the other points' coordinates with regard to the new reference point. The measurement result is show in Fig. 9.7. As we expect, the X and Y direction error decreases as the point index increases, since the last point is our

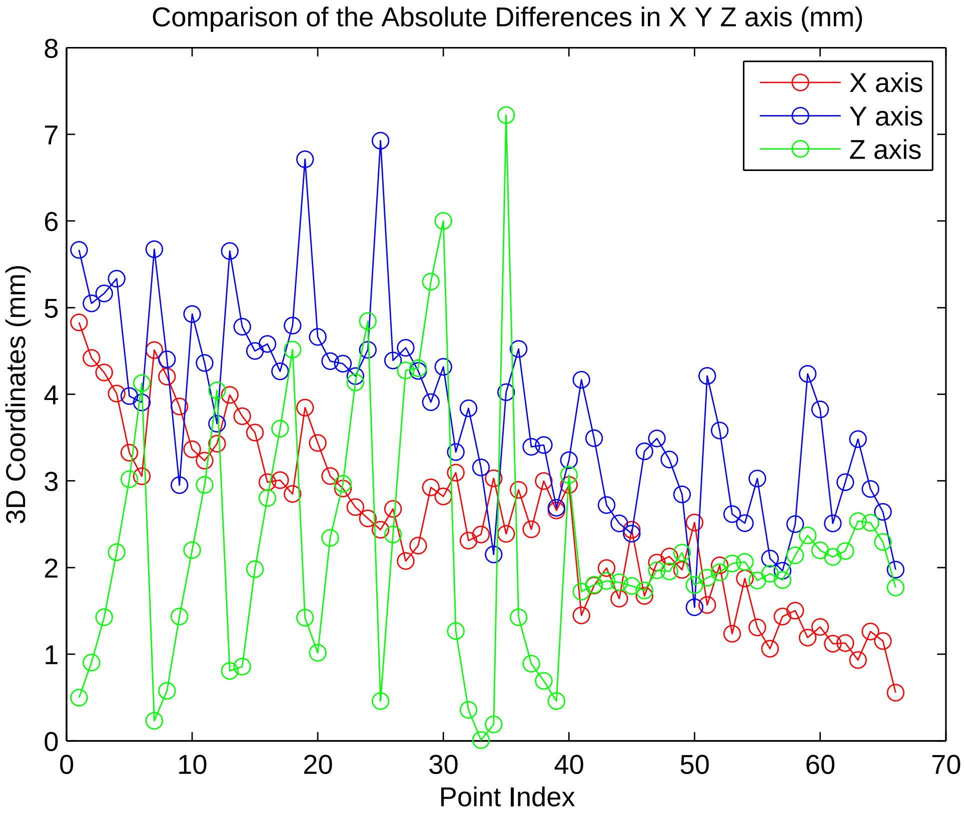


Fig. 9.7. Comparison of the absolute differences between the measured and reconstructed coordinates of the markers in X Y and Z axis, using point No.66 as reference point.

reference point.Therefore, we conclude that the point closer to the reference point has smaller error in X and Y direction and the point further away from the reference point has larger error due to our measurement procedure.

10. PERSPECTIVE CHANGE: AN APPLICATION USING 3D DEPTH MAP

3D projection can be cataloged into orthographic projection, weak perspective projection and perspective projection. 3D objects are captured by camera, or observed by human eyes in the form of perspective projection, where points further away from the image plane appears smaller, while points closer to the image plane appears larger. One example of the perspective model is the pinhole camera model. The origin of camera coordinate system is the center of projection. The ray that connects each point on the 3D object to the center of projection intersects with the image plane. The intersection of the ray and the image plane is the projection of that point from the object. Orthographic projection, in another way, does not have a center of projection. Orthographic projection only has a viewing direction instead. The ray that starts from the point on the 3D object and is parallel to the viewing direction intersects with image plane to form the projection of that 3D point. Therefore, orthographic preserves the actual dimensions of the object regardless the viewing distance, which is widely used among construction and engineering. Weak perspective projection is a limiting form of the perspective projection, which occurs when the depth of the object along viewing direction is small ($< 10\%$) compared to the viewing distance. In this case, perspective projection can be modeled as orthographic projection with a scalar that ensures that objects closer to the image plane appears smaller and objects further away from the image plane appears larger [48, 49]. In this chapter, we mainly use perspective projection and pinhole camera model.

In Chap. 6 we already introduced an application of 3D depth map which can change the perspective of an object using our first generation 3D capture system that contains one projector. In this section, we discuss a similar application for the dual-projector system.

10.1 Problem Definition and Theory

As shown in Fig. 10.1, a series of photos illustrate how different object-lens distances change the perspective of the face. Left and center column shows the perspective change and the corresponding focal length used. Varies focal lengths are only used to frame the person the same way. The right column shows exactly how the photographer capture the center column images, and how object-lens distance actually impact the perspective.

From Sec. 6.1, we already explain that the perspective is changed due to the object-lens distance. Figure. 10.2 illustrates how object-lens distance impact perspective of the object. Suppose green and red squares represent two points on the object. The original distances from lens to green square and red square are d_{go} and d_{ro}, respectively. Decrease the distances from lens to green square and red square to ${d_{go}}^{'}$ and ${d_{ro}}^{'}$, the corresponding points on the image plane are becoming larger and more distorted, comparing to the points on the original image plane, which are smaller and flatter.

