

2004

# Towards automated capture of 3D foot geometry for custom orthoses

Rajeev Madazhy

Louisiana State University and Agricultural and Mechanical College, [rmadaz1@lsu.edu](mailto:rmadaz1@lsu.edu)

Follow this and additional works at: [https://digitalcommons.lsu.edu/gradschool\\_theses](https://digitalcommons.lsu.edu/gradschool_theses)

 Part of the [Mechanical Engineering Commons](#)

---

## Recommended Citation

Madazhy, Rajeev, "Towards automated capture of 3D foot geometry for custom orthoses" (2004). *LSU Master's Theses*. 1440.  
[https://digitalcommons.lsu.edu/gradschool\\_theses/1440](https://digitalcommons.lsu.edu/gradschool_theses/1440)

This Thesis is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Master's Theses by an authorized graduate school editor of LSU Digital Commons. For more information, please contact [gradetd@lsu.edu](mailto:gradetd@lsu.edu).

**TOWARDS AUTOMATED CAPTURE OF 3D FOOT GEOMETRY FOR  
CUSTOM ORTHOSES**

**A Thesis**

**Submitted to the Graduate Committee of the  
Louisiana State University and  
Agricultural and Mechanical College  
in partial fulfillment of the  
requirement for the degree of  
Master of Science in Mechanical Engineering**

**in**

**The Department of Mechanical Engineering**

**by  
Rajeev Madazhy  
B.S., Bangalore University, India, 2000  
December, 2004**

***'Mata, Pita, Guru & Daivam (Mother, Father, Teacher & God).  
It is the best combination of the three that manifests the fourth'***

*-Sri Ramanujacharya, Philosopher (11th century AD).*

*Dedicated to my parents, and teachers throughout my academic life....*

## Acknowledgements

First, I would like to thank my parents, Vijayam and K. Venugopalan, for their love and support. My dad has always been an inspiration for me to pursue the path of science as a career. I have been able to achieve all of the goals I have set for myself due to their blessings and support. Additional thanks to my brother, Sujesh whose guidance and support over the years is cherished immeasurably.

I would like to express my sincere gratitude to my major professor and advisor Dr. Warren N. Waggenpack Jr. for lending his valuable time and patience to teach and guide me, without whom this work would not have been possible. His academic and professional guidance has vastly affected this research. I would also like to thank the members of my graduate committee, Doctors Michael Murphy and Dimitris Nikitopoulos. Thanks to Dr. Nikitopoulos for his suggestions on optimizing the camera calibration process.

I would like to thank Dr. Bill Coleman, who was vastly resourceful in providing required data necessary for this research. I must sincerely thank the senior design project members, Dwayne Bajoie, Alp Eminsel and Pablo Guardado for building the foot scanner utility frame, required to capture images of foot.

Of the members in the lab, I must thank Hemant Khatod, Venugopal Jogi, Rohan Panchadhar and Michael Crochet for their friendship and contributions throughout my graduate student life. It was a joy working with them and will always treasure the memorable moments we had during this time.

Last but not the least, I would like to thank the Mechanical Engineering Department, Louisiana State University for all the support I have received during this period.

## Table of Contents

<b>Acknowledgements.....</b>	<b>iii</b>
<b>List of Tables.....</b>	<b>vii</b>
<b>List of Figures.....</b>	<b>viii</b>
<b>Abstract.....</b>	<b>xi</b>
<b>Chapter 1: Introduction.....</b>	<b>1</b>
<b>Chapter 2: Background and Literature Review.....</b>	<b>3</b>
2.1 Introduction.....	3
2.2 Diabetic Neuropathy.....	3
2.3 Custom Orthoses.....	5
2.4 Casting for Foot Orthoses.....	6
2.5 Limitations of Manual Cast Process.....	8
2.6 Need for Automation of Foot Orthoses.....	8
2.7 Automated Orthotic Process.....	9
2.8 Limitations of Current Laser Foot Scanners.....	10
2.9 Mathematics of 3D Vision.....	11
2.9.1 Projective Geometry.....	12
2.9.1.1 Homogenous Coordinates and Other Definitions.....	12
2.9.1.2 Projective Plane.....	13
2.9.1.3 Projective Space.....	14
2.9.1.4 Summary.....	14
2.9.2 Affine Geometry.....	15
2.9.2.1 Affine Plane.....	15
2.9.2.1.1 Transformations.....	15
2.9.2.2 Affine Space.....	15
2.9.2.2.1 Transformations.....	16
2.9.3 Metric Geometry.....	16
2.9.3.1 Metric Plane.....	16
2.9.3.2 Metric Space.....	16
2.9.4 Euclidean Geometry.....	17
2.10 Conclusions.....	17
<b>Chapter 3: 3D Foot Scanner Utility Frame.....</b>	<b>18</b>
3.1 Introduction.....	18
3.2 Foot Scanner Utility Frame.....	18
3.3 Experimental Setup.....	21
3.4 Advantages of Using the Foot Scanner Utility Frame.....	21
3.5 Graphical User Interface to Generate 3D Point Geometry.....	21
3.6 Objectives and Novel Aspects of Current Method.....	22
3.7 Conclusions.....	23

<b>Chapter 4: Camera Calibration and Epipolar Geometry.....</b>	<b>24</b>
4.1 Introduction.....	24
4.2 Camera Model.....	24
4.3 Epipolar Geometry.....	26
4.4 Camera Calibration Matrix.....	27
4.5 Calibration Using a Planar Check Board.....	28
4.6 Calculating the Camera Calibration Matrix.....	30
4.7 Calibration Results.....	31
4.8 Conclusions.....	34
<b>Chapter 5: Feature Detection.....</b>	<b>35</b>
5.1 Introduction.....	35
5.2 Review of Earlier Works on Feature Detection.....	35
5.3 Corner Detection Using Harris-Stephens Algorithm.....	37
5.4 Drawbacks of Harris-Stephens Algorithm.....	38
5.5 Corner Detection Using Combined Harris-Stephens & Color Intensity Algorithm..	38
5.6 Results.....	38
5.6.1 Harris-Stephens Algorithm.....	38
5.6.2 Combined Harris-Stephens and Color Intensity Algorithm.....	41
5.7 Conclusions.....	42
<b>Chapter 6: Feature Matching.....</b>	<b>43</b>
6.1 Introduction.....	43
6.2 Review of Earlier Works.....	43
6.3 Results.....	49
6.3.1 Results Using Intensity Cross-correlation Algorithm.....	50
6.3.2 Results Using Scott, Longuet-Higgins' Algorithm.....	51
6.3.3 Results Using Pilu's Algorithm.....	52
6.3.4 Results Using Zhang's Algorithm by Recovering Epipolar Geometry.....	53
6.3.5 Results Using Fixed Configuration of Cameras.....	58
6.3.5.1 Case 1.....	58
6.3.5.2 Case 2.....	59
6.3.6 Feature Matching Using Homography Matrix.....	60
6.3.7 Results Using Homography Matrix.....	61
6.4 Conclusions.....	62
<b>Chapter 7: 3D Reconstruction.....</b>	<b>63</b>
7.1 Introduction.....	63
7.2 Assumptions Made to Eliminate Reconstruction Ambiguity.....	63
7.3 Steps Towards Euclidean Reconstruction.....	64
7.4 Eight Point Algorithm.....	64
7.5 3D Structure Computation.....	66
7.6 Results.....	68
7.6.1 Reconstruction of a Shelf-File.....	68
7.6.2 Reconstruction of a Scaled Box using a Reference Cube.....	70
7.6.3 Reconstruction of a Plaster Foot using a Reference Cube.....	74

7.6.4 Reconstruction of an Actual Foot Using a Reference Cube.....	76
7.6.5 Reconstruction of Varied Point Distances on Foot.....	78
7.7 Conclusions.....	81
<b>Chapter 8: Conclusions and Future Work.....</b>	<b>82</b>
8.1 Summary and Conclusions.....	82
8.2 Future Work.....	82
<b>Bibliography.....</b>	<b>84</b>
<b>Vita.....</b>	<b>88</b>

## List of Tables

Table 4.1 Calibration Result Set for Case 1.....	32
Table 4.2 Calibration Result Set for Case 2.....	34
Table 5.1 Results of Points Detected Using Harris-Stephens Algorithm.....	39
Table 5.2 Results of Points on Actual Foot with General Background Using Harris-Stephens Algorithm.....	40
Table 5.3 Results of Points Detected on Actual Foot with General Background Using Combined Algorithm.....	42
Table 6.1 (x, y) Coordinates of 8 Corresponding points in Image 1 and Image 2.....	54
Table 7.1 Distances between Points in Reference Cube.....	72
Table 7.2 Comparison of Scaled Distances and Actual Distances of the Object.....	73
Table 7.3 3D Scaled Coordinates (x,y,z) of the Foot.....	77
Table 7.4 Varied Point Sizes and Distances Marked on Foot.....	78
Table 7.5 Unscaled (x,y,z) Coordinates of 48 Points.....	80



## List of Figures

Figure 2.1 Foot Anatomy.....	5
Figure 2.2 (a-d) Manual Casts for Foot Orthoses.....	7
Figure 3.1 Examination Room.....	19
Figure 3.2 Utility Frame.....	19
Figure 3.3 Graphical User Interface (GUI).....	23
Figure 4.1 Perspective Projection.....	25
Figure 4.2 Epipolar Geometry.....	26
Figure 4.3 Illustration of Pixel Skew.....	28
Figure 4.4 (1-4) Steps Involved During the Calibration Process.....	29
Figure 4.5 Final Extracted Corners on the Chess Board.....	32
Figure 4.6 Second Trial Using a Different Chess Board.....	33
Figure 5.1 Marked Points Detected in an Ideal Image.....	39
Figure 5.2 Corners Detected Vs Threshold Value for Image 1.....	40
Figure 5.3 Result of Points Detected on Actual Foot Image with General Background.	40
Figure 5.4 Result of Points Detected on the Same Image Using Combined Algorithm..	41
Figure 6.1 Epipolar Geometry between Points in Images and the 3D Point in Space.....	48
Figure 6.2 Initial Matches Using Intensity Cross-Correlation.....	50
Figure 6.3 Plot of Search Window Size Vs Number of Matches Using Correlation Algorithm.....	51
Figure 6.4 Feature Matching Using Scott, Longuet-Higgins' Algorithm.....	51
Figure 6.5 Feature Matching Using Pilu's Algorithm.....	52
Figure 6.6 Initial Correspondence Using Intensity Cross-Correlation.....	53

Figure 6.7 Point 1 in Image 1 and Epipolar Line Passing Through Corresponding Point in Image 2.....	55
Figure 6.8 Point 2 in Image 1 and Epipolar Line Passing Through Corresponding Point in Image 2.....	55
Figure 6.9 Point 3 in Image 1 and Epipolar Line Passing Through Corresponding Point in Image 2.....	55
Figure 6.10 All 8 Points in Image 1 and Epipolar Lines Passing Through the Corresponding Points in Image 2.....	56
Figure 6.11 All Points in Image 1 and Epipolar Lines Passing Through the Corresponding Points in Image 2.....	56
Figure 6.12 Final Feature Matching after Applying Epipolar Constraint to Two Images.....	57
Figure 6.13 Case 1: Feature Matching of Points on Plaster Foot by Recovering Fundamental Matrix.....	58
Figure 6.14 Case 2: Feature Matching of Points on Plaster Foot by Recovering Fundamental Matrix.....	59
Figure 6.15 Feature Matching of Points on Plaster Foot Using Homography Matrix.....	62
Figure 7.1 Dimensions of Shelf-File.....	69
Figure 7.2 Marked Corners on Shelf-File on Both Images.....	69
Figure 7.3 3D Points of Shelf-File.....	70
Figure 7.4 Marked Points on Object and Reference Cube .....	71
Figure 7.5 3D points of Object and Reference Cube.....	71
Figure 7.6 Comparisons of Scaled Distances and Actual Distances of Box.....	73
Figure 7.7 3D Wire Frame Model of Scaled Object and Reference Cube.....	74
Figure 7.8 Marked Points on Plaster Foot and Reference Cube.....	74
Figure 7.9 3D Point Cloud of Plaster Foot and Reference Cube.....	75
Figure 7.10 Surface Interpolated 3D points of Plaster Foot and Reference Cube.....	75

Figure 7.11 Triangulated Patches between 3D Points of Plaster Foot.....	75
Figure 7.12 Surface Interpolated 3D points of Plaster Foot.....	75
Figure 7.13 Marked Points on Actual Foot and Reference Cube.....	76
Figure 7.14 3D Point Cloud of Actual Foot and Reference Cube.....	76
Figure 7.15 Surface Interpolated 3D points of Actual Foot and Reference Cube.....	76
Figure 7.16 Triangulated Patches between 3D Points of Actual Foot.....	77
Figure 7.17 Surface Interpolated 3D points of Actual Foot.....	77
Figure 7.18 Feature Detection of Marked Points using Harris Detector Algorithm.....	78
Figure 7.19 Feature Detection of Marked Points using Combined Algorithm.....	79
Figure 7.20 Feature Matching of Marked Points using Homography Matrix.....	79
Figure 7.21 Two Views of the Reconstructed 3D Points.....	80

## Abstract

This thesis presents a novel method of capturing 3D foot geometry from images for custom shoe insole manufacture. Orthopedic footwear plays an important role as a treatment and prevention of foot conditions associated with diabetes. Through the use of customized shoe insoles, a podiatrist can provide a means to better distribute the pressure around the foot, and can also correct the biomechanics of the foot.

Different foot scanners are used to obtain the geometric plantar surface of foot, but are expensive and more generic in nature. The focus of this thesis is to build 3D foot structure from a pair of calibrated images. The process begins with considering a pair of good images of the foot, obtained from the scanner utility frame. The next step involves identifying corners or features in the images. Correlation between the selected features forms the fundamental part of epipolar analysis. Rigorous techniques are implemented for robust feature matching. A 3D point cloud is then obtained by applying the 8-point algorithm and linear 3D triangulation method.

The advantage of this system is quick capture of foot geometry and minimal intervention from the user. A reconstructed 3D point cloud of foot is presented to verify this method as inexpensive and more suited to the needs of the podiatrist.

## Chapter 1

### Introduction

Orthopedic footwear plays an important role as a therapeutic and preventive measure against foot conditions associated with diabetes. Through the use of customized shoe insoles, a podiatrist can provide a means to better distribute the pressure around the foot, and can also correct the biomechanics of the foot. If caught early enough, orthopedic insoles can correct or prevent further complications from occurring. In order to construct customized shoe insoles, a 3D model of the patient's foot is required.

The current process for constructing orthopedic shoe insoles involves applying a plaster sock to capture the shape or negative cast of the foot. Next, from that sock, a plaster mold is made to provide a positive image of the foot. Finally the podiatrist fabricates the insole over the mold using specialty materials. The time to produce just a plaster sock of the feet is approximately one and half hour. The ideas described here support automating this process with a digital means of capturing the shape of the foot, thereby reducing the modeling time considerably.

Existing commercial scanners that generate 3D models are limited by high cost and/or too long a scan time for the intended application. This project seeks a cheaper and more specific alternative to generate a 3D model of the foot. Recently, a flexible 3D foot scanner utility frame [BAJOIE2003] was built. It is a flexible set up, positioning inexpensive cameras, providing ambient lighting conditions and hardware required to obtain pairs of distinct 2D images of the plantar surfaces of a patient's foot, for use in capturing 3D foot geometry.

The focus of this thesis is to build 3D foot structure from a pair of calibrated images. It consists of three basic steps. The process begins with considering a pair of good images obtained

from the scanner utility frame. The next step involves identifying corners or features in the image pair. Correlation between the features selected from these images forms the fundamental part of epipolar analysis. Rigorous techniques are implemented for robust feature matching. By recovering the epipolar geometry and estimating the homography matrix, all points on the foot in the first image are matched to the corresponding points in the second image. A 3D point cloud is then obtained by applying the 8-point algorithm and 3D triangulation [MA2001] algorithm. Results of the 3D point geometry of different foot samples are presented. All results are scaled to match dimensions of the model, by determining the scale factor in each case, using a structured model whose dimensions are known.

Chapter 2 begins with the introduction to diabetic neuropathy, aspects of the manual casting process, introduces the need for automation and reviews different automated techniques to achieve the objective. It also summarizes the mathematics of 3D vision. The foot scanner utility frame to capture the images of the foot and the software interface used in generating the 3D point geometry of foot is described in the third chapter. Those familiar with these topics may skip to chapter 4 which deals with camera calibration and determining the intrinsic calibration matrix. Chapter 5 describes the process of feature detection of points in the image using the combined Harris-Stephens and color intensity algorithm. Different aspects of point matching and choosing simplified Zhang's algorithm to carry out feature correspondence is described in the sixth chapter. The seventh chapter deals with Euclidean 3D reconstruction process with results of 3D point cloud as well as surface interpolated foot models are presented. The last chapter concludes the thesis with suggestions and a brief description of future work.

## Chapter 2

### Background and Literature Review

#### 2.1 Introduction

Diabetic neuropathy traditionally is considered progressive and irreversible, and will result in lower extremity ulceration and amputation in a segment of the diabetic population despite the best efforts to control serum glucose levels. Among preventive measures are using custom footwear which reduce discomfort, increase blood circulation and help achieve good contact support [VALMASSY1996].

Conventional methods of manufacturing custom insoles involve lesser accuracy of plantar surface, more time to manufacture and are labor intensive. Newer methods of using laser scanners and footpads reduce time and effort but cost more and are less accommodating of podiatrist's needs.

This chapter begins exploring the effects of diabetic neuropathy and the need for custom inserts in shoes to prevent leg amputation. It then describes the manual casting process needed to capture the plantar surface of foot and its drawbacks. A brief description of different existing automated methods to capture foot geometry is presented next. Lastly, an introduction to mathematics of 3D vision is developed.

#### 2.2 Diabetic Neuropathy

Diabetes can cause damage to the nerve and vascular supply in the feet and legs. Patients with neuropathy have reduced or no sensation and, therefore, might be unaware of any trauma to their feet caused by ill-fitting footwear or foreign objects in their shoes. Persons with diabetes often have circulation disorders (peripheral vascular disease) that can cause cramping in the calf or buttocks when walking.

Some statistics related to foot problems of diabetic patients include the following [ADA2002]:

- out of 4 Americans experience serious foot problems in their lifetime;
- About 60-70% of people with diabetes have mild to severe forms of diabetic nerve damage which in severe forms can lead to lower limb amputation;
- Approximately 56,000 Americans a year lose their foot or leg to diabetes.

Costs associated with diabetic peripheral neuropathy and its resulting complication were shown by Gordois in his paper in 1993[GORDOIS2003] and include:.

- 3182 patients underwent foot amputations in 2001 incurring a cost of \$40,000 for Type I diabetes;
- 39,242 patients underwent foot amputations in 2001 incurring a cost of approximately \$38,000 for Type II diabetes;
- Long term costs for treatment (1-3 years) are between \$40,000 - \$60,000 per patient.

The possible outcomes that result due to improper diabetic foot care begin with the insensitive foot which frequently collapses and widens. Repeated swelling and redness, mild to moderate aching, and an inability to adequately fit into regular shoes, often first herald this destructive condition. Neuropathy allows lesions to develop and go unrecognized because the normal warning sense of pain has been lost. Continued pressure or walking on the injured skin creates even further damage and the ulcer worsens. The open sore will frequently become infected and may even penetrate to bone, which may ultimately require foot amputation [VALMASSY1996].



General precaution techniques include avoiding bare foot exposure and using the right custom made insoles or shoe inserts. Corrective foot surgery is an option when the patient is generally in good health and has good circulation.

### 2.3 Custom Orthoses

Normal footwear does not accommodate the biomechanics of the foot [VALMASSY1996]. Every patient has a different foot shape and the peak stress distribution is different for different feet. They impair blood circulation around the plantar surface of foot, leading to numbness or loss of sensation.

Custom orthoses generally tend to decrease the amount of abnormal stress and strain on the lower extremity. Abnormalities secondary to joint malfunction and muscle tendinous malposition are successfully addressed via functional orthoses. A functional foot orthosis best achieves its goal by maintaining normal function at the level of the subtalar and metatarsal joints (see Fig. 2.1), thereby allowing improved functioning for the more distal and proximal articulations of the lower extremity.

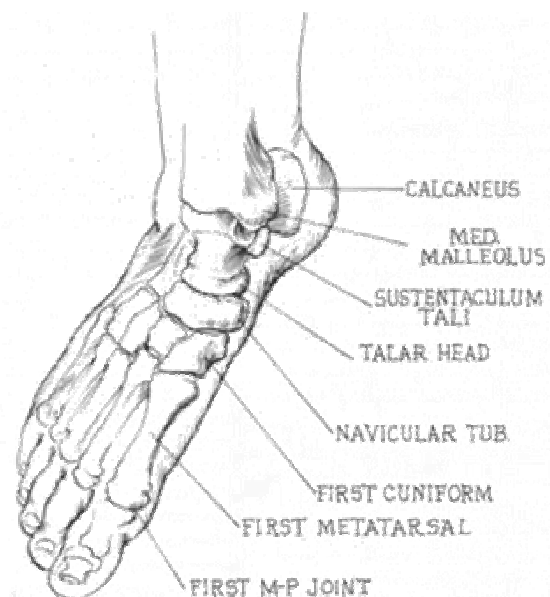


Fig. 2.1 Foot Anatomy (image from <http://www.savingfeet.com>)

As noted by Sgarlatto [SGARLATTO1972], properly fabricated orthoses help not only to improve abnormal foot function but also produce the following effects as listed below:

- Normal ankle joint dorsiflexion and plantar flexion;
- Normal knee flexion at the moment of heel contact for more efficient shock absorption;
- Proper hip flexion and extension;
- Efficient internal and external lower extremity motion;
- Proper subtalar and midtarsal joint pronation and supination.

Once properly fabricated, functional and accommodative orthoses have the ability to effectively reduce and eliminate painful excrescences and dermatologic hypertrophied lesions beneath weight bearing pressure points.

#### **2.4 Casting for Foot Orthoses**

The most essential part in this process is the ability to produce an accurate negative cast with the subtalar joint neutral and the midtarsal joint fully pronated about both axes of motion. Valmassy in his book [VALMASSY1996] notes that this is achieved by first positioning the patient on the examining chair or table with the knee slightly flexed (See Fig. 2.2a), which prevents undesirable movement as the foot is held in the casting position. Following the positioning process, plaster bandage splints (5 X 30 in.) are applied to the patient's foot. The plaster splints are then layered circumferentially around the medial, lateral, and plantar aspect of the foot. The foot is held in the proper position as plaster dries.

Once the impression is set, it is safe to remove the plaster cast from the foot. All contours seen in the plaster impression cast are compared with the foot. Once an acceptable cast is obtained, the impressions are allowed to dry sufficiently, before shipping to an orthotic laboratory.



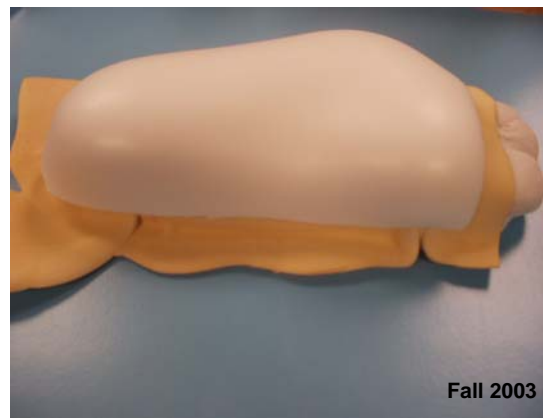
**(a) Podiatrist positions the foot**



**(b) Plaster is applied on the plantar surface of the foot**



**(c) Negative and positive cast of feet**



**(d) Custom insole manufacture**

**Fig. 2.2 Manual Casts for Foot Orthoses. (Courtesy: Dr. William Coleman, New Orleans)**

Once the laboratory receives the plaster sock negative, the orthotic fabrication process begins with inscribing a reference line, bisecting the posterior surface of the cast. A coating is applied to the interior surface to prevent the negative cast from adhering to the positive cast during the curing process. Liquefied plaster is then poured to fill interior of the cast, which is positioned to ensure the heel bisector remains vertical as it solidifies.

