



3D FACE RECONSTRUCTION AND ANIMATION

by

MOSTAFA ABOBAKR ABDEL-MAJED

B.Sc. Computer Science, Assiut University, 2006

A THESIS

Submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

Faculty of Computers and Information

UNIVERSITY OF ASSIUT

Assiut, Egypt.

2013

Supervised by:

Prof. Dr. Yousef Bassyouni Mahdy

Dr. Khaled Fathy Hussein

Examined by:

Prof. Dr. Adel Abo El-Magd Sewesy

Prof. Dr. Usama Sayed Mohammed

Prof. Dr. Yousef Bassyouni Mahdy

Dr. Khaled Fathy Hussein

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

IN THE NAME OF ALLAH, MOST GRACIOUS, MOST MERCIFUL.

Abstract

Shape reconstruction using structured lighting is considered one of the most reliable techniques for recovering object surface. Coded structured light is a technique based on the projection of light patterns on the measured surface. The projected patterns are coded in order to solve the correspondence problem. Each token of light is projected on the scene carrying a label which indicates where it comes from (i.e. this token represents projector image column, row or pixel). When the token is imaged by the camera, this label is read and decoded to obtain an unique match.

Using a calibrated projector-camera pair, a coded pattern is projected on the object and imaged by the camera. Since the pattern is coded, correspondences between image points in camera frame and points of the projected pattern in the projector frame can be easily found. These correspondences are used to triangulate and extract 3D information of the surface.

In this thesis, 3D human faces are reconstructed using structured lighting. Also, a novel projector calibration method is presented. It is based on passive stereo and triangulation. After face reconstruction, Optitrack facial capturing system is used to record facial expression of a real actor. The captured data is used to animate the reconstructed faces.

Acknowledgements

First of all, Praise be to *Allah*, the Cherisher and Sustainer of the worlds, Most Gracious, Most Merciful, All-Sustainer, Most Beneficent. The Just judge, All-Knower, the Living, the Self-subsisting, Eternal. There is no god but He. To Him is the ultimate return.

I am much indebted to *Prof.Dr. Yousef Bassyouni Mahdy* (Professor of Computers, Dean of Faculty of Computers and Information, Assiut University) for giving me the chance to be one of his graduate students. I would like to thank him for his insights and peculiar remarks all through our work. I would like to express my deepest thanks to him for his kind guidance and continuous observation. Without his help this thesis would not have been what it is.

I would like to express my deepest gratitude to *Dr. Khaled Fatehy*, without his patience, close guidance, and continual support this thesis would not have been possible. Meeting *Dr. Khaled* at the start of my postgraduate study was a turning point in my scientific career, he played an instrumental role in the formation and evolution of my scientific personality. Hours of scientific conversation with him in the course of this thesis were real pleasure.

I would also like to thank my colleague post-graduate students at the faculty of computers and information, the joyful scientific climate they formed helped me go through this period. Our conversations have always gave me different viewpoints on the topic and enriched my understanding.

Last, but by no means least, words are useless in expressing my gratitude to my parents, to whom i simply owe everything.

Mostafa Abobakr

2013

To my Grandfather Abdel-Majed Salem who I love very much.

Contents

	Page
Abstract	iii
Acknowledgements	iv
List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Introduction	1
1.2 Objective of the Thesis	3
1.3 Organization of the Thesis	3
2 Related Works	5
2.1 Introduction	5
2.2 3D Scanning Technology	5
2.2.1 Passive Methods	7
2.2.2 Active Methods	8
2.3 Structured Lighting	10
2.3.1 A Classification of Coding Strategies	10
2.3.1.1 Time-Multiplexing Strategy	14
2.3.1.2 Spatial Neighbourhood	16
2.3.1.3 Direct Codification	16

2.4	Calibration	17
2.4.1	Perspective Projection and Pinhole Model	17
2.4.1.1	The Ideal Pinhole Camera	18
2.4.1.2	The General Pinhole Camera	18
2.4.2	Camera Calibration	21
2.4.3	Projector Calibration	22
2.5	Facial Animation Techniques	24
2.5.1	Blend Shapes or Shape Interpolation	25
2.5.2	Parameterizations	26
2.5.3	Facial Action Coding System	26
2.5.4	Deformation Based Approaches	27
2.5.4.1	Free From Deformation	27
2.5.5	Physics Based Muscle Modelling	28
2.5.6	Performance Driven Facial Animation	28
2.5.6.1	Facial Motion Capture System	29
3	Building a 3D Scanner Using a Novel Projector Calibration Method	31
3.1	Introduction	31
3.2	Data Capture	32
3.2.1	Scanner Hardware	32
3.2.2	Structured Light Sequences	33
3.3	Decoding of Structured Light Sequences	37
3.4	Projector Calibration	38
3.4.1	System Overview	40
3.4.2	Correspondences' Matching	40
3.4.2.1	Pattern Displaying and Capturing	40
3.4.2.2	Corners' Extraction	43
3.4.3	Reconstruction by Triangulation	43

3.4.3.1	The Mathematics of Triangulation	45
3.4.4	Estimating Calibration Parameters	46
3.4.4.1	Definition of The Intrinsic Parameters	46
3.5	Camera Calibration	52
3.6	3D Point Cloud Reconstruction	53
3.7	Results and Performance	60
3.7.1	Projector Calibration Results	60
3.7.2	3D Scanner Results	63
4	Face Animation Using Facial Motion Capture System	70
4.1	Introduction	70
4.2	System Setup	71
4.2.1	System Overview	71
4.2.2	System Calibration	71
4.3	Facial Expression Capturing	73
4.3.1	Apply Face Markers	73
4.3.2	Facial Capture Wizard	73
4.4	Animation Results	77
5	Conclusions and Future Works	80
5.1	Conclusions	80
5.2	Future Works	81
	References	82
	Arabic Summary	

List of Tables

Table	Page
2.1 A Classification of Pattern Projection Techniques	13
2.2 The Sample Single Action Units	26
2.3 Example Sets of Action Units for Basic Expressions	27
3.1 The Standard Deviation of Error (StdDev) of The Coordinates of The Points	61

List of Figures

Figure	Page
2.1 Contact 3D Scanners Example:Coordinate Measuring Machine	6
2.2 Passive Methods for 3D Scanning	7
2.3 Active Methods for 3D Scanning	8
2.4 Non-Contact Active 3D Scanners	11
2.5 The Problem of Coded Structured Light with a Projector-Camera Pair . . .	12
2.6 The Temporary Codification Proposed by Posdamer and Altschuler	15
2.7 Perspective Projection Under The Pinhole Model	19
2.8 Linear Interpolation on Blend Shapes	25
2.9 Free Form Deformation	28
2.10 OptiTrack Facial Capture System	30
3.1 Structured Light for 3D Scanning	33
3.2 Structured Light Illumination Sequences	34
3.3 Comparison of Binary and Gray Code Structured Light Sequences	36
3.4 Decoding Structured Light Illumination Sequences	37
3.5 The Proposed System Diagram	39
3.6 Overview of The Proposed Method	41
3.7 System Components	42
3.8 Checkerboard Pattern in Projector Frame and The Extracted Corners	44

3.9	The 3D Points of The Checkerboard Pattern Reconstructed by Triangulation and The 2D Points of The Checkerboard Pattern in Projector Frame . . .	47
3.10	Camera Calibration Sequence Containing Multiple Views of a Checkerboard at Different Positions and Orientations	53
3.11	Camera Calibration Results	54
3.12	The World Coordinate System	55
3.13	Reconstruction by Triangulation	56
3.14	Reprojection Error of The Proposed Method and Sergio Fernandez et al. Method	62
3.15	3D Reconstruction of an LCD Screen	64
3.16	3D Reconstruction of a Flower Vase	65
3.17	3D Reconstruction of a Camera Lamp	66
3.18	Reconstructed Faces in The Form of 3D Cloud of Points (The first person) .	67
3.19	Reconstructed Face in The Form of 3D Cloud of Points (The second person)	68
3.20	Reconstructed Face in The Form of Surfaces	69
4.1	OptiTrack Expression Capturing System	71
4.2	Our System Overview	72
4.3	Calibration Types	73
4.4	System Calibration Steps	74
4.5	Setting Ground Plane and The Captured Volume	75
4.6	The Marker Placement Diagram	76
4.7	The Face Template	77
4.8	A Take Captured by The A/V Camera	78
4.9	The FBX Data File Imported in 3Ds Max	78
4.10	The Reconstructed Face After Animation	79

Chapter 1

Introduction

1.1 Introduction

The human visual ability to perceive depth looks like a puzzle. Humans perceive three-dimensional spatial information quickly and efficiently by using binocular stereopsis of eyes. A lot of research is involved in robot vision in order to obtain 3D information of the surrounding scene. Most of this research is based on modelling the stereopsis of humans by using two cameras as if they were two eyes. This method is known as stereo vision or passive stereo.

The stereo vision principle is based on obtaining the three dimensional position of an object point from the position of its projective points in both camera image planes. However, before evaluating 3D information, the mathematical models of both cameras have to be known. This step is called camera calibration and will be described in chapter 2. The most important problem in stereo vision is the determination of the pair of matching points in the two images (i.e the two pixels related to the same 3D world point), known as the correspondence problem, and it is also one of the most difficult problems to be solved.

Active stereo is a technique used for solving the correspondence problem found in passive stereo. One of the cameras used in passive stereo is replaced by an illumination

source such as data projector. Active optical devices are based on an illumination source, which produces some sort of structured illumination on the object to be scanned, and a sensor, which is typically a camera. The camera acquires images of the distorted pattern reflected by the object surface. In most cases the depth information is reconstructed by triangulation, given the known relative positions of the projector-camera pair.

System calibration is a key step in any 3D object reconstruction. Calibration means estimating the intrinsic parameters which model the optical characteristics and the internal geometry of the sensor, and the extrinsic parameters which model the position and orientation of the sensor with respect to a world co-ordinate system. Camera calibration methods have been extensively studied over the years [1, 2]. The calibration of a projector is more complicated than that of a camera because projectors cannot image the surface that they illuminate so that the correspondence between the 2D projected points and the 3D illuminated points cannot be made without the use of a camera. Also it is difficult to retrieve the co-ordinates of the 3D points because the calibrating pattern is projected and not attached to the world coordinate frame in general.

In this thesis, a novel projector calibration method is presented. It is based on passive stereo and triangulation. The key point of the proposed calibration method is to consider the projector as an inverse camera (mapping intensities of a 2D image into 3D rays) thus making the calibration of a projector the same as that of a camera. In this way, any standard calibration procedure for cameras can be used in order to calibrate the projector. So, the main concern of our method is to find the 3D points of the projected pattern in order to use them together with the 2D points of the projected image to finally obtain the intrinsic and extrinsic parameters of the projector.

1.2 Objective of the Thesis

The interest of this thesis is focused on structured light which has been considered as one of the most frequently used techniques in order to reduce the problems related to stereo vision. The deformation between the pattern projected into the scene and the one captured by the camera, permits to obtain three dimensional information of the illuminated scene. This technique has been widely used in many applications as: 3D object reconstruction, robot navigation, and quality control. Although, the projection of regular patterns solve the problem of points without matching, it does not solve the problem of multiple matching. To solve this problem, another structured light technique is used. This technique is based on the codification of the light projected on the scene in order to be used as a tool to obtain a unique match. Each token of light projected on the scene carries a label indicating from which position in the projected frame it comes. When the token is imaged by the camera, the label is read (decode the pattern) in order to solve the correspondence problem.

The objectives of this thesis is to reconstruct 3D human face using active optical scanner. Furthermore, a novel projector calibration method is presented. After the face reconstruction step, 3D motion capture data is used to animate the reconstructed faces. Motion capture is a powerful aid in computer animation and a supplement to the traditional key-frame animation. In motion capture system, the animator does not need to record the motion of every point on the actor, which will result in a huge computation pressure, but some key points that can represent the movement as a whole. The key points are mostly on the muscles on face, which can determine other points' locations in the movement.

1.3 Organization of the Thesis

The thesis consists of five chapters including this introductory chapter. The rest of this thesis is organized as follows. Chapter 2, Related Works, contains a survey of the 3D scanning

technology (passive methods and active methods), coded structured light, camera-projector calibration, and facial animation techniques. Chapter 3, Building a 3D Scanner Using a Novel Projector Calibration Method, discusses how to build a 3D scanner using a novel method for projector calibration. This method is based on passive stereo and triangulation. Chapter 4, Face Animation Using Facial Motion Capture System, discusses how to animate the reconstructed faces using facial motion capture data. Chapter 5, Conclusions and Future Works, concludes the presented work and discusses the future works followed by the references used in this thesis.

Chapter 2

Related Works

2.1 Introduction

This chapter discusses important topics related to 3D face reconstruction and animation. It consists of five sections including this introductory section. The rest of the chapter is organized as follows. Section 2.2, 3D Scanning Technology, discusses the 3D scanners types and benefits. Section 2.3, Structured Lighting, discusses how to use the structured light techniques for recovering the surface of objects. Section 2.4, Calibration, discusses the perspective Projection, the pinhole model, and camera-projector calibration methods. Finally, section 2.5 discusses facial modelling and animation techniques.

2.2 3D Scanning Technology

Three dimensional measurement is an important topic in computer vision, having different applications such as range sensor, industrial inspection of manufactured parts, object recognition, 3D map building, biometrics, clothing design, and others. The developed solutions are traditionally categorized into contact and non contact techniques [3]. Contact measurement techniques have been used for a long time in reverse engineering and indus-

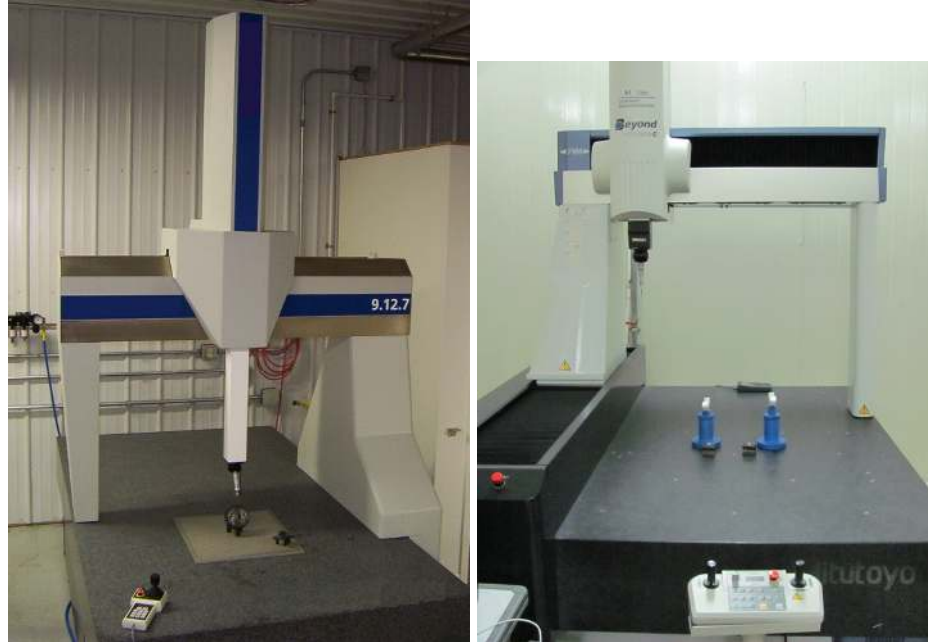


Figure 2.1: *Contact 3D Scanners Example: Coordinate Measuring Machine [6]*

trial inspections. Its working principle is based on a calibrated sensor that passes through the object and records the displacement caused at every position in the 3D space. Based on this, shape reconstruction can be done. Figure 2.1 shows a contact 3D scanner example. A CMM (coordinate measuring machine) is an example of a contact 3D scanner. It is used mostly in manufacturing and can be very precise. The disadvantage of CMMs is that it requires contact with the object being scanned. So, scanning the object might modify or damage it. This fact is very significant when scanning delicate or valuable objects such as historical artifacts. The other disadvantage of CMMs is that they are relatively slow compared to the other scanning methods. Physically moving the arm that the probe is mounted on can be very slow and the fastest CMMs can only operate on a few hundred hertz. In contrast, an optical system like a laser scanner can operate from 10 to 500 kHz. The main problems of contact techniques are its slow performance and high cost price [4], as well as the necessity of touching the object, which is not feasible for all applications. Non contact techniques were developed to cope with this problem, and have been studied widely. These techniques can be both active and passive [5].

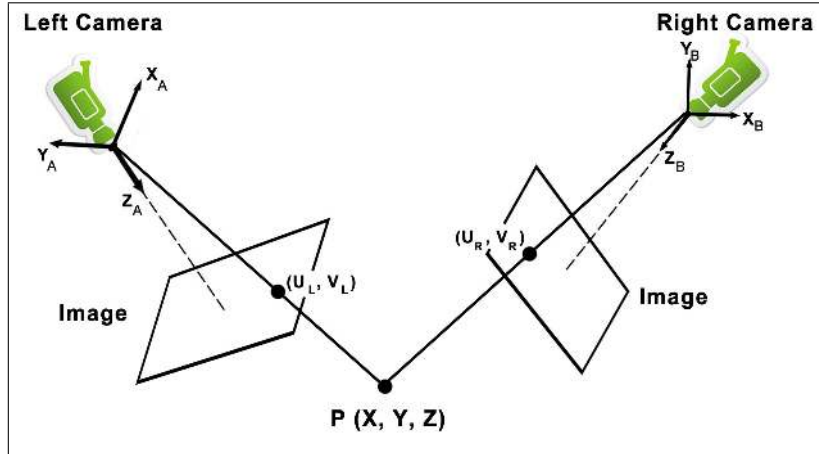


Figure 2.2: *Passive Methods for 3D Scanning.*

2.2.1 Passive Methods

Non-contact optical scanners can be categorized by the degree to which controlled illumination is required [5]. Passive scanners do not require direct control of any illumination source, instead relying completely on ambient light. Stereoscopic imaging is one of the most widely used passive 3D imaging systems, both in biology and engineering [7]. Mirroring the human visual system, stereoscopy estimates the position of a 3D scene point by triangulation [8]. First, the 2D projection of a given point is identified in each camera. Using known calibration objects, the imaging properties of each camera are estimated, allowing a single 3D line to be drawn from each camera's center of projection through the 3D point as shown in Fig(2.2). The intersection of these two lines is then used to recover the depth of the point.

Trinocular [9] and multi-view stereo [10] systems have been introduced to improve the accuracy and reliability of conventional stereoscopic systems. However, all such passive triangulation methods require correspondences to be found among the various viewpoints. Even for stereo vision, the development of matching algorithms remains an open and challenging problem in the field [11]. Flat or periodic textures prevent robust matching. While machine learning methods and prior knowledge are being presented to solve such prob-

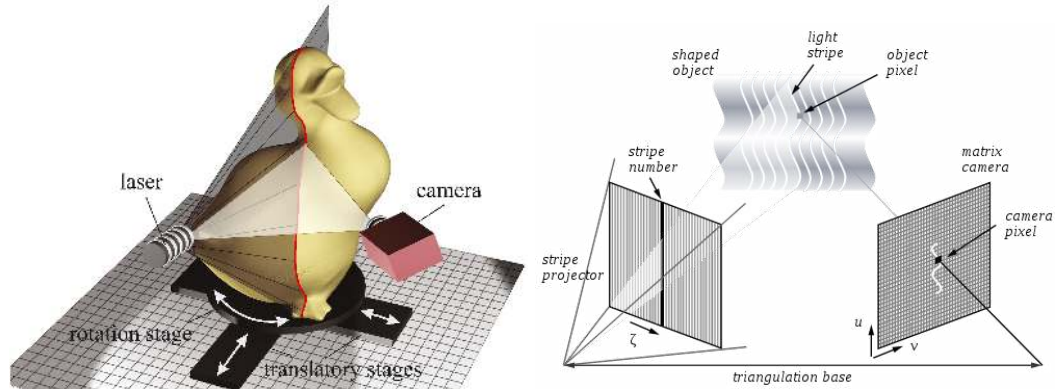


Figure 2.3: *Active Methods for 3D Scanning* [6].

lems, multi-view 3D scanning remains somewhat outside the domain of those concerned with accurate, reliable 3D measurement.

2.2.2 Active Methods

Active optical scanners use controlled illumination to overcome the correspondence problem. In comparison to non-contact and passive methods, active illumination is often more sensitive to surface material properties. Many active systems attempt to solve the correspondence problem by replacing one of the cameras, in a passive stereoscopic system, with a controllable illumination source as shown in Fig(2.3) [12].

During the 1970s, single-point laser scanning emerged. In this scheme, a series of fixed and rotating mirrors are used to raster scan a single laser spot across a surface. A digital camera records the motion of this "flying spot". The 2D projection of the spot defines, with appropriate calibration knowledge, a line connecting the spot and the camera's center of projection. The depth is recovered by intersecting this line with the line passing from the laser source to the spot, given by the known deflection of the mirrors. As a result, such single-point scanners can be seen as the optical equivalent of coordinate measuring machines [12].

As with CMMs, single-point scanning is a very slow process. With the development of low-cost, high-quality cameras in the 1980s, slit scanners emerged as a powerful alternative. In this design, a laser projector creates a single planar sheet of light. This "slit" is then mechanically swept across the surface. As before, the known deflection of the laser source defines a 3D plane. The depth is recovered by the intersection of this plane with the set of lines passing through the 3D stripe on the surface and the camera's center of projection [12].

A digital "structured light" projector can be used to eliminate the mechanical motion required to translate the laser stripe across the surface [13, 14, 15]. The projector could be used to display a single column (or row) of white pixels translating against a black background to replicate the performance of a slit scanner. It is capable of displaying 24-bit color images. Structured lighting sequences have been developed which allow the projector-camera correspondences to be assigned in relatively few frames. In general, the identity of each plane can be encoded spatially (i.e., within a single frame) or temporally (i.e., across multiple frames), or with a combination of both spatial and temporal encodings [16]. Both slit and structured lighting scanners are not suitable for scanning dynamic scenes. In addition, due to separation of the light source and camera, certain occluded regions will not be recovered.

Also, time-of-flight 3D laser scanner is an active scanner that uses laser light to get the 3D shape of the object [17]. At the heart of this type of scanner is a time-of-flight laser rangefinder. The laser rangefinder finds the distance of a surface by timing the round-trip time of a pulse of light. A laser is used to emit a pulse of light and the amount of time before the reflected light is seen by a detector is measured. Since the speed of light c is known, the round-trip time determines the travel distance of the light, which is twice the distance between the scanner and the surface. If t is the round-trip time, then distance is equal to $c.t/2$. Several economical time-of-flight depth cameras are now commercially

available, including Canestas CANESTAVISION [18] and 3DV's Z-Cam [19]. Figure 2.4 shows some examples for Non-contact active 3D scanners.

Figure (2.4a) shows a lidar scanner which may be used to scan buildings and rock formations to produce a 3D model. The lidar can aim its laser beam in a wide range: its head rotates horizontally, a mirror flips vertically. The laser beam is used to measure the distance to the first object on its path. Figure (2.4b) shows a hand-held 3D laser scanner which is used for 3D modelling. The scanner has a camera to accurately texture map the object[6].

