Structure from Motion Using a Single Camera

MUTHANA YASEEN NAWAF ISAWI

September 2016

STRUCTURE FROM MOTION USING A SINGLE CAMERA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
MUTHANA YASEEN NAWAF ISAWI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
INFORMATION TECHNOLOGY PROGRAM

September 2016

Title of the Thesis: **Structure from Motion Using a Single Camera.**

Submitted by **Muthana Yaseen Nawaf Isawi**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Halil TANYER EYÜBOĞLU
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Özlem DEFTERLI
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Reza HASSANPOUR
Supervisor

Examination Date: 19.09.2016

Examining Committee Members

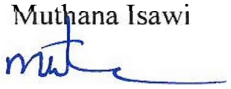
Assist. Prof. Dr. Reza HASSANPOUR (Çankaya Univ.)

Assoc. Prof. Dr. Tansel OZYER (TOBB Univ.)

Assoc. Prof. Dr. James LITTLE (Çankaya Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Muthana Isawi
Signature : 
Date : 19.09.2016

ABSTRACT

Structure from Motion Using a Single Camera

Isawi, Muthana Yaseen Nawaf

M.Sc., Department of Mathematics and Computer Science

Information Technology Program

Supervisor: Assist. Prof. Dr. Reza Hassanpour

September 2016, 55 pages

This thesis introduces a general survey of conversion algorithms, their advantages and disadvantages and a thorough explanation of the basic concepts in the field of 3D model reconstruction. The thesis concentrates, step by step, on the structures of motion technique and reconstruction of three-dimensional models from image pairs. The reconstruction process is carried out using a single calibrated camera and an algorithm based on only two views of a scene, the SFM technique based on detecting the correspondence points between the two images, and the epipolar inliers. All the experimental results have been computed using MATLAB (R2015b). By using the KLT algorithm we figure out the incompatibility of it with the widely-spaced images. Also, the ability of reducing the rate of reprojection error by removing the images that have the biggest rate of error. The experimental results are consisting from three stages. The first stage is done by using a scene with soft surfaces, the performance of the algorithm shows some deficiencies with the soft surfaces which are have few details. The second stage is done by using different scene with objects which have more details and rough surfaces, the algorithm results become more accurate than the first scene. The third stage is done by using the first scene of the first stage but after adding more details for surface of the ball in order to motivate the algorithm to detect more points, the results become more accurate than the results of the first stage. The experiments are showing the performance of the algorithm with different scenes and demonstrate the way of improving the algorithm. At last, we would like to mention that the aim of thesis is to figure out the depth information from two 2D images, and not to create 3D image from two 2D images.

Keywords: SFM, Conversion Algorithms, 2D into 3D, Computer Vision.

ÖZ

Tek Kamera Kullanımıyla Hareketten Yapı Oluşturma

Isawı, Muthana Yaseen Nawaf

Yüksek Lisans, Matematik ve Bilgisayar Bilimleri Anabilim Dalı

Bilgi Teknolojisi Programı

Danışman: Yrd. Doç. Dr. Reza Hassanpour

Eylül 2016, 55 sayfa

Bu tez dönüştürme algoritmaları, bunların avantajları, dezavantajları üzerine genel bir araştırma ile 3D modeli yeniden yapılandırma alanında detaylı bir açıklama sunmaktadır. Araştırma, hareket tekniğinin yapılarını ve görüntü eşlerinden üç boyutlu modelleri yeniden yapılandırma sürecini aşama anlatmaktadır. Yeniden yapılandırma süreci tek kalibreli kamera ve aynı sahnenin sadece iki görüntüsüne dayanan bir algoritma kullanılarak yürütülmüştür. SFM tekniği iki görüntünün bileşen noktalarını tespit etmeye dayanmaktadır. Tüm deneysel sonuçlar MATLAB (R2015b) kullanılarak hesaplanmıştır. KLT algoritması kullanılarak geniş aralıklı görüntülerin uyuşmazlığı açıklanmıştır. Ayrıca en büyük hata oranına sahip görüntülerin çıkarılmasıyla, yeniden projeksiyon oranının düşürülmesi sağlanmaktadır. Deneysel sonuçlar üç aşamadan oluşmaktadır. Birinci aşama, yumuşak yüzeyli sahne kullanılarak tamamlanmıştır. Algoritmanın performansı az detaylı yumuşak sahnelerde yetersizlik göstermektedir. İkinci aşama, daha fazla detayı olan sert yüzeyli nesneler kullanılarak tamamlanmıştır. Algoritma sonuçları birincisinden daha fazla doğruluk göstermektedir. Üçüncü aşama, ilk deneyseli sahneye daha fazla detay eklenerek gerçekleştirilmiş ve sonuçların ilk deneyden daha büyük bir doğruluk oranına sahip olduğu gözlemlenmiştir. Deneyler, algoritmanın farklı sahnelerdeki performansını göstermekte ve algoritmayı geliştirmek için yollar ortaya çıkarmaktadır. Sonuç olarak, tezin amacı iki boyutlu görüntülerden üç boyutlu görüntüler yaratmak değil, 2D görüntüler hakkında derin bir araştırma ortaya koymaktır.