10.2 Flowchart and Results

Figure. 10.3 shows the flowchart of perspective change using the 3D depth map we obtained from Sec. 8.3. After we get the 3D depth map of the object, we first convert the depth map to the object-lens (d_o)map. Then we interpolate the object-lens map (d_o) to the whole image use Shepard interpolation, and modified to a new object-lens map $({d_o}' = kd_o)$. The ratio k could be any positive number. If $k < 1$, then the object-lens distance decreases, the object become more distorted than the original. Otherwise, the object become flatter than the original. Use the new object-lens map $({d_o}')$,we can map each pixel on the original 2D image to the new image using Eq. 6.4. Last but not least, we fill in the missing pixels on the output image by locally averaging the neighbor pixels.

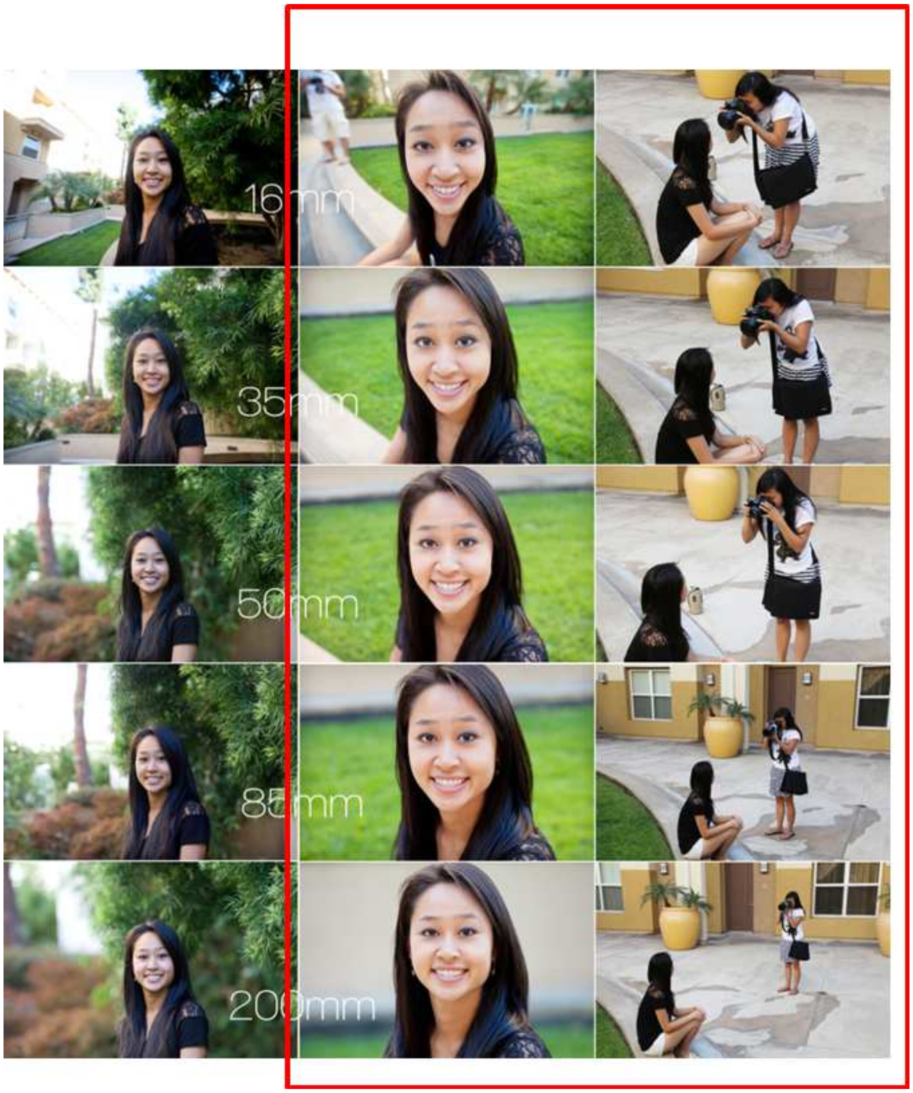
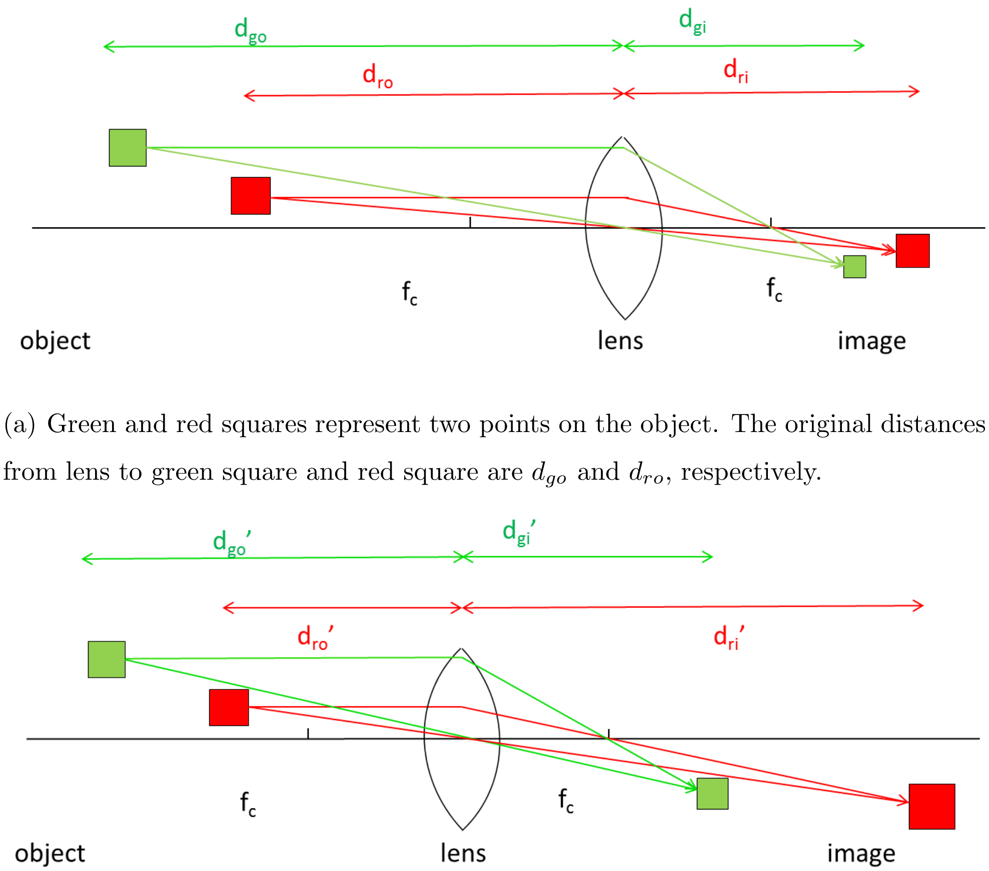