2.3 Structured Lighting

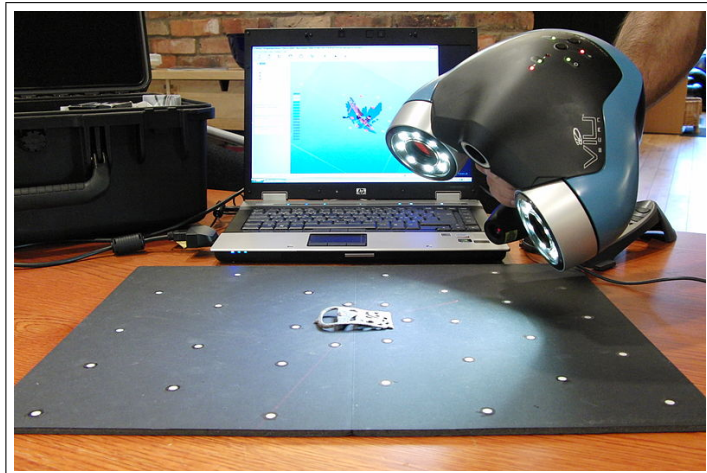
Coded structured light is considered one of the most reliable techniques for recovering the surface of objects [16]. This technique is based on projecting a light pattern and viewing the illuminated scene from one or more points of view. Since the pattern is coded, correspondences between image points and points of the projected pattern can be easily found. The decoded points can be triangulated and 3D information is obtained as shown in Fig (2.5).

2.3.1 A Classification of Coding Strategies

A coded structured light system is based on the projection of a single pattern or a set of patterns onto the measuring scene which is then viewed by a single camera or a set of cameras. The patterns are specially designed so that codewords are assigned to a set of pixels. Every coded pixel has its own codeword, so there is a direct mapping from the codewords to the corresponding coordinates of the pixel in the pattern. The codewords are simply numbers, which are mapped in the pattern by using grey levels, colour or geometrical representations. The larger the number of points that must be coded, the larger the codewords are and, therefore, the mapping of such codewords to a pattern is more difficult.



(a)



(b)

Figure 2.4: *Non-contact Active 3D Scanners* [6].

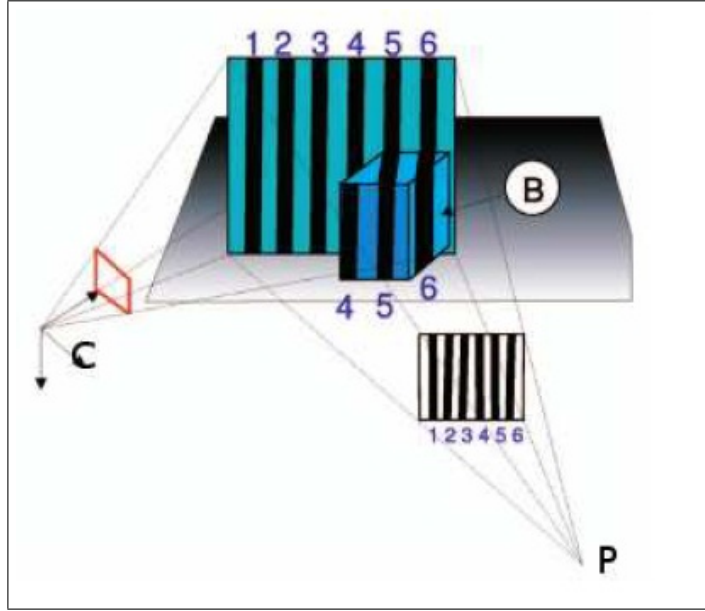
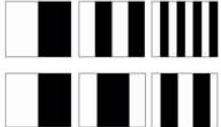
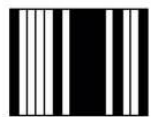

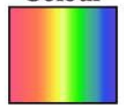


Figure 2.5: *The Problem of Coded Structured Light with a Projector-Camera Pair [20].*

Pattern projection techniques differ in the way in which every point in the pattern is coded and decoded, i.e. what kind of codeword is used, and whether it encodes a single axis or two spatial axis. Only a single axis can be encoded, since a 3D point can be obtained by intersecting two lines (i.e. when both pattern axis are coded) or intersecting one line (the one which contains a pixel of the camera image) with a plane (i.e. when a single pattern axis is coded).

The different coding structured light techniques existing until 1998 are analysed by Salvi et al. [21]. He did a comprehensive work of the different techniques until that date. This work is updated and expanded with the techniques existing by Pages et al. [16]. In his survey, a study of the different pattern codification strategies in structured light systems was performed. He presented a comprehensive table where all techniques were represented, and a classification table of all existing techniques until that date was done. The table classifies existing methods in time multiplexing, spatial neighbourhood and direct coding methods. He also compared some common characteristics like scene applicability (static or moving scenarios), pixel depth (black and white, garyscale or color vocabulary) and coding strategy

Table 2.1: A Classification of Pattern Projection Techniques [16].

Time-multiplexing	Binary codes 	Posdamer et al. Inokuchi et al. Minou et al. Trobina Valkenburg and McIvor Skocaj and Leonardis Rocchini et al.	✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓
	n-ary codes	Caspi et al. Horn and Kiryati	✓ ✓	✓ ✓	✓ ✓
	Gray code + Phase shifting	Bergmann Sansoni et al. Wiora Gühring	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓
	Hybrid methods	Kosuke Sato Hall-Holt and Rusinkiewicz	✓ ✓	✓ ✓	✓ ✓
Spatial Neighborhood	Non-formal codification	Maruyama and Abe Durdle et al. Ito and Ishii Boyer and Kak Chen et al.	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
	De Bruijn sequences 	Hügli and Maitre Monks et al. Vuylsteke and Oosterlinck Salvi et al. Lavoie et al. Zhang et al.	✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓
	M-arrays 	Morita et al. Petriu et al. Kiyasu et al. Spoelder et al. Griffin and Yee Davies and Nixon Morano et al.	✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓
Direct coding	Grey levels	Carrihill and Hummel Chazan and Kiryati Hung	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓
	Colour 	Tajima and Iwakawa Smutny and Pajdla Geng Wust and Capson Tatsuo Sato	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
	Scene applicability	<i>Static</i> <i>Moving</i>			
	Pixel depth	<i>Binary</i> <i>Grey levels</i> <i>Colour</i>			
	Coding strategy	<i>Periodical</i> <i>Absolute</i>			

(periodical or absolute). Table 2.1 shows a classification of pattern projection techniques according to [16]. The following subsections (Time-multiplexing strategy, Spatial neighbourhood and Direct codification) briefly describe the existing methods.

2.3.1.1 Time-Multiplexing Strategy

One of the most commonly used strategies is based on temporal coding. In this case, a set of patterns are successively projected onto the measuring surface. The codeword for a given pixel is usually formed by the sequence of illumination values for that pixel across the projected patterns. Therefore, the codification is called temporal because the bits of the codewords are multiplexed in time. In these techniques only two illumination levels are commonly used, which are coded as 0 and 1. Every pixel of the pattern has its own codeword formed by the sequence of 0s and 1s corresponding to its value in every projected pattern. Therefore, a codeword is obtained only when the sequence is completed.

Pages et al. [16] classified these techniques as follows: (a) techniques based on binary codes: a sequence of binary patterns is used in order to generate binary codewords; (b) techniques based on n-ary codes: a basis of n primitives is used to generate the codewords; (c) Gray code combined with phase shifting: the same pattern is projected several times, shifting it in a certain direction in order to increase resolution; (d) hybrid techniques: a combination of time-multiplexing and neighbourhood strategies.

Figure 2.6 shows Posdamer and Altschuler [22] temporary codification which will be used in chapter 3. This codification technique is an example of techniques based on binary codes. In these techniques only two illumination levels are commonly used, which are coded as 0 and 1. Every pixel of the pattern has its own codeword formed by the sequence of 0s and 1s corresponding to its value in every projected pattern. Therefore, a codeword is obtained only when the sequence is completed.

Posdamer and Altschuler were the first to propose the projection of a sequence of m patterns to encode 2^m stripes using a plain binary code. Therefore, the codeword associated

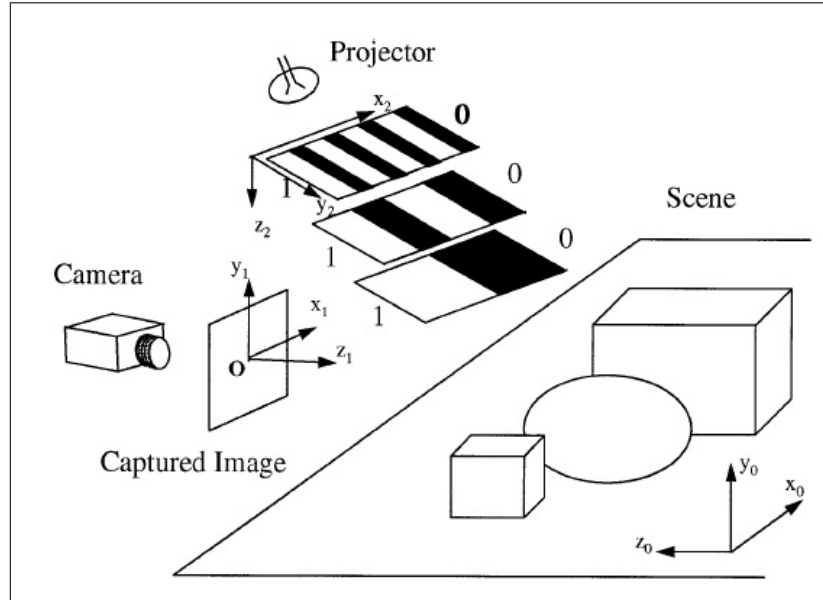


Figure 2.6: *The Temporary Codification Proposed by Posdamer and Altschuler [22].*

with each pixel is the sequence of 0s and 1s obtained from the m patterns, the first pattern being the one which contains the most significant bit. The symbol 0 corresponds to black intensity while 1 corresponds to full illuminated white. Therefore, the number of stripes increases by a factor of two for each consecutive pattern. Every stripe of the last pattern has its own binary codeword. Inokuchi et al. [23] improved the codification scheme of Posdamer and Altschuler by introducing Gray code instead of plain binary. The advantages of Gray code is that consecutive codewords have a Hamming distance of one, being more robust against noise.

Time-multiplexing was the first paradigm of coded structured light used to obtain 3D data from an unknown surface. The advantages of these techniques are the easy implementation, the high spatial resolution and the accurate 3D measurements that can be achieved. The main drawback is their inapplicability to moving surfaces since multiple patterns must be projected [16].

2.3.1.2 Spatial Neighbourhood

The techniques in this group tend to concentrate all the coding scheme in a unique pattern. The codeword that labels a certain point of the pattern is obtained from a neighbourhood of the points around it. However, the decoding stage becomes more difficult as the spatial neighbourhood cannot always be identified and reconstruction errors can arise. Normally, the visual features gathered in a neighbourhood are the intensity or colour of the pixels or groups of adjacent pixels around it. These spatial neighbourhood techniques can be classified as follows: (a) strategies based on non-formal codification: the neighbourhoods are generated intuitively; (b) strategies based on De Bruijn sequences: the neighbourhoods are defined using pseudorandom sequences; or (c) strategies based on M-arrays: extension of the pseudorandom theory to the 2-D case [16].

Spatial neighbourhood coding is the second big group of coded structured light techniques. The advantage compared with time-multiplexing is that such strategy permits, in most cases, moving surfaces to be measured. However, since the codification must be condensed in a unique pattern, the spatial resolution is lower.

2.3.1.3 Direct Codification

There are certain ways of creating a pattern so that every pixel can be labelled by the information represented on it. Therefore, the entire codeword for a given point is contained in a unique pixel. In order to achieve this, it is necessary to use either a large range of colour values or introduce periodicity. In theory, a high resolution of 3D information can be obtained. However, the sensitivity to noise is very high because the "distance" between "codewords", i.e. the colours used, is nearly zero. Moreover, the perceived colours depend not only on the projected colours, but also on the intrinsic colour of the measuring surface. This means, in most cases, that one or more reference images must be taken. Therefore, these techniques are not typically suitable for dynamic scenes.

Direct codification is usually constrained to neutral colour or pale objects. For this reason, it is necessary to perceive and identify the whole spectrum of colours, which requires a "tuning" stage that is not always easy to achieve[16]. These direct codification techniques can be classified into two groups of methods (a)codification based on grey levels: a spectrum of grey levels is used to encode the points of the pattern; (b) codification based on colour: these techniques take advantage of a large spectrum of colours.

2.4 Calibration

2.4.1 Perspective Projection and Pinhole Model

The pinhole model [24, 25] is simple and popular geometric model for cameras or projectors. It consists of a plane and a point external to that plane. The plane is called the image plane and the point is called the center of projection as shown in Fig (2.7a).

In a camera, every 3D point (except the center of projection) determines a unique line passing through the center of projection. If this line is not parallel to the image plane, then it must intersect the image plane in a single image point. In mathematics, this mapping from 3D points to 2D image points is called a perspective projection.

The geometry of a projector can be described with the same model because of the fact that light traverses this line in the opposite direction. That is, given a 2D image point in the projector's image plane, there must exist a unique line containing this point and the center of projection (since the center of projection cannot belong to the image plane). In summary, the projector is a camera inverse which means that light travels away from a projector along the line connecting the 3D scene point with its 2D perspective projection onto the image plane [26].

2.4.1.1 The Ideal Pinhole Camera

In the ideal pinhole camera shown in Fig (2.7b), the center of projection o is at the origin of the world coordinate system, with coordinates $(0, 0, 0)^t$, and the point q and the vectors v_1 and v_2 are defined as

$$[v_1|v_2|q] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.1)$$

Note that not every 3D point has a projection on the image plane. An arbitrary 3D point p with coordinates $(p^1, p^2, p^3)^t$ belongs to this plane if $p^3 = 0$, otherwise it projects onto an image point with the following coordinates.

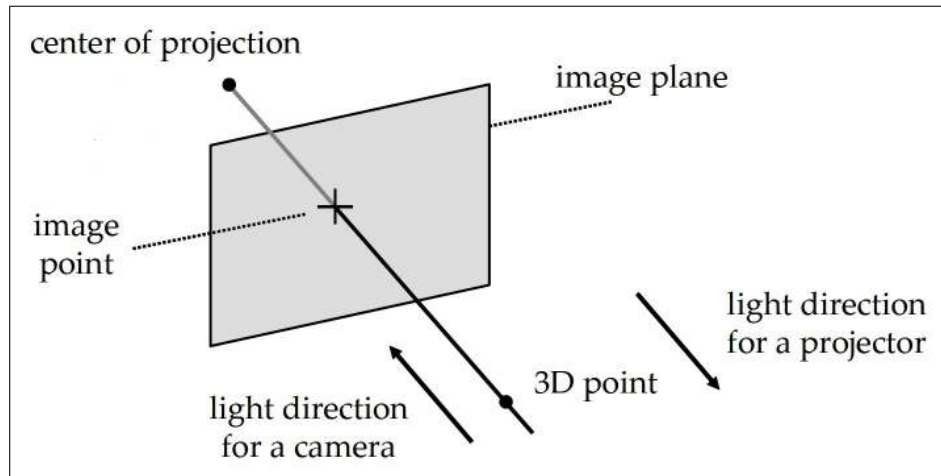
$$u^1 = p^1/p^3, u^2 = p^2/p^3 \quad (2.2)$$

The relation between the coordinates of a point and the image coordinates of its projection can be described in many ways; for example, the projection of a 3D point p with coordinates $(p^1, p^2, p^3)^t$ has image coordinates $u = (u^1, u^2, 1)$ if, for some scalar $\lambda \neq 0$, the relation will be

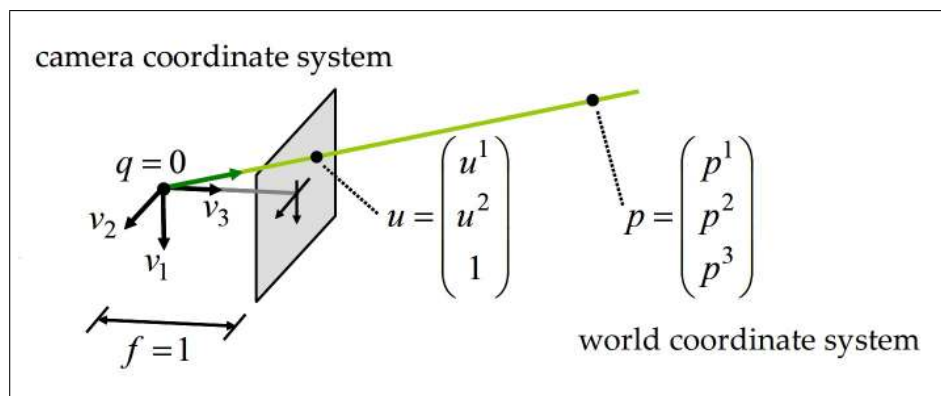
$$\lambda \begin{pmatrix} u^1 \\ u^2 \\ 1 \end{pmatrix} = \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix}. \quad (2.3)$$

2.4.1.2 The General Pinhole Camera

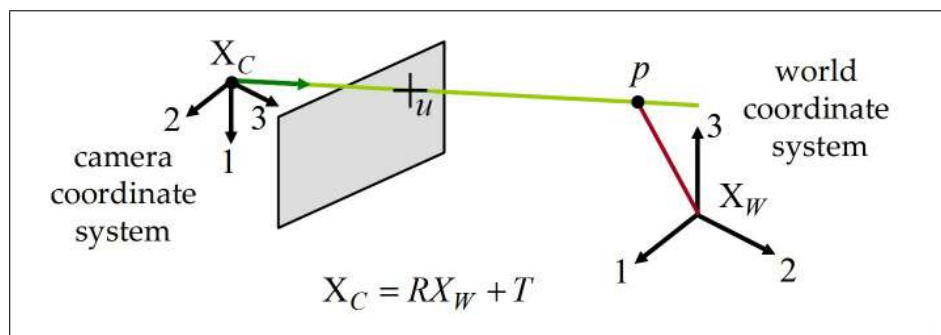
It is not necessarily that the center of a general pinhole camera is placed at the origin of the world coordinate system and it may be oriented. However, it does have a camera coordinate system attached to the camera, in addition to the world coordinate system as shown in Fig (2.7c). A 3D point p has world coordinates described by the vector $p_W = (p_W^1, p_W^2, p_W^3)^t$ and camera coordinates described by the vector $p_C = (p_C^1, p_C^2, p_C^3)^t$. These two vectors



(a)



(b)



(c)

Figure 2.7: (a) Perspective Projection Under The Pinhole Model. (b) The Ideal Pinhole Camera. (c) The General Pinhole Model. [26]

are related by a rigid body transformation specified by a translation vector $T \in \mathbb{R}^3$ and a rotation matrix $R \in \mathbb{R}^{3 \times 3}$, such that

$$p_C = Rp_W + T. \quad (2.4)$$

In camera coordinates, the relation between the 3D point coordinates and the 2D image coordinates of the projection is described by the ideal pinhole camera projection (i.e., Equation 2.3), with $\lambda u = p_C$. In world coordinates this relation becomes

$$\lambda u = Rp_W + T. \quad (2.5)$$

The parameters (R, T) are the extrinsic parameters of the camera, describe the location and orientation of the camera with respect to the world coordinate system. These parameters translate the coordinate of a point from the world coordinate to the camera coordinate.

Equation 2.5 assumes that the unit of measurement of lengths on the image plane is the same as for world coordinates, that the distance from the center of projection to the image plane is equal to one unit of length, and that the origin of the image coordinate system has image coordinates $u^1 = 0$ and $u^2 = 0$.

In practice, none of these assumptions hold. For example, lengths on the image plane are measured in pixel units, and in meters or inches for world coordinates, the distance from the center of projection to the image plane can be arbitrary, and the origin of the image coordinates is usually on the upper left corner of the image. In addition, the image plane may be tilted with respect to the ideal image plane. To overcome these limitations of the current model, a matrix $K \in \mathbb{R}^{3 \times 3}$ is introduced in the projection equations to describe intrinsic parameters as follows.

$$\lambda u = K(Rp_W + T) \quad (2.6)$$

The matrix K has the following form

$$K = \begin{pmatrix} fs_1 & fs_\theta & o^1 \\ 0 & fs_2 & o^2 \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.7)$$

where f is the focal length (i.e., the distance between the center of projection and the image plane). The parameters s_1 and s_2 are the first and second coordinate scale parameters, respectively. Note that such scale parameters are required since some cameras have non-square pixels. The parameter s_θ is used to compensate for a tilted image plane. Finally, $(o^1, o^2)^t$ are the image coordinates of the intersection of the vertical line in camera coordinates with the image plane. This point is called the image center or principal point. All intrinsic parameters (i.e., the matrix K) are independent of the camera pose.

The matrix K can be estimated once through a calibration procedure because it describes physical properties related to the mechanical and optical design of the camera. Image plane measurements can be normalized in pixel units by multiplying the measured image coordinate vector by K^{-1} , so that the relation between a 3D point in world coordinates and 2D image coordinates is described by Equation 2.5 [26].

2.4.2 Camera Calibration

Camera calibration requires estimating the parameters of the general pinhole model presented in section 2.4.1. The intrinsic parameters contains focal length, principal point, and the scale factors. The extrinsic parameters contains the rotation matrix and translation vector mapping between the world and camera coordinate systems. In total, 11 parameters (5 intrinsic and 6 extrinsic) must be estimated from a calibration sequence.

Calibration required recording a sequence of images of a calibration object, composed of a unique set of distinguishable features with known 3D displacements. Thus, each image of the calibration object provides a set of 2D-to-3D correspondences, mapping image

coordinates to scene points. Optimization over the set of 11 camera model parameters is required so that the set of 2D-to-3D correspondences are correctly predicted (i.e., the projection of each known 3D model feature is close to its measured image coordinates).

Many methods like Zhang [27] have been proposed to solve for the camera parameters given such correspondences. In this method, a planar checkerboard pattern is observed in two or more orientations. From this sequence, the intrinsic parameters can be separately solved. Afterwards, a single view of a checkerboard can be used to solve for the extrinsic parameters. This method is commonly used in computer graphics and vision publications.

2.4.3 Projector Calibration

Projector calibration has received increasing attention, because of the emergence of lower-cost digital projectors. A projector is simply the "inverse" of a camera, where 2D points on an image plane are mapped to outgoing light rays passing through the center of projection. Camera and projector calibrations are the necessary steps in any active computer vision systems, and therefore, various approaches and methods have been proposed to calibrate projectors.

One class of these approaches projects a calibration pattern onto a plane, "the wall", captures it by a camera, and then goes through the standard calibration work flow. It makes use of the idea which is based in considering the projector as an inverse camera which maps 2D image intensities into 3D rays [28, 29].

Sergio Fernandez et al. proposed a plane based calibration method of a projector-camera system. A checkerboard pattern is projected on a plane which contains another printed checkerboard. They recover 3D position for each projected corner using ray-plane intersection [30].

Ivan Martynov et al. also proposed a projector calibration method by inverting the standard camera calibration workflow. The calibration procedure requires a single camera, which does not need to be calibrated. The camera works as the sensor whether projected

dots and calibration pattern landmarks, such as the checkerboard corners, coincide. The 3D position for the projected dots is recovered by adjusting the projected dots to coincide with the landmarks and the final coordinates are used as inputs to a camera calibration method [31].

Another important class of the methods, including those referred to as Auto-calibration methods. These methods do not need a physical calibration target. Most auto-calibration methods can only estimate the extrinsic parameters [32] or require a calibrated camera [28], but recently many automatic methods have been proposed [33].