The orthosis is fabricated by first outlining the dimensions of the cast on the orthotic blank. Excess material around this perimeter can be removed to facilitate the pressing process. With acrylics or polyethylene foam-like materials, a convection oven can be used to evenly heat the blank to achieve the desired flexibility. Once sufficiently heated, the blank is aligned with the positive cast and formed under mechanical or vacuum pressure until cooled enough to hold its shape. Once formed, the orthosis is fashioned with a grinder until it meets the predetermined dimensions of the positive cast. The orthosis finish is obtained by polishing and buffing.

## **2.5 Limitations of Manual Cast Process**

The manual orthotic process is a time consuming and dirty process. It usually takes about 45 minutes to apply the plaster sock on the foot and takes around 15 – 20 minutes to dry and remove it from the foot. It then takes approximately about an hour to produce a positive foot image from the negative cast. The final orthotic is made through another lengthy process.

Apart from the time element, the geometry of the cast does not exactly define the plantar surface of the foot due to inherent errors in the process [SGARLATTO1971]. Sgarlatto suggested this is due to the unexpected drying nature of the material used, faster in certain areas of the foot compared to others, resulting in inaccurate geometry of the foot.

## **2.6 Need for Automation of Foot Orthoses**

The advantages of automating the process are numerous. Image based scanning takes considerably less time compared to the manual process allowing physicians to focus their time on additional patients. Current foot scanners (laser based foot scanners or image based foot scanners) available in the market capture the 3D foot geometry much more quickly and accurately than the manual process. Any necessary changes in foot scan are easier and faster than

using the manual process. Multiple scans can be performed to obtain an accurate plantar surface of foot.

## **2.7 Automated Orthotic Process**

Most automated processes begin with the negative scan of the foot taken using a scanner. The raw scan is then processed using a standard software interface to determine whether the plantar surface of the foot matches in close proximity to the points scanned. Redundant points can be removed and critical points that were not scanned can be interpolated from existing data. The final foot model is then sent to a CNC machine to mill the custom insole from a propylene board. Some general-purpose foot scanners that are currently used are described over the next few paragraphs.

### **• Polhemus FastSCAN**

FastSCAN is a hand held scanner [POLHEMUS2001] for which the setup time is minimal. However experience with this shows that it takes time, skill and proper lighting to get good scans. FormThotics, an insole manufacturer first used FastSCAN for orthotic purposes. Its software has features such as polygon reduction, which reduces redundant data from the model. It scans at a high level of detail throughout the foot, except for certain concavities near the toe area. The cost of the scanner is around \$30,000 per unit.

### **• INFOOT-foot scanner**

INFOOT foot scanner [KOUCHI2001] has 8 CCD cameras and 4 laser projectors. It is capable of scanning the foot in 8 seconds. Cross-sectional data can be measured at 1 mm intervals. 3D Coordinates of anatomical landmarks can also be recognized using special markers, which do not reflect the laser. After automatic recognition and detection of landmarks with INFOOT, seventeen distinct foot measurements can be extracted. The software interface uses a

Free Form Deformation (FFD) technique [BLOOMENTHAL1997] to deform the pre existing data to the target foot data. The scanner costs around \$40,000.

- **Foot Scanner**

Foot scanner [SHAPE GRABBER2002] creates a complete 3D image of foot. It takes about 17 seconds to scan an entire foot. The scan speed is 10590 points per second. The deployment time is around 10 minutes. This scanner is custom built for shoe insole manufacture.

- **Footcare Express**

Footcare Express [FOOTCARE2001] gives podiatrists details about the dynamic distribution of pressure on foot. All functionalities like scanner, software interface and the milling machine are packaged as one unit for obtaining maximum efficiency. This orthotic foot scanner is more expensive than most scanners (around \$45,000), but also produces an orthotic.

## **2.8 Limitations of Current Laser Foot Scanners**

There are several limitations associated with image based laser scanners. Some of the prominent ones are listed below:

- Current commercially available scanners cost around \$30,000 - \$45,000;
- Hand held scanners take too long a time to scan the foot, longer than a typical patient can hold their foot still;
- Some scanners are bulky and need considerable startup time;
- Effective insoles require proper positioning of foot and this problem is not addressed by any available scanners.

The method proposed in this thesis deals with capturing 3D foot geometry from 2D images using pairs of inexpensive cameras. All techniques related to 3D reconstruction are performed to obtain a reasonable set of 3D point data of foot. The advantages related to this

process include being cost effective and more suitable to the needs of the podiatrist. Further chapters in the thesis describe the process in detail.

## **2.9 Mathematics of 3D Vision**

This section introduces certain concepts required to understand different geometries related to the 3D reconstruction process. The entire 3D space can be described using 4 hierarchical levels of geometry – *Euclidean, Metric, Affine* and *Perspective*. The most general group is projective geometry, which forms the superset of all other groups. Successive subgroups include affine, then metric and lastly euclidean geometry.

Each of these geometries has an associated set of transformations, which leave certain data properties invariant. It is invariants, when recovered that allow for upgrading the model data to more specific or detailed level geometry. Each of these geometries beginning from Section 2.9.1 is explained in terms of their invariants and transformations.

Projective geometry allows for perspective projections, and as such models the imaging process very well. Having a model of this perspective projection, it is possible to upgrade the projective geometry later to Euclidean, via the affine and metric geometries. Algebraic and projective geometry forms the framework for most computer vision tasks, especially in the fields of *3D reconstruction from images* and *camera self-calibration*.

A classical text covering all aspects of projective and algebraic geometry is by Semple and Kneebone [SEMPLE1979]. Faugeras applies principles of projective geometry to 3D vision and reconstruction [FAUGERAS1993]. Other good introductions to projective geometry are by Mohr and Triggs [MOHR1996] and by Birchfield [BIRCHFIELD1988].

## 2.9.1 Projective Geometry

### 2.9.1.1 Homogenous Coordinates and Other Definitions

A point in projective space of  $n$ - dimensions,  $\mathbf{P}^n$ , is represented by a  $(n + 1)$  element column vector of coordinates  $\mathbf{x} = [x_1, \dots, x_{n+1}]^T$ . At least one of the  $x_i$  coordinates must be nonzero. Two points represented by  $(n + 1)$ -vectors  $x$  and  $y$  are considered equal if a nonzero scalar  $\lambda$  exists such that  $x = \lambda y$ . Equality between points is indicated by  $x \approx y$ . Since scale is not important in projective geometry, the vectors described above are called *homogenous coordinates* of a point. Homogenous points with  $x_{n+1} = 0$  are called *points at infinity* and are related to the affine geometry described in section 2.3.

A *collineation* or linear transformation of  $\mathbf{P}^n$  is defined as a mapping between projective spaces, which preserves collinearity of any set of points. This mapping is represented by a  $(m+1) \times (n+1)$  matrix  $H$ , for a mapping from  $\mathbf{P}^n \rightarrow \mathbf{P}^m$ .

A *projective basis* for  $\mathbf{P}^n$  is defined as any set of  $(n + 2)$  points of  $\mathbf{P}^n$ , such that no  $(n + 1)$  of them are linearly dependent. The set  $e_i = [0, \dots, 1, \dots, 0]^T$ , for  $i = 1 \dots n+1$ , where 1 is in the  $i^{\text{th}}$  position, and  $e_{n+2} = [1, \dots, 1, \dots, 1]^T$  form the *standard projective basis*. A projective point of  $\mathbf{P}^n$  represented by any of its coordinate vectors  $\mathbf{x}$  can be described as a linear combination of any  $n + 1$  points of the standard basis in Eqn. 2.1.

$$\mathbf{x} = \sum_{i=1}^{n+1} x_i e_i \quad (2.1)$$

Any projective basis can be transformed by a collineation into a standard projective basis [FAUGERAS1993]: “let  $[x_1, \dots, x_{n+2}]$  be  $n+2$  coordinate vectors of points in  $\mathbf{P}^n$ , no  $n + 1$  of which are linearly dependent, i.e., a projective basis. If  $e_1, \dots, e_{n+1}, e_{n+2}$  is the standard projective basis,



then there exists a nonsingular matrix  $A$  such that  $Ae_i = \lambda_i x_i$ ,  $i=1, \dots, n+2$  where  $\lambda_i$  are nonzero scalars; any two matrices with this property differ at most by a scalar factor.”

### 2.9.1.2 Projective Plane

The projective space  $\mathbf{P}^2$  is known as the projective plane. A point in  $\mathbf{P}^2$  is defined as a 3-vector  $\mathbf{x}=[x_1, x_2, x_3]^T$ , with  $(u, v) = (x_1/x_3, x_2/x_3)$  the Euclidean position on the plane. This 2D projective plane is placed in a 3D projective space. A line is also defined as a 3-vector  $l=[l_1, l_2, l_3]$  and having the form of Eqn. 2.2.

$$\sum_1^3 l_i x_i = 0 \quad (2.2)$$

This equation is called the *point equation*, which means that a point  $\mathbf{x}$  is represented by a set of lines through it, or this equation is also called the *line equation*, which means that a line  $l$  is represented by a set of all points, that satisfy the relation. These two statements show that there is an equivalence between points and lines in  $\mathbf{P}^2$ . This property is called the *principle of duality*. Any theorem or statement that is true for the projective plane can be reworded by substituting points for lines and lines for points, and the resulting statement will also be true.

The line vector  $l$  defining the line through two points  $\mathbf{x}$  and  $\mathbf{y}$  is

$$l = \mathbf{x} \times \mathbf{y}. \quad (2.3)$$

This is also sometimes calculated as follows:

$$l = [\mathbf{x}]_x \mathbf{y} \quad (2.4)$$

with

$$[\mathbf{x}]_x = \begin{bmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & x_1 \\ x_2 & -x_1 & 0 \end{bmatrix} \quad (2.5)$$

being the antisymmetric matrix of coordinate vector  $\mathbf{x}$  associated with the cross product. The intersection point of two lines is also defined by the cross product of the line vectors:  $\mathbf{x}=\mathbf{l}_1\times\mathbf{l}_2$ .

All the lines passing through a specific point form a *pencil of lines*. If two lines  $\mathbf{l}_1$  and  $\mathbf{l}_2$  in a 2D projective plane are distinct elements of this pencil, then all the other lines can be obtained as follows:

$$\mathbf{l} = \lambda_1\mathbf{l}_1 + \lambda_2\mathbf{l}_2 \quad (2.6)$$

where  $\lambda_1$  and  $\lambda_2$  are scalars, such that at least one is non-zero.

### 2.9.1.3 Projective Space

The space  $\mathbf{P}^3$  is known as the projective space. A point of  $\mathbf{P}^3$  is defined by a 4 numbers  $\mathbf{x}=[x_1,x_2,x_3,x_4]$ , not all zero. They form a coordinate vector  $\mathbf{x}$  defined up to a scale factor. In  $\mathbf{P}^3$  objects other than just points and lines are included, such as planes. A plane is also defined as a four set of numbers ( $\pi_1, \pi_2, \pi_3, \pi_4$ ). The equation of this plane is then

$$\sum_{i=1}^4 \pi_i x_i = 0. \quad (2.7)$$

A point  $x$  is located on a plane if the following equation is true.

$$\boldsymbol{\pi}^T \mathbf{x} = 0 \quad (2.8)$$

The structure which is analogous to the pencil of lines of  $\mathbf{P}^2$  is the *pencil of planes*, the set of all planes that intersect in a certain line.

### 2.9.1.4 Summary

Now that a framework for projective geometry has been created, it is possible to define the 3D Affine space as embedded in a projective space  $\mathbf{P}^3$ . In a similar way, the image plane of the camera is embedded in a projective space  $\mathbf{P}^2$ . Then a collineation exists which maps the 3D

space to the image plane,  $P^3 \rightarrow P^2$ , via a 3 x 4 matrix, as discussed in more detail in chapter seven on 3D Reconstruction.

## 2.9.2 Affine Geometry

The Affine geometry lies between the projective and metric geometries and contains more structure than the projective stratum, but less than the metric and Euclidean ones.

### 2.9.2.1 Affine Plane

The line in the 3D projective plane with all points  $x_4 = 0$  is called the *line at infinity* or  $l_\infty$ . It is represented by the vector  $l_\infty = [0,0,0,1]^T$ . The affine plane is embedded in the projective plane under a correspondence of  $A^2 \rightarrow P^2 : X = [X_1, X_2]^T \rightarrow [X_1, X_2, 1]^T$ . There “*is a one-to-one correspondence between the affine plane and the projective plane minus the line at infinity with equation  $x_3 = 0$* ” [FAUGERAS1993]. For a projective point  $x = [x_1, x_2, x_3]^T$  that is not on the line at infinity  $x_3 \neq 0$ , the affine parameters can be calculated as  $X_1 = \frac{x_1}{x_3}$  and  $X_2 = \frac{x_2}{x_3}$ .

#### 2.9.2.1.1 Transformations

A point  $x$  is transformed in the affine plane as follows:

$$A = BX + b \quad (2.9)$$

with  $B$  being a 2 x 2 matrix of rank 2, and  $b$  a 2 x 1 vector. These transformations form a particular subgroup which leaves the line at infinity invariant.

### 2.9.2.2 Affine Space

As in the previous section, the plane at infinity  $\pi_\infty$  has equation  $x_4 = 0$  and the affine space can be considered to be embedded in the projective space under a correspondence of  $A^3 \rightarrow P^3 : X = [X_1, X_2, X_3]^T \rightarrow [X_1, X_2, X_3, 1]^T$ . Then for each projective point  $x = [x_1, x_2, x_3, x_4]^T$

that is not in the plane at infinity  $x_4 \neq 0$ , the affine parameters can be calculated as  $X_1 = \frac{x_1}{x_4}$ ,

$$X_2 = \frac{x_2}{x_4} \text{ and } X_3 = \frac{x_3}{x_4}.$$

### 2.9.2.2.1 Transformations

Affine transformations of space ( $X^* = [T]X$ ) can be written exactly as in Equation (2.9), but with  $B$  being a 3x3 matrix of rank 3, and  $b$  a 3x1 vector.

$$T_A \approx \begin{bmatrix} B & b \\ 0_3^T & 1 \end{bmatrix} \quad (2.10)$$

The invariants of affine geometry are the points, lines and planes at infinity. These form an important aspect of camera calibration and 3D reconstruction. Obtaining the plane at infinity in a specific projective representation allows for an upgrade to an affine representation.

### 2.9.3 Metric Geometry

This geometry corresponds to the group of *similarities*. The transformations in this group are Euclidean transformations such as rotation and translation. The metric geometry allows for a complete reconstruction up to an unknown scale.

#### 2.9.3.1 Metric Plane

Affine transformations don't change line at infinity by definition, but to also preserve two points on that line called the *absolute points* or *circular points*. The circular points are two complex conjugate points lying on the line at infinity [SEMPLE1979].

#### 2.9.3.2 Metric Space

In metric space, affine transformations are adapted to leave the absolute conic invariant.

The absolute conic  $\Omega$  is obtained as the intersection of the quadric equation  $\sum_{i=1}^4 x_i^2 = 0$  with  $\pi_\infty$

$$\sum_{i=1}^4 x_i^2 = x_4 = 0. \quad (2.11)$$

Affine transformations which keep  $\Omega$  invariant are written as:

$$T_M \approx \begin{bmatrix} cC & b \\ \mathbf{0}_3^T & 1 \end{bmatrix} \quad (2.12)$$

where  $c > 0$  is any positive non zero scalar, and  $C$  is orthogonal:  $CC^T = I_{3 \times 3}$ . The absolute conic is the invariant variable of the metric geometry.

### 2.9.4 Euclidean Geometry

Euclidean Geometry is the same as metric geometry, the only difference being that the relative lengths are upgraded to absolute lengths. This means that the Euclidean transformation matrix is the same as in equation (2.12) without the scaling factor

$$T_E \approx \begin{bmatrix} C & b \\ \mathbf{0}_3^T & 1 \end{bmatrix}. \quad (2.13)$$

### 2.10 Conclusions

This chapter introduced the problems associated with diabetic neuropathy and the need for custom shoe inserts to prevent leg amputation. A brief description of the manual casting process was given along with its drawbacks which motivate the need to automate this process. It was noted that the downside using existing foot scanners were the costs associated with it and lack of applicability. A brief treatise on the mathematics of 3D vision was presented with description of different 3D geometries. Though sufficient details were given on projective and affine geometry, a euclidean reconstruction of the foot is considered in this thesis. More specific details related to euclidean reconstruction are dealt in chapter seven in the thesis.

## Chapter 3

### 3D Foot Scanner Utility Frame

#### 3.1 Introduction

This chapter introduces the basic framework of the entire work done applying the new method of 3D modeling of foot from images. Review of the foot scanner utility frame is described briefly and the description of the software interface (GUI) is laid out with its functionalities.

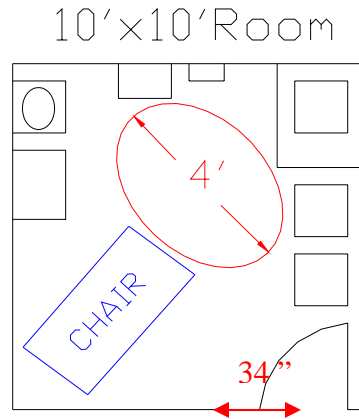
#### 3.2 Foot Scanner Utility Frame

Louisiana State University, Mechanical Engineering Department, was asked to design an inexpensive prototype system that will acquire sufficient 3D point geometry of feet for the use in the delivery of orthopedic shoe insoles. The project consists of two parts:

1. Capture, Extract, and Process 3D Point Geometry of the Feet
2. Design of the Utility Frame

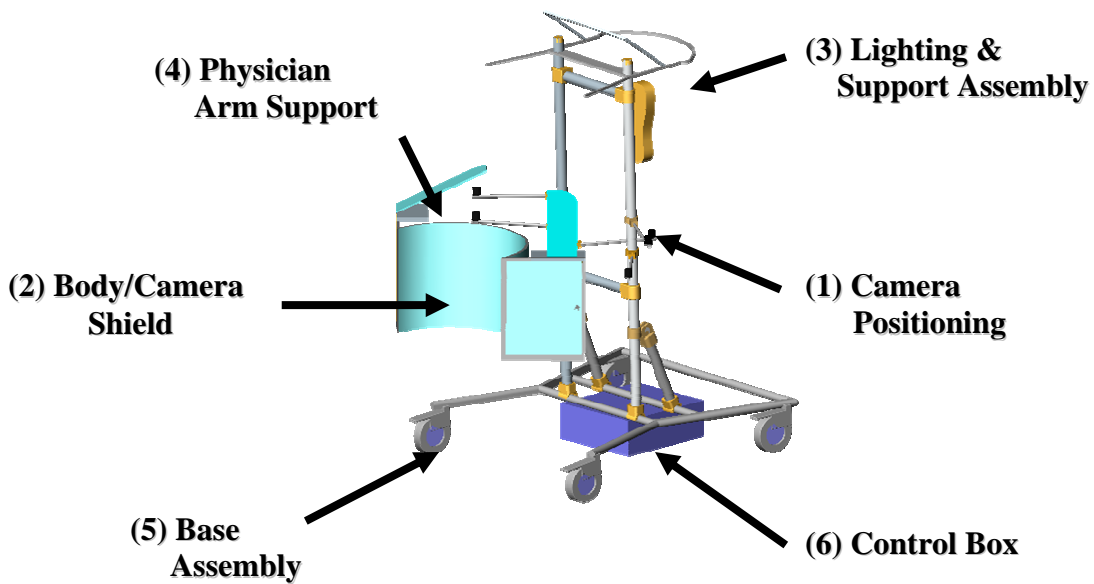
A mechanical engineering design team [BAJOIE2003], comprising three students was tasked with the design of the prototype scanner utility frame. The function of this prototype was to support the experimental determination of the best set and arrangement of inexpensive cameras to allow for each point on the foot to be captured by at least two cameras.

The primary operating environment for this system is a 10 ft x 10 ft clinical examination room. The actual workspace within this room is a small 4 ft diameter area in front of an adjustable clinical chair. The following figure shows the layout of the examination room.



**Fig. 3.1 Examination Room**

The prototype design consists of 6 major sub-assemblies depicted below.



**Fig. 3.2 Utility Frame**

The first sub-assembly of the utility frame is the *Camera Positioning* assembly. The functionality of this system is to allow for each point on the foot to be captured by at least two cameras, and to also provide a clear camera line of sight to the foot. The data points on the foot

that are needed are from the ball of the foot to the back of the heel, and up the lateral sides. Of these points, the heel is the most critical point on the foot. Utilizing a U-shaped configuration of five camera arms, which consist of varying lengths of 3/8" diameter aluminum rods and mini-ballhead camera mounts, five different unobstructed images of the foot can be acquired.

The next sub-assembly in the utility frame is the *Body/Camera Shield*. The first function of this assembly is to prevent the physician's lab coat and tie from interfering with the camera view, while the second function is to protect the camera arms from being bumped or knocked out of position. While isolating the cameras from the physician, the shield also provides adequate leg and body clearance to allow the physician to comfortably position his/her body.

The *Light and Support Assembly* is the third sub-assembly for the design. The functions of this assembly are to provide clear lines of sight for both the physician and the cameras, allow easy access of the foot, provide the adequate lighting intensity and contrast for image capture, and protect the physician's forehead. This assembly is composed of four parts: Support frame, light source, light shield/shroud, and forehead padding.

The fourth sub-assembly is the *Physician Arm Support Assembly*. The first function of this assembly is to provide a clear camera line of sight to the foot. The second function is to assist the physician in holding the patient's foot. This assembly is composed of the following parts: Polypropylene platform, foam padding, aluminum support bar, and L-bracket. The polypropylene platform supports the physicians arm, and holds back the physician's lab coat sleeve. Meanwhile, the foam padding provides extra comfort for the physician's arm. The support bar and L-bracket are used for strengthening and mounting of the assembly.

The *Base Assembly* is the fifth assembly to the design. The functions of the base assembly are to provide portability and stability for the frame, and support for the control box



assembly. The light weight base frame is made of aluminum and is maneuverable through the use of four medium duty casters.

The final sub-assembly is the *Control Box*. The control box, which is located under the base, provides the housing for the electrical components of the utility frame, such as a surge protector, USB Hubs, and electrical adaptors. It also isolates the wiring of these components and the cameras from the metallic frame. The control box provides a single power source cable and two USB outlet cables which connect to the PC.

### **3.3 Experimental Setup**

Preliminary setup consists of assembling the six sub components into one frame. All necessary connections are done and all wiring directed towards the control box to reduce clutter. Webcam software is started and video images are checked for clarity. Images from the webcams are then taken in a sequence manually to collect a series of 5 images. These are then stored in a common folder for the interface to generate the 3-D model of the foot.

### **3.4 Advantages of Using the Foot Scanner Utility Frame**

The foot scanner is unique in the sense that it does not use any lasers in its scanning process. It allows each point to be captured by at least two cameras. Provides an inexpensive means of capturing the geometry of foot. Unlike usual laser foot scanners, the capture period is less. It also gives an adequate environment for image capture. It is also versatile and adjustable. It is custom built to meet the needs of the podiatrist, by providing easy access to foot. It is portable and mobile in nature. Easy setup and teardown is possible because of the use of sub assemblies.