Fig. 10.1. A series of photos illustrate that different object lens distances change the perspective of the face. Left and center column shows the perspective change and the corresponding focal length used. Varies focal lengths are only used to frame the person the same way. The right column shows exactly how the photographer capture the center column images, and how object-lens distance actually impact the perspective. (photo from web)



(a) Green and red squares represent two points on the object. The original distances from lens to green square and red square are d_{go} and d_{ro}, respectively.

(b) Decrease the distances from lens to green square and red square to $d_{go}{}'$ and $d_{ro}{}'$, the corresponding points on the image plane are becoming larger and distorted

Fig. 10.2. Optics of how object-lens distance impact perspective of the object

The results of perspective change by decreasing the object-lens distance are shown in Fig. 10.4. As k decreases from 1(original image) to 0.85 and 0.55, we can see the object become more distorted and bigger. Similarly, the results of perspective change by increasing the object-lens distance are shown in Fig. 10.5. As k increases from 1(original image) to 10 and 100, we can see the object become flatter and smaller.

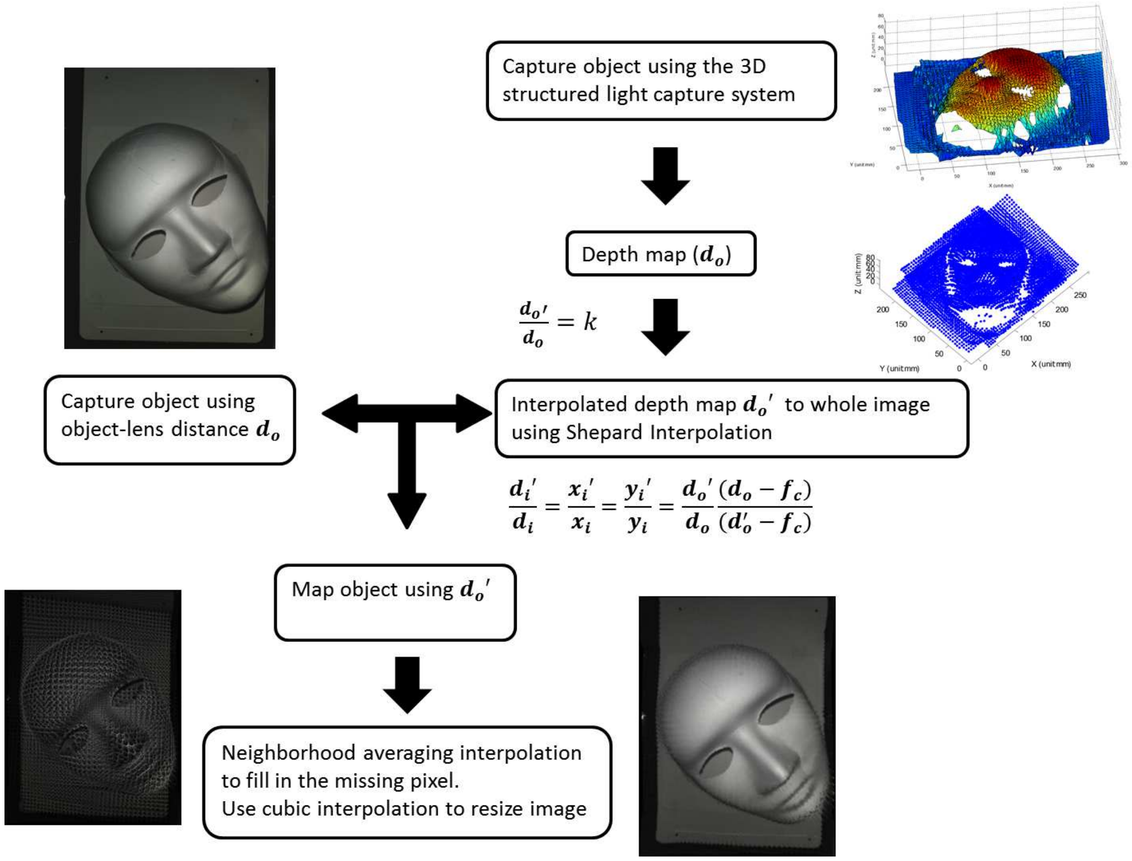
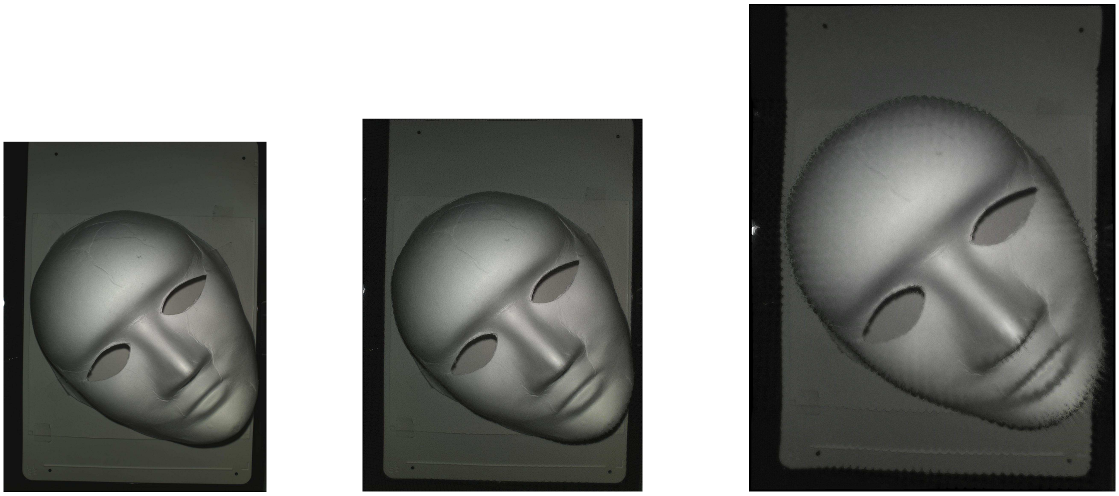
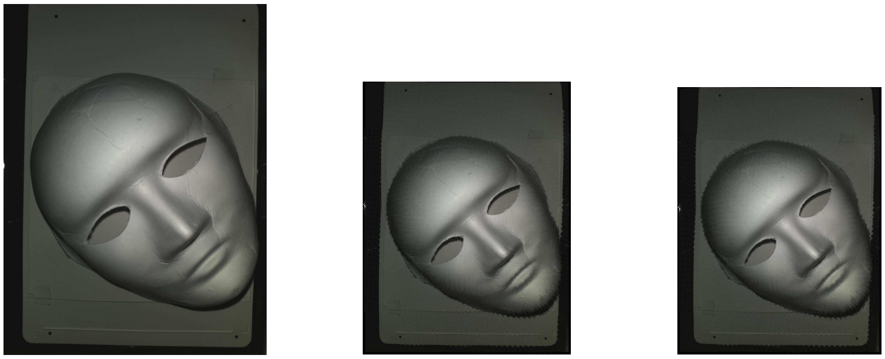