These methods are attractive choices because of their automatic processing, but there is always a need for highly accurate calibration in the structured light and active vision systems. The auto-calibration methods can solve the extrinsic parameters, but the intrinsic parameters should be solved by the inverted camera approach which use a physical calibration target, since this is accurate and should be done just once.

Furukawa and Kawasaki proposed a technique which uses structured light projection to calibrate the projector[34]. The correspondences are obtained using Gray code patterns, and the projector's intrinsic and extrinsic parameters are estimated using the epipolar constraints. The calibration depends on the non-linear optimization of an objective function, which needs good initial values of both intrinsic and extrinsic parameters.

Shuntaro Yamazaki and Masaaki Mochimaru propose a method for calibrating an active vision system, composed of a projector and a camera, using structured light projection. Unlike existing methods of self-calibration for projector-camera systems, their method estimates the intrinsic parameters of both the projector and the camera as well as extrinsic parameters except a global scale without any calibration apparatus such as a checkerboard. Their method is based on the decomposition of a radial fundamental matrix into intrinsic and extrinsic parameters [35].

Raskar and Berdsey propose a method for calibrating the focal length and the principal points of a projector by observing a planar surface using a pre-calibrated camera [36].

Okatani and Deguchi solve the projector calibration problem using multiple patterns projected onto planar surfaces [32].

2.5 Facial Animation Techniques

Animation of human faces has been an active research topic in Computer Graphics for three decades. Realistic animation of 3D character face represents one of the most difficult challenges in computer animation. The difficulties mainly come from three problems:

1. The representation of the face model. It is extremely difficult to model a face with a detailed geometry and natural-looking skin.
2. Motion control of facial expression. Facial expression involves psychology and physiology. Adding the complex interactions between the bone and the muscle, make realistic facial animation exceptionally difficult.
3. The sensitivity of viewers. As facial expression is one of the major ways we communicate with each other, people are very sensitive at the details of facial expressions. Even a subtle change in facial expression can strongly draw the viewers' attention [37].

Recent interest in facial modelling and animation is motivated by the increasing appearance of virtual characters in film and video and inexpensive desktop processing power. It is difficult to classify facial modelling and animation techniques because exact classifications are complicated by the lack of exact boundaries between methods and the fact that recent approaches often integrate several methods to produce better results.

Zhigang Deng and Junyong Noh [38] classify facial modelling and animation techniques into the following categories: Blend shape or Shape interpolation, Parameterizations, Facial Action Coding System based approaches, Deformation based approaches, Physics based muscle and Performance driven facial animation.

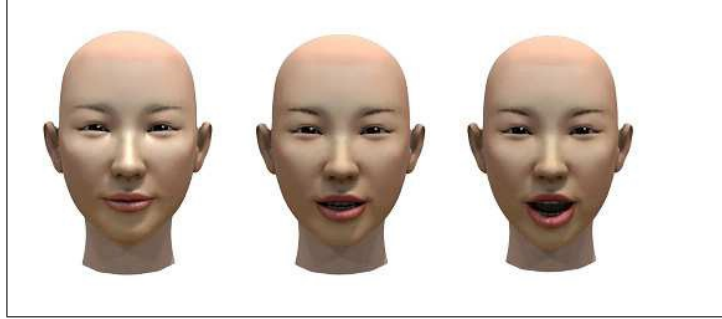


Figure 2.8: Linear Interpolation is performed on blend shapes. Left: Neutral pose, Right: "A" mouth shape, Middle: Interpolated shape[38].

2.5.1 Blend Shapes or Shape Interpolation

Shape interpolation (blend shapes, morph targets and shape interpolation) is the most sensitive and commonly used technique in facial animation practice. A blendshape model is simply the linear weighted sum of a number of topologically conforming shape primitives (See equation 2.8).

$$v_j = \sum_{k=1}^n w_k b_{kj} \quad (2.8)$$

where v_j is the j^{th} vertex of the resulting animated model, w_k is blending weight, b_{kj} is the j^{th} vertex of the k^{th} blendshape, and n is the number of blendshapes. The weighted sum can be applied to the vertices of polygonal models, or to the control vertices of spline models. The weights w_k can be determined automatically by algorithms [39] or manipulated by the animator. Projects such as the Stuart Little, Star Wars, and Lord of the Rings use this technique. It was also adopted in many commercial animation software packages such as Maya and 3D Studio Max. The simplest case is an interpolation between two key-frames at extreme positions over a time interval as shown in Fig (2.8).

Linear interpolation is often employed for simplicity [40], but a cosine interpolation function [41] or other variations such as spline can provide acceleration and deceleration effects at the beginning and end of an animation. When four key frames are involved, rather than two, bilinear interpolation generates a greater variety of facial expressions than

linear interpolation [42]. Bilinear interpolation, when combined with simultaneous image morphing, creates a wide range of facial expression changes [43].

2.5.2 Parameterizations

Parameterization techniques for facial animation [44, 45] overcome some of the limitations and restrictions of simple interpolations. Ideal parameterizations specify any possible face and expression by a combination of independent parameter values [46]. Unlike interpolation techniques, parameterizations allow explicit control of specific facial configurations. Combinations of parameters provide a large range of facial expressions with relatively low computational costs.

2.5.3 Facial Action Coding System

The Facial Action Coding System (FACS) is a description of the movements of the facial muscles and jaw/tongue derived from an analysis of facial anatomy [47]. FACS includes 44 basic action units (AUs). Combinations of independent action units generate facial expressions. For example, combining the AU1 (Inner brow raiser), AU6 (Cheek Raiser), AU12 (Lip Corner Puller), and AU14 (Dimpler) creates a happiness expression. Tables 2.2, 2.3 show the sample action units and the basic expressions generated by the actions respectively.

Table 2.2: The Sample Single Action Units

AU	FACS Name	AU	FACS Name	AU	FACS Name
1	Inner Brow Raiser	12	Lid Corner Puller	2	Outer Brow Raiser
14	Dimpler	4	Brow Lower	15	Lip Corner Depressor
5	Upper Lid Raiser	16	Lower Lip Depressor	6	Check Raiser
17	Chin Raiser	9	Nose Wrinkler	20	Lip Stretcher
23	Lip Tightener	10	Upper Lid Raiser	26	Jaw Drop

Table 2.3: Example Sets of Action Units for Basic Expressions

Basic Expressions	Involved Action Units
Surprise	AU 1, 2, 5, 15, 16, 20, 26
Fear	AU 1, 2, 4, 5, 15, 20, 26
Anger	AU 2, 4, 7, 9, 10, 20, 26
Happiness	AU 1, 6, 12, 14
Sadness	AU 1, 4, 15, 23

2.5.4 Deformation Based Approaches

Direct deformation defined on the facial mesh surface often produces quality animation. It ignores underlying facial anatomy or true muscle structures. Instead, the focus is on creating various facial expressions by manipulating the mesh. This category includes the free form deformations [48, 49].

2.5.4.1 Free From Deformation

Free form deformation (FFD) deforms volumetric objects by manipulating control points arranged in a three-dimensional cubic lattice [50]. As the control points are manipulated by moving from their positions in the lattice, it causes the control boxes to be bend, squashed, or twisted. The deformations of the FFD lattice are then automatically passed to the model. Consequently, the embedded object deforms accordingly as shown in Fig (2.9).

Extended free form deformation(EFFD) [51] allows the extension of the control point lattice into a cylindrical structure. Additional flexibility for shape deformation is achieved by using a cylindrical lattice compared to regular cubic lattices. Rational free form deformation (RFFD) adds extra degree of freedom in specifying deformations by incorporating weight factors for each control point. So, deformations are possible by not only changing the control point positions but also changing the weight factors. When all weights are equal to one, then RFFD becomes a FFD.

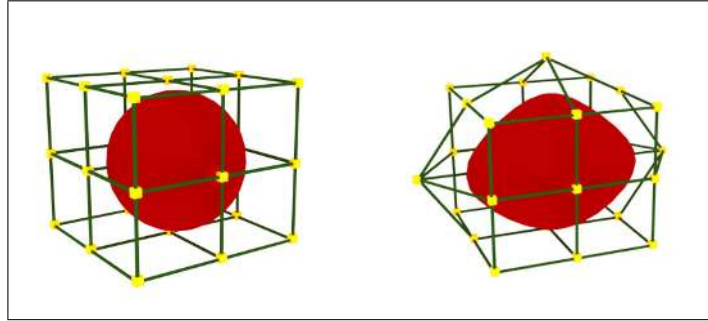


Figure 2.9: *Free Form Deformation. Controlling box and embedded object are shown. When controlling box is deformed by manipulating control points, the embedded object deforms accordingly.*

2.5.5 Physics Based Muscle Modelling

In this technique, the face is represented by its skin. Facial expressions are produced by applying abstract muscle actions to the skin [52]. Physics-based muscle models fall into three categories: mass spring systems, vector representations, and layered spring meshes.

Mass-spring methods propagate muscle forces in an elastic spring mesh that models skin deformation. The vector approach deforms a facial mesh using motion fields in delineated regions of influence. A layered spring mesh extends a mass spring structure into three connected mesh layers to model anatomical facial behavior more faithfully.

2.5.6 Performance Driven Facial Animation

The difficulties in controlling facial animations led to the performance driven approach where the facial animation is controlled by real motion of human face. These methods capture actual performers' movements and actions in order to use them to animate synthetic characters. This technique has many advantages where it save significant time and efforts and produce more realistic facial animation.

Performance driven facial animation contains many approaches which differ in how to get the motion data from the real actor. Most of these methods track facial markers from real actors and recover the 2D or 3D positions of these markers. Often the tracked 2D or 3D

feature motions are filtered or transformed to generate the motion data needed for driving a specific animation system. Motion data can be used to directly generate facial animation [53] or to infer AUs of FACS in generating facial expressions. The following subsection describes the motion capture systems.

2.5.6.1 Facial Motion Capture System

To produce quality animation, the use of 3D motion capture data is necessary. Six or seven high performance cameras are used to reconstruct the 3D marker locations on the face. Although this optical system is difficult to set up and expensive, the reconstructed data provide accurate timing and motion information.

In motion capture sessions, movements of one or more actors are sampled many times per second, early techniques used images from multiple cameras and calculate 3D positions, motion capture often records only the movements of the actor, not his or her visual appearance. This animation data is often mapped to a 3D model so that the model performs the same actions as the actor. Motion tracking or motion capture started as a photogrammetric analysis tool in biomechanics research in the 1970s and 1980s, and expanded into education, training, sports and recently computer animation for television, cinema, and video games. A performer wears markers (LED, magnetic or reflective markers, or combinations of any of these) near each joint to identify the motion by the positions or angles between the markers. Figure 2.10 shows OptiTrack facial capture system [54].



Figure 2.10: *OptiTrack Facial Capture System.*

Chapter 3

Building a 3D Scanner Using a Novel Projector Calibration Method

3.1 Introduction

A 3D scanner is a device that scan a real-world object or environment to get its shape or surface and possibly its appearance (i.e. color). The collected data can then be used to construct three dimensional models. Collected 3D data is useful for a wide variety of applications. These devices are used extensively by the entertainment industry in the production of movies and video games. Other common applications of this technology include industrial design, reverse engineering and prototyping, quality control/inspection, and documentation of cultural artifacts.

The purpose of a 3D scanner is usually to create a point cloud of geometric samples on the surface of the subject. These points can then be used to extrapolate the shape of the subject (a process called reconstruction). If color information is collected at each point, then the colors on the surface of the subject can also be determined.

3D scanners share several traits with cameras. Like cameras, they have a cone-like field of view, and like cameras, they can only collect information about surfaces that are

not obscured. While a camera collects color information about surfaces within its field of view, a 3D scanner collects distance information about surfaces within its field of view. The "picture" produced by a 3D scanner describes the distance to a surface at each point in the picture. This allows the three dimensional position of each point in the picture to be identified.

For most situations, a single scan will not produce a complete model of the subject. Multiple scans, even hundreds, from many different directions are usually required to obtain information about all sides of the subject. These scans have to be brought in a common reference system, a process that is usually called alignment or registration, and then merged to create a complete model. This whole process, going from the single range map to the whole model, is usually known as the 3D scanning pipeline[55].

In this chapter a structured light scanner is described. It consists of one or more digital cameras and a single projector. It also discusses a novel method for projector calibration which is based on passive stereo and triangulation.

3.2 Data Capture

3.2.1 Scanner Hardware

The proposed 3D Scanner consists of two digital cameras and a single digital projector as shown in Fig (3.1). The object will eventually be reconstructed by ray-plane triangulation, between each camera ray and a plane corresponding to the projector column (and/or row) illuminating that point on the surface. The cameras and projector should be arranged to ensure that no camera ray and projector plane meet at small incidence angles. A "diagonal" placement of the cameras, as shown in the figure, ensures that both projector rows and columns can be used for reconstruction.

A wide variety of digital cameras and projectors can be selected for the proposed 3D scanner. While low-cost webcams will be sufficient, access to raw imagery will eliminate

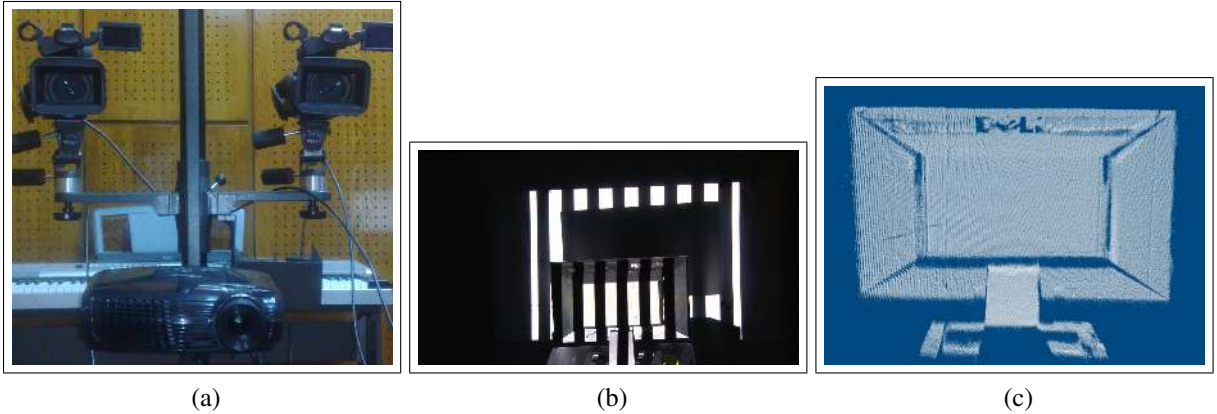


Figure 3.1: *Structured Light for 3D Scanning.* (a) A structured light scanning system containing a pair of digital cameras and a single projector. (b) An image of an object illuminated by bit planes of a Gray code structured light sequence. (c) A reconstructed 3D points cloud.

decoding errors introduced by compression artifacts. The selected camera must be supported by the development environment. For example, if the MATLAB Image Acquisition Toolbox is used, then any DCAM-compatible FireWire camera or webcam with a Windows Driver Model (WDM) or Video for Windows (VFW) driver will work [56]. If OpenCV is used, a list compatible cameras is maintained on the OpenCV wiki [57]. Almost any digital projector can be used, since the operating system will simply treat it as an additional display. The proposed 3D scanner contains a single data projector (Optoma EH020) with a resolution of 1024x768 pixels, two cameras (Sony nxcam) and a frame grabber (DeckLink Studio) digitizing images at 1920x1080 pixels with 24 bits per pixel (RGB).

3.2.2 Structured Light Sequences

The primary job of the projector is to eliminate the mechanical motion required in point and slit scanners [58]. The projector can be used to display a single column (or row) of white pixels translating against a black background. Since the projector resolution is 1024x768, 1024 (or 768) images would be required to assign the correspondences between camera

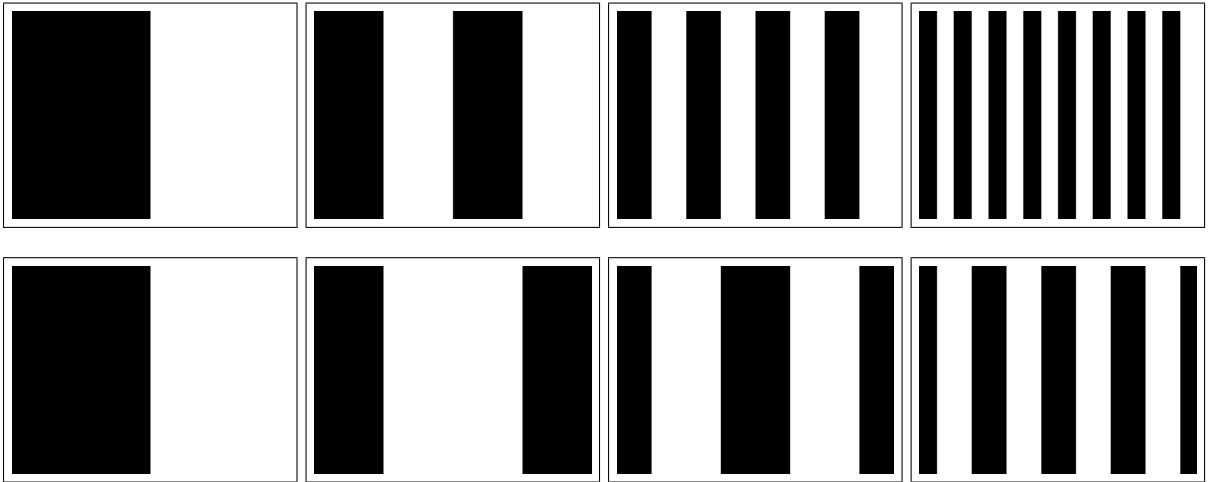


Figure 3.2: *Structured Light for 3D Scanning. Structured Light Illumination Sequences.* (Top row, left to right) The first four bit planes of a binary encoding of the projector columns, ordered from most to least significant bit. (Bottom row, left to right) The first four bit planes of a Gray code sequence encoding the projector columns.

pixels and projector columns (or rows). After establishing the correspondences and calibrating the system, a 3D point cloud is reconstructed using familiar ray-plane triangulation.

Since 24-bit color images can be projected, one would expect that there exists a sequence of coded patterns, besides a simple translation of a single stripe, that allows the projector-camera correspondences to be assigned in relatively few frames. In general, the identity of each plane can be encoded spatially (i.e., within a single frame) or temporally (i.e., across multiple frames), or with a combination of both spatial and temporal encodings. There are benefits and drawbacks to each strategy. For instance, purely spatial encodings allow a single static pattern to be used for reconstruction, enabling dynamic scenes to be captured. Section 2.3 gives more details about structured light codification.

The proposed 3D Scanner uses purely temporal encodings (i.e., the structured light codification is across multiple frames). While such patterns are not well-suited to scan dynamic scenes, they have the benefit of being easy to decode and are robust to surface texture variations, producing accurate reconstructions for static objects.

Posdamer and Altschuler [22] proposed the first simple binary structured light sequence in 1981. As shown in Fig (3.2), the binary encoding consists of a sequence of binary images in which each frame is a single bit plane of the binary representation of the integer indices for the projector columns (or rows). For example, column 546 in our prototype has a binary representation of 1000100010 (ordered from the most to the least significant bit). Similarly, column 546 of the binary structured light sequence has an identical bit sequence, with each frame displaying the next bit. Assigning an accurate projector column/row to camera pixel correspondence is the most important step, otherwise triangulation artifacts will lead to large reconstruction errors.

Inokuchi et al. [23] proposed Gray codes as one alternative to the simple binary encoding in 1984. Figure (3.3) shows a comparison of binary (top) and Gray code (bottom) structured light sequences. Each image represents the sequence of bit planes displayed during data acquisition. Image rows corresponding to the bit planes encode the projector columns, assuming a projector resolution of 1024x768, ordered from most to least significant bit (from top to bottom). As shown in Fig (3.3), the Gray code can be obtained by reflecting, in a specific manner, the individual bit-planes of the binary encoding. Algorithms (3.2.1 and 3.2.2) show the Pseudo codes for converting between binary and Gray codes. For example, column 546 in our implementation has a Gray code representation of 1100110011, as given by BIN2GRAY. The key property of the Gray code is that two adjacent code words (e.g., adjacent columns in the projected sequence) only differ by one bit (i.e., adjacent codes have a Hamming distance of one). As a result, the Gray code structured light sequence tends to be more robust to decoding errors than a simple binary encoding.

It is easy to generate binary codes and gray codes structured light sequences. If the projected image width and height are w and h respectively, the structured light sequence contains $2^{\lceil \log_2 w \rceil} + 2^{\lceil \log_2 h \rceil} + 2$ uncompressed images. The first two images consist of an all-white and an all-black image, respectively. The next $2^{\lceil \log_2 w \rceil}$ images contain the bit planes of the binary sequence encoding the projector columns, interleaved with the

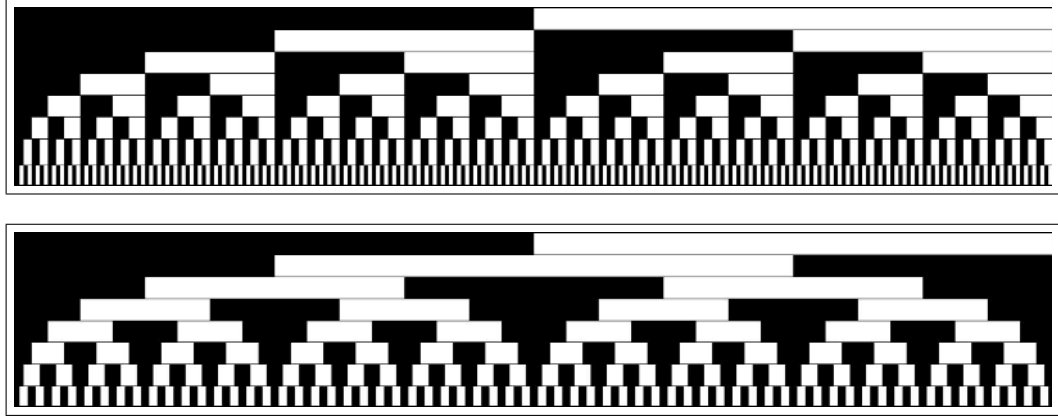


Figure 3.3: *Comparison of Binary (top) and Gray Code (bottom) Structured Light Sequences.*

binary inverse of each bit plane (to assist in decoding). The last $2^{\lceil \log_2 h \rceil}$ images contain a similar encoding for the projector rows.