### **3.5 Graphical User Interface to Generate 3D Point Geometry**

The Graphical User Interface is built using MATLAB<sup>®</sup>. The functionality of the interface is to pick the set of images from the folder taken by the cameras and generate 3D point geometry of

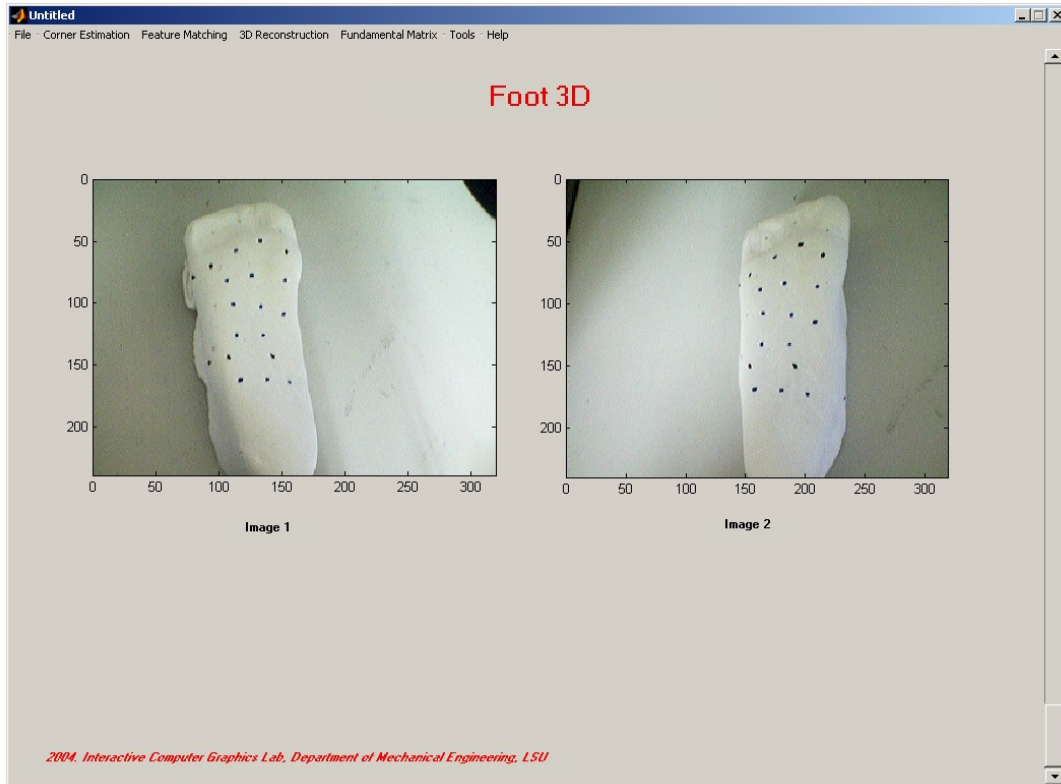
foot using set of sequential techniques. It is completely interactive and needs minimal supervision. The steps involved using the interface to generate the model is:

- Camera Calibration is done to estimate the intrinsic parameters of the camera. This involves using the calibration tool on a set of images which contain a test rig such as a planar chess board;
- Corner detection is done on a pair of images using the algorithm to detect marked corners or points;
- Feature matching is the next process to match corresponding points in their respective images;
- Robust estimation of fundamental matrix is performed using 8 corresponding points;
- 8 point algorithm is executed to determine the extrinsic parameters of the cameras;
- 3D triangulation algorithm is executed to extract the 3D point data from images.

All these steps are dealt in more detail in successive chapters.

### **3.6 Objectives and Novel Aspects of Current Method**

One of the important objectives as required by the podiatrist was to extract 3D geometry of the foot inexpensively. It involves using economically priced web cameras to capture images. Simple Foot Scanner Utility Frame to house the cameras and necessary equipments. It is customized to suit the needs of the podiatrist. The entire functionality is built in the software interface which generates 3D points from those marked on foot. Apart from that, the frame is built for easy setup and teardown. It is sterile and adaptable for the hospital environment.



**Fig. 3.3 Graphical User Interface (GUI)**

### **3.7 Conclusions**

This chapter reviews the custom built foot scanner utility frame to capture images of foot. The basic components of the foot scanner are dealt in detail. A brief description of the image capturing process is given. Mention of the software interface to generate 3D foot geometry is made with description of each of the functionalities involved.

## Chapter 4

### Camera Calibration and Epipolar Geometry

#### 4.1 Introduction

Camera calibration is one of the most important steps in the 3D reconstruction process in order to extract the metric information from 2D images. By calibrating the camera and determining its intrinsic parameters, the 3D coordinates can be mapped in world space to 2D coordinates in the image. This information is necessary to reconstruct the 3D point from images (reverse mapping). This chapter introduces the basic pinhole camera model for capturing an image from the 3D model. It also describes the epipolar geometry that exists when there are two images of the same 3D model in space. A detailed description of the calibration algorithm is given to determine the intrinsic camera parameters - focal length  $f$ , center of projection  $c$  and radial distortion  $k$ .

#### 4.2 Camera Model

A camera is often described using the *pinhole model* [FAUGERAS1993]. A collineation exists in 3D projective geometry, which maps the projective space to the camera's retinal plane (see Fig. 4.1). Then the coordinates of a 3D point  $\mathbf{M} = [X, Y, Z]^T$  in a Euclidean world coordinate system and the retinal image coordinates  $\mathbf{m} = [u, v]^T$  are related by the following equation

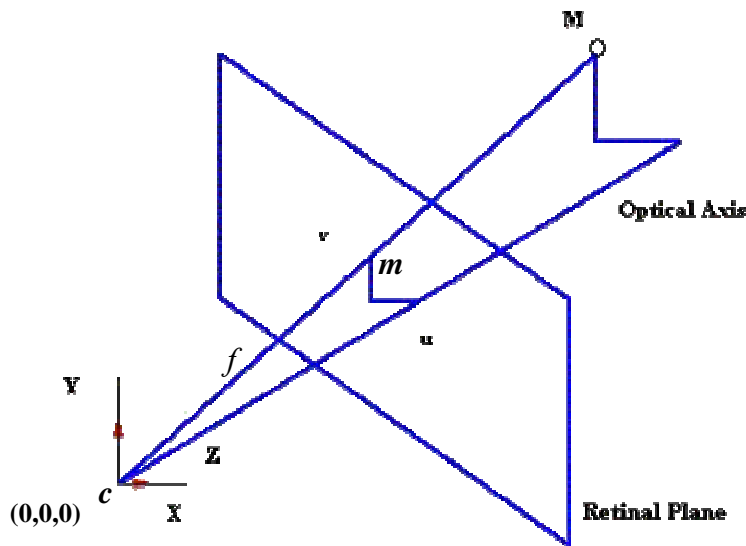
$$s \mathbf{m} = \mathbf{P} \mathbf{M} \quad (4.1)$$

where  $s$  is a scale factor,  $\mathbf{m} = [u, v, 1]^T$  and  $\mathbf{M} = [U, V, Z, 1]^T$  are the homogenous coordinates of vector  $\mathbf{m}$  and  $\mathbf{M}$ , and  $\mathbf{P}$  is a 3x4 matrix representing the collineation  $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ .  $\mathbf{P}$  is called the perspective projection as illustrated in Fig. 4.1. It shows the case where the projection center is placed at the origin of the world coordinate frame and the retinal plane is at  $Z=f=1$ .

Then  $u$  and  $v$  are defined as follows:

$$u = \frac{fX}{Z}, v = \frac{fY}{Z} \quad \forall \quad P = \begin{bmatrix} I_{3 \times 3} & \mathbf{0}_3 \end{bmatrix}. \quad (4.2)$$

The optical axis passes through the center of projection (camera)  $C$  and is orthogonal to the retinal plane. The principal point  $c$ , is the intersection of the optical axis with the line passing through point  $M$  and the retinal plane. The focal length  $f$  is the distance between the center of projection and the retinal plane.

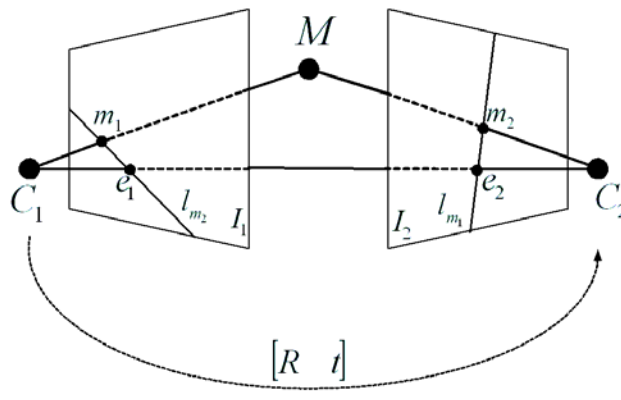


**Fig. 4.1 Perspective Projection**

A world coordinate system is usually defined with the positive  $Y$ - direction pointing upwards, the positive  $X$ -direction pointing to the right, and the positive  $Z$ -direction pointing into the retinal plane. The retinal plane forms part of epipolar geometry (described in section 4.3) and the image plane is internal to the camera. The transformation which maps the retinal plane from the image plane is brought about by the camera calibration matrix (described in section 4.4)

### 4.3 Epipolar Geometry

Epipolar geometry exists between two camera systems. With reference to Fig 4.2, the two cameras are represented by coordinate systems  $C_1$  and  $C_2$ . Points  $m_1$  and  $m_2$  are the image points of the 3D point  $M$ . Epipoles  $e_1$  and  $e_2$  are the intersections with both images planes of the line joining the two cameras  $C_1$  and  $C_2$ . These epipolar points are the projection of the respective cameras in the opposite image planes. The plane formed with the three points  $\langle C_1 M C_2 \rangle$  is called the *epipolar plane*. The lines  $l_{m1}$  and  $l_{m2}$  are called the *epipolar lines* and are formed when the epipoles and image points are joined.



**Fig. 4.2 Epipolar Geometry**

The point  $m_2$  is constrained to lie on the epipolar line  $l_{m1}$  of point  $m_1$ . This is called the *epipolar constraint*. To visualize it differently: the epipolar line  $l_{m1}$  is the intersection of the epipolar plane mentioned above with the second image plane  $I_2$ . This means that image point  $m_1$  can correspond to any 3D point (even points at infinity) on the line  $\langle C_1 M \rangle$  and that the projection of  $\langle C_1 M \rangle$  in the second image  $I_2$  is the line  $l_{m1}$ . All epipolar lines of the points in the first image pass through the epipole  $e_2$  and form thus a pencil of planes containing the baseline  $\langle C_1 C_2 \rangle$ . The above definitions are symmetric in a way such that the point of  $m_1$  must lie on the epipolar line  $l_{m2}$  of point  $m_2$ . Expressing the epipolar constraint algebraically, the following equation needs to be satisfied in order for  $m_1$  and  $m_2$  to be matched:

$$m_2^T F m_1 = 0 \quad (4.3)$$

where  $F$  is a 3x3 matrix called the *fundamental matrix*. The following equation also holds:

$$l_{m_1} = F m_1 \quad (4.4)$$

since the point  $m_2$  corresponds to point  $m_1$  belongs to the line  $l_{m_1}$  [LUONG1996]. The role of the images can be reversed and then

$$m_1^T F^T m_2 = 0. \quad (4.5)$$

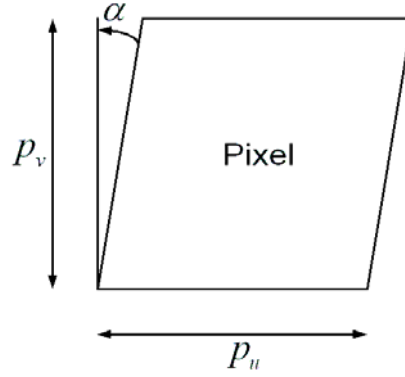
which shows that the fundamental matrix is changed to its transpose.

#### 4.4 Camera Calibration Matrix

The camera calibration matrix, denoted by  $K$ , contains the intrinsic parameters of the camera used in the imaging process.

$$K = \begin{bmatrix} \frac{f}{p_u} & \frac{(\tan \alpha)f}{p_v} & u_0 \\ 0 & \frac{f}{p_v} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

The focal length  $f$ , gives the metric distance measured in pixels ( in the image coordinate system), used later to map the 3D point from two corresponding points in the images. The values  $p_u$  and  $p_v$  represent the width and height of the pixels in the image and  $\alpha$ , the skew angle (Fig. 4.3).



**Fig. 4.3 Illustration of Pixel Skew**

It is possible to simplify  $K$  as:

$$K = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

where,  $f_u$  and  $f_v$  are the focal lengths measured in width and height of the pixels,  $s$  represents the pixel skew and the ratio  $f_u:f_v$  characterizes the aspect ratio of the camera.

The camera calibration matrix to transform points from the retinal plane to points on the image plane given by

$$m = K m_{\text{ret}}. \quad (4.8)$$

#### 4.5 Calibration Using a Planar Check Board

There are several advantages using a 2D surface for calibration. It is easy to mark the dimensions on a 2D surface as compared to 3D surface. In addition, measurements are easy using a 2D surface. The complexity associated with 2D rigs is less compared to 3D surfaces.

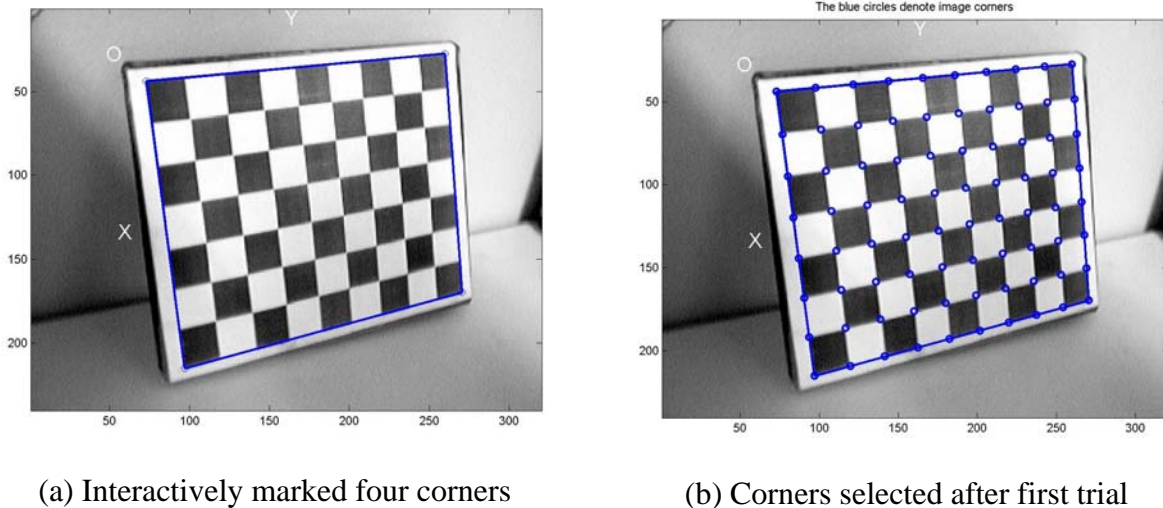
Besides, the accuracy in the calibration results obtained using 3D rigs is higher compared to 3D rigs. In this thesis, a chess board is chosen as a 2D calibration rig. It is chosen because, it



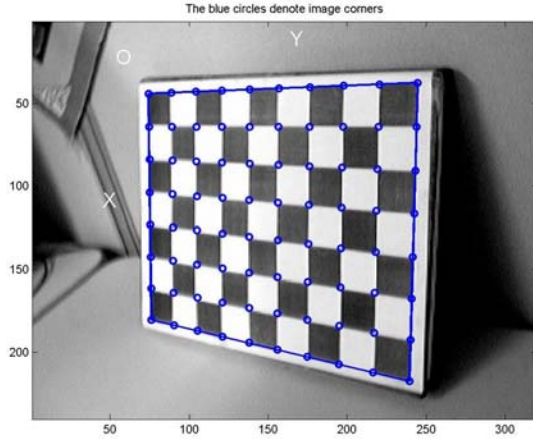
has distinct black and white squares with equal size. The chess board also makes it easy to mark the corners interactively.

The calibration is done using an existing Camera Calibration Toolbox [CALTECH1999].The process begins with selecting a 2D chess board as a test rig. The dimensions of the chess board are measured. This includes the dimensions of each square (see Fig. 4.4) as well as the number of squares along the length and breadth of the chess board. The focal length of the camera is fixed during the entire calibration process. Twenty images of the chess board from different vantage points are then taken and stored in a folder.

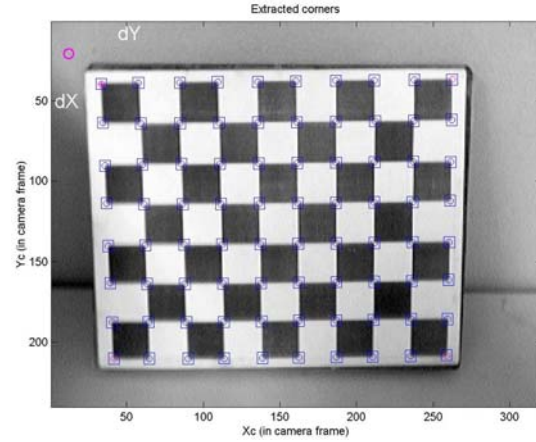
The toolbox program requires at least twenty images of the chess board in different positions to get a robust result of the calibration matrix. On every image the four corners of the chess board are marked interactively. The program then assigns all corners associated with the chess board. This process is repeated for all selected images and a best estimate is evaluated for the intrinsic parameters of the camera.



**Fig. 4.4 Steps Involved During the Calibration Process (fig. continued)**



(c) Corners selected during one of 'n' trials



(d) Final extracted corners on Chess Board

#### 4.6 Calculating the Camera Calibration Matrix

A treatise on estimating the Camera Calibration Matrix is shown in the book *An Invitation to 3-D Vision, From Images to Geometric Models* [MA2001]. The final equation that maps the coordinates of the 3D object to pixel coordinates in the image is given by:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.9)$$

where:

$x'$  &  $y'$  =  $x$  &  $y$  image pixel coordinates,

$s_x$  &  $s_y$  = scale in  $x$  and  $y$  directions,

$s_\theta$  = skew in  $x$  and  $y$  directions,

$o_x$  &  $o_y$  = center of projection in  $x$  and  $y$  coordinates in pixels,

$f$  = focal length in pixels,

$X, Y, Z$  = coordinates of the 3-D object in world coordinates,

$\lambda$  = arbitrary independent scale factor.

Equation 4.9 can be simplified as noted in Equation 4.10.

$$\lambda x' = K\Pi_0 X = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.10)$$

The constant 3x4 matrix  $\Pi_0$  represents the perspective projection. The upper triangular 3x3 matrix  $K$  collects all parameters that are “intrinsic” to a particular camera, and is therefore called the *intrinsic parameter matrix*, or the *calibration matrix* of the camera. The above parameters can be calculated using the Camera Calibration Toolbox [CALTECH1999].

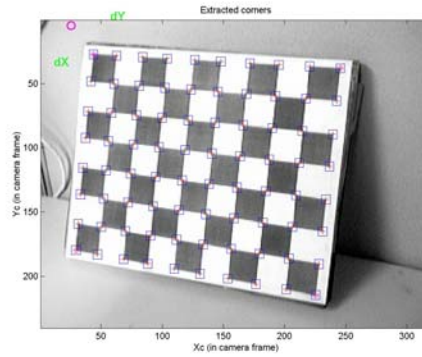
#### 4.7 Calibration Results

The camera used in this work is an Intel<sup>®</sup> CS 330 Pro Video PC Camera. The technical specifications are

- CCD Sensor for enhanced image quality and low-light performance,
- 640 x 480 VGA Resolution,
- 50 degree field of view,
- Focus distance of 10 cm to infinity.

Two test rigs are considered to verify and obtain an accurate calibration matrix values. The first case involves a rig rectangular in shape, similar to a chess board. In the second case, a chess board is considered having equal number of squares in it.

## Case 1:



**Fig. 4.5 Final Extracted Corners on the Test Rig**

### Dimensions of the chess board:

*Length of the rig in Y direction (mm) = 270, Breadth of the rig in X direction (mm) = 210*

*Number of squares along X direction = 7, Number of squares along Y direction = 9*

### Calibration Results:

The results obtained are the intrinsic parameters of the camera. They include focal length  $f$  along  $u$  and  $v$  directions in pixels. Principal point or center of projection  $c_u$  and  $c_v$  in pixels and skew of the pixel  $\alpha$  in degrees. They are listed in Table 4.1.

**Table 4.1 Calibration Result Set for Case 1**

<i>Focal length along <math>u</math> direction <math>f_u</math> (in pixels)</i>	408.98
<i>Focal length along <math>v</math> direction <math>f_v</math> (in pixels)</i>	407.61
<i>Principal point coordinates <math>c_u</math> (in pixels)</i>	154.5
<i>Principal point coordinates <math>c_v</math> (in pixels)</i>	122.5
<i>Skew <math>\alpha</math> (in degrees)</i>	0

$$K_{case1} = \begin{bmatrix} 408.98 & 0 & 154.5 \\ 0 & 407.61 & 122.5 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

It is noted that the values of  $K$  matrix are in pixels.

### Case 2:

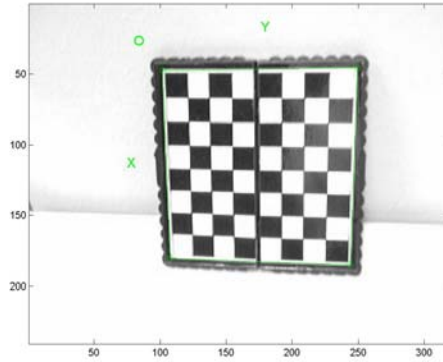
To check the accuracy of the results obtained, another calibration rig was chosen with different dimensions (see Fig. 5.6). Twenty images were taken and the calibration algorithm was applied to those images. The results obtained are as follows:

#### Dimensions of the chess board:

*Length of chess board along Y direction (mm) = 144,*

*Breadth of chess board along X direction (mm) = 144*

*Number of squares along X direction = 8, Number of squares along Y direction = 8*



**Fig. 4.6 Second Trial Using a Different Chess Board**

#### Calibration Results:

The results obtained are the intrinsic parameters of the camera. They include focal length  $f$  along  $u$  and  $v$  directions in pixels. Principal point or center of projection  $c_u$  and  $c_v$  in pixels and skew of the pixel  $\alpha$  in degrees. The results for case 2 are listed in Table 4.2.

**Table 4.2 Calibration Result Set for Case 2**

<i>Focal length along <math>u</math> direction <math>f_u</math> (in pixels)</i>	412.61
<i>Focal length along <math>v</math> direction <math>f_v</math> (in pixels)</i>	403.28
<i>Principal point coordinates <math>c_u</math> (in pixels)</i>	156.4
<i>Principal point coordinates <math>c_v</math> (in pixels)</i>	119.2
<i>Skew <math>\alpha</math> (in degrees)</i>	0

Therefore from Eqn. 4.7,  $\mathbf{K}$  matrix becomes

$$K_{case2} = \begin{bmatrix} 412.61 & 0 & 156.4 \\ 0 & 403.28 & 119.2 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

where all values in  $\mathbf{K}$  matrix are in pixels.

Comparing the values of  $\mathbf{K}$  in equation 4.11 and 4.12, we see that all parameter values are within a range of  $\pm 5$  pixel accuracy. The small differences in values are attributed to the corners being picked in the algorithm interactively.

#### 4.8 Conclusions

The camera calibration process and determining the calibration matrix has been presented in this chapter. This process is among the important steps that are undertaken in the 3D reconstruction process. The application of the calibration matrix  $\mathbf{K}$  will be shown in Chapter 7, where it is used in reconstruction of foot geometry.

## Chapter 5

### Feature Detection

#### 5.1 Introduction

Feature point selection plays an important role in the entire 3D reconstruction process. This is a process where marked points on a photo are detected by the algorithm. These points in the image are then reconstructed later as 3D points in space. This chapter outlines the existing methods to detect points on the image and their drawbacks. A composite technique is suggested to robustly detect points utilizing color intensity information in addition to gradient changes that occur near marked points in the image.

#### 5.2 Review of Earlier Works on Feature Detection

Several techniques have been proposed to detect points or features. There are algorithms, which detect features such as points, and also which detect edges. A few major works are reviewed in this section.

One of the earliest works was done by Kitchen & Rosenfield [KITCHEN1982], where they used intensity variation techniques to detect corners in an image. Their results are based on grey level intensity variations that occur in  $x$  &  $y$  directions in the image. Any change in the gradient in both the directions results in detecting a point or corner.

Earlier work done on detecting the corners used the method of segmentation, where the image was partitioned based on some common shapes and then attention focused on those segments for corner detection. The work done by the above authors eliminates this process of segmentation by using the method of gradients to detect the corners. The drawbacks using this method are not obtaining a clear set of results for a more defined image. The algorithm also detects unwanted corners. Besides that, all results use grey scale images. But nevertheless this

method marked the beginning of using grey scale gradients in corner detection and also eliminating the process of segmentation.

Ellen C. Hildreth [HILDRETH1985] introduced the concept of edge detection, based on earlier works on corner detection. Her research was based on recovering physical properties of objects in the scene, such as the location of object boundaries and the structure, color and texture of object surfaces in the image. The method outlined in this manuscript is similar to the procedure of evaluating grey scale intensities in the process of feature detection. This algorithm was used to determine the shape and properties of an object under consideration. One noted contribution of her research includes applying the smoothing function on intensity gradients. Drawbacks of using this method with respect to corner detection are the absence of a local intensity correction needed to resolve ambiguous corners.