Capture object using the 3D structured light capture system
Depth map (d_o)
$\frac{d_o'}{d_o} = k$
Capture object using object-lens distance d_o
Interpolated depth map d_o' to whole image using Shepard Interpolation
$\frac{d_i'}{d_i} = \frac{x_i'}{x_i} = \frac{y_i'}{y_i} = \frac{d_o'}{d_o}\frac{(d_o - f_c)}{(d_o' - f_c)}$
Map object using d_o'
Neighborhood averaging interpolation to fill in the missing pixel.
Use cubic interpolation to resize image
Z (unit mm)
Y (unit mm)
X (unit mm)

Fig. 10.3. First, we capture the 2D reference image and 3D depth map of the object using 3D capture system. Convert the depth map to the object-lens (d_o) map. Then we interpolate the object-lens map (d_o) to the whole image use Shepard interpolation, and modified to a new object-lens map ($d_o{}' = kd_o$). Use the new object-lens map ($d_o{}'$),we can map each pixel on the original 2D image to the new image and fill in the missing pixels on the output image by locally averaging the neighbor pixels.



(a) Original image (b) $\frac{{d_o}'}{d_o} = 0.85$ (c) $\frac{{d_o}'}{d_o} = 0.55$

Fig. 10.4. Decrease the object-lens distance, the object on the image plane becomes distorted and larger



(a) Original image (b) $\frac{{d_o}'}{d_o} = 10$ (c) $\frac{{d_o}'}{d_o} = 100$

Fig. 10.5. Decrease the object-lens distance, the object on the image plane becomes flatter and smaller

11. SUMMARY

We present a novel design [50] using low-cost, readily available components to build a 3D capture system that has sufficient resolution for home and small business use. Compare to the state of art, the image capture and reconstruction processes are also time efficient since only a single shot of the projected M-array pattern is requires. The design and analysis of the dual-projector 3D capture system we present is the fundamental for next step including decoding the M-array pattern (Fig. 11.1) and reconstruction the surface [51] as well as the application using 3D depth information for paper flattening and shading correction [52]. The paper flattening results are shown in Fig. 11.2.

We describe an analytical approach to predicting the achievable resolution of the reconstructed 3D object based on differentials and small signal theory, and an experimental procedure for validating that the system under test meets the specification for reconstructed object resolution that are predicted by our analytical model. The experimental procedure is based on comparing results from the analytical sensitivity analysis, the results from the simulation codes, and analysis of the image data captured by our system.

We also present a complete reconstruction example using straight line pattern to demonstrate the functionality of our system. We introduce the pattern design, world coordinates calculation and data collecting and fusing procedure in detail. We also analyze the achievable reconstruction accuracy and explain the possible reasons for the errors. Last but not least, we show an application that can change the perspective of the object by altering the object-lens distance, which is not achievable without the 3D reconstruction data.