Algorithm 3.2.1: BIN2GRAY(B)

comment: Compute Gray Code G from Binary code B

$n \leftarrow \text{length}[B]$ $G[1] \leftarrow B[1]$

for $i \leftarrow 2$ **to** n

do $G[i] \leftarrow B[i - 1] \text{ xor } B[i]$

return (G)

Algorithm 3.2.2: GRAY2BIN(G)

comment: Compute Binary Code B from Gray code G

$n \leftarrow \text{length}[G]$ $B[1] \leftarrow G[1]$

for $i \leftarrow 2$ **to** n

do $B[i] \leftarrow B[i - 1] \text{ xor } G[i]$

return (B)

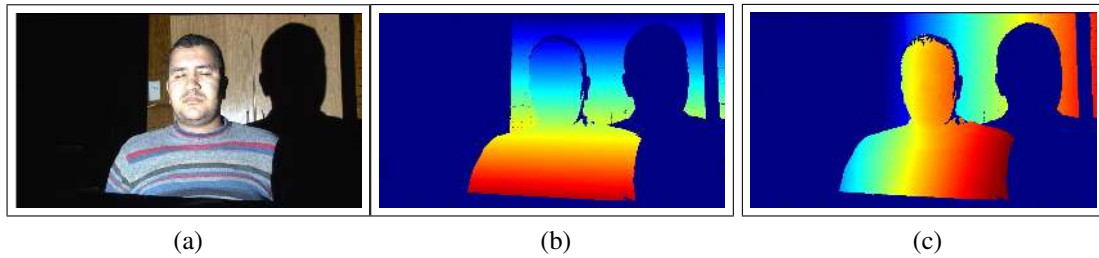


Figure 3.4: *Decoding Structured Light Illumination Sequences. (a) Camera image captured while projecting an all white frame. (b) The decoding results for a Gray code structured light sequence, with projector row and camera pixel correspondences. (c) Similar decoding results for projector column correspondences.*

3.3 Decoding of Structured Light Sequences

This section describes the decoding process for the structured light sequences described in the previous section. For each camera, it must be determined whether a given pixel is directly illuminated by the projector in each displayed image. If it is illuminated in any given frame, then the corresponding code bit is set high, otherwise it is set low. By decoding the received bit sequences for each camera pixel, the decimal integer index of the corresponding projector column (and/or row) can then be recovered. Figure (3.4) shows the decoding structured light illumination sequences represented using a jet colormap in MATLAB (jet ranges from blue to red, and passes through the colors cyan, yellow, and orange). Points that cannot be assigned a correspondence are shown in black. An intensity threshold is used to determine whether a given pixel is illuminated. For instance, $\lceil \log_2 w \rceil + 2$ images could be used to encode the projector columns, with the additional two images consisting of all-white and all-black frames. The average intensity of all-white and all-black frames could be used to assign a per-pixel threshold; the individual bit planes of the projected sequence could then be decoded by comparing the received intensity to the threshold.

Since certain points on the surface may only receive indirect illumination scattered from directly-illuminated points, the scattered light may cause a bit error, in which an unilluminated point appears illuminated due to scattered light. Depending on the specific structured light sequence, such bit errors may produce significant reconstruction errors in the 3D point cloud. One solution is to project each bit plane and its inverse. While the frames and its inverse are projected to encode the projector columns, the decoding process is less sensitive to scattered light, since a variable per-pixel threshold can be used. So, a bit is determined to be high or low depending on whether a projected bit-plane or its inverse is brighter at a given pixel.

3.4 Projector Calibration

In the past, 3D shape reconstruction process was based on passive stereo which do not require direct control of any illumination source, instead relying entirely on light. Nowadays, 3D shape reconstruction is based on active stereo which replace one camera with a projector. The projector plays an important part in solving the correspondence problem. It projects coded patterns on the scanned object. By capturing the deformed pattern using cameras, the correspondences between image pixels and projector (columns-rows) can be found easily. To do this, the projector must be calibrated.

In this section, the problem of projector calibration is solved by passive stereo and triangulation. The proposed system consists of two cameras, projector, and planner board. A checkerboard pattern is projected on the board and then captured by the two cameras. Using triangulation, the corresponding 3D points of the projected pattern is computed. In this way, having the 2D projected points in the projector frame and its 3D correspondences (calculated using triangulation) the system can be calibrated using a standard camera calibration method. A data projector has been calibrated by this method and accurate results have been achieved.

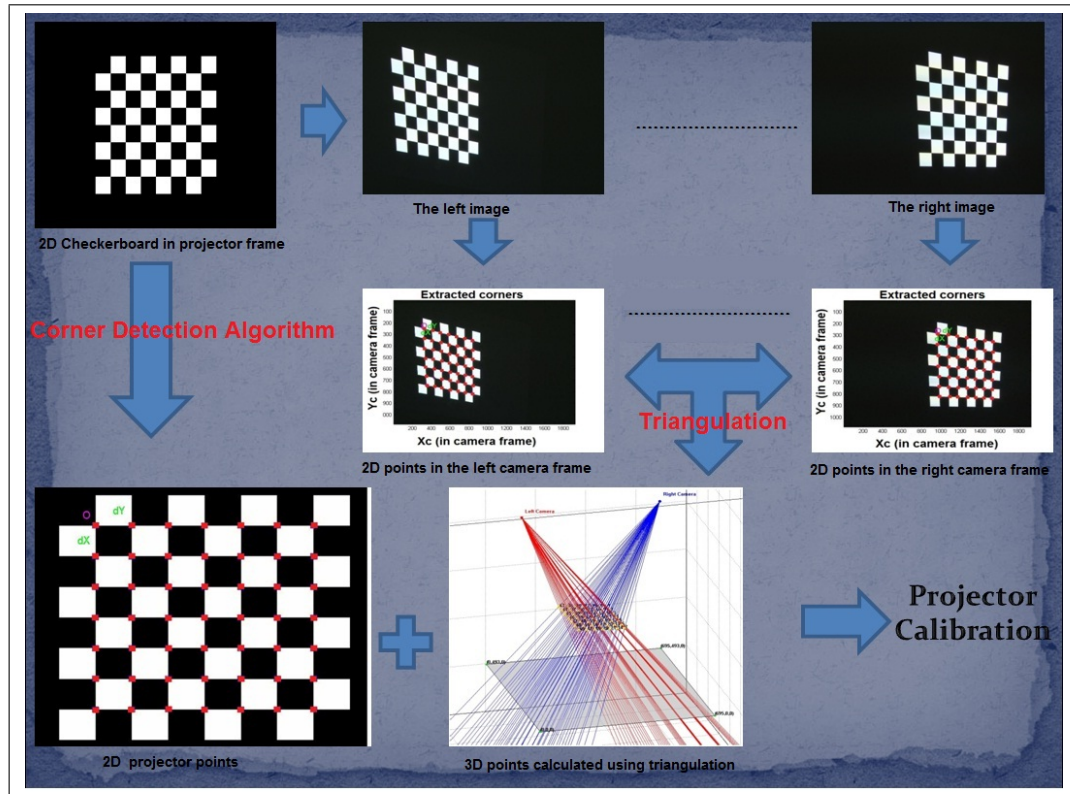


Figure 3.5: The Proposed System Diagram.

The new idea in this method is the use of passive stereo (i.e. two cameras) to calibrate the projector which is a main component in active stereo. The projector can display a calibration pattern (2D checkerboard pattern image contains 49 corners in projector frame). The problem is how to calculate the 3D points in the world coordinate system correspondent to these 49 2D corners. First, Projector is not like a camera. It can not take images of the projected calibration pattern so, a camera must be used to capture what the projector displays. Second, the calibrating pattern is projected and not attached to the world coordinate frame. Any horizontal surface such as white board must be used to receive the projected calibrating pattern. Figure (3.5) shows a diagram which explains the proposed method.

3.4.1 System Overview

The proposed system consists of four major steps as shown in Fig (3.6). The first step is pattern displaying and capturing. In this step, the projector displays a checkerboard pattern on the white board in full screen mode. The two cameras capture this pattern and store the images. The white board is placed in different positions and this step is repeated.

The second step is corners' extraction and correspondences' matching. In this step, the 2D corners in every pair of images are extracted. So, there are many sets of correspondences between the two cameras. In the third step, the 3D points of the right/left correspondences are reconstructed using triangulation. Finally, the projector parameters are estimated and the calibration step is done.

3.4.2 Correspondences' Matching

This section describes how to get the correspondence points. It consists of the first two steps in the proposed system. The first step, Pattern Displaying and Capturing, a checkerboard pattern is projected on the white board and captured by the two cameras. The second step, Corners' Extraction, the corners of all captured images are extracted.

3.4.2.1 Pattern Displaying and Capturing

The proposed system consists of two calibrated cameras (left-right), white board, and projector as shown in Fig (3.7). The two cameras are calibrated using Zhang's method [27], a flexible new technique to easily calibrate a camera. It only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. This procedure consists of a closed-form solution, followed by a nonlinear refinement based on the maximum likelihood criterion. This algorithm was implemented in Matlab Camera Calibration Toolbox

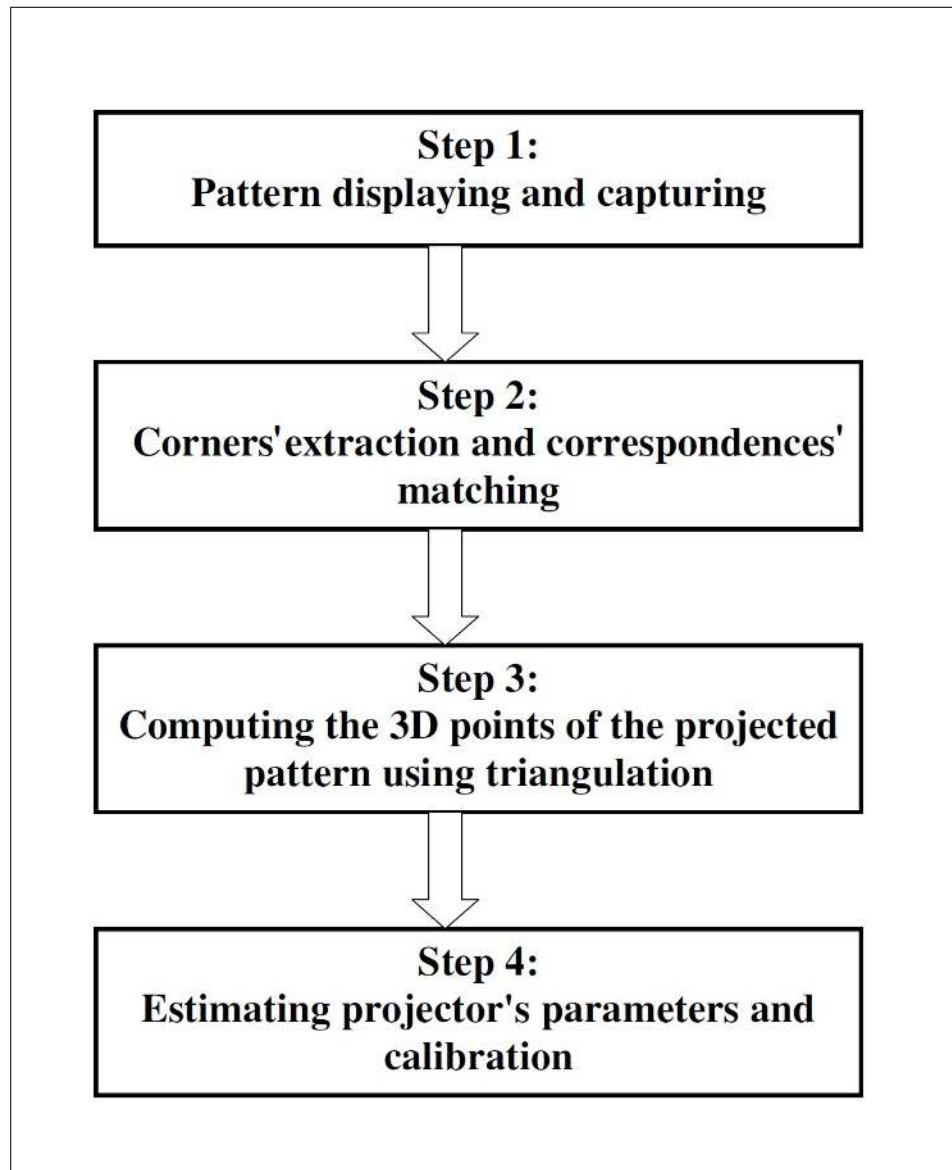
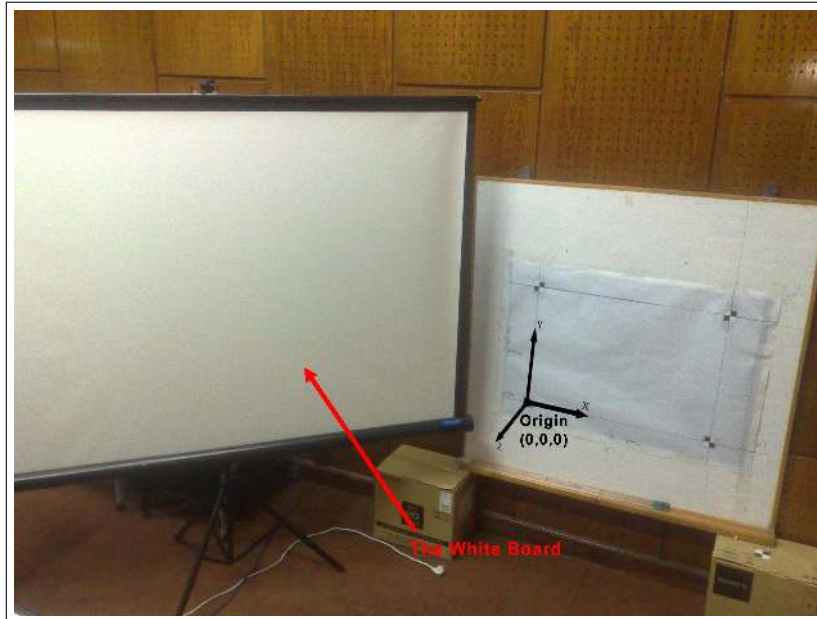


Figure 3.6: Overview of The Proposed Method



(a)



(b)

Figure 3.7: (a) *The White Board and The World Coordinate System.* (b) *The Two Cameras and The Projector.*

[59] by Jean-Yves Bouguet and C++ in Intel OpenCV library [60]. These two libraries are probably the most widely used tools for camera calibration nowadays.

The projector displays a calibration pattern (2D checkerboard pattern image contains 49 corners in projector frame) as shown in Fig (3.8a). Figures (3.8b and 3.8c) show this calibration pattern falling on the white board and seen from the two cameras. The white board is moved in many positions. In every position, the calibration pattern is captured by the two cameras. Now, there are many pairs of images (left-right) and we are ready for the next step.

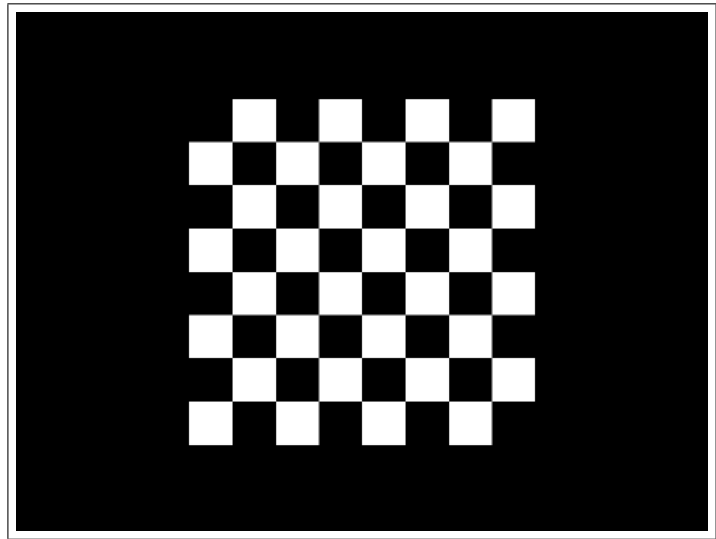
3.4.2.2 Corners' Extraction

After capturing the patterns, the 2D corners in all (left-right) image pairs are extracted [61]. First, the right image is displayed and the four extreme corners on the projected checkerboard pattern are clicked clockwise or counter-clockwise starting with any corner. When the left image is displayed, the same clicking mechanism must be used. Figures (3.8d and 3.8e) show the extracted corners (i.e., the correspondences).

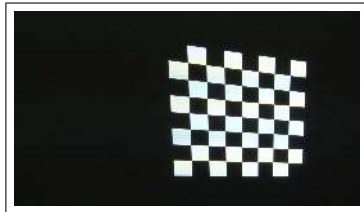
3.4.3 Reconstruction by Triangulation

This is the most important step in the proposed system. The 3D coordinate values of every left-right corners extracted from images captured in the previous step can be constructed using triangulation.

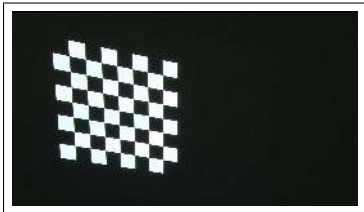
The projector is modelled as an inverse camera (i.e., one in which light travels in the opposite direction as usual). Under this model, calibration proceeds in a similar manner as with cameras. Rather than photographing fixed checkerboards, known checkerboard patterns are projected and their distorted appearances are captured by the cameras when reflected from a diffuse rigid object (i.e., the white board) [28]. This approach has the advantage of being a direct extension of Zhangs calibration algorithm for cameras.



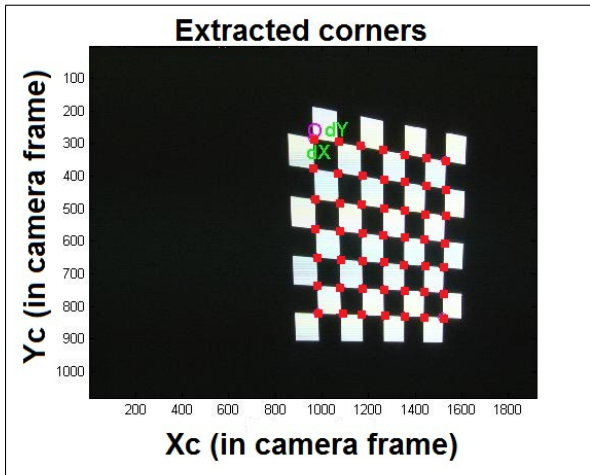
(a)



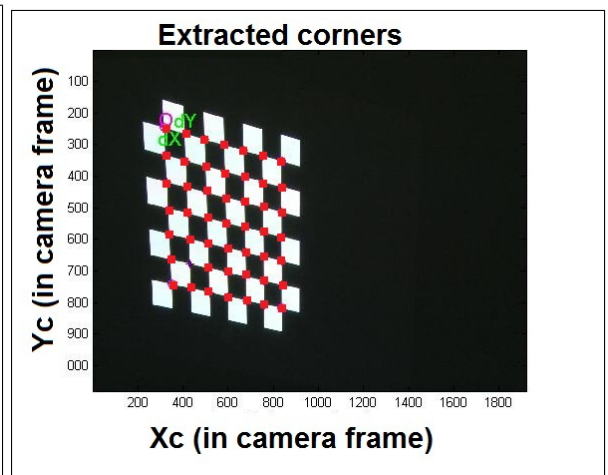
(b)



(c)



(d)



(e)

Figure 3.8: (a) Checkerboard Pattern in Projector Frame. (b) The Left View. (c) The Right View. (d) The Left Extracted Corners. (e) The Right Extracted Corners.

3.4.3.1 The Mathematics of Triangulation

Under the pinhole camera model described in subsection 2.4.1, each corner in the left image creates a ray (a unique line containing this image point and the center of projection), and also the corresponding corner in the right image. The intersection of these two rays is the 3D value related to these corners. So, the 3D values of all the corners of the projected 2D checkerboard pattern can be calculated.

Let \mathbf{x}_1 and \mathbf{x}_2 be a 2D point correspondence where \mathbf{x}_1 is a corner in the left image and \mathbf{x}_2 is the corresponding corner in the right image in homogeneous coordinates. Let \mathbf{P}_1 and \mathbf{P}_2 be the two projection matrices for the left and right cameras respectively. The 3D point location \mathbf{X} is given as follows

$$\lambda_1 \mathbf{x}_1 = \mathbf{P}_1 \mathbf{X}, \quad (3.1)$$

$$\lambda_2 \mathbf{x}_2 = \mathbf{P}_2 \mathbf{X}. \quad (3.2)$$

Multiplying both sides of the equations with the cross-product of each point results

$$\mathbf{x}_1 \times \mathbf{P}_1 \mathbf{X} = [\mathbf{x}_1 \times] \mathbf{P}_1 \mathbf{X} = 0 \quad (3.3)$$

$$\mathbf{x}_2 \times \mathbf{P}_2 \mathbf{X} = [\mathbf{x}_2 \times] \mathbf{P}_2 \mathbf{X} = 0 \quad (3.4)$$

where the skew-symmetric matrices are used $[\mathbf{x}_i \times]$ to replace the cross product

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}_\times] \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \mathbf{b}. \quad (3.5)$$

Each 2D point provides two independent equations for a total of three unknowns. The overconstrained system can be solved by stacking the first two equations for each point in a matrix A and computing the least-squares solution for $AX = 0$ which can be easily solved by Singular Value Decomposition (SVD)[62]. Applying SVD to A yields the

decomposition $A = UDV$. The homogeneous least-squares solution corresponds to the least singular vector, which is given by the last column of V . Algorithm (3.4.1) shows the triangulate function discussed in this section.

Now, the 3D coordinates of the projected checkerboard pattern is calculated as shown in Fig(3.9a) and the 2D corners of the projected pattern in the projector frame can be obtained by extracting the corners of the image projected by the projector as shown in Fig(3.9b). Finally, the system is ready for calibration.

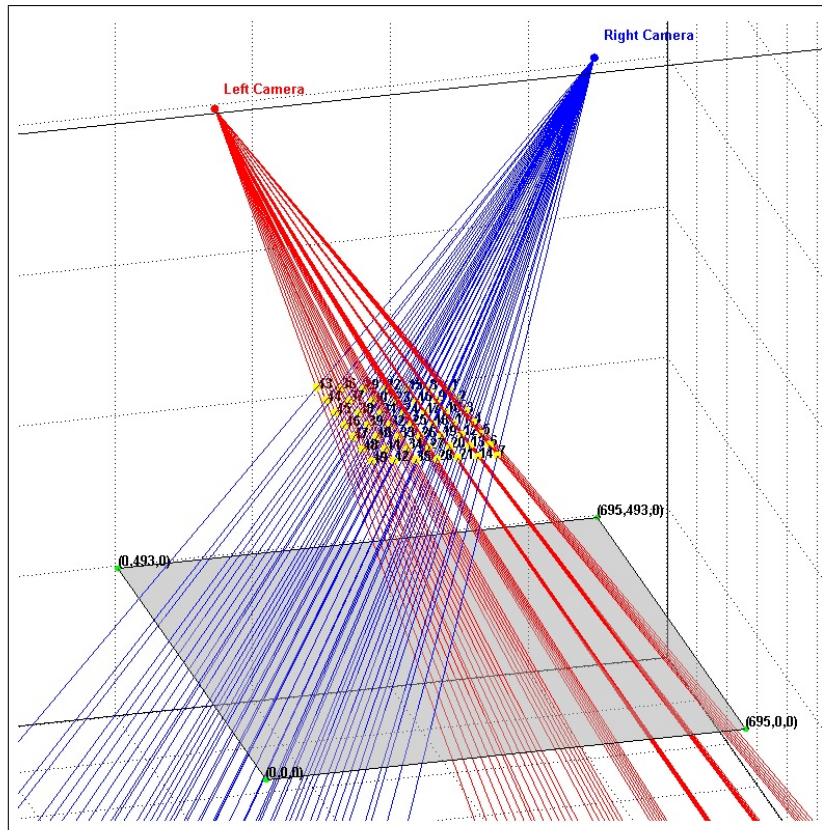
3.4.4 Estimating Calibration Parameters

Modelling the projector as camera inverse lets us use any camera calibration methods to calibrate our projector like Zhangs method. Camera (and also projector) calibration requires estimating the parameters of the general pin-hole model presented in section 2.4.1.2. This includes the intrinsic parameters (focal length, principal point, and the scale factors) as well as the extrinsic parameters (the rotation matrix and translation vector mapping between the world and camera coordinate systems). In total, 11 parameters (5 intrinsic and 6 extrinsic) must be estimated from a calibration sequence. In practice, a lens distortion model must be estimated as well. Algorithm (3.4.2) shows the projector calibration method used to calibrate the projector.