One of the most prominent contributions in the field of corner detection was presented by C Harris & M Stephens [HARRIS1988]. They first review the earlier work done by Kitchen & Rosenfield. They introduced the concept of a corner having intensity gradient change in both  $x$  and  $y$  directions, which would be detected as a feature. In addition to that, they used the technique of local auto-correlation to increase the accuracy of detection. They also discussed methods of tracking features defined as discrete (isolated points in the image which could be detected using the earlier technique of Kitchen & Rosenfield), which have certain drawbacks. Methods to detect features, which form a continuum like texture, or edge pixels, are also discussed in great detail. Using the concept of edge detection, corners also could be detected using intersection property of two edges. Important drawbacks include detecting more corners than required from the image. Apart from that, the algorithm is less effective in a known environment, where corners are marked on the object with known distances. This algorithm does



most of the corner detection needed for current project. However an improvement combining the algorithm with information on color intensity is detailed in further sections of this chapter.

### 5.3 Corner Detection Using Harris-Stephens Algorithm

Initial detection of corners is performed using the Harris-Stephens algorithm [HARRIS1988]. The entire image (RGB format) is converted into a 2 dimensional matrix of grey scale pixel intensities. A sub window or mask of dimensions 7 by 7 pixels is selected to examine the existence of corners. A minimum threshold  $\tau$  is defined for the window of pixels taken as:

$$\tau > 0 \quad (5.1)$$

for 7x7 window size. The value of the threshold can be changed accordingly, based on the kind of features to be detected. For most cases,  $\tau$  is around 0.04 – 0.1. The gradient matrix  $G$  for all pixels within the window is calculated as:

$$G = \sum_{x=-3}^{x=3} \sum_{y=-3}^{y=3} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (5.2)$$

where,  $I_x = \partial I / \partial x$ ,  $I_y = \partial I / \partial y$  and  $w(x, y)$  is the coordinate in consideration.

The threshold value for this gradient matrix is computed as:

$$C(G) = \det(G) + k \times \text{trace}^2(G) \quad (5.3)$$

where,  $k$  is a small scalar, chosen between 0.04 – 0.06. Different choices of  $k$ , results in favoring gradient variation in one or more directions. Features with significant variation in both directions ( $x$  &  $y$ ), will enable  $C(G)$  to be greater than minimum threshold  $\tau$ . These are then selected as features and stored in a matrix.

To detect points at the edges or boundaries of the image, the window size is chosen from 0 to 7 instead of -3 to +3 perpendicular to the edge considered. For example, if the edge is along  $y$  direction, then the  $x$  values of the window at points on that edge would begin from 0 to 7.

## 5.4 Drawbacks of Harris-Stephens Algorithm

One of the important drawbacks of using this algorithm is detecting too many corners. It also works only on grey scale images, hence not taking advantage of color information of marked feature points. To make this algorithm work in a more controlled environment, color intensity of the features can be used in conjunction with the algorithm. The next section deals with the application of combined Harris-Stephens and color intensity algorithm to enhance corner extraction and reduce selecting extraneous points.

## 5.5 Corner Detection Using Combined Harris-Stephens & Color Intensity Algorithm

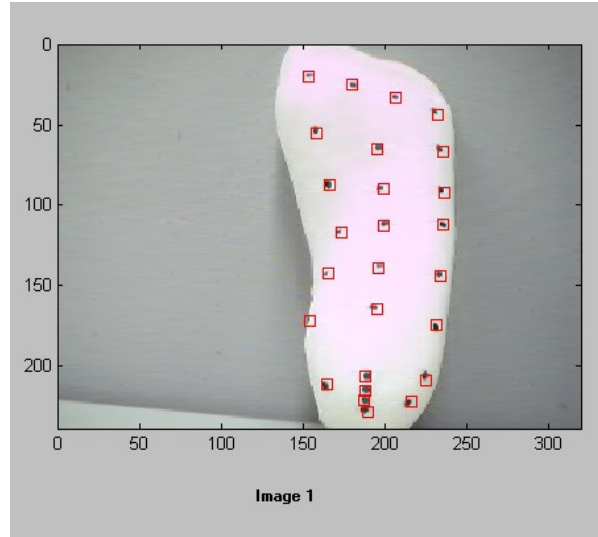
The points marked on the foot have a certain color intensity value described in terms of its combination of RGB (Red, Blue and Green) values. This step is performed by taking a sample image of the foot with marked points of user specified color. Points from the image are picked interactively and the RGB value of the marked color is determined using a standard image processing function in Matlab [MATHWORKS2004]. Color values for marked points would vary under different lighting conditions but lie within a bandwidth. Bandwidth here is defined as a range of acceptable range of values in all three primary colors (RGB).

The process of detecting points (marked on the foot) on the image begins with the initial selection of corners using Harris-Stephens algorithm which are the input for a color intensity algorithm. The RGB values of input corners from the Harris-Stephens algorithm are recorded. Corners that lie within the specified bandwidth are kept as feature points and others are discarded. The final set of points so obtained is stored.

## 5.6 Results

### 5.6.1 Harris-Stephens Algorithm

A representative set of results obtained using Harris-Stephens algorithm.

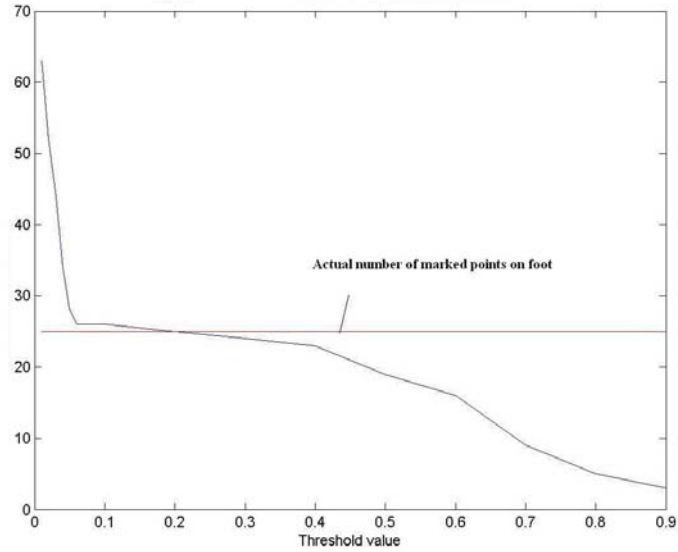


**Fig. 5.1 Marked Points Detected in an Ideal Image**

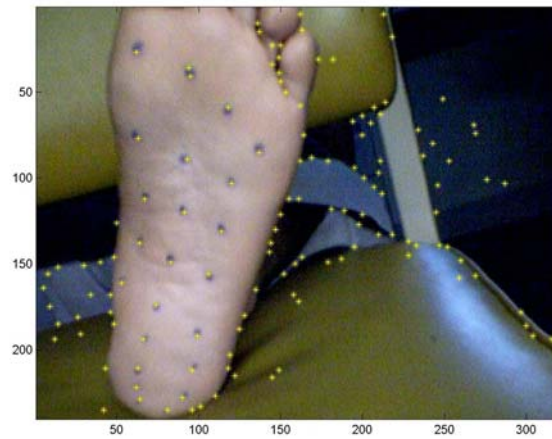
**Table 5.1 Results of Points Detected Using Harris-Stephens Algorithm**

<i>Tracking Window Size <math>w_x</math> (in pixels)</i>	7
<i>Tracking Window Size <math>w_y</math> (in pixels)</i>	7
<i>Threshold Value <math>\tau</math></i>	0.06
<i>Actual marked points on foot</i>	26
<i>Number of Points Detected</i>	26

A plot of corners detected vs. threshold value (sensitivity to threshold value) is shown in Fig. 5.2. It is observed that for an ideal image with a uniform background, Harris-Stephens algorithm detects almost all required points marked on the foot. Considering a more general background, such as one shown in Fig. 5.3, the algorithm detects a significant number of unwanted features in the background as well as on the foot.



**Fig. 5.2 Marked Points Detected Vs Threshold Value for Image 1**



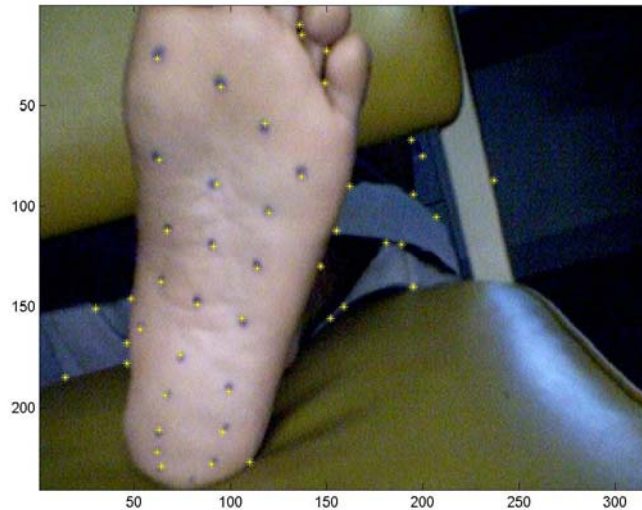
**Fig. 5.3 Result of Points Detected on Actual Foot Image with a General Background**

**Table 5.2 Results of Points Detected on Actual Foot with General Background Using Harris-Stephens Algorithm**

<i>Tracking Window Size <math>w_x</math> (in pixels)</i>	7
<i>Tracking Window Size <math>w_y</math> (in pixels)</i>	7
<i>Threshold Value <math>\tau</math></i>	0.07
<i>Actual marked points on foot</i>	25
<i>Number of Points Detected</i>	129

### 5.6.2 Combined Harris-Stephens and Color Intensity Algorithm

An initial set of feature points are obtained using Harris-Stephens algorithm. The color used to mark on the foot is entered as a parameter in the algorithm. RGB values of all points detected are stored. The selected points are filtered through the color bandwidth. The same image is used for comparison with the previous result obtained. The results obtained does not depend upon texture, complexion and other features of the foot which otherwise may get detected using Harris-Stephens algorithm.



**Fig. 5.4 Result of Points Detected on Same Image Using Combined Algorithms**

It is noted from Fig. 5.4 that, there are certain extraneous points detected in the image besides the actual marked points on foot. This is because a more general background was used to test the algorithm. There are points which satisfy the initial threshold value to qualify as points being detected by Harris-Stephens algorithm. They also happen to pass the color bandwidth to finally appear on Fig. 5.4. A more defined background of a specific environment can eliminate these points and only detect marked points on foot.

**Table 5.3 Results of Points Detected on Actual Foot with General Background Using Combined Algorithm**

<i>Tracking Window Size <math>w_x</math> (in pixels)</i>	7
<i>Tracking Window Size <math>w_y</math> (in pixels)</i>	7
<i>Threshold Value <math>\tau</math></i>	0.07
<i>Actual marked points on foot</i>	25
<i>Number of marked points detected</i>	45
<i>Reduced number of points</i>	84
<i>Color marked on foot</i>	blue
<i>RGB value</i>	[10 10 255]
<i>Bandwidth <math>[5 &lt; R &lt; 15, 5 &lt; G &lt; 15, 240 &lt; B &lt; 255] \pm [5 \ 5 \ 5]</math> pixels</i>	

## 5.7 Conclusions

The chapter begins with the review of earlier works done in corner detection. A brief overview of the Harris-Stephens algorithm is presented, where it is shown to be inefficient in this process. The current work utilizes the color information to eliminate a large number of extraneous points not on the foot. Preliminary results show an improvement over the existing Harris-Stephens algorithm which works on grey scale intensities of pixels. More comprehensive results with different foot images are shown in the succeeding chapters. However as seen in Fig. 5.4, certain extraneous points were also detected. This was because the points were detected in Harris-Stephens algorithm and passed the bandwidth to form feature points. But this limitation can be reduced by using a uniform background and controlled lighting environment.

## Chapter 6

### Feature Matching

#### 6.1 Introduction

Feature based matching consists of finding corresponding features in different images that represent the same features in the observed scene. This is the second and the most important step in 3D reconstruction process. Several methods are reviewed in this chapter and a combined method is presented to more robustly match features or corners in the images. This step assists in the calculation of the fundamental matrix that is required to estimate the extrinsic parameters of the camera system.

#### 6.2 Review of Earlier Works

Feature matching is the most difficult and often most fragile component of the entire 3D reconstruction pipeline. Real scenes often present occlusions, specular reflections, cast shadows, changing illumination and other factors that make feature matching across wide baselines difficult. Different techniques were developed to overcome each of these challenges.

One of the earliest methods uses cross-correlation intensity [MA2001]. Here a pixel window of user specified size is chosen and gross intensity value calculated for the neighborhood. This process is repeated for all selected points identified as corners in a pair of images.

The algorithm then calculates a cross-correlation intensity for each point with all points in the second image. All correlation values are normalized and stored in a matrix. Their values range between  $-1$  to  $1$ .

Any point pair from two images, which have a cross correlation value close to  $1$ , are identified as a match i.e., they represent the image of the same element in the scene, and in this

thesis the points marked on the foot. Points which have cross-correlation values approaching  $-1$ , are deduced to be not correlated and ignored. This step begins the arduous task of feature matching.

An important drawback of this process is that points, which have the same correlation due to similar color intensity, may not really be the same point in the scene. However this algorithm still assists in providing a reasonable collection of matches which can be taken as input in a second stage of process.

The detailed description of the cross-correlation algorithm is as follows. Coordinates of candidate feature points are stored in a matrix. A correlation window of size  $(2n+1) \times (2m+1)$  is centered on each corner detected in the first of the two images, where  $n$  and  $m$  are sizes specified by user. The correlation window is placed around this point in the second image and for all the corners falling inside this area, a correlation score is calculated with the following equation

$$C_{ij} = \frac{\sum_{W(x)} (I_1(\tilde{x}) - \bar{I}_1)(I_2(\tilde{x}) - \bar{I}_2)}{\sqrt{\sum_{W(x)} (I_1(\tilde{x}) - \bar{I}_1)^2 \sum_{W(x)} (I_2(\tilde{x}) - \bar{I}_2)^2}} \quad (6.1)$$

where:

$\bar{I}_1, \bar{I}_2$  = Mean intensity across all pixels for image 1 and image 2 respectively

$I_1(\tilde{x})$  = Intensity value of window of pixels in image 1

$I_2(\tilde{x})$  = Intensity value of window of pixels in image 2

$W(x)$  = correlation window size  $(2n+1) \times (2m+1)$ .

Any set of corners or points which have their cross-correlation intensity value close to 1 is considered a match, and the ones close to  $-1$  are considered a mismatch. Nevertheless, the



ability of the algorithm in selecting matched features is poor and includes matching ambiguous corners.

Scott & Longuet-Higgins [SCOTT1991] proposed a new method of using distances among the points apart from utilizing the color intensities to match corners in two images. An important feature of the algorithm is its implementation, founded on eigenvector solution, which involves no explicit iterations. Proximity of corners is also given importance apart from color intensity values. A brief description of the algorithm is given below.

Let  $I$  and  $J$  be two images, containing a set of  $m$   $I_i(i=1\dots m)$  and  $n$  features  $J_j(j=1\dots n)$ , respectively, which are to be put in one-to-one correspondence. The algorithm begins with building the *proximity matrix*  $\mathbf{G}$  of the two feature sets where each element  $G_{ij}$  is Gaussian-weighted distance between two feature locations  $I_i$  and  $J_j$ :

$$G_{ij} = e^{-r_{ij}^2 / 2\sigma^2} \quad i=1\dots m, j=1\dots n \quad (6.2)$$

where  $r_{ij} = |I_i - J_j|$  is the Euclidean distance as if both images are overlaid in the same coordinate axis.  $G_{ij}$  decreases monotonically from 1 to 0 with distance. The parameter  $\sigma$  controls the degree of interaction between the two sets of features. A small value of  $\sigma$  enforces local interactions, while a larger value permits more global interactions. Singular Value Decomposition (SVD) [PRESS1992] is performed on  $\mathbf{G}_{ij}$ .

$$G = UDV^T \quad (6.3)$$

$U \in M_m$  and  $V \in M_n$  are orthogonal matrices and the diagonal matrix  $D \in M_{m,n}$  contains the positive singular values along its diagonal elements  $D_{ii}$  in descending order. If  $m < n$ , only the first  $m$  elements of  $U$  have any significance [MA2001]. Matrix  $\mathbf{D}$  is converted to a new matrix  $\mathbf{E}$  obtained by replacing every diagonal element  $D_{ii}$  with 1 and then a new product is computed.

$$P = UDV^T \quad (6.4)$$

This new matrix  $P \in M_{m,n}$  has the same shape as the proximity matrix  $\mathbf{G}$  and has the interesting property of “*amplifying good pairings and attenuating bad ones*” [SCOTT1991]. If  $P_{ij}$  is both the greatest element in its row and the greatest element in its column, then the two different features  $I_i$  and  $J_j$  are regarded as being 1:1 correspondence to each other, otherwise they compete unsuccessfully and are discarded. This algorithm is rooted into the solution of subspace rotations problem known as “orthogonal Procrustes problem” [GOLUB1983].

Pilu [PILU1997] proposed to add intensity correlation values along with proximity to determine effective matching. The author worked on the weakness of Scott & Longuet-Higgins’s algorithm where only distances were considered as criteria for feature matching. Pilu added the intensity correlation matrix along with proximity matrix to evaluate  $\mathbf{G}$ .

Pilu’s algorithm begins with using Equation 6.1 to evaluate the intensity cross-correlation for points in two images.  $r_{ij} = |I_i - I_j|$ , the Euclidean distance, between the points is calculated and correlation weighted proximity matrix is estimated as:

$$G_{ij} = [e^{-(C_{ij}-1)^2/2\gamma^2}]e^{-r_{ij}^2/2\sigma^2} \quad (6.4)$$

where  $e^{-(C_{ij}-1)^2/2\gamma^2}$  is the Gaussian weighted function of the correlation  $C_{ij}$  in which  $\gamma$  determines how quickly its values decrease with a diminishing  $C_{ij}$ . This new correspondence strength can be seen as a *correlation weighted proximity*. The values of  $\mathbf{G}$  still range from 0 to 1 and the selection criterion is the same as in the previous algorithm. However, this new correspondence strength embodies the *similarity* between features and is therefore much more selective than just proximity as in Equation (6.2).

The black box nature of this algorithm is an important limitation. It is not possible to embed any non-pair wise constraint, such as the disparity gradient, or the geometric ordering

constraint [POLLARD1985]. A better approach is to recover the epipolar geometry and the fundamental matrix to perform one-to-one stereo correspondence.

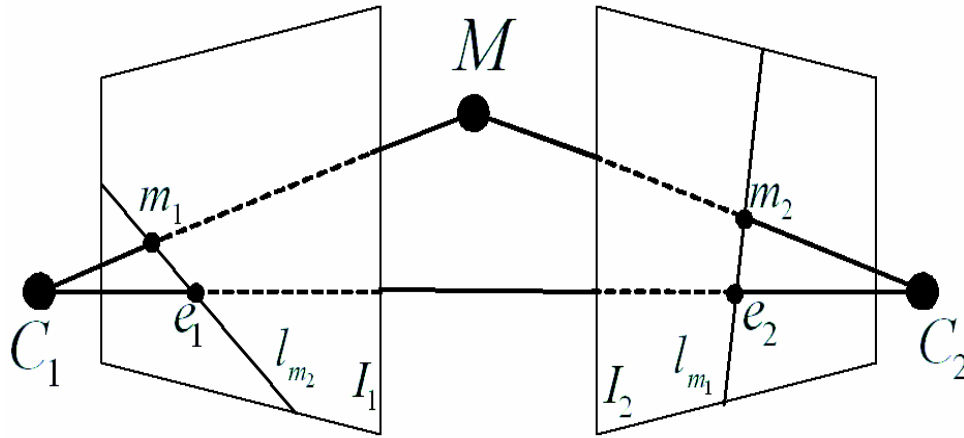
Shapiro and Brady [SHAPIRO1992] used an Eigen vector approach for feature-based correspondence. The algorithm copes with rotations and translations in the image plane, yet still give a dense correspondence. It basically uses the same concept as that of Scott & Longuet-Higgins's algorithm. Here the object could be rotated before taking the second image to get the same results that would be obtained using Scott & Longuet-Higgins's algorithm.

One of the more recent approaches is that used by Zhang and Faugeras [ZHANG1994], where they recover the epipolar geometry by estimating the fundamental matrix to evaluate correct matches from an initial set of matches. The idea underlying this approach is to use the correlation technique to obtain an initial set of matches and then apply robust Least Median of Squares (LMedS) to discard false matches. Then the epipolar geometry is recovered by estimating the Fundamental Matrix and more matches are eventually found.

The process begins with the estimation of at least eight correct correspondences using either the correlation intensity algorithm or Scott & Longuet-Higgins algorithm. Unlike the original algorithm which Zhang implements using sub-sampling techniques and Least Median of Squares (LmedS) technique to determine eight matches correctly, the method adopted in this thesis is simplified by manually choosing eight corresponding corners from the initial results.

Earlier techniques adopt information either using the intensity of the pixels or measuring distances between the points to determine initial matches. This method involves recovering the epipolar geometry between the images and the 3D point in question. This is reversing the process of calibrating the camera, where 3D information is utilized to evaluate the mapping between 3D points to their 2D coordinates in the images.

The input is at least eight good correspondences from two images, which contain information about how the two images are positioned relative to each other in 3D world coordinate system. Since epipolar geometry (see Fig. 6.1) only depends on internal parameters of the camera and relative pose, the scene structure is of no relevance to determine the matches.



**Fig. 6.1 Epipolar Geometry between Points in Two Images and 3D Point in Space**

The fundamental matrix  $\mathbf{F}$  encapsulates this intrinsic geometry. It is a  $3 \times 3$  matrix of rank 2. If a point in 3-space  $\mathbf{X}$  is imaged as  $\mathbf{x}$  in the first view, and  $\mathbf{x}'$  in the second view, then the image points satisfy the relation

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0. \quad (6.4)$$

Though the fundamental matrix is independent of scene structure, it can be computed from correspondence of imaged scene points alone, without requiring knowledge of the cameras' internal parameters or relative pose.  $\mathbf{F}$  maps a point in one image to a corresponding epipolar line in the other image. A full treatise of robust, algebraic estimation of  $\mathbf{F}$  can be found in Zhang's manuscript [ZHANG2001b]. A minimum of seven points are necessary to compute  $\mathbf{F}$  though 8 points could be used to finally corroborate the estimated value of  $\mathbf{F}$ . Once  $\mathbf{F}$  has been determined, all points in image 2 are used to test the epipolar constraint condition in Equation

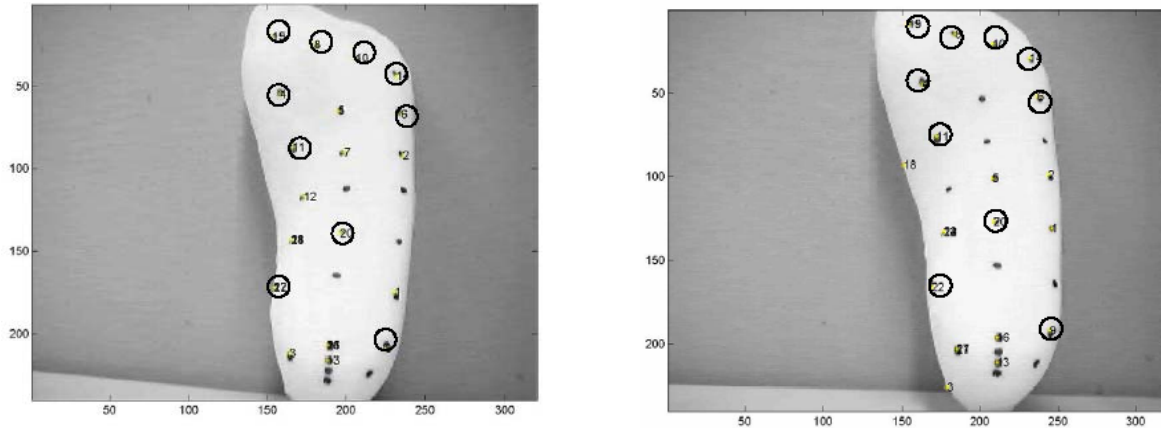
6.4. The points, which satisfy the condition, are treated as a match otherwise they are ignored. This method proves more effective as a solution for feature correspondence compared to earlier methods.