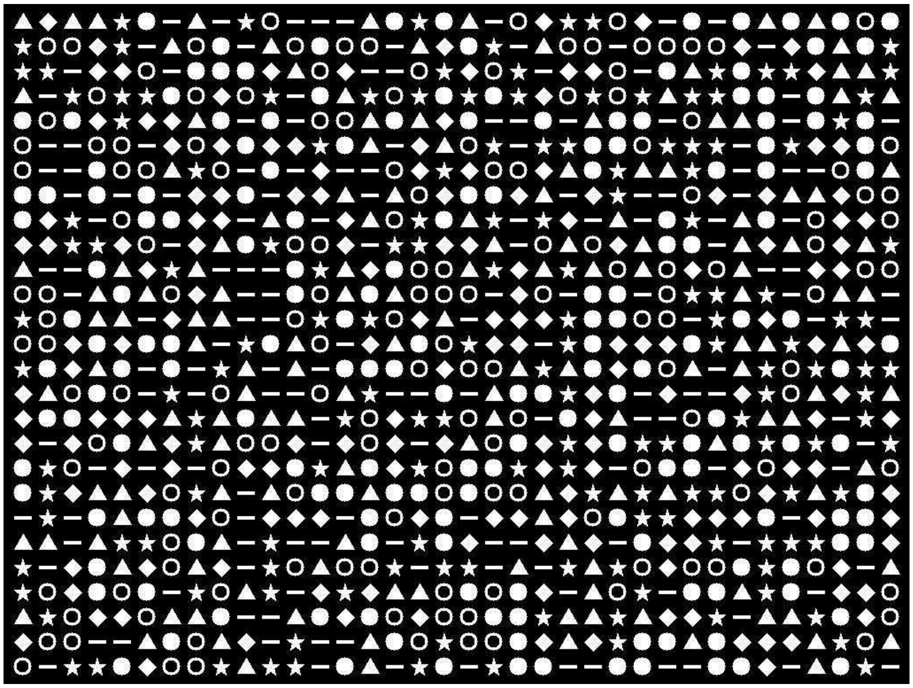
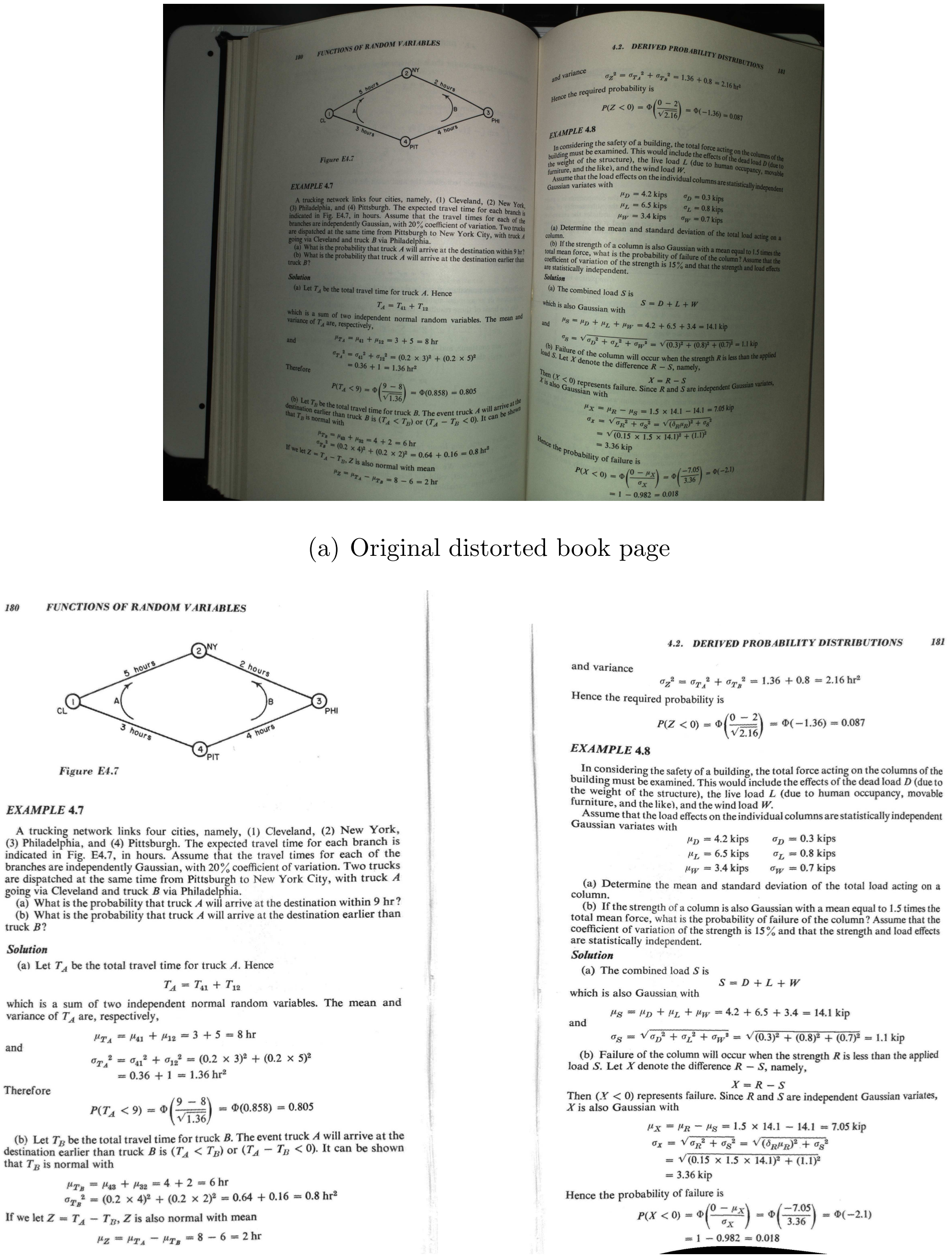