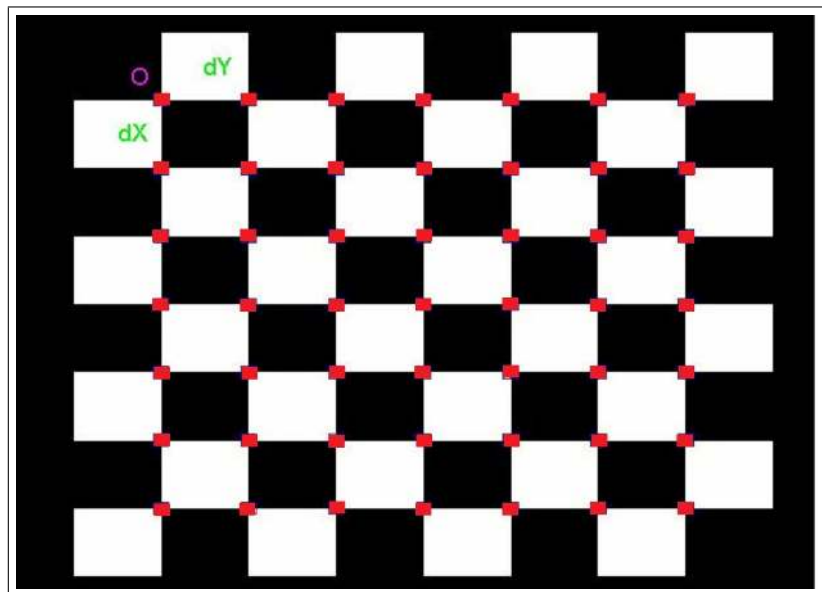
3.4.4.1 Definition of The Intrinsic Parameters

The list of intrinsic parameters:

- (1) Focal length: The focal length in pixels is stored in the 2x1 vector fc .
- (2) Principal point: The principal point coordinates are stored in the 2x1 vector cc .
- (3) Skew coefficient: The skew coefficient defining the angle between the x and y pixel axes is stored in the scalar $alphaC$.
- (4) Distortions: The image distortion coefficients (radial and tangential distortions) are stored in the 5x1 vector kc .



(a)



(b)

Figure 3.9: (a) The 3D Points of The Checkerboard Pattern Reconstructed by Triangulation.(b) The 2D Points of The Checkerboard Pattern in Projector Frame.

Let P be a point in the camera coordinate system. It has a vector value $XXc = [Xc; Yc; Zc]$. Let us project that point on the image plane according to the intrinsic parameters ($fc, cc, alphaC, kc$).

Let x_n be the normalized (pinhole) image projection:

$$\mathbf{x}_n = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.6)$$

Let $r^2 = x^2 + y^2$. After including the lens distortion, the new normalized point coordinate x_d is defined as follows:

$$\mathbf{x}_d = \begin{bmatrix} x_d(0) \\ x_d(1) \end{bmatrix} = \left(1 + kc(0)r^2 + kc(1)r^4 + kc(4)r^6\right)x_n + dx \quad (3.7)$$

where dx is the tangential distortion vector:

$$\mathbf{dx} = \begin{bmatrix} 2kc(2)xy + kc(3)(r^2 + 2x^2) \\ kc(2)(r^2 + 2y^2) + 2kc(3)xy \end{bmatrix} \quad (3.8)$$

Therefore, the 5-vector kc contains both radial and tangential distortion coefficients. This distortion model was first introduced by Brown in 1966 and called "Plumb Bob" model (radial polynomial + "thin prism"). The tangential distortion is due to "decentering", or imperfect centering of the lens components and other manufacturing defects in a compound lens[63].

Once distortion is applied, the final pixel coordinates $x_{pixel} = [x_p; y_p]$ of the projection of P on the image plane is:

$$\begin{cases} x_p = fc(0) + (x_d(0) + alphaC * x_d(1)) + cc(0) \\ y_p = fc(1)x_d(1) + cc(1) \end{cases}$$

Therefore, the pixel coordinate vector x_{pixel} and the normalized (distorted) coordinate vector x_d are related to each other through the linear equation:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = KK \begin{bmatrix} x_d(0) \\ x_d(1) \\ 1 \end{bmatrix} \quad (3.9)$$

where KK is known as the camera matrix, and defined as follows:

$$KK = \begin{bmatrix} fc(0) & alphaC * fc(0) & cc(0) \\ 0 & fc(1) & cc(1) \\ 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

The proposed system uses two cameras in the calibration stage instead of one camera to increase the accuracy of our system. Adding the second camera will not increase the cost of the system. Because there are many systems such as [26] and [64] which use two cameras in the 3D reconstruction stage, but they do not use them in the calibration stage. In the 3D reconstruction stage, every camera with the projector will reconstruct parts of the scanned object which are not seen from the other camera. Merging these parts together will reconstruct the whole object in only few scans.

Algorithm 3.4.1: TRIANGULATE($CamRPoints$, $CamLPoints$, PR , PL)

comment: Calculate the set of 3D points given 2D matches and two projection matrices

Input $\left\{ \begin{array}{l} CamRPoints: \text{The 2D points in the right camera} \\ CamLPoints: \text{The 2D points in the left camera} \\ PR: \text{3x4 right camera projection matrix} \\ PL: \text{3x4 left camera projection matrix} \end{array} \right.$

Output $\left\{ \begin{array}{l} Reconstructed3D : \text{The set of 3D points} \end{array} \right.$

Set NumPoints equals to the size of CamRPoints

Set i equals 1

comment: x_{Rx} is the x component of x_R , x_{Ry} is the y component of x_R , x_{Lx} is the x component of x_L and x_{Ly} is the y component of x_L

foreach point x_R in CamRPoints and its correspondence x_L in CamLPoints **do**

comment: first compute the 4x4 matrix J such that $JX = 0$

Set J(1,1) = PR(3,1) * x_{Rx} - PR(1,1); **Set** J(1,2) = PR(3,2) * x_{Rx} - PR(1,2);

Set J(1,3) = PR(3,3) * x_{Rx} - PR(1,3); **Set** J(1,4) = PR(3,4) * x_{Rx} - PR(1,4);

Set J(2,1) = PR(3,1) * x_{Ry} - PR(2,1); **Set** J(2,2) = PR(3,2) * x_{Ry} - PR(2,2);

Set J(2,3) = PR(3,3) * x_{Ry} - PR(2,3); **Set** J(2,4) = PR(3,4) * x_{Ry} - PR(2,4);

Set J(3,1) = PL(3,1) * x_{Lx} - PL(1,1); **Set** J(3,2) = PL(3,2) * x_{Lx} - PL(1,2);

Set J(3,3) = PL(3,3) * x_{Lx} - PL(1,3); **Set** J(3,4) = PL(3,4) * x_{Lx} - PL(1,4);

Set J(4,1) = PL(3,1) * x_{Ly} - PL(2,1); **Set** J(4,2) = PL(3,2) * x_{Ly} - PL(2,2);

Set J(4,3) = PL(3,3) * x_{Ly} - PL(2,3); **Set** J(4,4) = PL(3,4) * x_{Ly} - PL(2,4);

Set [U, S, V] = Singular Value Decomposition(J);

Set X_i the last column of V

Increment i

End

comment: Normalize the four values of every element in X

foreach value P in X **do**

Set $P_1 = P_1/P_4$

Set $P_2 = P_2/P_4$

Set $P_3 = P_3/P_4$

Add P to Reconstructed3D

End

Return Reconstructed3D

End Algorithm

Algorithm 3.4.2: PROJECTOR CALIBRATION($PR, PL, NumImages$)

comment: Calibrate a projector and estimate the intrinsic parameters (KK, fc, cc, kc, and alphaC)

Input {
PR: 3x4 right camera projection matrix
PL: 3x4 left camera projection matrix
NumImages: The number of correspondence images(left-right)

Output {
fc: Focal length
cc: Principal point
kc: Distortion coefficients
alphaC: Skew coefficient
KK: Camera matrix

for i=1 to NumImages **do**

comment: The first step: get the left/right 2D correspondence points

Read the i^{th} right image in RightImage

Set CamRPoints = Extract the Corners of RightImage

Read the i^{th} left image in LeftImage

Set CamLPoints= Extract the Corners of LeftImage

Set X_i = triangulate(CamRPoints,CamLPoints,PR,PL)

End

comment: Get the 2D points of the projected checkerboard in projector frame

Read the projected checkerboard image in ProjectedImage

Set ProjectedPoints = Extract the Corners of ProjectedImage

comment: Calibration Step

Using Zhangs Calibration Method

The initialization step:

Compute a closed-form solution for the calibration parameters

The nonlinear optimization step:

Minimize the total reprojection error (in the least squares sense) over all the calibration parameters

End Algorithm

3.5 Camera Calibration

Camera calibration is one of the main steps in the proposed 3D scanner. This section describes how to calibrate the camera and get its intrinsic and extrinsic parameters using the Camera Calibration Toolbox for MATLAB.

The intrinsic parameters are estimated by viewing several images of a planar checkerboard with various poses as shown in Fig (3.10). After corner extraction, the toolbox runs the main camera calibration. Calibration is done in two steps: first initialization, and then nonlinear optimization. The initialization step computes a closed-form solution for the calibration parameters not including any lens distortion. The non-linear optimization step minimizes the total reprojection error (in the least squares sense) over all the calibration parameters. The optimization is done by iterative gradient descent with an explicit (closed-form) computation of the Jacobian matrix. Figure 3.11 shows the camera calibration results and its reprojection error.

The extrinsic parameters (rotation matrix and translation vector) which determine the position and orientation of the camera co-ordinate system with respect to the world co-ordinate system can be computed using the `compute_extrinsic` function in the Camera Calibration Toolbox. This function computes the extrinsic parameters attached to a 3D structure (4 3D points) given its projection on the image plane and the intrinsic camera parameters (f_c , c_c and k_c).

Figure 3.12 shows the image used in computing the extrinsic parameters. The four checkerboard markers are the 3D structure. The lower-left checkerboard marker is the

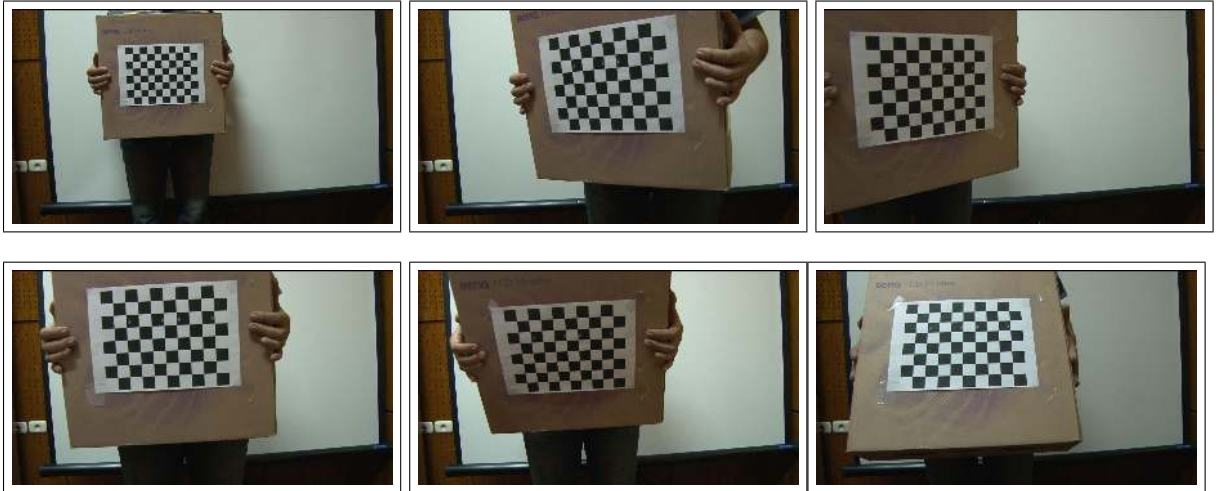


Figure 3.10: *Camera Calibration Sequence Containing Multiple Views of a Checkerboard at Different Positions and Orientations.*

world coordinate origin with values $(0,0,0)$. First, the four checkerboard markers are clicked counter-clockwise starting from the lower left one. The 2D values of the markers are computed using the corner extraction engine. The 3D values are calculated according to the horizontal and the vertical distances between the checkerboard markers. In the proposed 3D scanner, the horizontal and vertical distances are equal to 695 (mm) and 493 (mm) respectively. So the 3D values are Lower-Left $(0, 0, 0)$, Lower-Right $(695, 0, 0)$, Upper-Right $(695, 493, 0)$ and Upper-Left $(0, 493, 0)$.

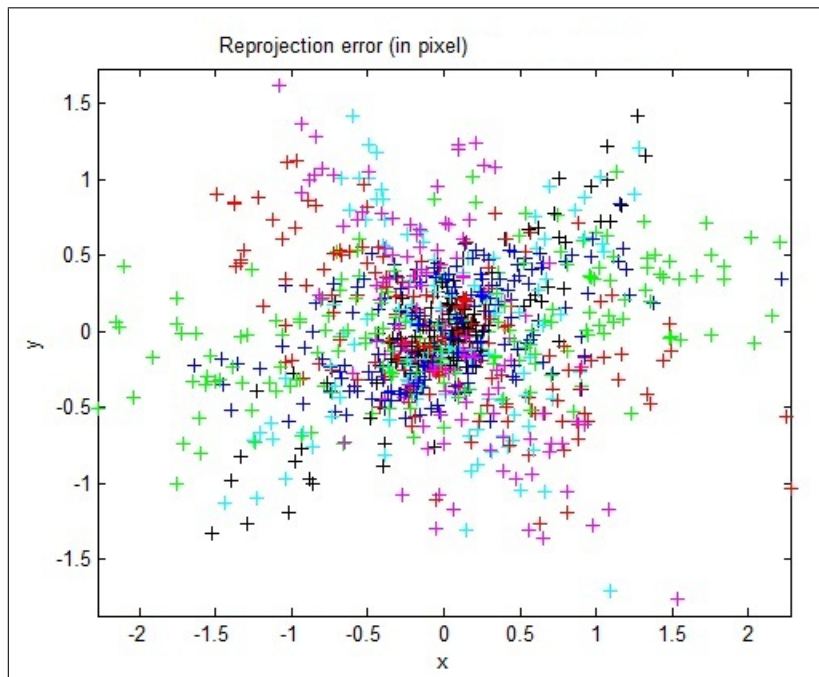
3.6 3D Point Cloud Reconstruction

The decoded set of camera and projector correspondences can be used to reconstruct a 3D point cloud. Several reconstruction schemes can be implemented using the sequences described in Section 3.2.2. The projector column correspondences can be used to reconstruct a point cloud using ray plane triangulation as shown in Fig (3.13a). A second point cloud can be reconstructed using the projector row correspondences. Finally, the projector pixel to camera pixel correspondences can be used to reconstruct the point cloud using ray-ray

Calibration results after optimization:

Focal Length :	fc = [3666.74739 3659.77596]
Principal point :	cc = [926.61547 482.31001]
Skew:	alpha_c = [0.0000]
Distortion:	kc = [0.24564 0.69784 -0.01266 -0.0039 0.0000]
Pixel error:	err = [0.68536 0.47711]

(a)



(b)

Figure 3.11: (a) *The Camera Calibration Results (Intrinsic Parameters)*. (b) *Camera Reprojection Error*.

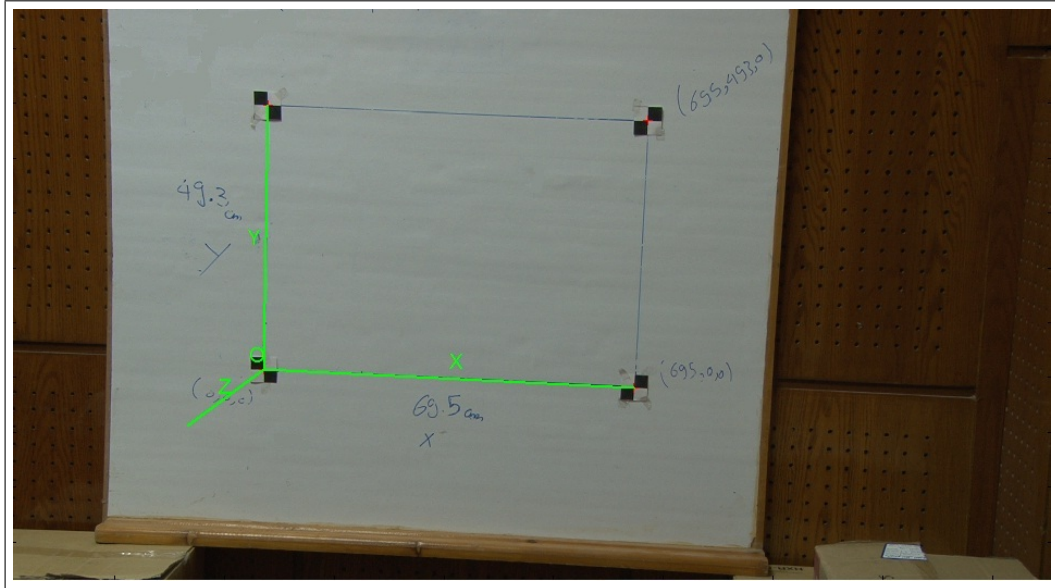


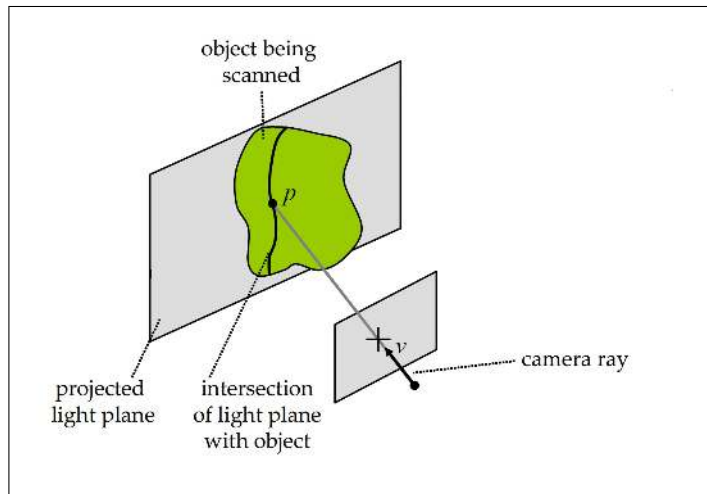
Figure 3.12: *The World Coordinate System.*

triangulation (i.e., by finding the closest point to the optical rays defined by the projector and camera pixels) as shown in Fig (3.13b). Figure 3.13c shows how the approximate intersection of two rays is defined.

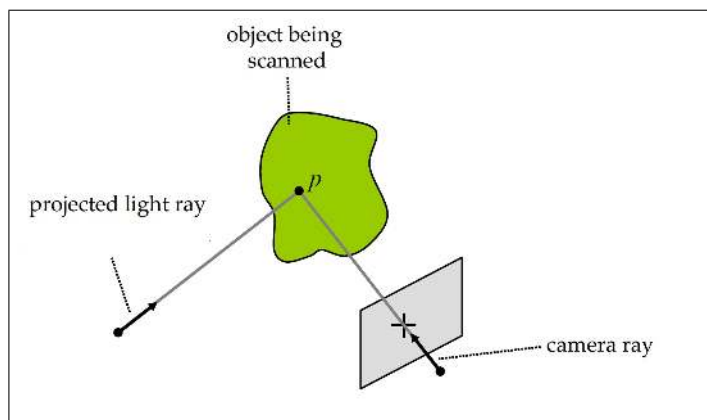
Given the transformation matrix, which relates an object point with its projection on the captured image and on the projector image, respectively, and the 2D co-ordinates of the two projections of the same 3D object point, the co-ordinate of this object point can be computed as follows.

Let the 3D object point co-ordinate be

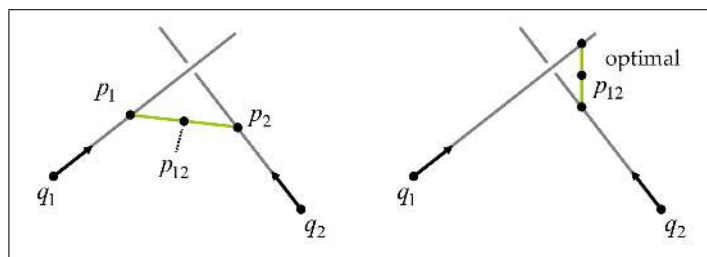
$$\begin{pmatrix} X_{pw} \\ Y_{pw} \\ Z_{pw} \end{pmatrix},$$



(a)



(b)



(c)

Figure 3.13: *Reconstruction by Triangulation. (a) Triangulation by Line-Plane Intersection. (b) Triangulation by Line-Line Intersection. (c) The Approximate Intersection of Two Rays.*

its projection on the projector image and on the captured image, respectively are

$$x_1 = \begin{pmatrix} xp_1 \\ yp_1 \\ 1 \end{pmatrix}, x_2 = \begin{pmatrix} xp_2 \\ yp_2 \\ 1 \end{pmatrix},$$

and the two projection matrices of the projector (P_1) and the camera (P_2) are

$$P_1 = \begin{pmatrix} P_{111} & P_{112} & P_{113} & P_{114} \\ P_{121} & P_{122} & P_{123} & P_{124} \\ P_{131} & P_{132} & P_{133} & P_{134} \end{pmatrix}, P_2 = \begin{pmatrix} P_{211} & P_{212} & P_{213} & P_{214} \\ P_{221} & P_{222} & P_{223} & P_{224} \\ P_{231} & P_{232} & P_{233} & P_{234} \end{pmatrix},$$

then

$$\begin{pmatrix} x_{p1} \\ y_{p1} \\ 1 \end{pmatrix} = \begin{pmatrix} P_{111} & P_{112} & P_{113} & P_{114} \\ P_{121} & P_{122} & P_{123} & P_{124} \\ P_{131} & P_{132} & P_{133} & P_{134} \end{pmatrix} \begin{pmatrix} Xp_W \\ Yp_W \\ Zp_W \\ 1 \end{pmatrix} \quad (3.11)$$

and

$$\begin{pmatrix} x_{p2} \\ y_{p2} \\ 1 \end{pmatrix} = \begin{pmatrix} P_{211} & P_{212} & P_{213} & P_{214} \\ P_{221} & P_{222} & P_{223} & P_{224} \\ P_{231} & P_{232} & P_{233} & P_{234} \end{pmatrix} \begin{pmatrix} Xp_W \\ Yp_W \\ Zp_W \\ 1 \end{pmatrix} \quad (3.12)$$

After multiplying and rearranging the variables, the following equations can be obtained

$$(P_{111} - P_{131}x_{p1}) Xp_W + (P_{112} - P_{132}x_{p1}) Yp_W + (P_{113} - P_{133}x_{p1}) Zp_W = P_{134}x_{p1} - P_{114} \quad (3.13)$$

$$(P_{121} - P_{131}y_{p1}) Xp_W + (P_{122} - P_{132}y_{p1}) Yp_W + (P_{123} - P_{133}y_{p1}) Zp_W = P_{134}y_{p1} - P_{124} \quad (3.14)$$

$$(P_{211} - P_{231}x_{p2}) X_{pW} + (P_{212} - P_{232}x_{p2}) Y_{pW} + (P_{213} - P_{233}x_{p2}) Z_{pW} = P_{234}x_{p2} - P_{214} \quad (3.15)$$

$$(P_{221} - P_{231}y_{p2}) X_{pW} + (P_{222} - P_{232}y_{p2}) Y_{pW} + (P_{223} - P_{233}y_{p2}) Z_{pW} = P_{234}y_{p2} - P_{224} \quad (3.16)$$

So, arranging the 2D point co-ordinates and the object point co-ordinates in matrix form, the relation can be expressed as,

$$PN = F \quad (3.17)$$

where the matrices are

$$P = \begin{pmatrix} P_{111} - P_{131}x_{P1} & P_{112} - P_{132}x_{P1} & P_{113} - P_{133}x_{P1} \\ P_{121} - P_{131}y_{P1} & P_{122} - P_{132}y_{P1} & P_{123} - P_{133}y_{P1} \\ P_{211} - P_{231}x_{P2} & P_{212} - P_{232}x_{P2} & P_{213} - P_{233}x_{P2} \\ P_{221} - P_{231}y_{P2} & P_{222} - P_{232}y_{P2} & P_{223} - P_{233}y_{P2} \end{pmatrix} \quad (3.18)$$

$$N = \begin{pmatrix} X_{pW} \\ Y_{pW} \\ Z_{pW} \end{pmatrix} \quad (3.19)$$

$$F = \begin{pmatrix} P_{134}x_{P1} - P_{114} \\ P_{134}y_{P1} - P_{124} \\ P_{234}x_{P2} - P_{214} \\ P_{234}y_{P2} - P_{224} \end{pmatrix} \quad (3.20)$$

Finally, the 3D object co-ordinate can be obtained by

$$N = (P^t P)^{-1} P^t F \quad (3.21)$$

Also, the 3D object point can be calculated using SVD (Singular Value Decomposition). As the object co-ordinates depend on the correct association of the projected point

(x_{p1}, y_{p1}) and the captured image point (x_{p2}, y_{p2}) , any mistake in the correspondence establishment leads to an error in the object point co-ordinates determination. A simple per-point RGB color can be assigned by sampling the color of the all-white camera image for each 3D point. Reconstruction artifacts can be further reduced by comparing the reconstruction produced by each of these schemes to get better results.