### 6.3 Results

This section reviews results obtained from different methods described earlier. A plastic foot model is used with points marked on it. All algorithms use the same set of images. As a convention, the image to the left is designated as image1 and image to the right as image 2 in all figures. The images are taken by moving the same camera in two different positions. Circled points in both the images indicate a match for all figures. The camera used is Intel<sup>®</sup> CS330 whose specifications are listed in Section 4.7. Section 6.3.5 uses two cameras of the same product specification. The cameras are fixed to the foot scanner in a specific configuration to capture all points on plaster foot.

Initial set of results include applying the intensity cross-correlation algorithm (section 6.3.1) to obtain point matches. Distance proximity matrix related matching is performed by implementing Scott, Longuet-Higgins (section 6.3.2) and Pilu's algorithm (section 6.3.3). Robust matching of points on two images is done using simplified Zhang's algorithm, where the epipolar geometry is recovered by estimating the fundamental matrix. Final set of results of one-to-one point correspondence of points is obtained by evaluating the homography matrix of the stereo image pair.

### 6.3.1 Results Using Intensity Cross-Correlation Algorithm

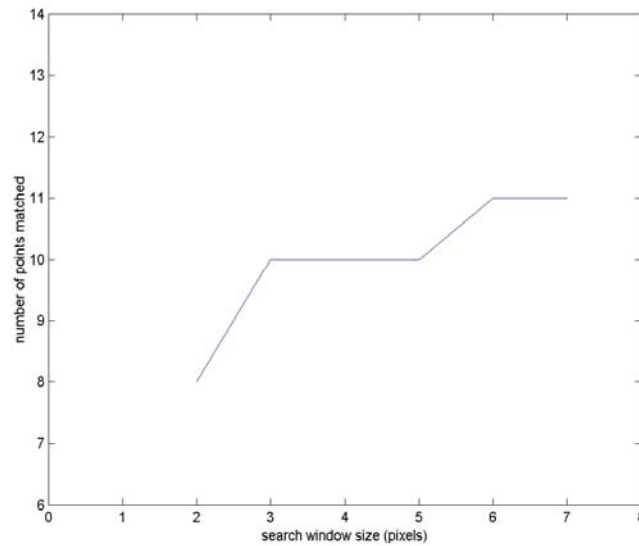


**Fig. 6.2 Initial Matches Using Intensity Cross-Correlation. (Circled points indicate a match)**

Preliminary Feature Correspondence using intensity cross-correlation.

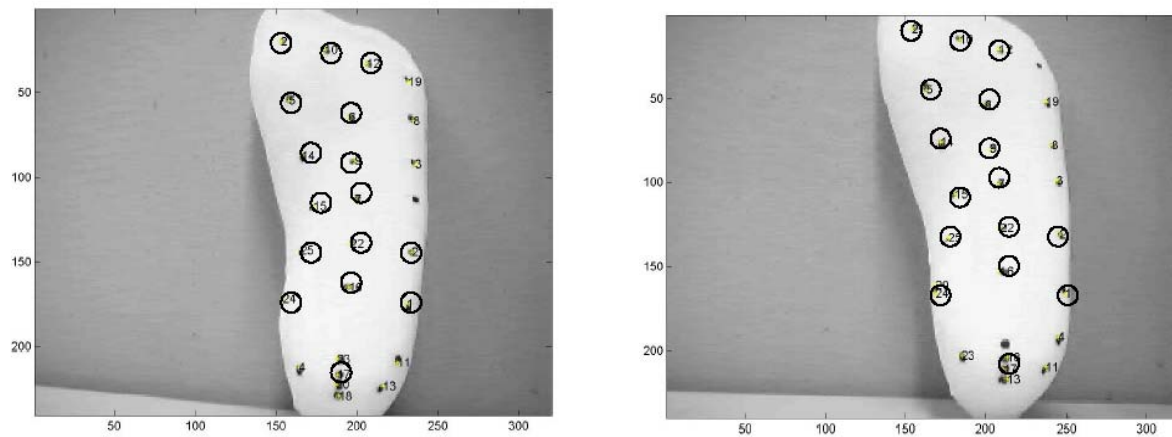
- Actual number of marked points on foot model = 26
- Number of good matches: 10.
- Percentage of good matches: 38%
- Number of ambiguous matches: 9
- Number of points not matched: 7
- Representative search window size: 7 x 7 in pixels

The plot shown in Fig. 6.3 is indicative of different results obtained when the search window size is changed. For small window size such as 2 (in pixels) fewer matches are obtained as compared to window size 7. Increasing the window size farther usually becomes time intensive computationally, but does not increase the number of matches significantly.



**Fig. 6.3 Plot of Search Window Size vs. Number of Matches Using Correlation Algorithm**

### 6.3.2 Results Using Scott, Longuet-Higgins' Algorithm



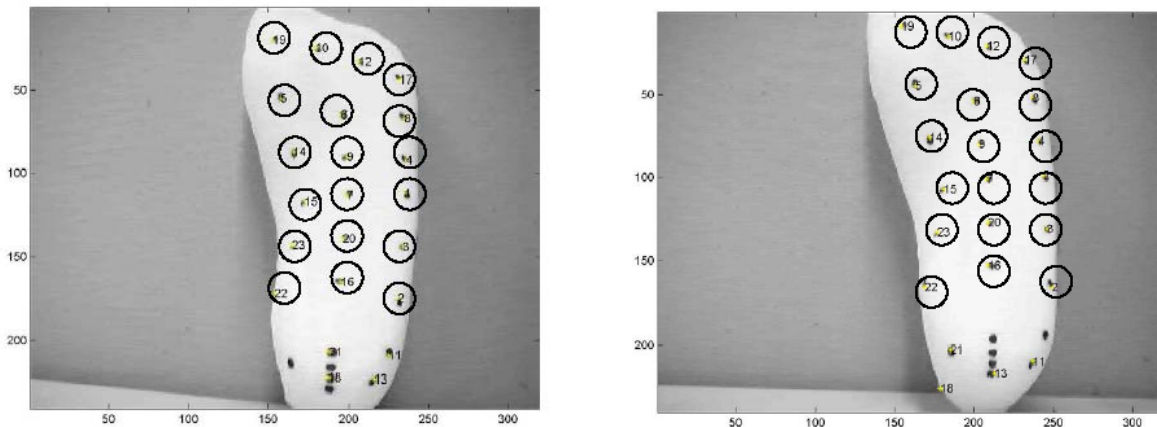
**Fig. 6.4 Feature Matching Using Scott, Longuet-Higgins' Algorithm (Circled points indicate a match)**

Comparing Fig. 6.3 and Fig. 6.4, it is noted that more matches are obtained using Scott, Longuet-Higgins algorithm. The results are listed as follows.

Feature Correspondence using proximity method.

- Actual number of marked points on foot model = 26
- Number of good matches: 16
- Percentage of good matches: 61%
- Number of ambiguous matches: 9
- Number of points not matched: 0

### 6.3.3 Results Using Pilu's Algorithm



**Fig. 6.5 Feature Matching Using Pilu's Algorithm (Circled points indicate a match)**

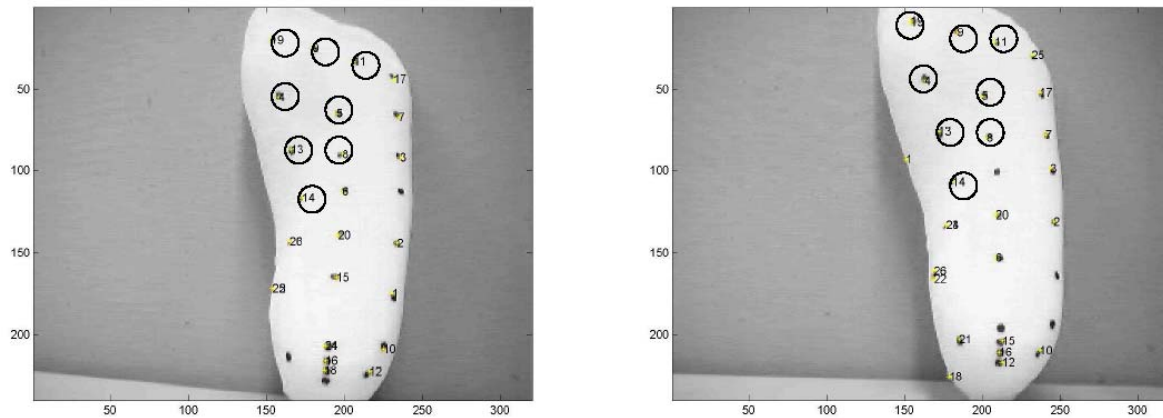
Feature Correspondence using correlation weighted Gaussian proximity matrix method.

- Actual number of marked points on foot model = 26
- Number of good matches: 19.
- Percentage of good matches: 73%
- Number of ambiguous matches: 4
- Number of points not matched: 2
- Representative search window size: 7 x 7 in pixels



### 6.3.4 Results Using Zhang's Algorithm by Recovering Epipolar Geometry

The method used in the thesis is a simplified Zhang's algorithm to obtain more robust matching. The details of which are discussed in Section 6.2. The process begins with an initial matching using intensity cross-correlation technique as described in Section 6.3.1. At least 8 points are picked manually from the result. These points serve as input to determine the fundamental matrix. The epipolar constraint is applied to determine the best match for all points in image 1 to their corresponding points in image 2 by evaluating the fundamental matrix.



**Fig. 6.6 Initial Correspondence using Intensity Cross-Correlation (Circled points indicate a match)**

Initial results are obtained using intensity cross-correlation algorithm. Eight points are selected interactively to determine the fundamental matrix from Fig. 6.6. Fundamental matrix is then estimated using Zhang's method [ZHANG2001]. All points in image 2 are checked for each point in image 1 for epipolar constraint condition. Point in image 2 which satisfies the condition is termed a match to the point in image 1. Coordinates of the initial eight correspondences are listed in Table 6.1.

**Table 6.1 (x, y) Coordinates of 8 Corresponding Points in Image 1 and Image 2**

Point No.	Coordinates of x and y in image 1 (pixels)		Coordinates of x and y in image 2 (pixels)	
	$x_1$	$y_1$	$x_2$	$y_2$
19	156	21	156	9
9	182	26	184	15
11	207	35	210	22
4	159	55	162	44
5	197	65	201	55
13	167	89	173	79
8	198	91	204	79
14	172	118	179	108

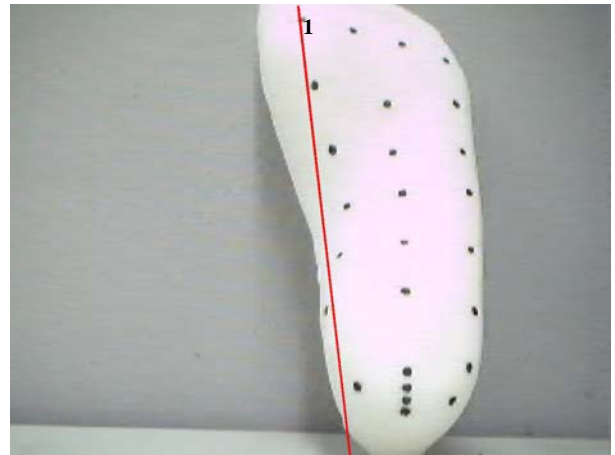
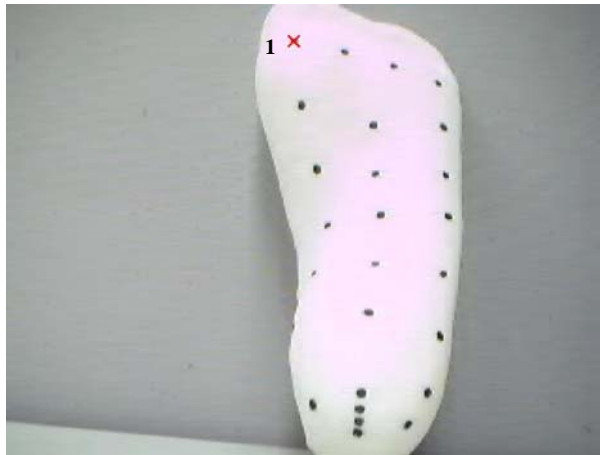
The values of the matrix are as follows.

$$F = \begin{bmatrix} 0 & -0.0001 & 0.038 \\ 0 & 0 & 0.0248 \\ -0.0037 & 0.0254 & -1.2986 \end{bmatrix} \quad (6.4)$$

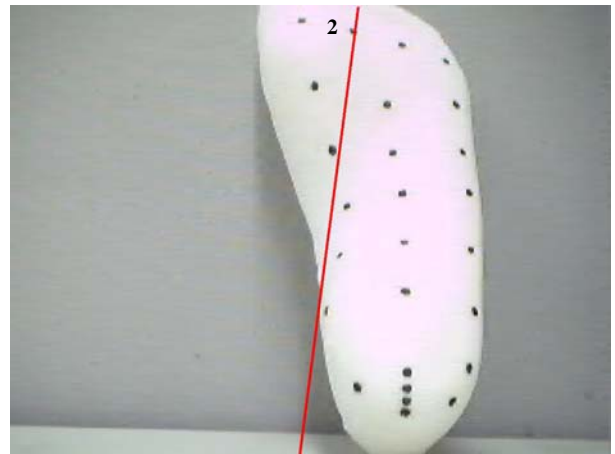
Matrix  $F$  has rank 2 and its determinant is 0. This is because  $F$  maps a point in Image 1 to its epipolar line in Image 2. Epipolar lines are constructed in Image2 using the given set of points in Image 1 and  $F$  using the following equation:

$$l' = FX_1 \quad (6.5)$$

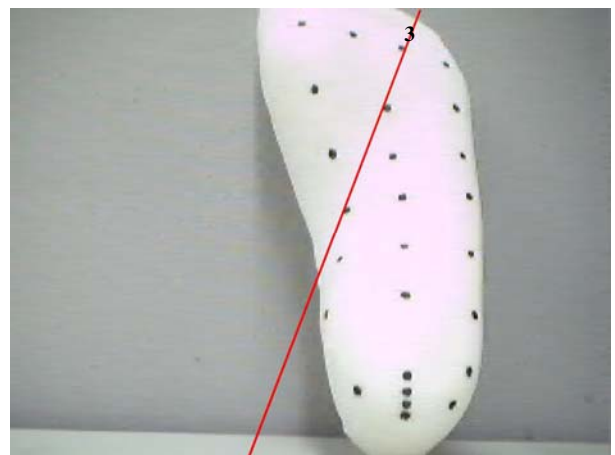
where  $l'$  is the epipolar line in image 2 associated with point  $X_1$  in image 1. Epipolar lines are defined to pass through the corresponding points in image 2. Epipolar lines for 3 of the 8 chosen points are shown in the Fig. 6.7 – Fig. 6.10.



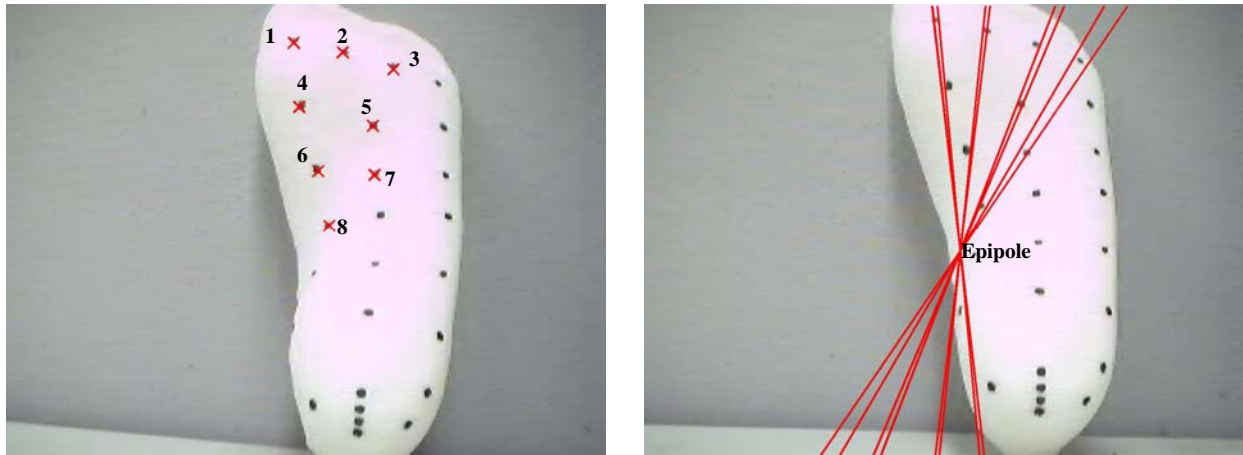
**Fig. 6.7 Point 1 in Image 1 and Epipolar Line passing through Corresponding Point in Image 2**



**Fig. 6.8 Point 1 in Image 1 and Epipolar Line passing through Corresponding Point in Image 2**

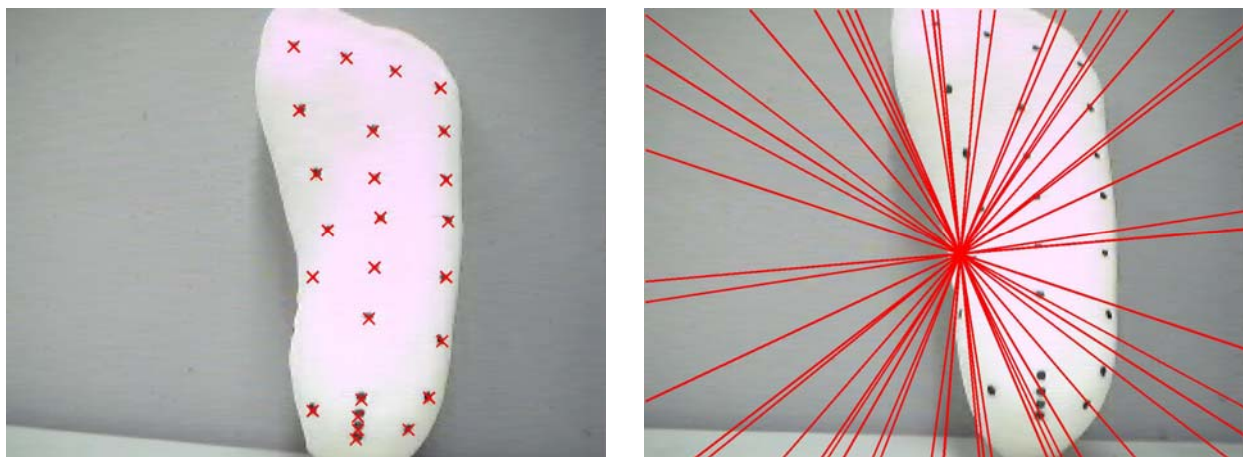


**Fig. 6.9 Point 1 in Image 1 and Epipolar Line passing through Corresponding Point in Image 2**



**Fig. 6.10 All 8 Points in Image 1 and Epipolar Lines passing through The Corresponding Points in Image 2**

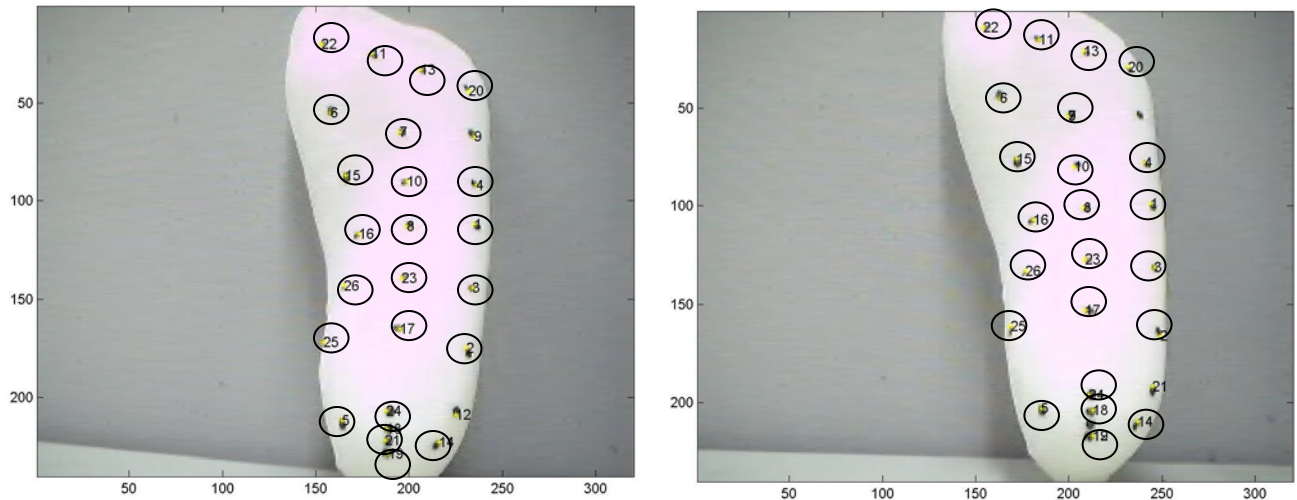
What can be observed from Fig. 6.7 is that all epipolar lines intersect at one point which is known as the **Epipole**. This characteristic point is also defined as the “*point of intersection of the line joining the camera centers with the image plane*” [MA2001]. All points in image 1 can now be matched to image 2 using the epipolar constraint given in Eqn. 6.5.



**Fig. 6.11 All Points in Image 1 and Epipolar Lines passing through the Corresponding Points in Image 2**

The correspondence algorithm begins by applying the epipolar constraint to each of the feature points in image 1 to all points in image 2. As points do not exactly lie on the epipolar

line, residual error is used to find the closest point to that line. The point number and the coordinates are also stored. This point then corresponds to point 1 in image1. The same step is repeated  $n$  times, where  $n$  is the number of points in image 1. The final matrix is then arranged in the same order as the points in image 1 and correlated feature points are plotted as shown in Fig. 6.12.



**Fig. 6.12 Final Feature matching after Applying Epipolar Constraint to Two Images**

Final feature correspondence by recovering the epipolar geometry:

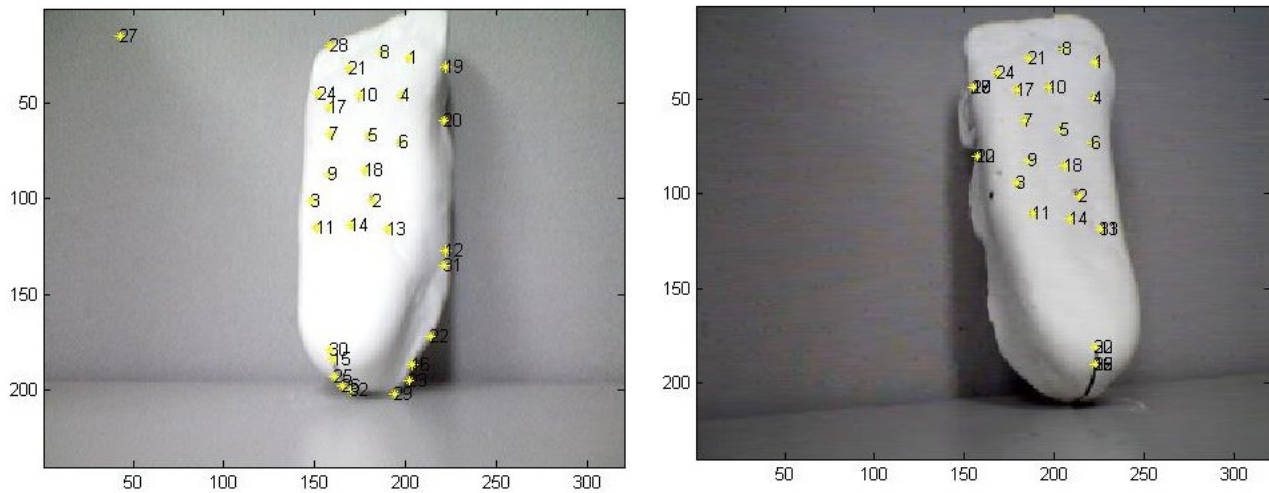
- Actual number of marked points on foot model = 26
- Number of good matches: 23
- Percentage of good matches: 88%
- Number of ambiguous matches: 1
- Number of points not matched: 1
- Representative search window size: 7 x 7 in pixels.

It is observed that, the results obtained here are a significant improvement from Pilu's algorithm. This method partially depends on the quality of the images or intensity variations to correspond the points unlike the correlation algorithm.