Fig. 11.1. M-array pattern designed by Yang Lei [51],is used for establish the correspondence between projector image and camera image to reconstruct the 3D surface


NY
2
5 hours
2 hours
A
B
1
CL
3
PHI
3 hours
4 hours
4
PIT

180 *FUNCTIONS OF RANDOM VARIABLES*

Figure E4.7

***EXAMPLE* 4.7**

A trucking network links four cities, namely, (1) Cleveland, (2) New York, (3) Philadelphia, and (4) Pittsburgh. The expected travel time for each branch is indicated in Fig. E4.7, in hours. Assume that the travel times for each of the branches are independently Gaussian, with 20% coefficient of variation. Two trucks are dispatched at the same time from Pittsburgh to New York City, with truck *A* going via Cleveland and truck *B* via Philadelphia.

(a) What is the probability that truck *A* will arrive at the destination within 9 hr?

(b) What is the probability that truck *A* will arrive at the destination earlier than truck *B*?

Solution

(a) Let T_A be the total travel time for truck *A*. Hence

$$T_A = T_{41} + T_{12}$$

which is a sum of two independent normal random variables. The mean and variance of T_A are, respectively,

$$\mu_{T_A} = \mu_{41} + \mu_{12} = 3 + 5 = 8 \text{ hr}$$

and

$$\sigma_{T_A}{}^2 = \sigma_{41}{}^2 + \sigma_{12}{}^2 = (0.2 \times 3)^2 + (0.2 \times 5)^2$$
$$= 0.36 + 1 = 1.36 \text{ hr}^2$$

Therefore

$$P(T_A < 9) = \Phi\left(\frac{9 - 8}{\sqrt{1.36}}\right) = \Phi(0.858) = 0.805$$

(b) Let T_B be the total travel time for truck *B*. The event truck *A* will arrive at the destination earlier than truck *B* is $(T_A < T_B)$ or $(T_A - T_B < 0)$. It can be shown that T_B is normal with

$$\mu_{T_B} = \mu_{43} + \mu_{32} = 4 + 2 = 6 \text{ hr}$$
$$\sigma_{T_B}{}^2 = (0.2 \times 4)^2 + (0.2 \times 2)^2 = 0.64 + 0.16 = 0.8 \text{ hr}^2$$

If we let $Z = T_A - T_B$, Z is also normal with mean

$$\mu_Z = \mu_{T_A} - \mu_{T_B} = 8 - 6 = 2 \text{ hr}$$

4.2. *DERIVED PROBABILITY DISTRIBUTIONS* 181

and variance

$$\sigma_Z{}^2 = \sigma_{T_A}{}^2 + \sigma_{T_B}{}^2 = 1.36 + 0.8 = 2.16 \text{ hr}^2$$

Hence the required probability is

$$P(Z < 0) = \Phi\left(\frac{0 - 2}{\sqrt{2.16}}\right) = \Phi(-1.36) = 0.087$$

***EXAMPLE* 4.8**

In considering the safety of a building, the total force acting on the columns of the building must be examined. This would include the effects of the dead load *D* (due to the weight of the structure), the live load *L* (due to human occupancy, movable furniture, and the like), and the wind load *W*.

Assume that the load effects on the individual columns are statistically independent Gaussian variates with

$$\mu_D = 4.2 \text{ kips} \qquad \sigma_D = 0.3 \text{ kips}$$
$$\mu_L = 6.5 \text{ kips} \qquad \sigma_L = 0.8 \text{ kips}$$
$$\mu_W = 3.4 \text{ kips} \qquad \sigma_W = 0.7 \text{ kips}$$

(a) Determine the mean and standard deviation of the total load acting on a column.

(b) If the strength of a column is also Gaussian with a mean equal to 1.5 times the total mean force, what is the probability of failure of the column? Assume that the coefficient of variation of the strength is 15% and that the strength and load effects are statistically independent.

Solution

(a) The combined load *S* is

$$S = D + L + W$$

which is also Gaussian with

$$\mu_S = \mu_D + \mu_L + \mu_W = 4.2 + 6.5 + 3.4 = 14.1 \text{ kip}$$

and

$$\sigma_S = \sqrt{\sigma_D{}^2 + \sigma_L{}^2 + \sigma_W{}^2} = \sqrt{(0.3)^2 + (0.8)^2 + (0.7)^2} = 1.1 \text{ kip}$$

(b) Failure of the column will occur when the strength *R* is less than the applied load *S*. Let *X* denote the difference $R - S$, namely,

$$X = R - S$$

Then $(X < 0)$ represents failure. Since *R* and *S* are independent Gaussian variates, *X* is also Gaussian with

$$\mu_X = \mu_R - \mu_S = 1.5 \times 14.1 - 14.1 = 7.05 \text{ kip}$$
$$\sigma_X = \sqrt{\sigma_R{}^2 + \sigma_S{}^2} = \sqrt{(\delta_R\mu_R)^2 + \sigma_S{}^2}$$
$$= \sqrt{(0.15 \times 1.5 \times 14.1)^2 + (1.1)^2}$$
$$= 3.36 \text{ kip}$$

Hence the probability of failure is

$$P(X < 0) = \Phi\left(\frac{0 - \mu_X}{\sigma_X}\right) = \Phi\left(\frac{-7.05}{3.36}\right) = \Phi(-2.1)$$
$$= 1 - 0.982 = 0.018$$

(a) Original distorted book page

(b) Flattened left page

(c) Flattened right page

Fig. 11.2. Paper flattening results using our first generation 3D capture prototype

REFERENCES

REFERENCES

[1] F. Blais, “Review of 20 years of range sensor development,” *Journal of Electronic Imaging*, vol. 13(1), pp. 231–243, 2004.