Algorithm 3.6.1: 3D RECONSTRUCTION(x_1, x_2, PR, PL)

comment: Reconstruct a 3D point X from two 2D correspondence points(x_1, x_2)

Input $\left\{ \begin{array}{l} x_1: \text{The 2D points in the right image} \\ x_2: \text{The 2D points in the left image} \\ PR: \text{3x4 right camera projection matrix} \\ PL: \text{3x4 left camera(projector) projection matrix} \end{array} \right.$

Output $\left\{ \begin{array}{l} X : \text{The 3D points reconstructed from } x_1 \text{ and } x_2 \end{array} \right.$

Set x_{p1} equals to the x component of x_1

Set y_{p1} equals to the y component of y_1

Set x_{p2} equals to the x component of x_2

Set y_{p2} equals to the y component of y_2

comment: P_{Rjk} is the j^{th} row and k^{th} column element in P_R

comment: P_{Ljk} is the j^{th} row and k^{th} column element in P_L

Set P = [

$$\begin{array}{lll} P_{R11} - P_{R31}x_{P1} & P_{R12} - P_{R32}x_{P1} & P_{R13} - P_{R33}x_{P1}; \\ P_{R21} - P_{R31}y_{P1} & P_{R22} - P_{R32}y_{P1} & P_{R23} - P_{R33}y_{P1}; \\ P_{L11} - P_{L31}x_{P2} & P_{L12} - P_{L32}x_{P2} & P_{L13} - P_{L33}x_{P2}; \\ P_{L21} - P_{L31}y_{P2} & P_{L22} - P_{L32}y_{P2} & P_{L23} - P_{L33}y_{P2}] \end{array}$$

Set F = [

$$\begin{array}{l} P_{R34}x_{P1} - P_{R14}; \\ P_{R34}y_{P1} - P_{R24}; \end{array}$$

$$P_{L34}x_{P2} - P_{L14};$$

$$P_{L34}y_{P2} - P_{L24}]$$

Set X = inverse(transpose(P)*P)*transpose(P)*F

Return X

End Algorithm

3.7 Results and Performance

This section discusses the results obtained when reconstructing objects using the proposed 3D scanner. It consists of two subsections. Subsection 3.7.1 discusses the results of the proposed projector calibration method and subsection 3.7.2 discusses the 3D scanner results.

3.7.1 Projector Calibration Results

This section shows the results obtained when calibrating the projector using the proposed method. The setup used for the tests was the same one used for the calibration step as in figure 3.7b. It consists of a data projector (Optoma EH020) with a resolution of 1024×768 pixels, two cameras (Sony nxcam) and a frame grabber (DeckLink Studio) digitizing images at 1920×1080 pixels with 24 bits per pixel (RGB). The method run on an Intel Core2 Duo CPU at 3.00GHz.

In order to measure the performance of the proposed method, the reprojection error function is used. The reprojection error is a geometric error corresponding to the image distance between a projected point and a measured one. It is used to quantify how closely an estimate of a 3D point X recreates the point's true projection x_1 . More precisely, let P be the projection matrix of a camera and x_2 be the image projection of X , i.e. $x_2 = PX$. The reprojection error of X is given by $d(x_1, x_2)$, where $d(x_1, x_2)$ denotes the Euclidean distance between the image points represented by vectors x_1 and x_2 .

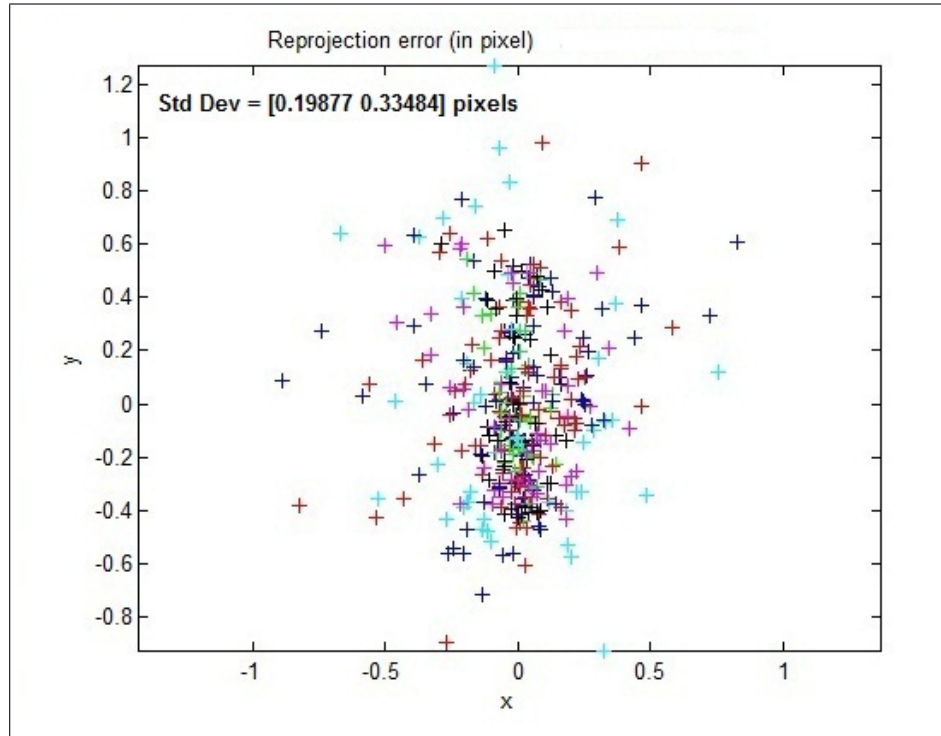
Table 3.1: The Standard Deviation of Error (StdDev) of The Coordinates of The Points

Pattern No.	StdDev of the proposed method (mm)	StdDev of Sergio Fernandez et al. method (mm)
1	0.0904	0.1213
2	0.1001	0.1295
3	0.0881	0.1303
4	0.0894	0.1311
5	0.0931	0.1341
Average	0.0922	0.1292

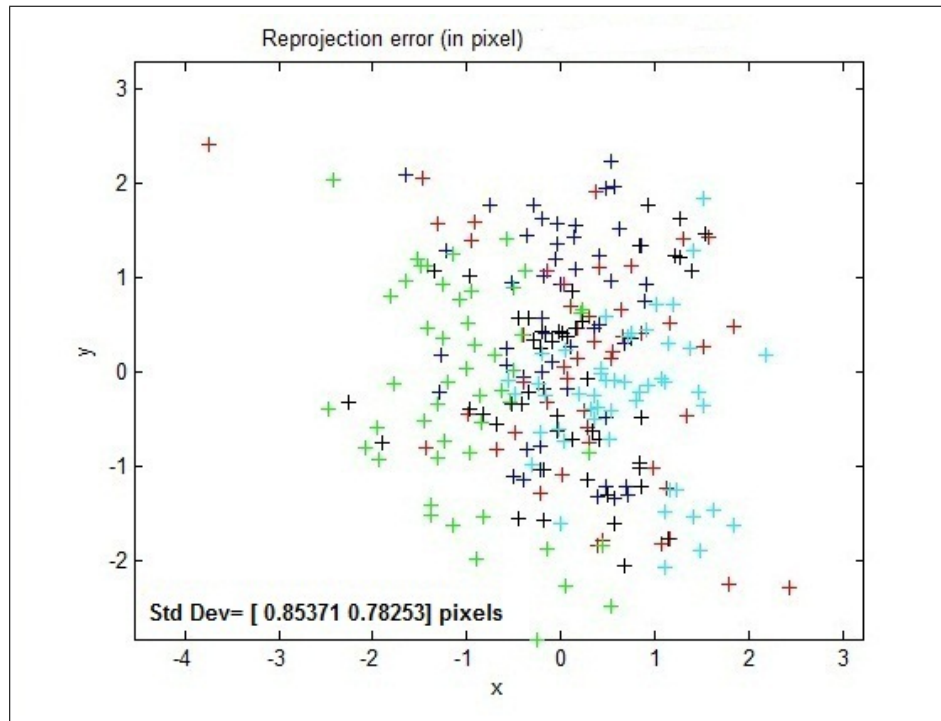
The reprojection error function uses the 3D calibration pattern points (49 3D checkerboard corners) calculated using triangulation, the correspondence 2D points calculated using the corner detection algorithm (the measured 49 2D points), and the estimated projector calibration parameters. The 49 3D points are reprojected on the projector frame using the estimated projector calibration parameters. The reprojection error function is the difference in pixels between the reprojected 2D points (the projected 49 2D points) and the 2D points calculated using the corner detection algorithm (the measured 49 2D points). The error value is represented by a cross(+). Every 49 crosses with the same color represent the errors of one calibration pattern points.

As it can be seen from figures (3.14a and 3.14b), the proposed method is more accurate than Sergio Fernandez et al. method [30]. The proposed method's average standard deviation of the error is [0.19877 0.33484] pixels while it is [0.85371 0.78253] pixels for Sergio Fernandez et al. method [30].

Distance measurement is carried out to evaluate the performance of the proposed calibration method. An 8×8 checkerboard pattern (49 corners) is projected on the whiteboard. The 3D coordinates corresponding to these corners calculated using both camera-camera triangulation and camera-projector triangulation are obtained. The whiteboard is moved in five different positions and in every position the 3D coordinates of the pattern corners are calculated. Table 3.1 shows results obtained by calibrating the projector using the pro-



(a)



(b)

Figure 3.14: (a) *Reprojection Error of The Proposed Method.* (b) *Reprojection Error of Sergio Fernandez et al. [30] Method.*

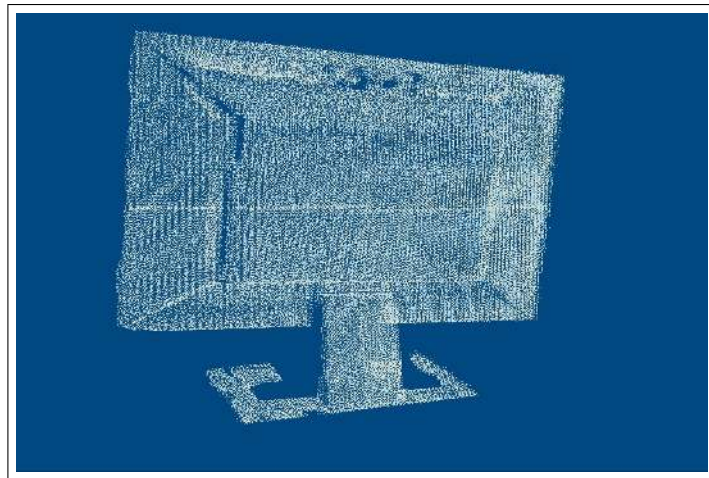
posed method compared with the results obtained by calibrating the projector using Sergio Fernandez et al. method [30].

3.7.2 3D Scanner Results

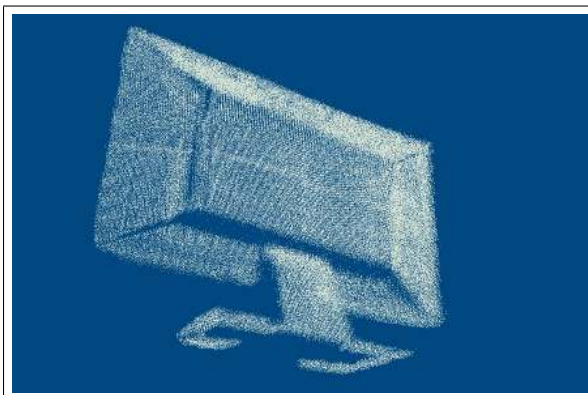
This section shows the results obtained when testing the 3D scanner which is based on the proposed projector calibration method discussed in section 3.4. First, in order to demonstrate the applicability and efficiency of the proposed projector calibration method, 3D reconstruction of three different objects using the proposed 3D scanner has been performed as shown in Fig(3.15 - 3.17). Second, two 3D human faces in the form of 3D point cloud are reconstructed using the proposed 3D scanner as shown in Fig(3.18 and 3.19). Figure (3.20) shows the first person's reconstructed face in surface form. The quality of the reconstructed faces is good.



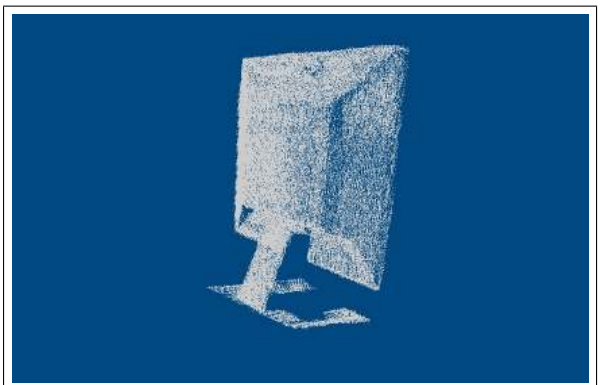
(a)



(b)



(c)

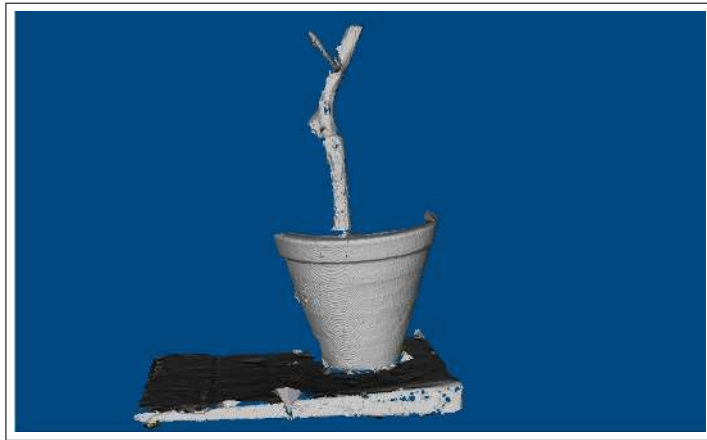


(d)

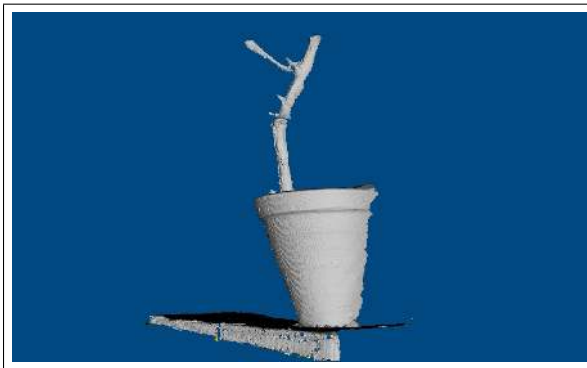
Figure 3.15: (a) The LCD screen to be scanned. (b,c,d) Three views for the reconstructed 3D cloud of points.



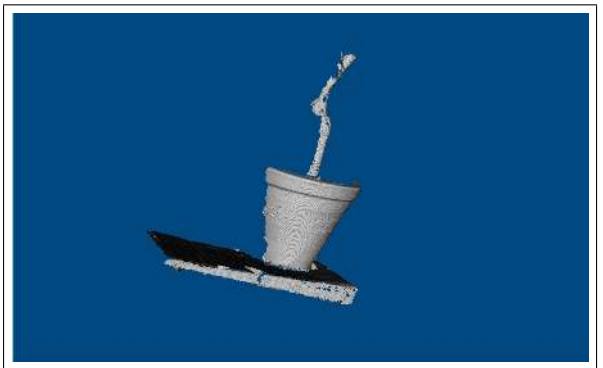
(a)



(b)

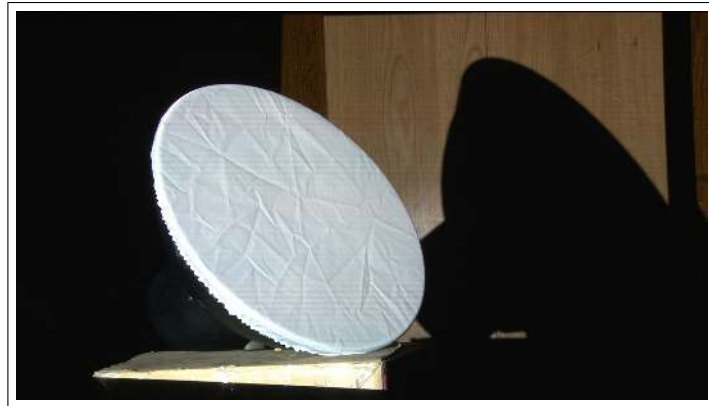


(c)

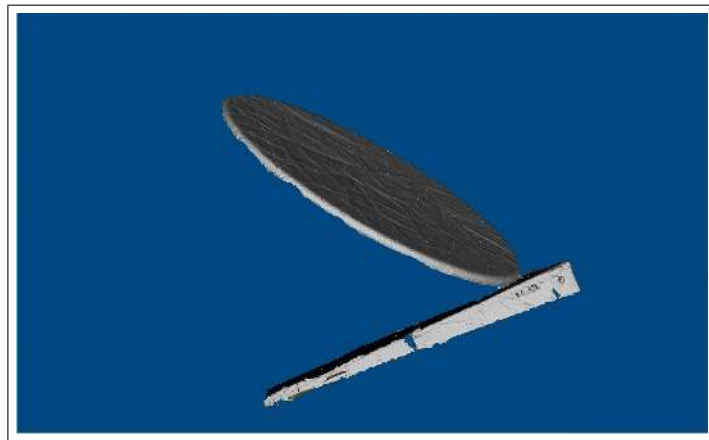


(d)

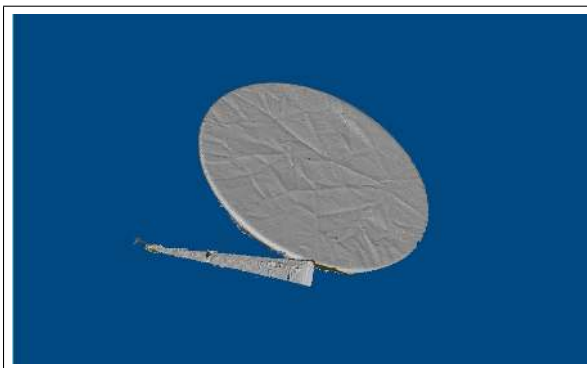
Figure 3.16: (a) The flower vase to be scanned. (b,c,d) Three views for the reconstructed surfaces.



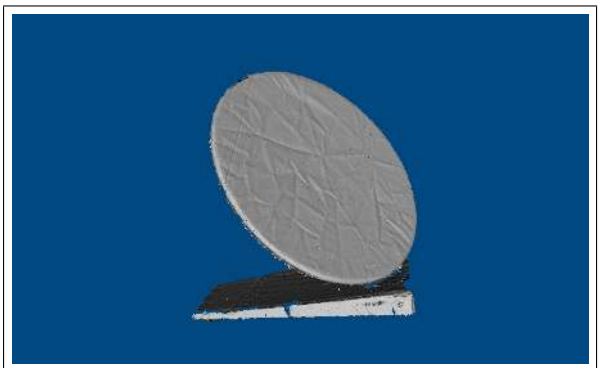
(a)



(b)

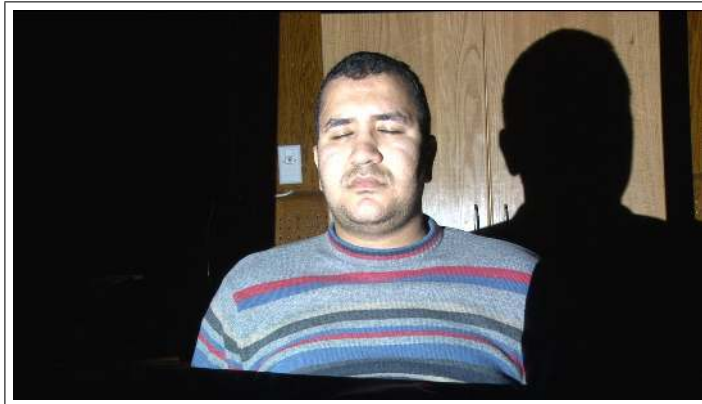


(c)

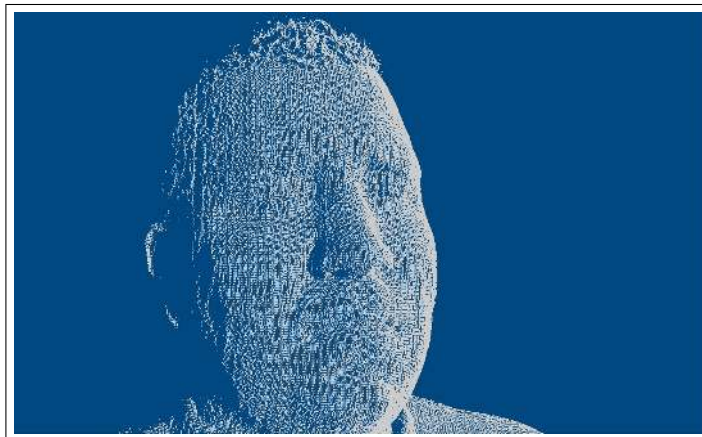


(d)

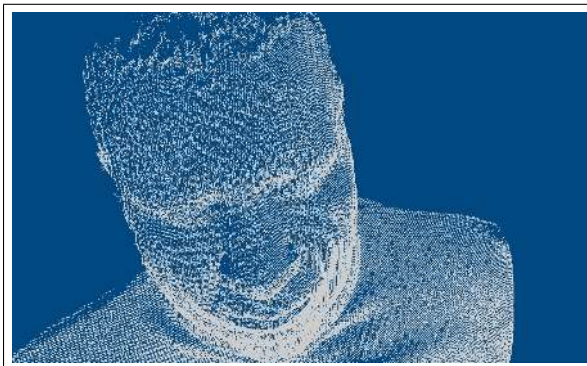
Figure 3.17: (a) The camera lamp holder to be scanned. (b,c,d) Three views for the reconstructed surfaces.