### 6.3.5 Results Using Fixed Configuration of Cameras

This section deals with fixing the positions of two cameras in a particular configuration and obtaining different images. This is to verify the fundamental matrix having the same value, because it depends, only on the camera matrix and relative pose between the cameras. The experiment was carried out fixing the cameras at the optimal location to capture marked points on the foot model. Two cameras were used in the process.

#### 6.3.5.1 Case 1



**Fig. 6.13 Case 1: Feature matching of Points on Plaster Foot by Recovering Fundamental Matrix**

In Case 1, points were detected on the foot. It is seen that more points were detected than marked on image 1. But by evaluating the fundamental matrix and performing correspondence, most points marked on foot were matched correctly. The results are given below.

- Actual number of marked points on foot model = 17
- Number of good matches: 16.
- Percentage of good matches: 94%
- Number of ambiguous matches: 0
- Number of points not matched: 0

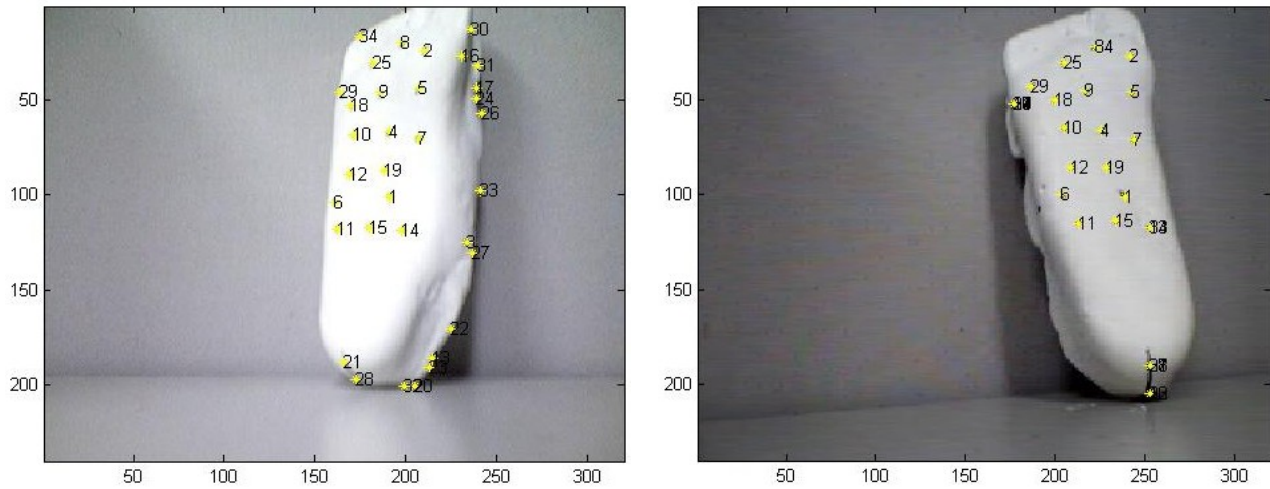
- Number of points with multiple correspondence: 1
- Representative search window size: 7 x 7 in pixels.

The fundamental matrix obtained in case 1 is given below:

$$F_{Case1} = \begin{bmatrix} 0 & 0.0001 & -0.0059 \\ -0.0001 & 0 & -0.013 \\ -0.0111 & 0.0202 & -1.3809 \end{bmatrix} \quad (6.6)$$

$F_{Case1}$  has rank 2 and its determinant is 0 as expected.

### 6.3.5.2 Case 2



**Fig. 6.14 Case 2: Feature matching of Points on Plaster Foot by Recovering Fundamental Matrix**

In Case 2, unlike case 1, the fundamental matrix was not evaluated using an initial set of matches. Instead,  $F_{Case1}$  was used for point correspondence. It is seen that most points on the foot were matched correctly using the same fundamental matrix. This shows that  $F$  remains the same for a fixed configuration of cameras. Since the cameras were fixed to a foot scanner frame, just one trial is needed to estimate  $F$ . All subsequent pairs of images can use the same matrix to perform feature correspondence. The results for case 2 are given below.

- Actual number of marked points on foot model = 17

- Number of good matches: 16.
- Percentage of good matches: 94%
- Number of ambiguous matches: 0
- Number of points not matched: 0
- Number of points with multiple correspondence: 1

The representative search window size: 7 x 7 in pixels. The fundamental matrix obtained after using at least eight correlated points to check for accuracy is:

$$F_{Case2} = \begin{bmatrix} 0 & 0.0001 & -0.0068 \\ -0.0001 & -0.0001 & -0.0231 \\ -0.0103 & 0.0239 & -1.4925 \end{bmatrix} \quad (6.7)$$

Comparing Eqns. 6.6 and 6.7, it is seen that most values in the matrices are the same. Small variations occur due to corner detection errors.

### 6.3.6 Feature Matching Using Homography Matrix

One of the important results obtained by recovering the epipolar geometry is determining the extrinsic parameters (rotational matrix  $R$  and translational matrix  $T$ ) of the camera. From this a direct mapping of one point in its image to its corresponding point in the second image is possible through what is known as the homography matrix. Determining  $R$  and  $T$  is dealt in more detail in Chapter 7 by means of implementing the eight-point algorithm [LONGUET-HIGGINS1981].

The points in two images ( $\mathbf{x}_1, \mathbf{x}_2$ ) are given for a point  $p \in P$  with respect to two camera frames as input. The coordinate transformation between the two frames are then given as,

$$\mathbf{x}_2 = R \mathbf{x}_1 + T, \quad (6.8)$$

where,  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are the coordinates of  $p$  relative to camera frames 1 and 2, respectively. Since the two points in two images satisfy the epipolar constraint,



$$\mathbf{x}_2^T E \mathbf{x}_1 = 0. \quad (6.9)$$

where,  $E$  is called the essential matrix (see Section 7.6). However, for points on the same plane  $P$ , their images will share an extra constraint apart from the epipolar constraint.

Let  $N = [n_1, n_2, n_3]^T$  be the normal unit vector of plane  $P$  with respect to the first camera frame, and let  $d > 0$  denote the distance from the plane  $P$  to the optic center of the first camera [MA2001]. Then,

$$N^T \mathbf{x}_1 = n_1x + n_2y + n_3z = d \Leftrightarrow (1/d) N^T \mathbf{x}_1 = 1 \quad (6.10)$$

Therefore Eqn. 6.8 becomes,

$$\mathbf{x}_2 = (R + T(1/d) N^T) \mathbf{x}_1. \quad (6.11)$$

The matrix,

$$H = R + T(1/d) N^T, \quad (6.12)$$

is called the homography matrix since it denotes a linear transformation from  $\mathbf{x}_1$  to  $\mathbf{x}_2$  given as,

$$\mathbf{x}_2 = \lambda H \mathbf{x}_1 \quad (6.13)$$

where  $\lambda$  is the scale factor from the image pair considered.

This important result is used in this section to determine the homography matrix of a stereo image pair after obtaining the relative pose between the cameras. This would finally ensure the best feature mapping of all points in image 1 to 2.

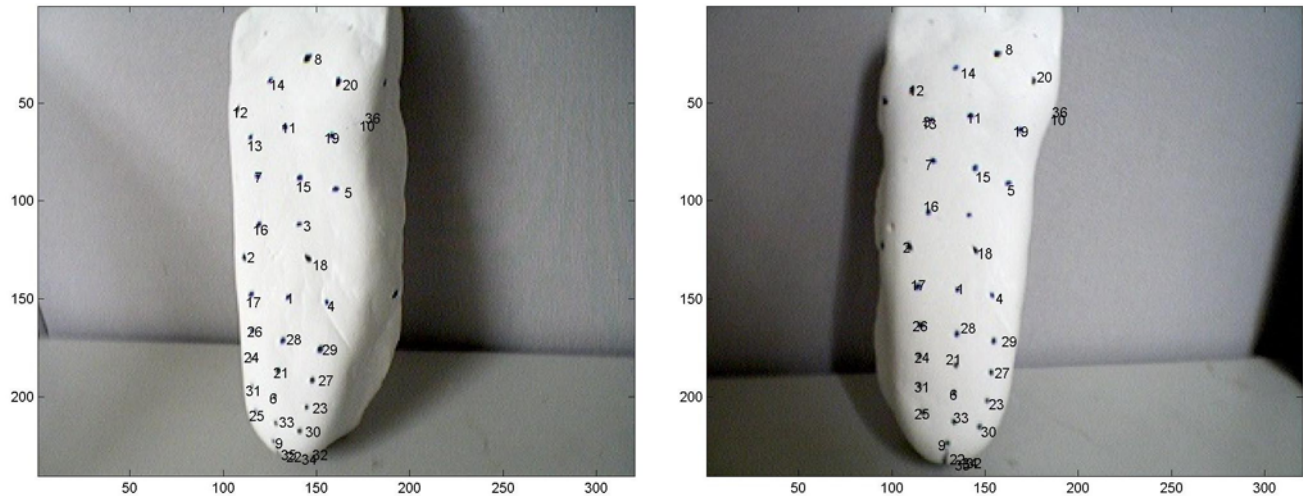
### 6.3.7 Results Using Homography Matrix

A pair of images of the same plaster foot is taken. Eight initial correspondences are determined using the intensity cross-correlation algorithm. The homography matrix is then evaluated using Eqn. 6.12.

$$H = \begin{bmatrix} -0.754 & 0.0048 & 18.51 \\ -0.1044 & -0.5625 & 16.5476 \\ -0.0006 & -0.0002 & -0.482 \end{bmatrix} \quad (6.14)$$

Using the matrix as obtained above, all detected points in image 1 can be correlated to the same points in image 2. The scale factor for point 1 obtained was,

$$\lambda = \begin{bmatrix} 1.88 \\ 1.9 \\ 1.88 \end{bmatrix} \quad (6.15)$$



**Fig. 6.15 Feature Matching of Points on Plaster Foot using Homography Matrix**

It is observed that only one point has a mismatch. All other points detected on image1 are matched to their corresponding points in image 2.

#### 6.4 Conclusions

This chapter takes an overview of different methods used in feature matching. It also gives perspective on the drawbacks of each method and suggesting modified Zhang's method of recovering the epipolar geometry as a means for robust feature matching. To perform complete feature point correspondence, sub sampling and LMedS techniques [ZHANG2001] could be utilized besides recovering epipolar geometry.

## Chapter 7

### 3D Reconstruction

#### 7.1 Introduction

This chapter outlines the steps involved in obtaining 3D point geometry of foot in a stereo image pair. Considered the final process in the reconstruction pipeline, it relies completely on the data provided in the earlier stages. The process to reconstruct the 3D geometry needs camera calibration information, feature points detected in a pair of images and at least eight corresponding point pairs from the two images. A series of steps is performed, beginning with determining the essential matrix, which contains the relative pose of the pair of cameras. Once the relative pose or extrinsic parameters of the pair of cameras is determined, linear 3D triangulation method is used to compute the 3D coordinates in space. First, the 3D points of an unscaled structured model are reconstructed. It is followed by reconstruction of scaled 3D points by using a reference cube which is outlined in section 7.6.2.

#### 7.2 Assumptions Made to Eliminate Reconstruction Ambiguity

In this section inherent ambiguities in 3D reconstruction are discussed and assumptions made to eliminate them. Without some knowledge of a scene's placement with respect to a 3D coordinate frame, it is generally not possible to reconstruct the absolute position or scale of a scene from a pair of views [HARTLEY2004]. This is independent of any knowledge which may be available about the internal parameters of the cameras, or their relative placement. The best depth estimation of the scene in 3D space is obtained when the two images are placed at vantage points where one image has both known rotational and translational displacement with respect to the other. The assumptions made to eliminate any reconstruction ambiguity are:

- Both the images are not coplanar or parallel with respect to one of the image coordinate systems.
- Both images do not have a pure translation component among each other. In other words, image 2 does not lie on the same  $Z$  plane as image 1 in one of the image coordinate systems.
- There is no projective ambiguity since internal parameters of the cameras as well as the relative pose between them are known (the cameras are fixed in the foot scanner and they have a definite configuration between them). Projective ambiguity occurs when the angle between the lines joining the image points and the 3D point in space is some is not a definite value.

### 7.3 Steps Towards Euclidean Reconstruction

As mentioned in the earlier section, the cameras are positioned in a particular configuration in the foot scanner. This helps evaluate the relative pose in terms of translation  $T$  and rotation  $R$  of one of the cameras with respect to the other. This transformation is evaluated using the eight point algorithm. For subsequent trials, matrices  $R$  &  $T$  remain the same. Calibrating the camera also helps to define a better structured scene in 3D space. Though introductory concepts and transformations of projective and affine geometry were reviewed in Chapter 2, the approach here is to use both the extrinsic and intrinsic parameters of the camera to obtain a Euclidean 3D reconstruction.

### 7.4 Eight Point Algorithm

The eight point algorithm for computing the essential matrix was introduced by Longuet-Higgins [LONGUET-HIGGINS1981]. In his work, the essential matrix was used to compute the structure of a scene from two views with calibrated cameras. Among the advantages of the eight

point algorithm are that it is linear, hence computation is fast and stable. The important property of the essential matrix is that it conveniently encapsulates the epipolar geometry of the imaging configuration. Later work by Hartley [HARTLEY1997] showed that the same algorithm could be extended to determine the fundamental matrix if a pair of uncalibrated images was used, but in this case only up to a projective reconstruction.

The work in this thesis uses the eight point algorithm to recover rotational  $R$  and translational  $T$  matrices between a pair of cameras, since calibrated images are used as input. A general treatise on the algorithm is succinctly described in *Invitation to Computer Vision* [MA2001].

For a given set of image correspondences  $(\mathbf{x}_1^j, \mathbf{x}_2^j), j = 1, 2, \dots, n$  ( $n \geq 8$ ), this algorithm recovers  $(R, T) \in S^3$ , where  $S^3$  is defined as the image coordinate system of camera 1. First an approximation of the essential matrix is constructed as  $\chi = [\mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^n]^T$  from correspondences  $\mathbf{x}_1^j$  and  $\mathbf{x}_2^j$  as in Equation 7.1.

$$\mathbf{a}^j = \mathbf{x}_1^j \otimes \mathbf{x}_2^j. \quad (7.1)$$

Vector  $\mathbf{E}^s$  is defined such that  $\|\mathbf{a}^j \mathbf{E}^s\|$  is minimized. This is done by performing Singular Value Decomposition on  $\mathbf{a}^j$  given by,

$$\text{SVD of } \mathbf{a}^j = U_\chi D_\chi V_\chi^T, \quad (7.2)$$

and define  $\mathbf{E}^s$  to be the ninth column of  $V_\chi$ . The nine elements of  $\mathbf{E}^s$  are unstacked into a 3x3 matrix  $E_p$ . SVD on  $E_p$  gives,

$$E = U \text{diag}(\sigma_1, \sigma_2, \sigma_3) V^T, \quad (7.3)$$

where,  $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$  and  $U, V \in 3$  D orthogonal group. The projection onto the normalized essential space is evaluated by replacing  $\sigma_1, \sigma_2, \sigma_3$  with  $[1, 1, 0]$ . Therefore  $E$  now becomes,

$$E = U \Sigma V^T, \quad (7.4)$$

where,  $\Sigma = \text{diag}\{1,1,0\}$ . Displacement parameters ( $R$  and  $T$ ) are recovered from the essential matrix given by,

$$R = UR_z^T \left( \pm \frac{\pi}{2} \right) V^T, \quad \hat{T} = UR_z^T \left( \pm \frac{\pi}{2} \right) \Sigma U^T, \quad (7.5)$$

where,

$$R_z^T \left( \pm \frac{\pi}{2} \right) = \begin{bmatrix} 0 & \pm 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.6)$$

and,

$$\hat{T} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}. \quad (7.7)$$

The matrix  $\hat{T}$  is always rank deficient, i.e., rank =2. The number of points, eight, assumed by the algorithm, is mostly for convenience and simplicity of presentation. Matrix  $E$  (as a function of  $(R,T)$ ) has only a total of five degrees of freedom: three for rotation and two for translation (up to a scalar factor). Many subsequent works introduced different algorithms [MAYBANK1993] to work on lesser points (seven to five points), but it was generally concluded that the eight point algorithm works reasonably well when adequate information of the 3D configuration between the cameras and their parameters are defined [HARTLEY1997].

## 7.5 3D Structure Computation

This section describes the computation of a 3D point in space, given its image in two views and the camera matrices of those views. It is assumed that there are errors only in the measured image coordinates and not in the camera calibration matrix.

The common process of reconstruction of structure by back-projecting rays from the measured image points will result in bad results because the rays will not intersect in general. It

is therefore necessary to obtain an optimal solution to 3D triangulation by reprojecting the points in the second image and minimizing the correspondence error.

The measured points  $x_1$  and  $x'_2$  in image 1 and 2 may have errors associated with them. When the rays from the points are back-projected, they may be skew to each other and may not intersect at a point  $X_1$  as desired. Also, the image points may not exactly satisfy the epipolar constraint,  $x'^T F x = 0$ . To obtain a successful set of feature correspondences, we need to minimize the residue of the equation  $x'^T F x$ . Once a set of good points have been established, 3D linear triangulation can be done to obtain the 3D point in space.

The eight-point algorithm described in Section 7.4 uses an input of eight point correspondences and returns the relative pose (rotation and translation) between the two cameras. In terms of images and depths, the rigid body equation is given by,

$$\lambda_2^j x_x^j = \lambda_1^j R x_1^j + \gamma T, j=1,2,\dots,n \quad (7.8)$$

where,  $\lambda_1$  and  $\lambda_2$  are the structural scales and  $\gamma$  is defined as the motion scale for the two camera system. For each point,  $\lambda_1, \lambda_2$  are its depths with respect to the first and second camera frames respectively. One of them is therefore redundant; for instance, if  $\lambda_1$  is known, then  $\lambda_2$  is simply a function of  $(R, T)$ . Hence  $\lambda_2$  can be eliminated from Equation. 7.8 by multiplying both sides by  $\hat{x}_2$ , which yields,

$$\lambda_1^j \hat{x}_2 R x_1^j + \gamma \hat{x}_2 T = 0, j=1,2,\dots,n. \quad (7.9)$$

This is equivalent to solving the linear equation,

$$M^j \lambda_j = \begin{bmatrix} \hat{x}_2^j R x_1^j, \hat{x}_2^j T \end{bmatrix} \begin{bmatrix} \lambda_1^j \\ \gamma \end{bmatrix} = 0. \quad (7.10)$$

where,  $M^j = \begin{bmatrix} \hat{x}_2^j R x_1^j, \hat{x}_2^j T \end{bmatrix}$  and  $\lambda_j = \begin{bmatrix} \lambda_1^j \\ \gamma \end{bmatrix}$ , for  $j = 1, 2, \dots, n$ .

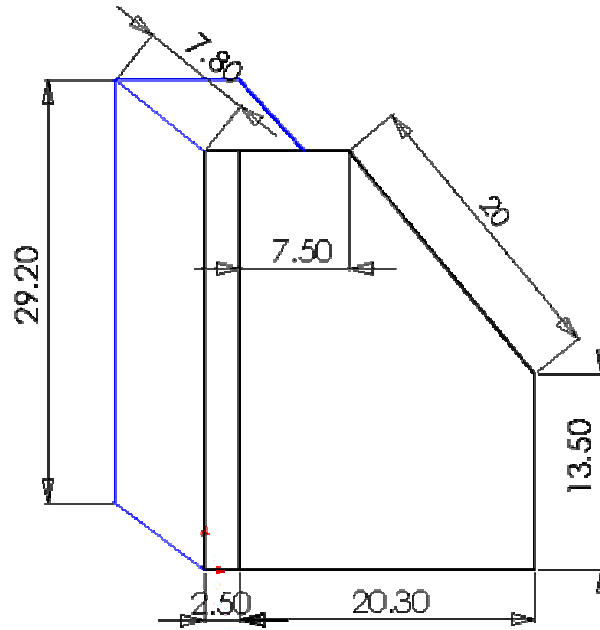
In order to have a unique solution, the matrix  $M^j$  needs to be of rank 1. In other words, the point  $P$  lies on the line connecting the two optical centers  $o_1$  and  $o_2$ . Equation 7.10 determines all the unknown depths *up to a single universal scale* [MA2001]. The linear least-squares estimate of  $\lambda$  is the Eigenvector of  $M^T M$  that corresponds to its smallest Eigenvalue. It is also noted that the 3D points reconstructed have coordinates with respect to the image coordinate system of image 1.

## 7.6 Results

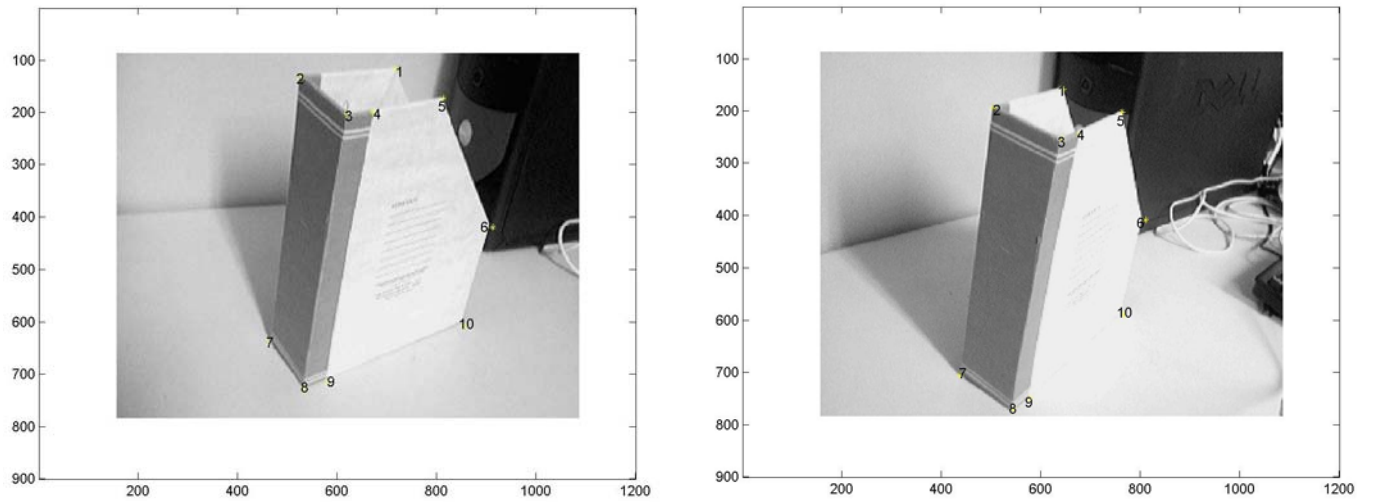
### 7.6.1 Reconstruction of a Shelf-File

This section shows the reconstruction of a simple structured object such as a shelf-file. The process begins with point detection (in this case the corners of the box). Point correspondence among the points in two images is performed next and extrinsic parameters of the camera system are estimated using eight-point algorithm. A linear 3D triangulation method is used to reconstruct the corners of the box on a 3D coordinate system, which is the coordinate system of image 1. The actual dimensions of the file are as shown in Fig. 7.1. The resulting 3D point cloud of the reconstructed model as shown in Fig. 7.3 is not scaled. Techniques to scale the model are discussed with results in section 7.6.2.