[2] A. Gueziec and R. Hummel, “Exploiting triangulated surface extraction using tetrahedral decomposition,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 1(4), pp. 328–342, 1995.

[3] Ø. Skotheim, J. O. Nygaard, J. Thielemann, and T. Vollset, “A flexible 3d vision system based on structured light for in-line product inspection,” in *Electronic Imaging 2008.* International Society for Optics and Photonics, 2008, pp. 680 505–680 505.

[4] B. Kusnoto and C. A. Evans, “Reliability of a 3d surface laser scanner for orthodontic applications,” *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 122, no. 4, pp. 342–348, 2002.

[5] A. Bell, A. Ayoub, and P. Siebert, “Assessment of the accuracy of a three-dimensional imaging system for archiving dental study models,” *Assessment*, vol. 30, no. 3, 2003.

[6] L. Kovacs, A. Zimmermann, G. Brockmann, M. Gühring, H. Baurecht, N. Papadopulos, K. Schwenzer-Zimmerer, R. Sader, E. Biemer, and H. Zeilhofer, “Three-dimensional recording of the human face with a 3d laser scanner,” *Journal of plastic, reconstructive & aesthetic surgery*, vol. 59, no. 11, pp. 1193–1202, 2006.

[7] K. Qian, X. Ma, F. Fang, and H. Yang, “3d environmental mapping of mobile robot using a low-cost depth camera,” in *Mechatronics and Automation (ICMA), 2013 IEEE International Conference on.* IEEE, 2013, pp. 507–512.

[8] K. Wang and Q. Yu, “Accurate 3d object measurement and inspection using structured light systems,” in *Proceedings of the 12th International Conference on Computer Systems and Technologies*, ser. CompSysTech ’11. New York, NY, USA: ACM, 2011, pp. 221–227.

[9] Z. Taha, M. A. Aris, Z. Ahmad, M. H. A. Hassan, and N. N. Sahim, “A low cost 3d foot scanner for custom-made sports shoes,” *Applied Mechanics and Materials*, vol. 440, pp. 369–372, 2014.

[10] K. Johnson, “Contact mechanics, 1985,” *Cam-bridge University Press, Cambridge*, 1974.

[11] G. Peggs, A. Lewis, and R. Leach, “Measuring in three dimensions at the mesoscopic level,” in *Proc. ASPE Winter Topical MeetingMachines and Processes for Micro-scale and Meso-scale Fabrication, Metrology and Assembly (FL, USA, January 2003)*, 2003, pp. 53–7.

[12] A. Küng, F. Meli, and R. Thalmann, “Ultraprecision micro-cmm using a low force 3d touch probe,” *Measurement Science and Technology*, vol. 18, no. 2, p. 319, 2007.

[13] M. Michihata, Y. Takaya, and T. Hayashi, “Nano position sensing based on laser trapping technique for flat surfaces,” *Measurement Science and Technology*, vol. 19, no. 8, p. 084013, 2008.

[14] H. Ji, H. Hsu, L. Kong, and A. Wedding, “Development of a contact probe incorporating a bragg grating strain sensor for nano coordinate measuring machines,” *Measurement science and technology*, vol. 20, no. 9, p. 095304, 2009.

[15] W. Dremel, G. Hausler, and M. Maul, “Triangulation with large dynamical range,” in *1986 Quebec Symposium.* International Society for Optics and Photonics, 1986, pp. 182–187.

[16] C. Chen and A. Kak, “Modeling and calibration of a structured light scanner for 3-d robot vision,” in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 807–815.

[17] F. Blais, M. Rioux, and J.-A. Beraldin, “Practical considerations for a design of a high precision 3-d laser scanner system,” in *1988 Dearborn Symposium.* International Society for Optics and Photonics, 1988, pp. 225–246.

[18] E. H. Adelson and J. Y. A. Wang, “Single lens stereo with a plenoptic camera,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 99–106, 1992.

[19] R. G. Dorsch, G. Häusler, and J. M. Herrmann, “Laser triangulation: fundamental uncertainty in distance measurement,” *Applied Optics*, vol. 33, no. 7, pp. 1306–1314, 1994.

[20] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg *et al.*, “The digital michelangelo project: 3d scanning of large statues,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques.* ACM Press/Addison-Wesley Publishing Co., 2000, pp. 131–144.

[21] F. Chen, G. M. Brown, and M. Song, “Overview of three-dimensional shape measurement using optical methods,” *Optical Engineering*, vol. 39, no. 1, pp. 10–22, 2000.

[22] J. Park, G. N. DeSouza, and A. C. Kak, “Dual-beam structured-light scanning for 3-d object modeling,” in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on.* IEEE, 2001, pp. 65–72.

[23] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[24] J. Starck and A. Hilton, “Model-based multiple view reconstruction of people,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on.* IEEE, 2003, pp. 915–922.