(a)



(b)



(c)

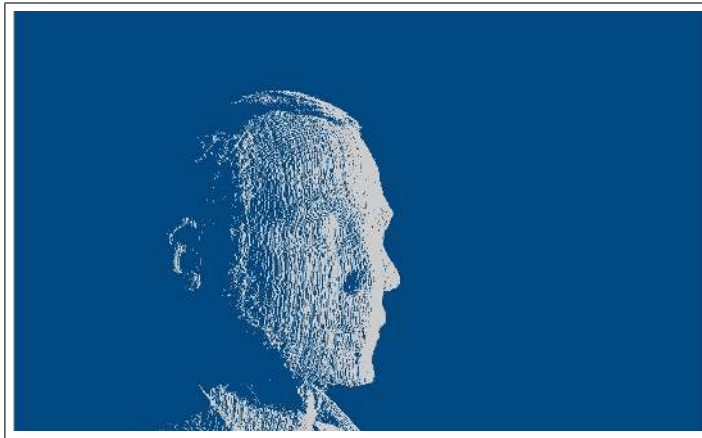


(d)

Figure 3.18: *Reconstructed faces in the form of 3D cloud of points:(a) The person real face. (b,c,d) Three views for the reconstructed 3D cloud of points.*



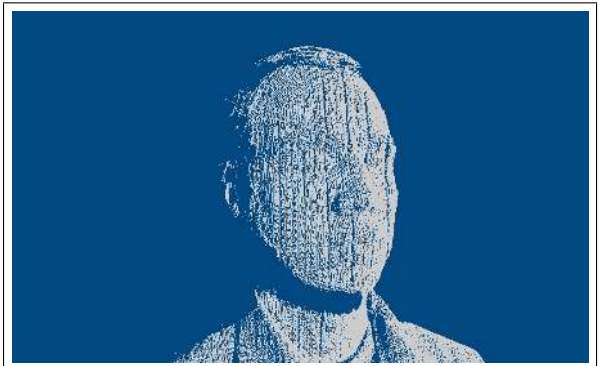
(a)



(b)



(c)



(d)

Figure 3.19: *Reconstructed face in the form of 3D cloud of points:(a) The person real face. (b,c,d) Three views for the reconstructed 3D cloud of points.*

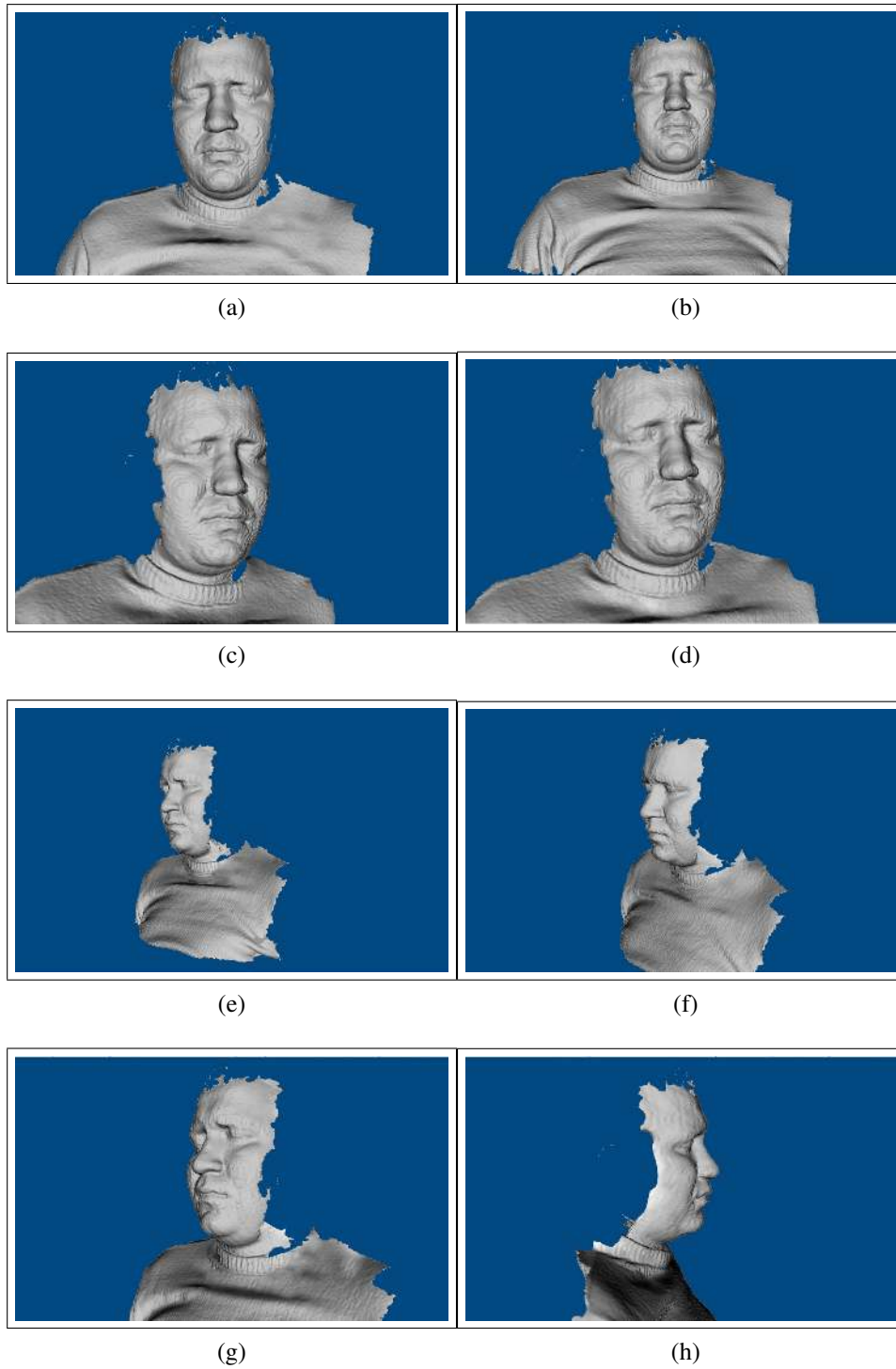


Figure 3.20: *Reconstructed Face in The Form of Surfaces.*

Chapter 4

Face Animation Using Facial Motion Capture System

4.1 Introduction

Facial Motion Capture is the process of electronically converting the movements of a person's face into a digital data using cameras or laser scanners. This data may then be used to produce CG (computer graphics) computer animation for movies, games, or real-time avatars. Because the motion of CG characters is derived from the movements of real people, it results in more realistic and natural computer character animation than if the animation were created manually. This chapter describes how to use the OptiTrack Expression Capturing System [54] to generate motion capture data and animate the human faces reconstructed in chapter 3 accordingly.

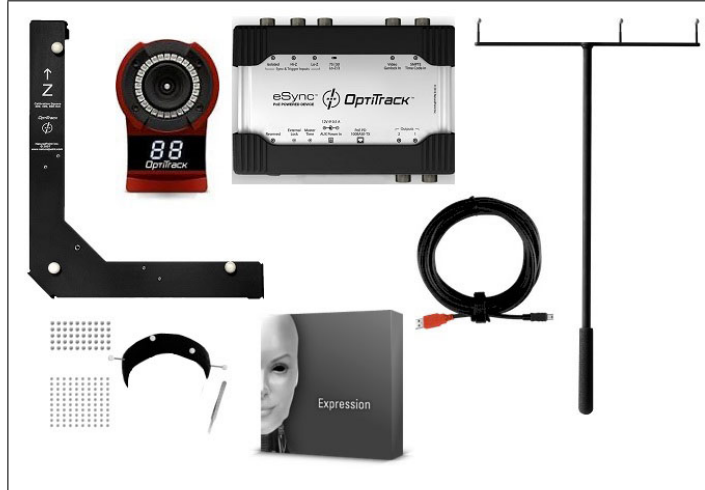


Figure 4.1: *OptiTrack Expression Capturing System.*

4.2 System Setup

4.2.1 System Overview

This section describes in details the OptiTrack Expression Capturing System. This system is a part of the multimedia lab in the faculty of Computers and Information at Assiut University. It consists of seven cameras, two hubs, cables, Facial Marker Set, Three Markers Wand , Calibration Square and Arena Expression software as shown in Fig(4.1). Cameras should be placed around the capture volume so that the markers will be within view of at least two cameras at all times. Place three cameras in a semicircle above head height and three cameras in a semicircle at chest level. All six cameras should be pointed at the individual's nose, about two feet from his/her face. The last camera will be used as A/V camera as shown in Fig(4.2). After positioning the cameras, they must be calibrated.

4.2.2 System Calibration

The calibration engine in Arena software can be used to calibrate the system. The cameras should not move once calibrated. If any camera is moved, tracking accuracy will be affected



Figure 4.2: *Our System Overview.*

and the cameras should be recalibrated. First, the cameras can be selected from the Camera menu. The calibration engine wizard starts from Wizard menu. The wizard will ask for the calibration type as shown in Fig(4.3).

There are three calibration types:

- 1) "Full Calibration" to capture new calibration data and make new calibration.
- 2) "Previously Recorded data" to calibrate using old calibration data.
- 3) "Set Ground Plane". Setting ground plane will be explained later.

After choosing the calibration type, the cameras must be adjusted to remove or block unwanted markers. The calibration step consists of waving the 3-Marker calibration wand in the capture volume. While waving the wand, every camera records the 3-markers positions as shown in Fig(4.4). When the overall quality becomes "very high", the calculation step can be started. The final step in the calibration is "Set Ground Plane". This step determines the z-axis of the system. The calibration square with its 3-markers is captured by the cameras. Make the z-axis towards the cameras as shown in Fig(4.5a). After finishing calibration, the capture volume preview will be displayed as shown in Fig(4.5b).

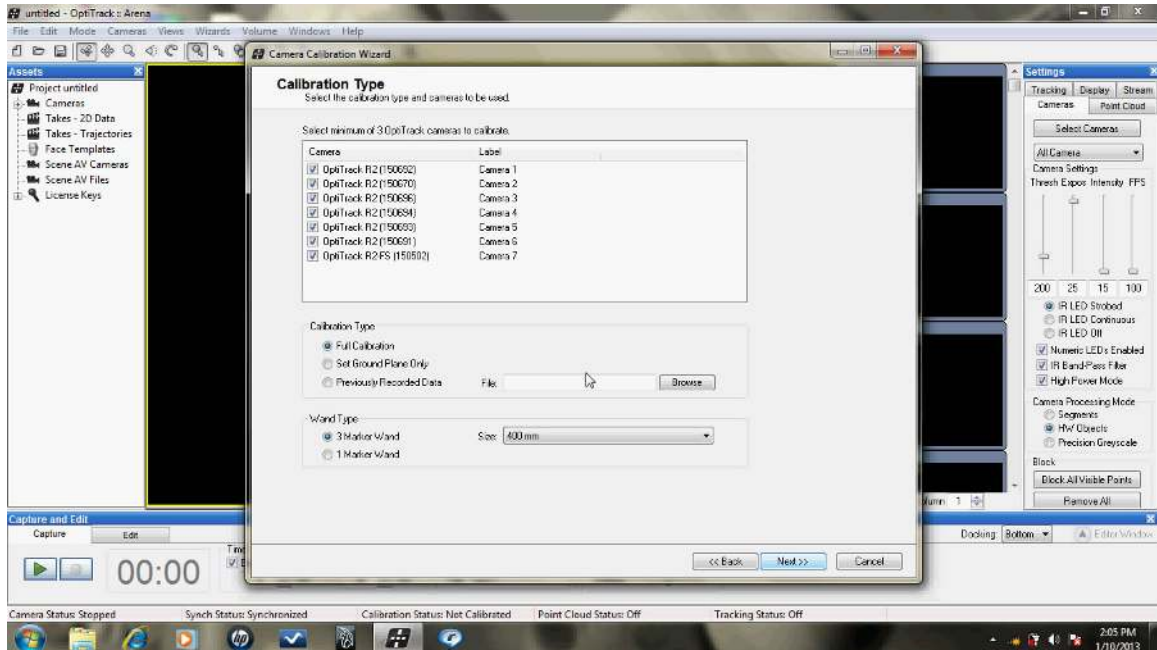


Figure 4.3: *Calibration Types.*

4.3 Facial Expression Capturing

This section describes how to use the OptiTrack system to get facial expression data. The output of this step is an FBX or a C3D files containing the positions of the animated markers.

4.3.1 Apply Face Markers

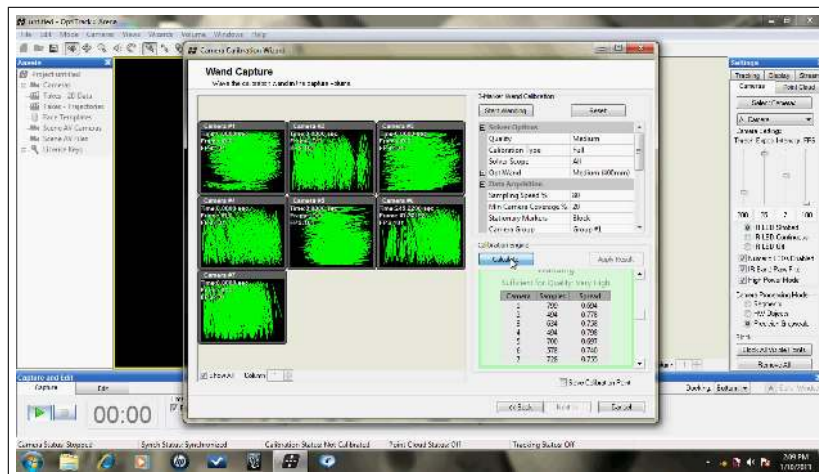
Arena expression software allows us to choose how many markers the animator wants to track. Figure 4.6 shows the marker placement diagram. Fewer number of markers can be used or marker positions can be changed according to the animator need.

4.3.2 Facial Capture Wizard

From Wizards menu, choose the Facial Capture Wizard. This wizard helps us create a face template from marker point data. There are three facial template types in Arena:

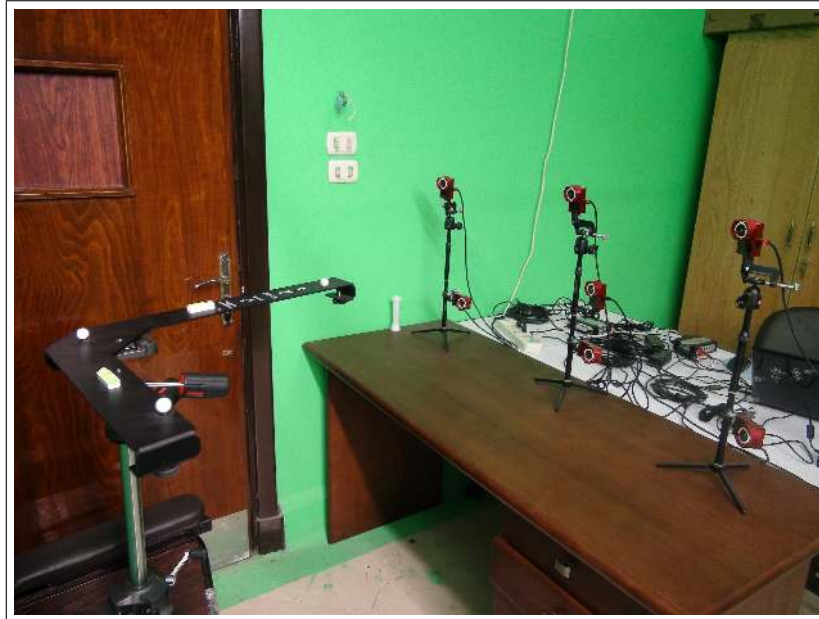


(a)

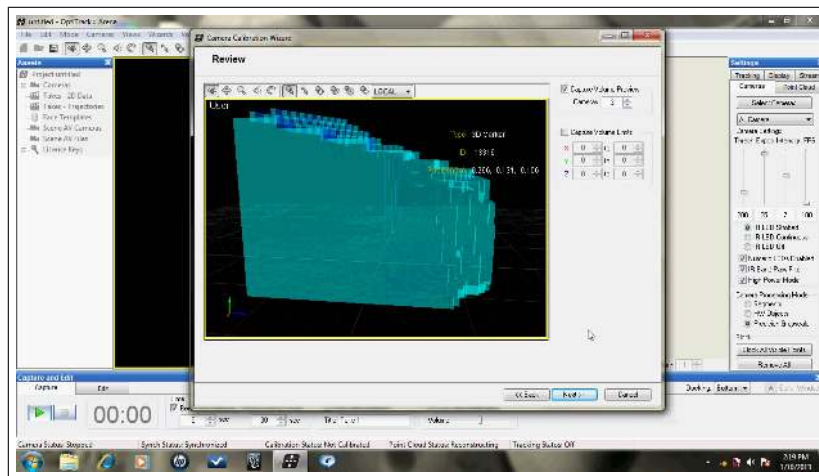


(b)

Figure 4.4: (a) The Waving Step. (b) The Recorded Data



(a)



(b)

Figure 4.5: (a) Setting The Ground Plane. (b) The Captured Volume.

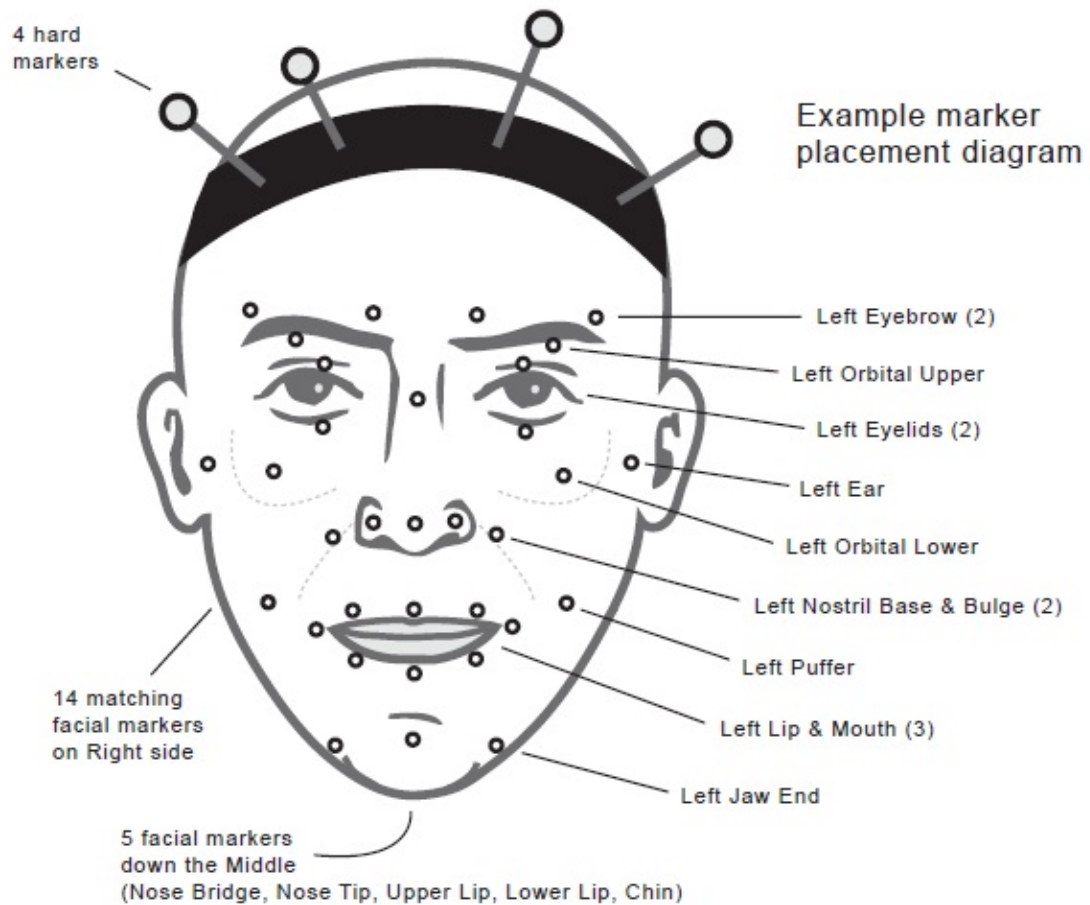


Figure 4.6: *The Marker Placement Diagram*[54].

- 1) 23 face markers + 4 or more head markers.
- 2) 33 face markers + 4 or more head markers.
- 3) 37 face markers + 4 or more head markers.
- 4) Custom.

The fourth template type allows us to choose the number of markers and their positions according to the required animation. Figure 4.7 shows the positions of the markers on the actor face and on the face template simultaneously. After creating the template, you can start recording the animation. From the capture panel, click the button "Start recording a

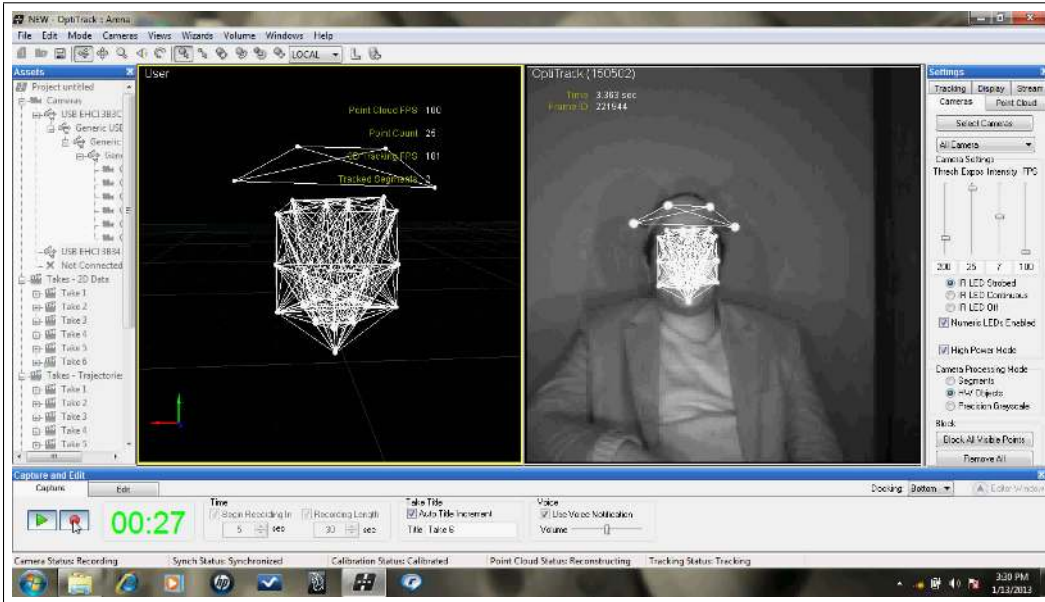


Figure 4.7: *The Face Template.*

take”. The take’s recording length can be determine. Figure 4.8 shows the output of the A/V camera for a take.