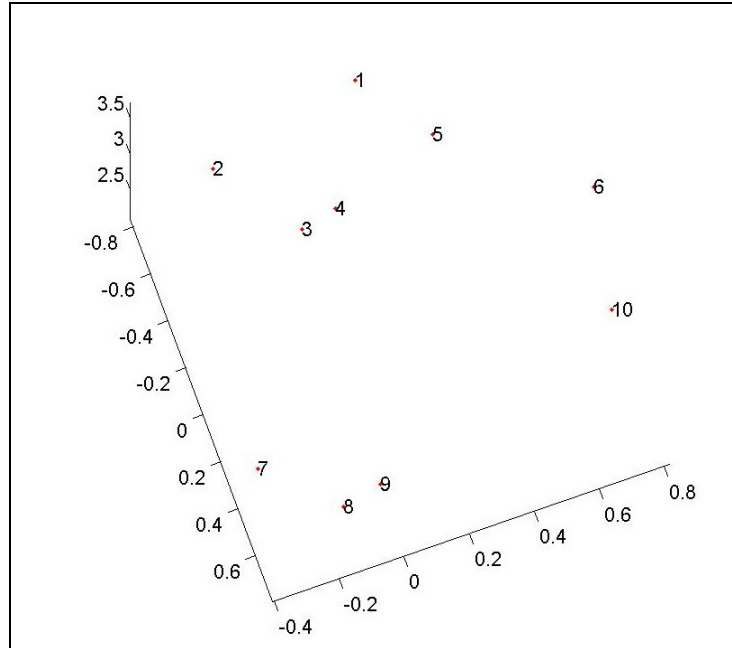




**Fig. 7.1 Dimensions of Shelf-File**



**Fig. 7.2 Marked Corners on Shelf-File on Both Images**

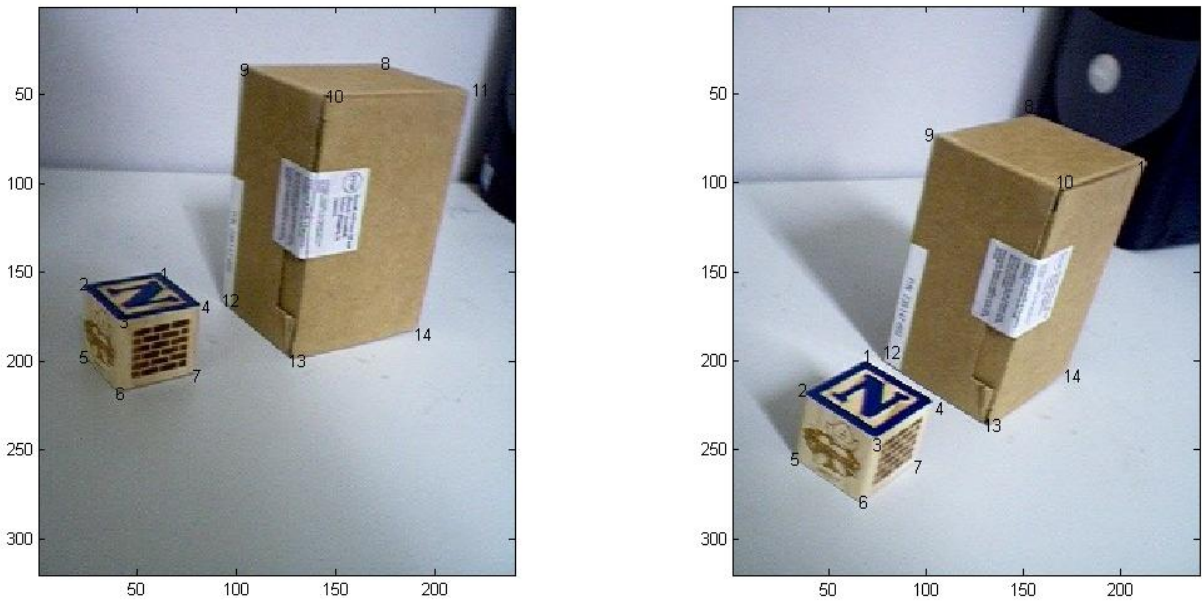


**Fig. 7.3 3D points of Shelf-File**

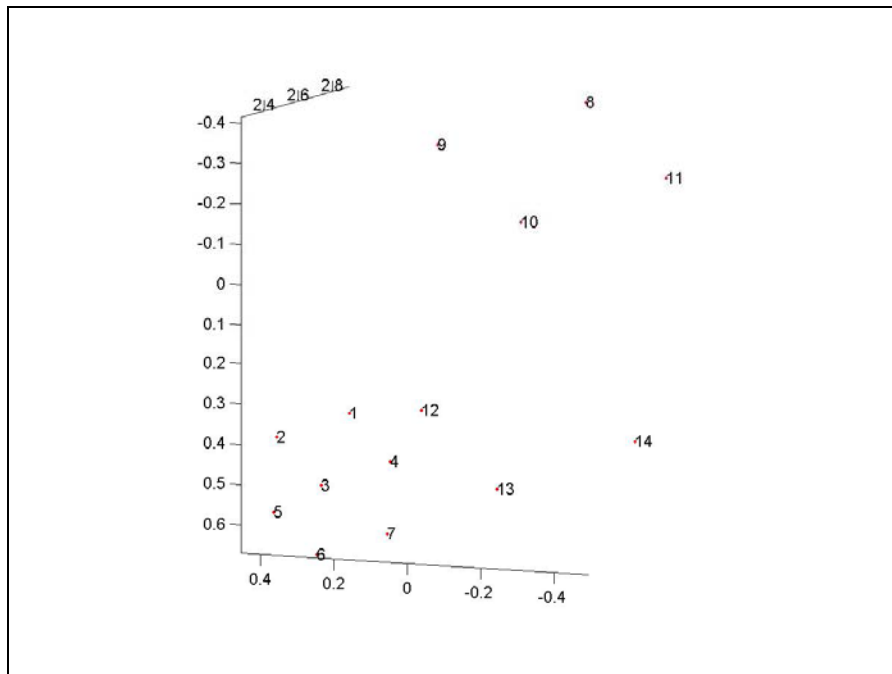
### 7.6.2 Reconstruction of a Scaled Box Using a Reference Cube

This section describes the process of scaling the 3D point data to absolute metric values. This is performed by placing a reference entity besides the object, to be reconstructed. The reference object dimensions are measured. In this thesis, a cube of edge length 4.5 cm is considered for simplicity (see Figure 7.4). All the steps leading to feature matching corresponding points are performed as described in earlier chapters. Points of the cube are also detected and matched. The final 3D point cloud is reconstructed using stereo image pair.

The 3D point distances of the cube obtained from reconstruction are determined. From this, a scale factor is evaluated. The same scale factor is used to scale the reconstructed model to its metric units. Results for a simple reconstructed box are shown in Fig. 7.5.



**Fig. 7.4 Marked Points on Object and Reference Cube**



**Fig. 7.5 3D points of Object and Reference Cube**

**Table 7.1 Distances between Points in Reference Cube**

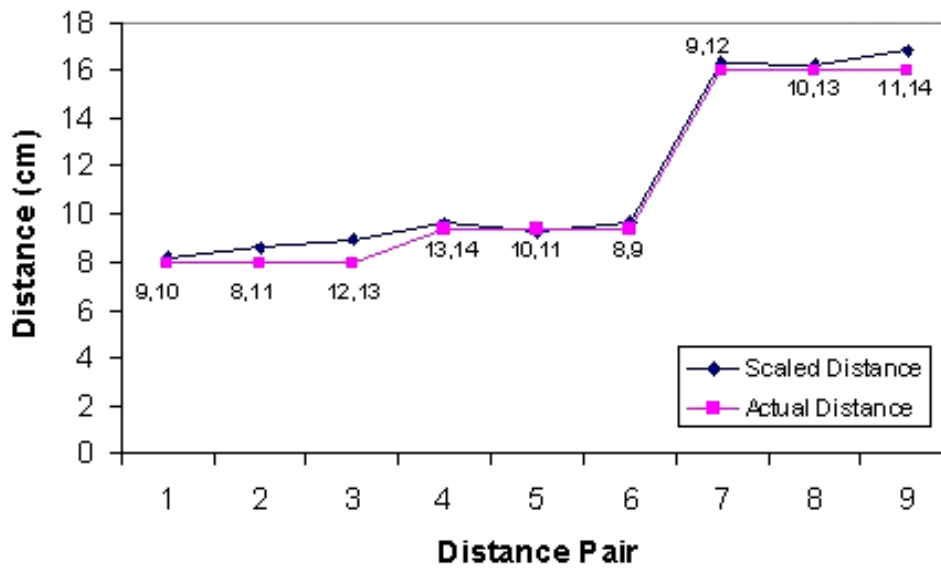
Distance Pair (points)	Distance $d$
1,2	0.23
2,3	0.24
3,4	0.23
5,6	0.23
6,7	0.24
1,4	0.24
2,5	0.23
3,6	0.23
4,7	0.24

$$d_{3D} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (7.11)$$

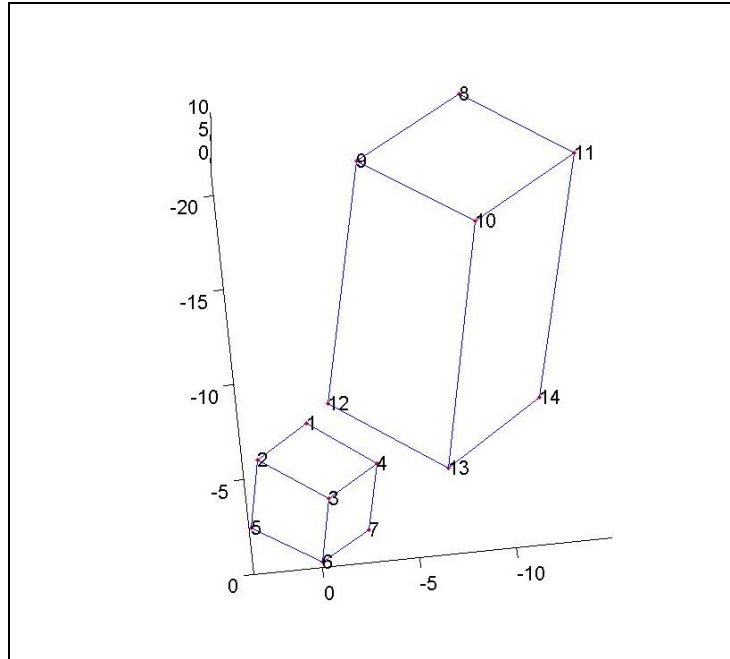
The Euclidean distance measured between points is given by Equation 7.11. Actual distance between the points in the reference cube is  $d_a=4.5$  cm. From Table 7.1, the averaged distance between all measured points is  $d_{avg}=0.23$ . The associated scale factor is then estimated as  $s= d_a/d_{avg} = 19.56$ . The same scale is applied to the distances of the reconstructed box. The final scaled distances and the measured distances of the box are tabulated in Table 7.2. A plot of actual distances and scaled distances are also shown in Fig. 7.6. It is noted that, the small differences in values are due to point detection errors that occur during feature detection algorithm. All errors lie between 1.27 – 9.82%. The same reference cube is placed besides a foot during image capture. The process as described in this section is repeated to obtain a scaled 3D model of the foot.

**Table 7.2 Comparison of Scaled Distances and Actual Distances of the Object**

Distance Pair (Points)	Scaled Distance $d_s$ (cm)	Actual Distance $d_a$ (cm)
9,10	8.24	8
11,8	8.6	8
12,13	8.93	8
13,14	9.62	9.4
10,11	9.28	9.4
8,9	9.71	9.4
9,12	16.31	16
10,13	16.26	16
11,14	16.81	16



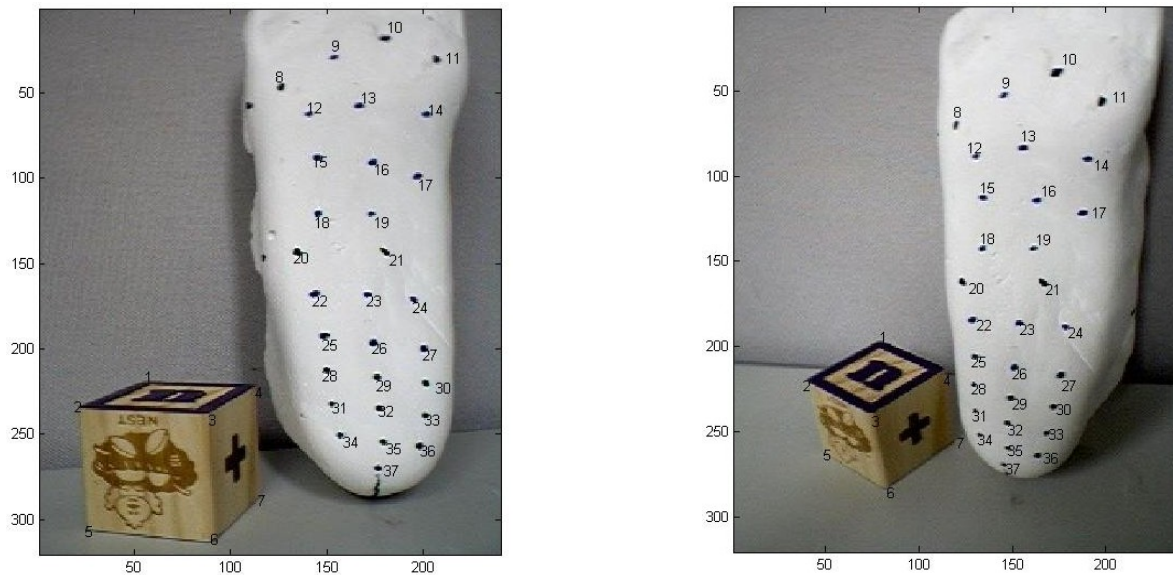
**Fig. 7.6 Comparisons of Scaled Distances and Actual Distances of Box**



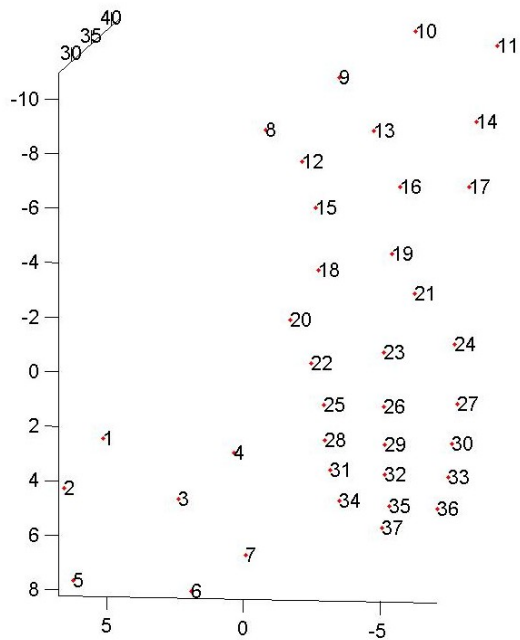
**Fig. 7.7 3D Wire Frame Model of Scaled Object and Reference Cube**

### 7.6.3 Reconstruction of a Plaster Foot Using a Reference Cube

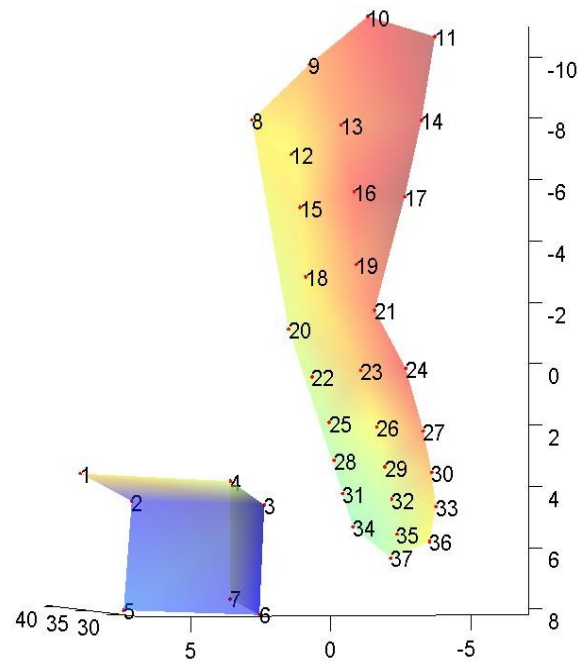
The process as described in section 7.6.2 is used to reconstruct plaster foot geometry. The same reference cube whose dimensions are known is placed besides the foot. The scale factor obtained here was 24.48. The scaled 3D points of plaster foot are shown in Fig. 7.9.



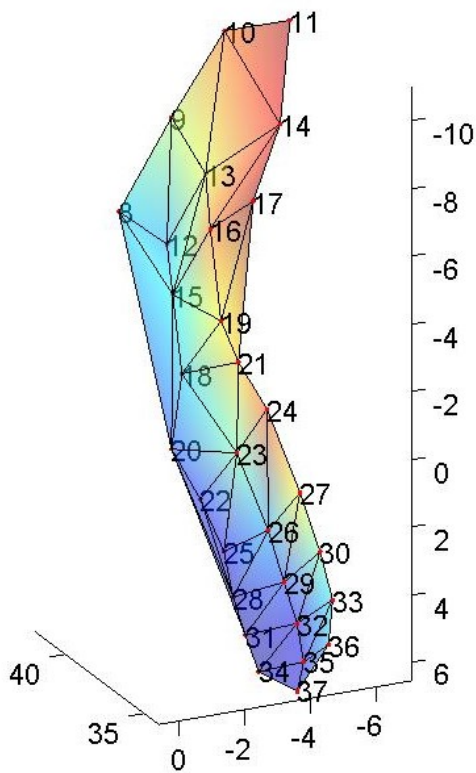
**Fig. 7.8 Marked Points on Plaster Foot and Reference Cube**



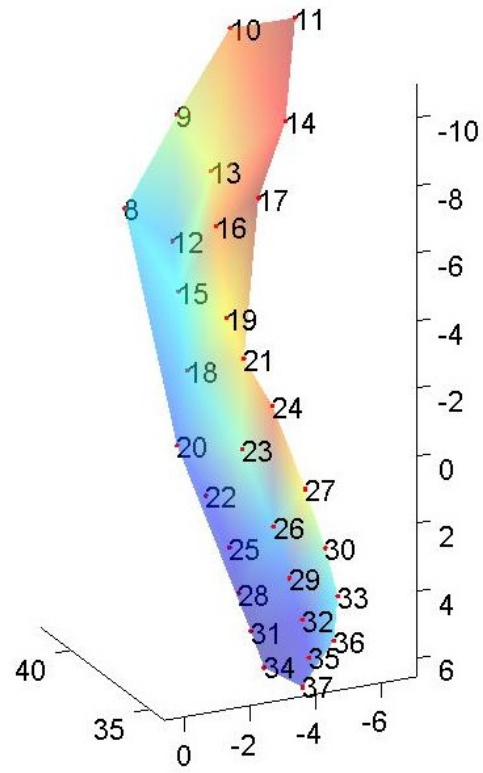
**Fig. 7.9 3D Point Cloud of Plaster Foot and Reference Cube**



**Fig. 7.10 Surfaced Interpolated 3D Points of Plaster Foot and Reference Cube**



**Fig. 7.11 Triangulated Patches between 3D Points of Plaster Foot**



**Fig. 7.12 Surface Interpolated 3D points of Plaster Foot**

### 7.6.4 Reconstruction of an Actual Foot Using a Reference Cube

The same process of image capture, feature detection and matching is performed on the stereo pair images of a real foot. The scale factor obtained here was 17.11. The scaled 3D points of the actual foot are shown in Fig. 7.13.

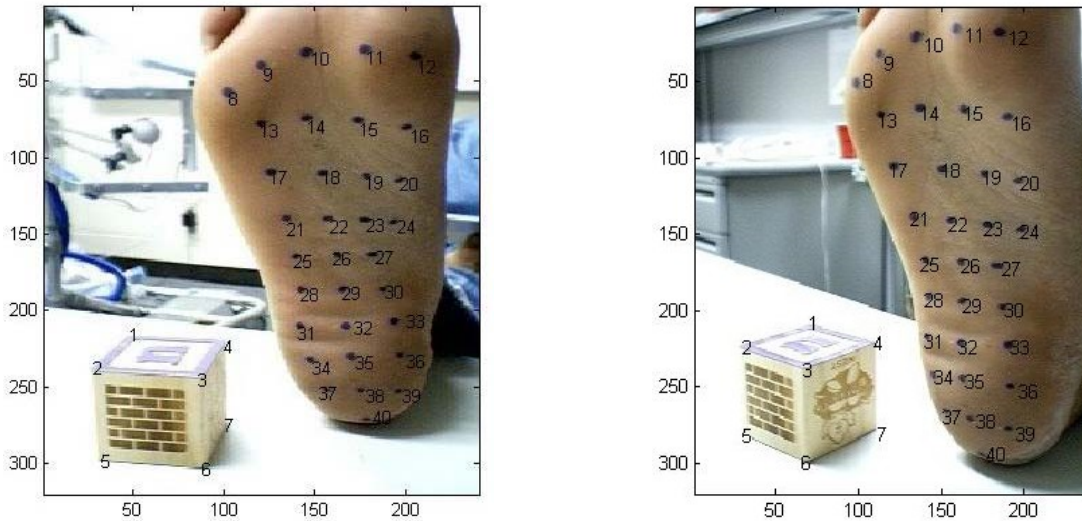


Fig. 7.13 Marked Points on Actual Foot and Reference Cube

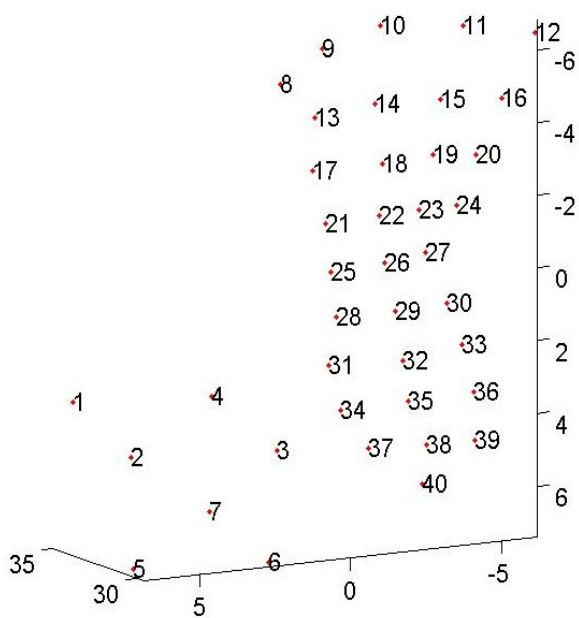


Fig. 7.14 3D Point Cloud of Actual Foot and Reference Cube

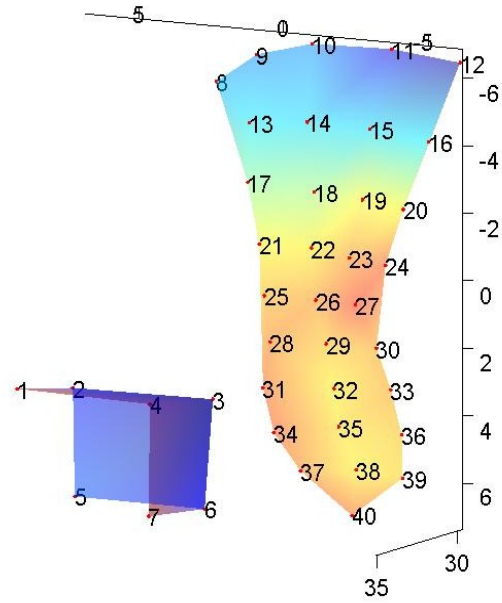
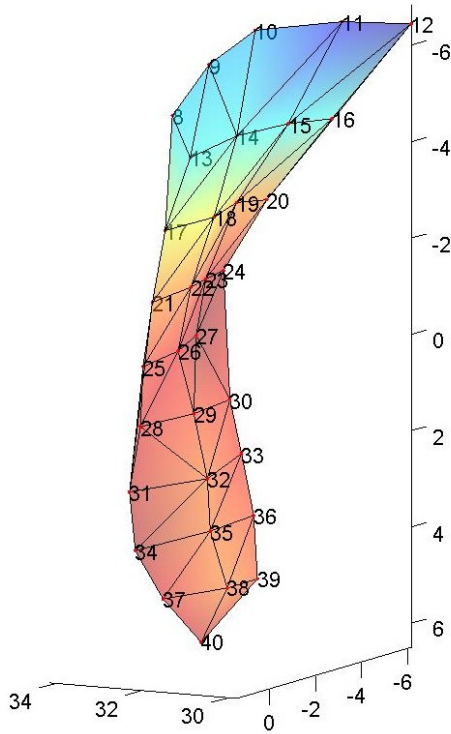
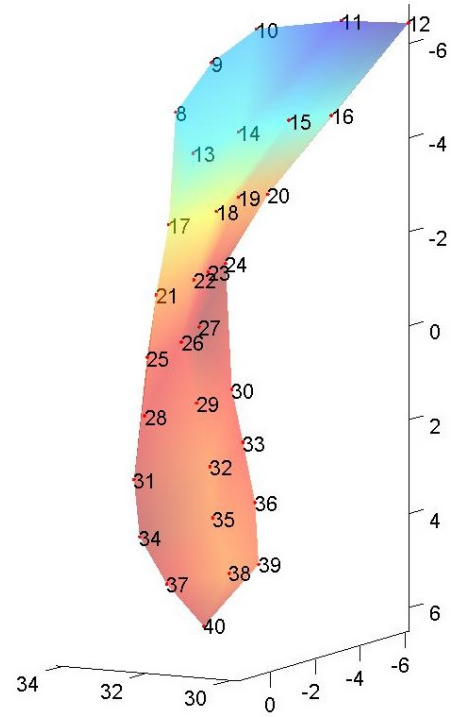