[25] R. Yang, M. Pollefeys, and S. Li, “Improved real-time stereo on commodity graphics hardware,” in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW’04. Conference on.* IEEE, 2004, pp. 36–36.

[26] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” *Computer Science Technical Report CSTR*, vol. 2, no. 11, 2005.

[27] J. G. D. Franca, M. A. Gazziro, A. N. Ide, and J. H. Saito, “A 3d scanning system based on laser triangulation and variable field of view,” in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 1. IEEE, 2005, pp. I–425.

[28] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” in *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 519–528.

[29] D.-I. B. Hagebeuker and P. Marketing, “A 3d time of flight camera for object detection,” *Optical 3-D Measurement Techniques*, 2007.

[30] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, “3d shape scanning with a time-of-flight camera,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1173–1180.

[31] F.-H. Cheng, C.-T. Lu, and Y.-S. Huang, “3d object scanning system by coded structured light,” in *Electronic Commerce and Security (ISECS), 2010 Third International Symposium on*. IEEE, 2010, pp. 213–217.

[32] Z. Zhang, “Microsoft kinect sensor and its effect,” *MultiMedia, IEEE*, vol. 19, no. 2, pp. 4–10, 2012.

[33] O. Wulf and B. Wagner, “Fast 3d scanning methods for laser measurement systems,” in *International conference on control systems and computer science (CSCS14)*, 2003, pp. 2–5.

[34] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, “A state of the art in structured light patterns for surface profilometry,” *Pattern recognition*, vol. 43, no. 8, pp. 2666–2680, 2010.

[35] J. Salvi, J. Pages, and J. Batlle, “Pattern codification strategies in structured light systems,” *Pattern Recognition*, vol. 37, no. 4, pp. 827–849, 2004.

[36] J. Posdamer and M. Altschuler, “Surface measurement by space-encoded projected beam systems,” *Computer graphics and image processing*, vol. 18, no. 1, pp. 1–17, 1982.

[37] M. Maruyama and S. Abe, “Range sensing by projecting multiple slits with random cuts,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 6, pp. 647–651, 1993.

[38] P. Vuylsteke and A. Oosterlinck, “Range image acquisition with a single binary-encoded light pattern,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 2, pp. 148–164, 1990.

[39] H. Morita, K. Yajima, and S. Sakata, “Reconstruction of surfaces of 3-d objects by m-array pattern projection method,” in *Computer Vision., Second International Conference on*. IEEE, 1988, pp. 468–473.

[40] T. Etzion, "Constructions for perfect maps and pseudorandom arrays," *Information Theory, IEEE Transactions on*, vol. 34, no. 5, pp. 1308–1316, 1988.

[41] D. Lanman and G. Taubin, "Build your own 3D scanner: 3D photography for beginners," in *SIGGRAPH '09: ACM SIGGRAPH 2009 courses.* New York, NY, USA: ACM, 2009, pp. 1–87.

[42] J. Geng, "Structured-light 3D surface imaging: a tutorial," *Adv. Opt. Photo*, vol. 3(2), pp. 128–160, 2011.

[43] Y. Lei, P. Majewicz, K. Bengtson, L. Li, and J. Allebach, "Composite target for camera-based document/object capture system," *Journal of Imaging Science and Technology*, vol. 57, pp. 030 404–1–030 404–14, 2013.

[44] X. Chen, J. Xi, Y. Jin, and J. Sun, "Accurate calibration for a camera–projector measurement system based on structured light projection," *Optics and Lasers in Engineering*, vol. 47, no. 3, pp. 310–319, 2009.

[45] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22(11), pp. 1330–1334, 2000.

[46] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.

[47] D. Shepard, "A two-dimensional interpolation function for irregularly spaced data," in *Proceedings of the 1968 ACM National Conference*, 1968, pp. 517–524.

[48] J. L. Mundy, A. Zisserman *et al.*, *Geometric invariance in computer vision.* MIT press Cambridge, 1992, vol. 92.

[49] Z. Pizlo, Y. Li, T. Sawada, R. M. Steinman *et al.*, *Making a machine that sees like us.* Oxford University Press, 2014.

[50] J. Dong, K. R. Bengtson, B. F. Robinson, and J. P. Allebach, "Low-cost structured-light based 3d capture system design," in *IS&T/SPIE Electronic Imaging.* International Society for Optics and Photonics, 2014, pp. 90 130B–90 130B.

[51] Y. Lei, K. Bengtson, L. Li, and J. Allebach, "Design and decoding of an M-array pattern for low-cost structured light 3D reconstruction systems," in *IEEE International Conference on Image Processing*, 2013.

[52] C.-T. Wu and J. P. Allebach, "Shading correction of camera captured document image with depth map information," in *IS&T/SPIE Electronic Imaging.* International Society for Optics and Photonics, 2015, pp. 939 505–939 505.

VITA

VITA

Jing Dong is a Ph.D. candidate in Electrical and Computer Engineering at Purdue University, under the supervision of Prof. Jan P. Allebach. She received her B.S. degree in Electrical Engineering in 2009 from Xi'an Jiaotong University, Xi'an, China and her M.S. degree in Electrical and Computer Engineering in 2011 from Purdue University, West Lafayette. She worked as a consultant for a start-up coffee company in the spring of 2013, correcting camera lens distortion and generating orthophoto to measure particle sizes based on cellphone captured images. She was also an imaging science intern at Hewlett-Packard Labs, Palo Alto, CA, during the summer of 2014, working on building a document retrieval system for cellphone captured images. Jing's research interests include computer vision and image processing.