After the take, which is a 2D data, is recorded, it must be trajectorized. Trajectorization means convert the 2D data takes using the calibration data into 3D data takes. These takes can be exported either as FBX or C3D files which can be imported in any mesh editing software like Autodesk 3ds Max or Maya. Figure 4.9 shows the FBX data imported in 3ds Max. This data - marker positions - can be used to animate a rigged face easily.

4.4 Animation Results

In this section, the results of the 3D face animation step are shown. The facial expressions are captured using the Optitrack Expression Capturing System described in section 4.3. Figure 4.10 shows the face animated with different expressions.



Figure 4.8: A Take Captured by The A/V Camera.

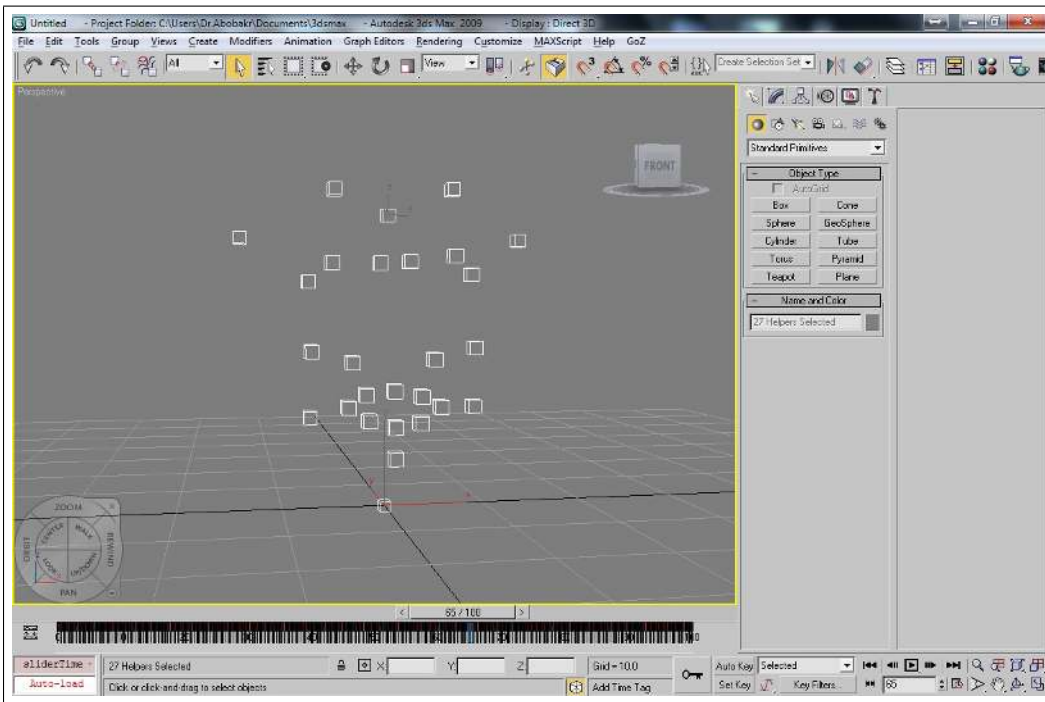


Figure 4.9: The FBX Data File Imported in 3Ds Max.

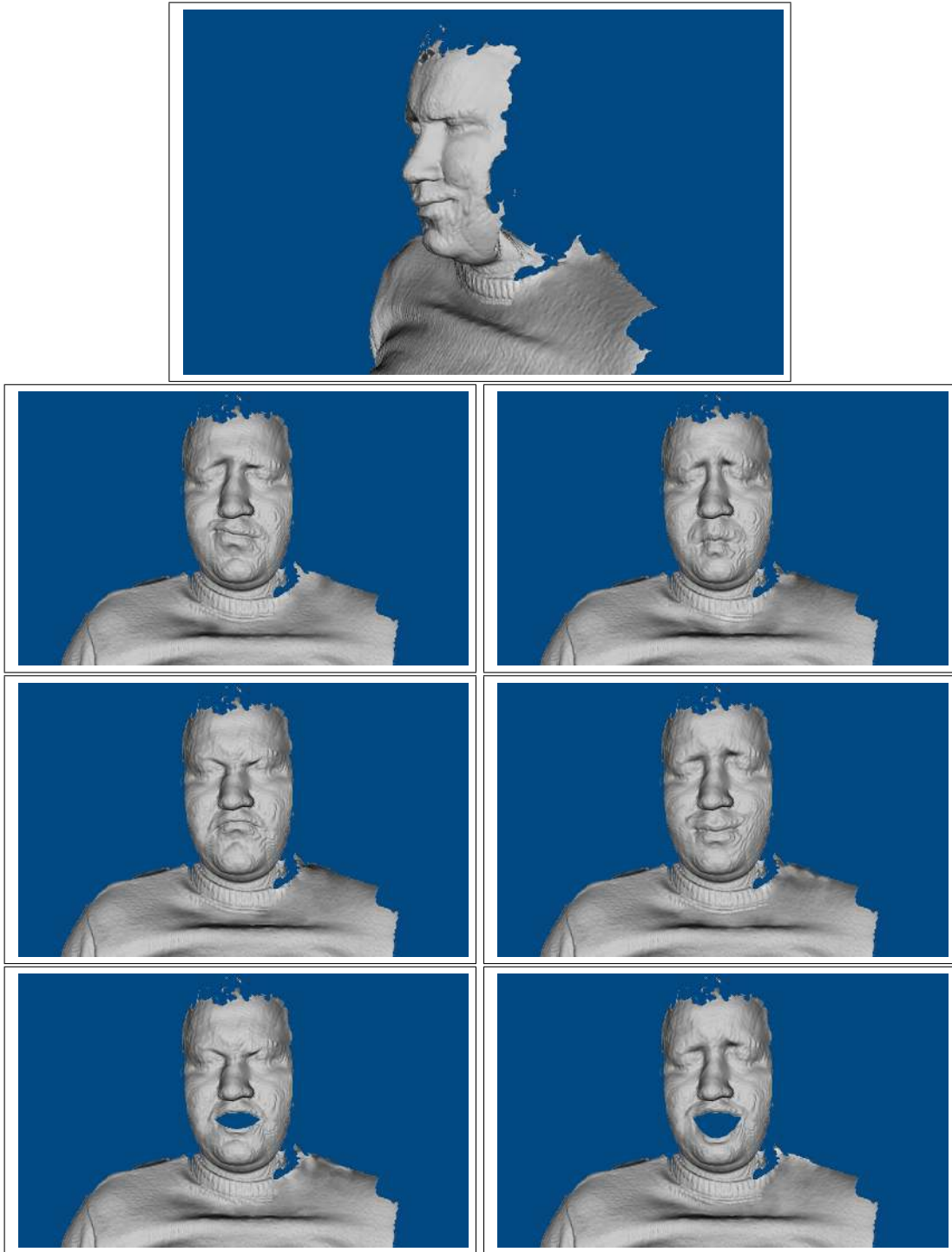


Figure 4.10: *The Reconstructed Face After Animation.*

Chapter 5

Conclusions and Future Works

5.1 Conclusions

In this thesis, 3D face is reconstructed using structured lighting. Also, an easy and accurate projector calibration method which based on passive stereo and triangulation is described. The simplicity of the method comes from considering the projector as an inverse camera and thus making the calibration of a projector the same as that of a camera for which there already exists well and accurate established methods. Projector calibration is more complicated than camera calibration because projectors cannot capture the surface that they illuminate, so a camera must be used to make the correspondence between the 2D projected points and the 3D illuminated points. Since the calibrating pattern is projected and not attached to the world coordinate system, it is difficult to retrieve the co-ordinates of the 3D points. Projecting a checkerboard pattern on a white board and capturing it from two points of view helps us to solve the problem and compute the 3D co-ordinates of the projected pattern corners by using Triangulation.

In order to verify the correctness and the accuracy of our calibration method, a simple reconstruction of different objects has been performed. Our method is more accurate than Sergio Fernandez et al. method. The use of two cameras increases the accuracy of our

method. The two cameras will help us in the 3D shape reconstruction stage. Every camera with the projector will reconstruct parts of the scanned object which not seen from the other camera. The whole object will be constructed in only few scans by merging these parts together. So, adding the second camera will not increase the cost of our system. After the reconstruction step, the 3D faces are animated using motion capture system.

5.2 Future Works

We have only scratched the surface of 3D reconstruction, there are still many research directions that we would like to examine in the future:

We would like to design a new color structured light pattern (Spatial Direct Codification). It will be used to reconstruct the face by only one image. So, we will use it to reconstruct the face and capture the animation in the same time. Also, we would like to design a new registration algorithm that will be used to merge many face scans to reconstruct a complete face.

References

- [1] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision". New York, NY, USA: Cambridge University Press, 2 ed., 2003.
- [2] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, "An invitation to 3-D vision: from images to geometric models", vol. 26 of Interdisciplinary Applied Mathematics. Springer, 2004.
- [3] B. Curless, "From range scans to 3D models," ACM SIGGRAPH Computer Graphics, vol. 33, no. 4, pp. 38–41, 1999.
- [4] J. Vanherzeele, P. Guillaume, and S. Vanlanduit, "Fourier fringe processing using a regressive fourier-transform technique," Optics and lasers in engineering, vol. 43, no. 6, pp. 645–658, 2005.
- [5] R. Ahlers and J. A. Lu, "Stereoscopic vision: an application-oriented overview," in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 1194, pp. 298–308, 1990.
- [6] http://en.wikipedia.org/wiki/3D_scanner, Last accessed: March 2013.
- [7] S. T. Barnard and M. A. Fischler, "Computational stereo," ACM Computing Surveys (CSUR), vol. 14, no. 4, pp. 553–572, 1982.
- [8] M. Land and D. Nilsson, "Animal eyes", Oxford University Press, 2004.
- [9] R. Vaillant and O. Faugeras, "Using extremal boundaries for 3-D object modeling," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, pp.157–173, feb 1992.
- [10] Z. A. HARTLEY R. I., "Multiple View Geometry in Computer Vision", Cambridge University Press, 2004.
- [11] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 519 – 528, june 2006.
- [12] F. Blais, "Review of 20 years of range sensor development," Journal of Electronic Imaging, vol. 13, no. 1, pp. 231–240, 2004.
- [13] K. G. Harding and L. H. Bieman, "High-speed moire contouring methods analysis," in Proceedings of SPIE, vol. 3520, p. 27, 1998.
- [14] J. JALKIO, R. KIM, and S. CASE, "Three dimensional inspection using multistripe structured light," Optical engineering, vol. 24, no. 6, pp. 966–974, 1982.
- [15] F. M. Wahl, "A coded light approach for depth map acquisition," in Mustererkennung 1986, pp. 12–17, Springer, 1986.

- [16] J. Salvi, J. Pages, and J. Battle, "Pattern codification strategies in structured light systems," *Pattern Recognition*, vol. 37, no. 4, pp. 827–849, 2004.
- [17] R. A. Jarvis, "A laser time-of-flight range scanner for robotic vision," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 5, pp. 505–512, 1983.
- [18] S. Hsu, S. Acharya, A. Rafii, and R. New, "Performance of a time-of-flight range camera for intelligent vehicle safety applications," *Advanced Microsystems for Automotive Applications 2006*, pp. 205–219, 2006.
- [19] G. IDAN and G. YAHAV, "Three-dimensional imaging in the studio and elsewhere," in *SPIE proceedings series*, vol. 4298, pp. 48–55, Society of Photo-Optical Instrumentation Engineers, 2001.
- [20] S. F. Navarro, "3D reconstruction of object shape using structured light", A Thesis Submitted for the Degree of MSc Erasmus Mundus in Vision and Robotics (VIBOT). University of Girona, 2009.
- [21] J. Battle, E. Mouaddib, and J. Salvi, "Recent progress in coded structured light as a technique to solve the correspondence problem: a survey," *Pattern recognition*, vol. 31, no. 7, pp. 963–982, 1998.
- [22] J. Posdamer and M. Altschuler, "Surface measurement by space-encoded projected beam systems," *Computer Graphics and Image Processing*, vol. 18, no. 1, pp. 1 – 17, 1982.
- [23] S. Inokuchi, K. Sato, and F. Matsuda, "Range imaging system for 3-D object recognition," in *Proceedings of the International Conference on Pattern Recognition*, pp. 806–808, 1984.
- [24] O. Faugeras, "Three dimensional computer vision: A geometric viewpoint", the MIT Press, 1993.
- [25] N. Ayache, "Artificial vision for mobile robots: stereo vision and multisensory perception", The MIT Press, 1991.
- [26] D. Lanman and G. Taubin, "Build your own 3D scanner: 3D photography for beginners," in *ACM SIGGRAPH 2009 Courses, SIGGRAPH '09*, (Louisiana, USA), pp. 8:1–8:94, ACM, 2009.
- [27] Z. Zhang, "A flexible new technique for camera calibration," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, Nov. 2000.
- [28] M. Kimura, M. Mochimaru, and T. Kanade, "Projector calibration using arbitrary planes and calibrated camera," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, (Minneapolis, Minnesota, USA), pp. 1–2, IEEE, June 2007.
- [29] F. Sadlo, T. Weyrich, R. Peikert, and M. Gross, "A practical structured light acquisition system for point-based geometry and texture," In *Proceedings of the Second Eurographics/IEEE VGTC conference on Point-Based Graphics*, pp. 89-98, Eurographics Association, 2005.
- [30] S. Fernandez and J. Salvi, "Planar-based camera-projector calibration," in *2011 7th International Symposium on Image and Signal Processing and Analysis (ISPA)*, (Dubrovnik, Croatia), pp. 633–638, Sept. 2011.

- [31] I. Martynov, J.-K. Kamarainen, and L. Lensu, "Projector calibration by "inverse camera calibration",," in Proceedings of the 17th Scandinavian conference on Image analysis, SCIA'11, (Berlin, Heidelberg), pp. 536–544, Springer-Verlag, 2011.
- [32] T. Okatani and K. Deguchi, "Autocalibration of a projector-camera system," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 12, pp. 1845–1855, 2005.
- [33] J. Drar'eni, S. Roy, and P. Sturm, "Geometric video projector auto-calibration," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, (Miami, Florida, USA), pp. 39–46, IEEE, 2009.
- [34] R. Furukawa and H. Kawasaki, "Dense 3D reconstruction with an uncalibrated stereo system using coded structured light," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 107–107, IEEE, 2005.
- [35] S. Yamazaki, M. Mochimaru, and T. Kanade, "Simultaneous self-calibration of a projector and a camera using structured light," in 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, (Colorado, USA), pp. 60–67, IEEE, 2011.
- [36] R. Raskar and P. Beardsley, "A self-correcting projector," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, (Kauai, HI, USA), pp. II–504, IEEE, 2001.
- [37] S. Chung and K. K. Bell, "Facial animation: A survey based on artistic expression control," Art Research, vol. 2, no. 1, pp. 131–162, 2009.
- [38] Z. Deng and J. Noh, "Computer Facial Animation: A Survey", Springer, 2007.
- [39] Z. Deng, P. Chiang, P. Fox, and U. Neumann, "Animating blendshape faces by cross-mapping motion capture data," in Proceedings of the 2006 symposium on Interactive 3D graphics and games, (California, USA), pp. 43–48, 2006.
- [40] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin, "Synthesizing realistic facial expressions from photographs," in Proceedings of the 25th annual conference on Computer graphics and interactive techniques, (Orlando, FL, USA), pp. 75–84, ACM, 1998.
- [41] K. Waters and T. Levergood, "An automatic lip-synchronization algorithm for synthetic faces," in Proceedings of The second ACM international conference on Multimedia, (San Francisco, California, USA), pp. 149–156, ACM, 1994.
- [42] F. PARKE, "A parametric model for human faces," PhD thesis, University of Utah, 1974.
- [43] K. Arai, T. Kurihara, and K. Anjyo, "Bilinear interpolation for facial expression and metamorphosis in real-time animation," The Visual Computer, vol. 12, no. 3, pp. 105–116, 1996.
- [44] M. M. Cohen and D.W. Massaro, "Modeling co-articulation in synthetic visual speech," Model and Technique in Computer Animation, pp. 139 –156, 1993.
- [45] F. Parke, "Parameterized models for facial animation," Computer Graphics and Applications, IEEE, vol. 2, no. 9, pp. 61–68, 1982.

- [46] F. I. Parke and K. Waters, "Computer facial animation", AK Peters (Wellesley, Mass.), 1996.
- [47] P. Ekman and W. Friesen, "Facial action coding system," CA: Consulting Psychologists Press, vol. 12, pp. 271–302, 1978.
- [48] S. Coquillart, "Extended free-form deformation: a sculpturing tool for 3D geometric modeling," SIGGRAPH Computer Graphics, vol. 24, pp. 187–196, Sept. 1990.
- [49] P. Kalra, A. Mangili, N. M. Thalmann, and D. Thalmann, "Simulation of facial muscle actions based on rational free form deformations," Computer Graphics Forum, vol. 11, no. 3, pp. 59–69, 1992.
- [50] T. Sederberg and S. Parry, "Free-form deformation of solid geometric models," in ACM Siggraph Computer Graphics, vol. 20, pp. 151–160, ACM, 1986.
- [51] S. Coquillart, "Extended free-form deformation: a sculpturing tool for 3D geometric modeling," in ACM SIGGRAPH Computer Graphics, vol. 24, pp. 187–196, ACM, 1990.
- [52] F. Scheepers, R. E. Parent, W. E. Carlson, and S. F. May, "Anatomy-based modeling of the human musculature," in Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97, (New York, NY, USA), pp. 163–172, ACM Press/Addison-Wesley Publishing Co., 1997.
- [53] I. Essa, S. Basu, T. Darrell, and A. Pentland, "Modeling, tracking and interactive animation of faces and heads using input from video," in Computer Animation '96. Proceedings, (Geneva, Switzerland), pp. 68 –79, Jun. 1996.
- [54] OptiTrack facial capture system. <http://www.naturalpoint.com/optitrack/>. Last accessed: March 2013.
- [55] F. Bernardini and H. Rushmeier, "The 3D model acquisition pipeline," in Computer Graphics Forum, vol. 21, pp. 149–172, Wiley Online Library, 2002.
- [56] MATHWORKS, INC, "Image Acquisition Toolbox", www.mathworks.com/products/imaq/.
- [57] OpenCV wiki. <http://opencv.willowgarage.com/wiki/>, Last accessed: Jan 2013.
- [58] J.-Y. Bouguet and P. Perona, "3D photography on your desk", <http://www.vision.caltech.edu/bouguet/ICCV98/>, Last accessed: March 2013.
- [59] J. Y. Bouguet, "Camera calibration toolbox for Matlab", http://www.vision.caltech.edu/bouguet/calib_doc/ , Last accessed: March 2013.
- [60] Intel OpenCV Computer Vision Library (C++). <http://opencv.org/downloads.html>, Last accessed: March 2013.
- [61] Z.Wang, Z.Wang, and X. Xu, "Extraction of the corner of checkerboard image," in 7th World Congress on Intelligent Control and Automation, pp. 6789–6792, 2008.
- [62] G. Strang, "Linear Algebra and its Applications", Orlando, FL: Harcourt Brace Jovanovich College Publishers, 3rd ed., 1988.
- [63] D. C. Brown, "Decentering distortion of lenses," Photometric Engineering, vol. 32, no. 3, pp. 444–462, 1966.

- [64] M. Yamazaki and G. Xu, "3D reconstruction of glossy surfaces using stereo cameras and projector-display," in 2010 IEEE Conference on Computer Vision and Pattern Recognition, (San Francisco, CA, USA), June 2010.

الملخص العربي للرسالة

يعتبر إعادة بناء وجه ثلاثي الأبعاد وإجراء الحركة عليه موضوعاً حيويًا في علمي الرؤية بالحاسب ومعالجة الصور بما لها من عدد كبير من التطبيقات المحورية في هذين العلمين وغيرها. ويندرج هذا التطبيق تحت ما يسمى بالرؤية المزدوجة حيث يتم تمثيل كيفية رؤية عيني الإنسان للأشكال باستخدام زوج من الكاميرات.

قديمًا كانت عملية الرؤية المزدوجة تتم باستخدام كاميرتين وكان يسمى هذا الأسلوب بالرؤية المزدوجة السلبية. هذا الأسلوب يحتوي على مشكلة كبيرة وهي مشكلة التوافق بين النقاط المتشابهة الموجودة في كلا صور الكاميرتين لأن معرفة البعد الحقيقي للنقطة يأتي من معرفة صورة هذه النقطة في كلا الصورتين. عند حدوث خطأ ولو طفيف في معرفة صورة هذه النقطة في كلا الصورتين لا يتم حساب النقطة الأصلية بطريقة صحيحة.

أما الآن فيتم استخدام أسلوب آخر يسمى الرؤية المزدوجة النشطة حيث يتم استبدال إحدى الكاميرتين في الأسلوب القديم بجهاز عرض بيانات. يقوم هذا الجهاز بإسقاط مجموعة من الصور التي تحتوي على شفرات معينه تتيح لنا التأكد من توافق النقاط الموجودة في الصورة المسقطه (النقاط بالنسبة لجهاز العرض) مع النقاط الموجودة في الصورة التي تم التقاطها بالكاميرا. وبذلك يكون حساب النقطة الأصلية في الشكل دقيق وصحيح.

المشكلة في هذا الأسلوب هو ضرورة معايرة جهاز العرض ومعرفة خصائصه الداخلية (خصائص العدسة) وكذلك الخصائص الخارجية (مكانه بالنسبة لنقطة القياس والمحاور الأساسية الأصلية). هذه المشكلة تكمن صعوبتها في أن معايرة جهاز العرض ليس كمعايرة الكاميرا التي تم وضع طرق كثيرة من قبل لمعايرتها. ذلك لأن جهاز العرض لا يمكنه تصوير الأشكال التي أمامه كما تفعل الكاميرات.

بعد بناء الوجه الثلاثي الأبعاد تم تحريك هذا الوجه باستخدام حركة حقيقية تم تسجيلها من جهاز يقوم بملاحظة علامات على وجه إنسان حقيقي والاحتفاظ بحركة العلامات وتطبيقها على الوجه الذي تم بناؤه.

وتتضمن الرسالة خمسة فصول. يحتوي الفصل الأول على مقدمة عن الرسالة وكيفية تنظيمها. والفصل الثاني يتكلم عن الأبحاث القديمة والحديثة المتعلقة بموضوع الرسالة. وتم شرح كيفية بناء الوجه الثلاثي الأبعاد وإجراء الحركة عليه تفصيليا وايضا شرح طريقة جديدة لمعايرة جهاز العرض في كلاً من الفصل الثالث والرابع. وتضمن الفصل الخامس والأخير عن ملخص عام قصير للرسالة وبعض الإتجاهات البحثية المستقبلية.



إعادة بناء وجه ثلاثي الأبعاد وإجراء الحركة عليه

مصطفى أبوبكر عبدالمجيد سالم

بكالوريوس الحاسبات والمعلومات - قسم علوم الحاسب

جامعة أسيوط - 2006

رساله مقدمه الى كلية الحاسبات والمعلومات - جامعة أسيوط

كجزء من متطلبات الحصول على درجة الماجستير في الحاسبات و المعلومات (علوم الحاسب)

جامعة أسيوط

أسيوط - مصر

2013

لجنة المناقشه:

أ. د. عادل أبوالمجد سويسي

أ. د. أسامة سيد محمد

أ. د. يوسف بسيوني مهدي

د. خالد فتحى حسين

لجنة الاشراف:

أ. د. يوسف بسيوني مهدي

د. خالد فتحى حسين