Fig. 7.15 Surfaced Interpolated 3D Points of Actual Foot and Reference Cube





**Fig. 7.16 Triangulated Patches between 3D Points of Actual Foot**



**Fig. 7.17 Surface Interpolated 3D points of Actual Foot**

**Table 7.3 3D Scaled Coordinates (x,y,z) of the Foot**

<b>Point</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>
<b>x</b>	-5.48	-6.37	-6.83	-6.74	-6.44	-4.41	-4.59	-4.55	-4.35	-2.79	-2.72
<b>y</b>	1.44	0.14	-1.75	-4.05	-6.16	0.15	-1.84	-3.98	-6.06	-0.24	-2.60
<b>z</b>	31.26	31.10	31.04	30.24	29.77	31.53	31.51	31.46	31.56	32.32	32.46
<b>Point</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>
<b>x</b>	-2.72	-2.56	-1.12	-1.18	-1.08	-1.03	0.30	0.25	0.19	1.61	1.57
<b>y</b>	-4.57	-6.03	-1.04	-2.90	-4.50	-5.95	-1.47	-3.36	-4.98	-1.81	-3.58
<b>z</b>	32.98	33.05	33.05	33.15	33.66	34.02	33.49	33.69	34.13	33.74	33.44
<b>Point</b>	<b>30</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>	<b>37</b>	<b>38</b>	<b>39</b>	<b>40</b>
<b>x</b>	1.53	2.96	2.92	2.73	4.24	4.03	4.04	5.34	5.28	5.35	6.50
<b>y</b>	-5.39	-1.65	-3.72	-5.88	-2.08	-3.89	-6.19	-2.89	-4.44	-6.18	-4.71
<b>z</b>	33.57	33.91	33.20	33.56	34.00	33.22	33.44	33.79	33.13	33.35	33.88

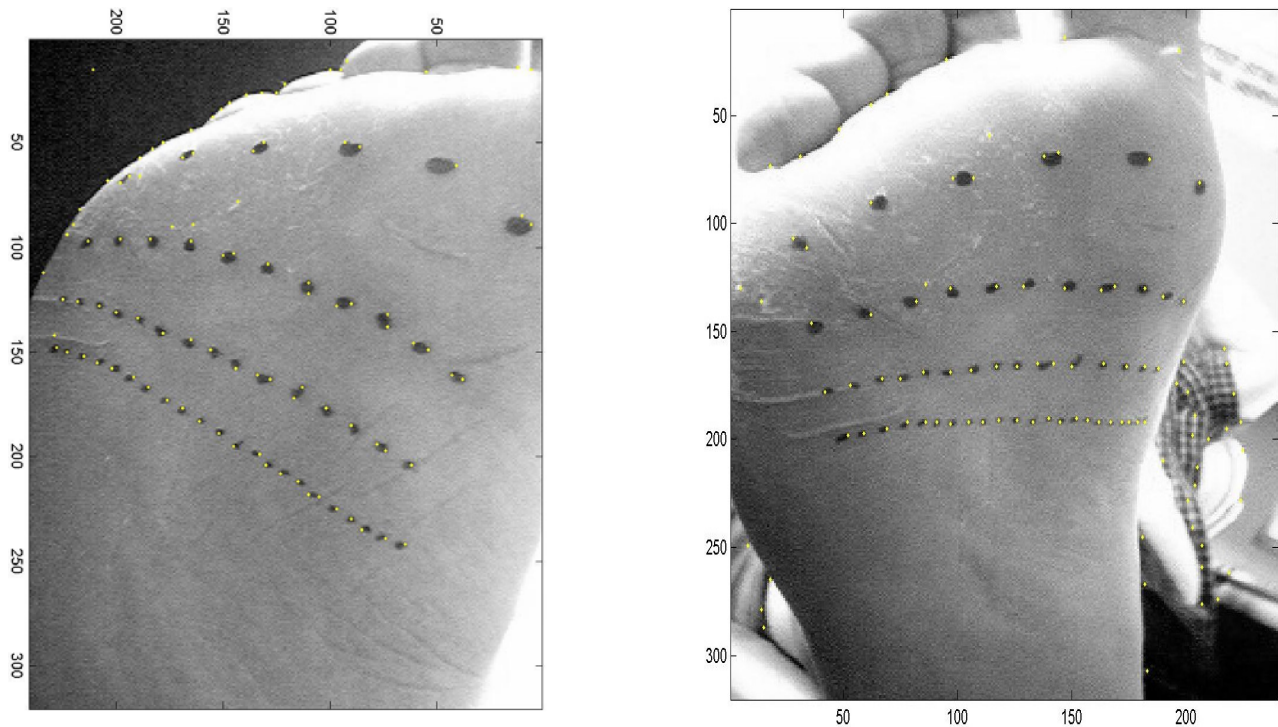
### 7.6.5 Reconstruction of Varied Point Distances on Foot

A test run was performed to reconstruct 3D points with varied distances marked on foot. This is implemented to ensure the required point resolution in detecting, feature matching and reconstructing the 3D points. The approximate diameter and the distance between the points are given in Table. 8.3. All values are within  $\pm 0.5$  mm.

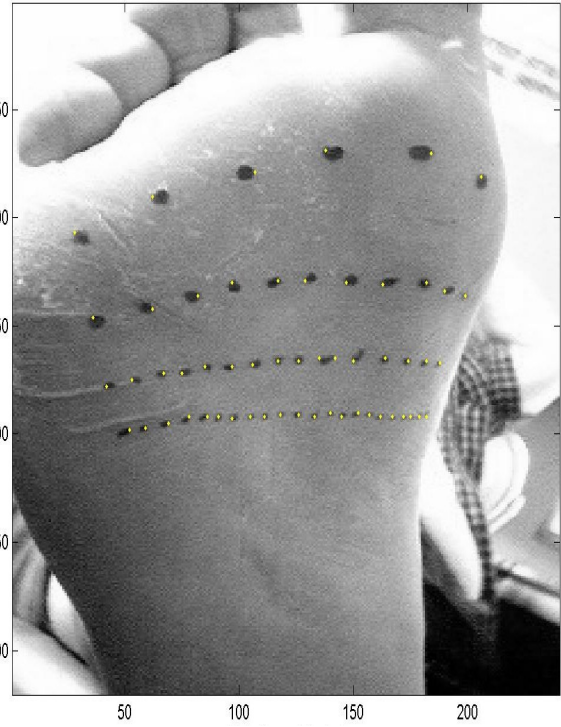
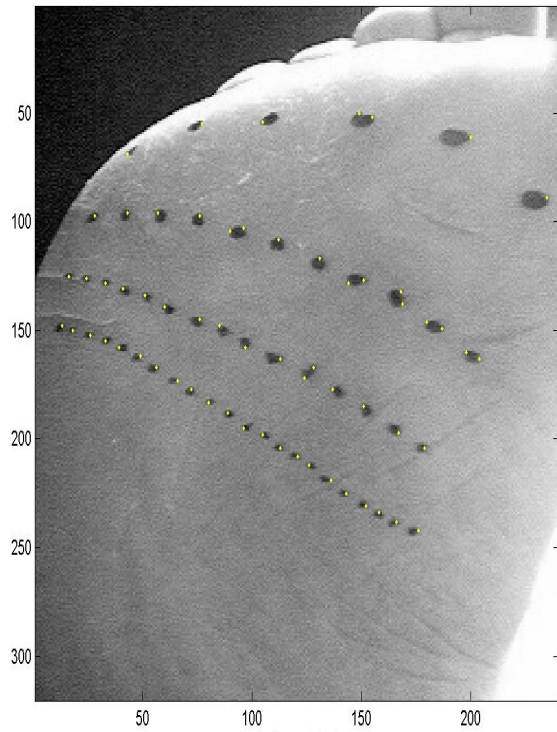
**Table 7.4. Varied Point Sizes and Distances Marked on Foot**

Dotted Line	Distance (mm)	Diameter(mm)	No. of dots
1	15	3	6
2	6	1.5	11
3	3	1	15
4	2	<1	22

The results are shown from Fig. 7.20 to Fig. 7.23.



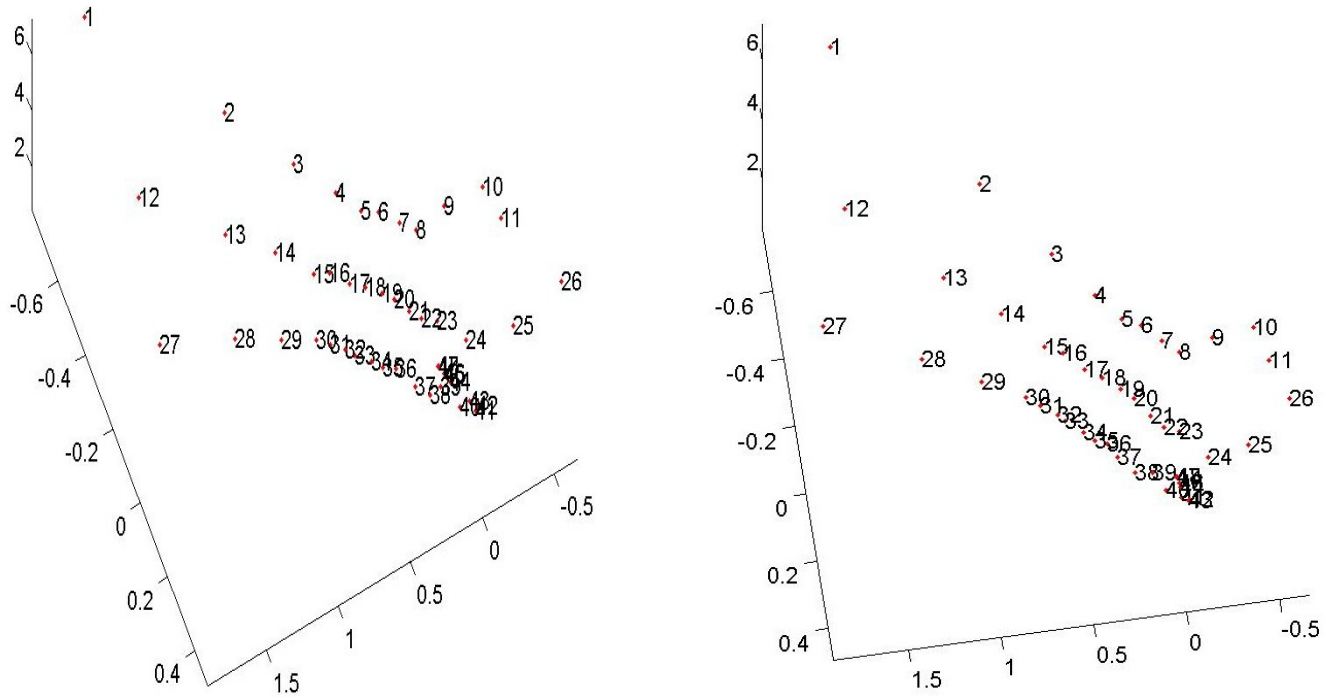
**Fig. 7.18 Feature Detection of Marked Points using Harris Detector Algorithm**



**Fig. 7.19 Feature Detection of Marked Points using Combined Algorithm**



**Fig. 7.20 Feature Matching of Marked Points using Homography Matrix**



**Fig. 7.21 Two Views of the Reconstructed 3D Points**

**Table 7.5 Unscaled (x,y,z) Coordinates of 48 Points**

<b>Point</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
<b>x</b>	-0.79	-0.53	-0.39	-0.31	-0.25	-0.23	-0.18	-0.14	-0.12	-0.06	0.02	-0.33
<b>y</b>	1.54	0.81	0.47	0.25	0.13	0.03	-0.07	-0.15	-0.32	-0.53	-0.58	1.60
<b>z</b>	6.17	3.89	2.80	2.16	1.91	1.89	1.79	1.81	2.44	3.28	3.00	5.95
<b>Point</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>
<b>x</b>	-0.23	-0.18	-0.12	-0.11	-0.07	-0.05	-0.03	0.00	0.02	0.03	0.08	0.15
<b>y</b>	1.09	0.80	0.59	0.49	0.38	0.29	0.20	0.14	0.06	-0.01	-0.08	-0.21
<b>z</b>	4.37	3.47	2.82	2.70	2.45	2.36	2.19	2.07	1.65	1.39	1.71	1.58
<b>Point</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>
<b>x</b>	0.32	0.49	0.06	0.05	0.05	0.05	0.08	0.10	0.14	0.17	0.20	0.24
<b>y</b>	-0.38	-0.55	1.83	1.29	0.97	0.74	0.67	0.58	0.55	0.46	0.41	0.35
<b>z</b>	3.74	7.12	6.62	4.98	3.97	3.26	3.25	3.10	3.30	3.12	3.19	3.42
<b>Point</b>	<b>37</b>	<b>38</b>	<b>39</b>	<b>40</b>	<b>41</b>	<b>42</b>	<b>43</b>	<b>44</b>	<b>45</b>	<b>46</b>	<b>47</b>	<b>48</b>
<b>x</b>	0.36	0.36	0.33	0.43	0.47	0.42	0.34	0.21	0.16	0.15	0.11	0.11
<b>y</b>	0.33	0.24	0.13	0.10	0.02	-0.03	-0.05	-0.05	-0.06	-0.06	-0.06	-0.06
<b>z</b>	4.32	3.85	3.33	3.91	3.96	3.36	2.43	1.31	0.94	0.80	0.56	0.53

It is noticed that all points are detected for varying distances and size in Fig. 7.20. All extraneous points are eliminated using combined algorithm as seen in Fig. 7.21. Feature matching is done using homography matrix (Fig. 7.22) and 3D reconstruction of all points is performed as shown in Fig. 7.23. It is noted from Table 8.5 that the points are not scaled.

## 7.7 Conclusions

This chapter takes an overview of 3D reconstruction of a foot from a pair of images. Steps included, determining the relative pose (extrinsic parameters) between the cameras using eight-point algorithm, and triangulating the 3D point in space to obtain a complete reconstruction of the model, using linear triangulation method. A reference cube was placed besides every image pair considered for reconstruction. This assisted in evaluating the necessary scale used to reconstruct all objects including foot. A plaster foot was first used to reconstruct all points marked on it. Next, 3D reconstruction of an actual foot was accomplished from a stereo image pair. All output geometry included a 3D point cloud, triangulated patches from 3D points, surface interpolated data from 3D points and the 3D coordinates of the points.

## Chapter 8

### Conclusions and Future Work

#### 8.1 Summary and Conclusions

Orthopedic footwear plays an important role as a treatment and preventive measure against foot conditions associated with diabetes. Through the use of customized shoe insoles, a podiatrist can provide a means to better distribute the pressure around the foot, and can also correct the biomechanics of the foot. The work outlined in this thesis details a method of capturing 3D foot geometry by utilizing a pair of images, results in the entire project being inexpensive and more suited to the needs of the podiatrist, than commercially available scanners today.

The current work has reviewed different algorithms, in different phases of reconstruction process. Careful consideration was given as to how each one of them would be effective and the best algorithm(s) were chosen to achieve the objective.

The work presented here is a novel technique to capture 3D foot geometry from images. However there are several challenges faced to make this project commercial. Among them are a more robust method to calibrate the camera, better feature detection methods to capture the marked points on foot and a standardized method to scale the points by calibrating the distances and the size of the reference object. A statistical error analysis needs to be performed to get a better picture of all algorithms used, with respect to their stability and accuracy.

#### 8.2 Future Work

The current work of reconstructing the 3D foot geometry was restricted to a pair of images, for simplicity. A more robust and accurate scheme would be to involve five cameras in the process of image capture. This assists in determining all parameters, fundamental matrix and the relative pose between the cameras more accurately than in a stereo image pair. A more robust

technique to mark points on foot would be to project a laser/color light grid on the foot. It could be an  $n \times n$  grid of squares which encompasses the plantar surface of the foot. This increases the percentage of detected features and gives more points to reconstruct the 3D geometry of foot.

The 3D point coordinates of the foot obtained using MATLAB, could be converted to a different format, such as VRML (Virtual Reality Modeling Language) or STL (Stereo Lithography). These formats are recognized by commercial softwares, which can be used to create models either from a 3D printer or a CNC machine. This could complete the cycle of creating a template of the shoe insert. Finally, the entire process can be streamlined by using standard digital high resolution cameras, obtaining a quick sequence of five images of foot and increasing the process of point detection and matching by implementing it in a more controlled environment.

## Bibliography

- [ADA2002] Fact Sheet, American Diabetes Association, 2002.
- [BAJOIE2003] Dwayne Bajoie, Alp Eminsel, Pablo Guardado, *3D Foot Scanner Utility Frame*, Senior Design Project, Department of Mechanical Engineering, Louisiana State University, 2003.
- [BIRCHFIELD1988] S. Birchfield, *An Introduction to Projective Geometry*, Technical Report, Stanford University, March 1998.
- [BLOOMENTHAL1997] Jules Bloomenthal, Chandrajit Bajaj, *Introduction to Implicit Surfaces (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*, ISBN: 155860233X, 1997.
- [CALTECH1999] Computer Vision Research Group, *Camera Calibration Toolbox for Matlab*, Department of Electrical Engineering, California Institute of Technology, 1999.
- [FAUGERAS1993] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, ISBN: 0262061589, 1993
- [FOOTCARE2001] Footcare Express Scanner, *Footcare Express Inc.*, Miami, FL, USA, 2001.
- [GOLUB1983] G. Golub, C. V. Loan, *Matrix Computations*. North Oxford Academic, ISBN: 0801854148, 1983.
- [GORDOIS2003] Gordoio, P. Scuffham, A. Sheare, A. Oglesby & J. A. Tobian, *The Health Care Costs of Diabetic Peripheral Neuropathy in the U.S.A.* Journal of Diabetes Care, Volume 26, Number 6, June 2003.
- [HARRIS1988] C Harris & M Stephens, *A Combined Corner & Edge Detector*, Proceedings, 4<sup>th</sup> Alvey Vision Conference, 1988.
- [HARTLEY1995] Richard I. Hartley, *A linear method for reconstruction from lines and points*, Proc. International Conference on Computer Vision, pages 882-887, 1995.



- [HARTLEY1997a] Richard I. Hartley, *In Defense of the Eight-Point Algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No.6, June 1997.
- [HARTLEY1997b] Richard I. Hartley, P.Sturm. *Triangulation*, Computer Vision and Image Understanding, 68, November 1997.
- [HARTLEY2004] Richard I. Hartley, Andrew Zisserman, *Multiple View Geometry in computer vision*, Cambridge University Press, ISBN: 0521540518, 2004.
- [HEYDEN1995] A. Heyden. *Reconstruction from image sequences by means of relative depths*, Proc. 5<sup>th</sup> International Conference on Computer Vision, Boston, Cambridge, MA, June 1995.
- [HILDRETH1985] Ellen C. Hildreth, *Edge Detection*, Technical Report, Artificial Intelligence Lab, MIT, September 1985.
- [HUANG1989] T.S. Huang, O.D. Faugeras. *Some properties of the E-matrix in two-view motion estimation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 11, 1989.
- [KITCHEN1982] L. Kitchen, A. Rosenfield, *Grey Level Corner Detection*, Pattern Recognition Letters, 1982
- [KOUCHI2001] M. Kouchi, M. Mochimaru, *Development of a Low Cost Foot Scanner for a Custom Shoe Making System*. Proc. of the 5<sup>th</sup> Symp. On Footwear Biomechanics, Zurich, Switzerland, 2001.
- [L-HIGGINS1981] H.C. Longuet-Higgins, *A Computer Algorithm for Reconstructing a Scene From Two Projections*, Nature, vol. 293, pp. 133–135, Sept 1981.
- [LUONG1996] Q.T. Luong, O. Faugeras, *The Fundamental matrix: theory, algorithms, and stability analysis*. The International Journal of Computer Vision, 1(17):43–76, 1996.

- [MA2001] Ma, Soatto, Kosecka, Sastry. *An Invitation to Geometric 3-D Vision, From Images to Geometric Models*. Springer, ISBN: 0387008934, 2001
- [MATHWORKS2004] Matlab 7.0, *R14*, Mathworks Inc., Boston, MA, 2004
- [MAYBANK1993] S.J. Maybank, *Theory of Reconstruction from Image Motions*, Springer-Verlag, Berlin, ISBN: 0387555374, 1993.
- [MOHR1996] R. Mohr, B. Triggs, *Projective Geometry for Image Analysis*, International Symposium of Photogrammetry and Remote Sensing, Vienna, July 1996.
- [PILU1997] M. Pilu, *A Direct Method for Stereo Correspondence based on Singular Value Decomposition*, IEEE International Conference of Computer Vision and Pattern Recognition, 1997.
- [POLHEMUS2001] FastSCAN Cobra, 6 DOF hand held scanner, Polhemus, Colchester, VT 05446, USA, 2001.
- [POLLARD1985] S. Pollard, J. Mayhew, J. Frisby, *PMF: A stereo correspondence algorithm using a disparity gradient limit*, Perception, (14), 1985
- [PRESS1992] William H. Press, Brian P. Flannery, Saul A. Teukolsky, William Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, ISBN: 0521431085, 1992.
- [SINCLAIR1992] D. A. Sinclair, *Experiments in Motion and Correspondence*. PhD thesis, University of Oxford, 1992.
- [SCOTT1991] G. Scott, H. Longuet-Higgins, *An algorithm for associating the feature of two patterns* Proc. Royal Society of London, 1991.
- [SEMPLE1979] J.G. Semple, G.T Kneebone, *Algebraic Projective Geometry*, Oxford University Press, ISBN: 0198503636, 1979.

- [SGARLATTO1972] Sgarlatto T, *A Compendium of Podiatric Biomechanics*. Board of Trustees of the California College of Podiatric Medicine Corporation, California College of Podiatric Medicine, San Francisco, 1972.
- [SHAPEGRABBER2002] Foot Scanner, *Shape Grabber Inc.*, Ontario, Canada, 2002.
- [SHAPIRO1992] Larry S Shapiro, Michael Brady *Feature-based correspondence: an Eigen vector approach*, *Image & Vision Computing*, Volume 10, Issue 5, June 1992.
- [VALMASSY1996] Ronald L. Valmassy, *Clinical Biomechanics of the Lower Extremities*, Mosby-Year Book, 1<sup>st</sup> Edition, ISBN: 0801679869, 1996.
- [WU1990] Wu KK, *Foot Orthoses: Principles and Clinical Applications*, Baltimore, Williams & Wilkins, pp 49-142, 1990.
- [XU1996] G. Xu, Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition*, Kluwer Academic Publishers, 1996.
- [ZHANG1994] Zhengyou Zhang, Rachid Deriche, Oliver Faugeras, Quan-Tuan Luong , *A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry*, *Artificial Intelligence Journal*, 1994.
- [ZHANG1998] Z. Zhang, *Determining the epipolar geometry and its uncertainty – a review*, *International Journal of Computer Vision*, 27, March 1998.
- [ZHANG2001] Zhengyou Zhang, Charles Loop, *Estimating Fundamental Matrix by Transforming Image Points in Projective Space*, *Computer Vision and Image Understanding*, Vol. 82, No. 2, May 2001.

## Vita

Rajeev Madazhy hails from one of world's top ten technology hubs, Bangalore, India. He did his schooling at Poorna Prajna High School in Bangalore. He received his Bachelor of Engineering degree in mechanical engineering from Bangalore University, India, in 2000. He received 10<sup>th</sup> rank in the university and graduated with academic excellence. After his graduation, he worked as systems engineer in Wipro Technologies, a prominent software services industry (Listed in NYSE as WIT) from August 2000 to December 2001. He then joined the graduate program at Louisiana State University, Baton Rouge (LSU), in January 2002. He worked on different aspects of 3D reconstruction of foot from a pair of images, which was part of his thesis research. He is a candidate for the degree of Master of Science in mechanical engineering to be awarded at the commencement of December, 2004.