Anahtar Kelimeler: SFM, Dönüştürme Algoritmaları, 2D'den 3D'ye, Bilgisayar Görüntüsü

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Asist. Prof. Dr. Reza Hassanpour for his supervision, special guidance, suggestion, and encouragement through the development of this thesis.

It is a pleasure to express my thanks to Dr. James little for his support and to my family and specially to my wife for their valuable support.

Finally, I would like to express my thanks to Eda Nur Timur from the English department for her help in the abstract translation from the English to the Turkish.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

2D	Two-Dimensional
3D	Three-Dimensional
WCS	World Coordinate System
CCS	Camera Coordinate System
SFM	Structure from Motion
P3P	Perspective Three Problem
DLT	Direct Linear Transform
SSD	Sum of Square Differences
PSF	Point Spread Function
SFS	Shape from Shading
SFT	Shape from Texture
EM	Estimation Maximization
PDF	Probability Depth Function
MRF	Markov Random Field
SLAM	Simultaneous Localization and Mapping
GPS	Global Position System
INS	Inertial Sensor

CHAPTER I

INTRODUCTION

1.1 **Problem Definition**

The ability of the vision of living creatures in receiving the real world as a three-dimensional scene motivates pioneers of the computer vision community to determine methods to simulate this ability. The solutions to this problem are divided into two groups, the first by acquiring a three-dimensional model directly from the real world by using special cameras such as a stereoscopic dual-camera with the ability to generate a three-dimensional model directly from a real-world scene. The second is by using two-dimensional data as inputs for algorithms designed particularly for the conversion of two-dimensional models into three-dimensional models. The role of these algorithms is to reconstruct a three-dimensional model based on the structure of the two-dimensional data which is missing the third dimension (the depth information) of the real world. The missing depth information is the result of the inadequacy of the traditional camera to obtain the third dimension from a captured scene, hence the role of algorithms to overcome this problem.

1.2 **Why Do We Need to Convert the Two-Dimensional into Three-Dimensional?**

In general, there is more than one reason to convert two-dimensional images into three-dimensional models. The enormous amount of two-dimensional data in the past and the present in addition to the traditional devices for capturing scenes from the real world are the most important reasons. At this point, we see a trend where the role of conversion algorithms from 2D to 3D for generating three-dimensional models is becoming more popular. The accuracy of these algorithms, which differ from each other, depends on elements such as time consumption and the precision of the output model [1] [2].

1.3 **Challenges Facing Conversion Techniques**

The challenges facing the techniques of conversion from the two-dimensional model to the three-dimensional model are divided into two groups. The first group covers every algorithm and a number of problems which must be solved by applying these algorithms. The second group of challenges involves specific types of algorithms considered to be high quality conversion techniques.

The first group of challenges includes three tasks which are solvable with every conversion algorithm. These tasks include [3][4]:

- **Apportionment of depth:** the determination of the range of allowed depth, the value of the depth value that should be matched to the screen location ("Intersection Point" Location), the allowed space ranges for objects on the screen according to the observer determining the three types of parallax known as zero parallax (on the screen), positive parallax (behind the screen), and negative parallax (in front of the screen).
- **Check of convenient disparity:** to avoid eye strain and the effects of nausea, the disparity must be convenient for the eyes without too much parallax or contradictory depth cues.
- **Padding of the exposed regions:** the objects in the original two-dimensional images may be partly or entirely occluded by the foreground, and should be uncovered (made visible) in the three-dimensional model.

The second group (as shown below) of these challenges could be named as typical problems, which require high quality conversion algorithms in order to execute them. Those problems such as:

- Semi-transparent objects such as glass
- Repercussion
- Foggy translucent objects
- Thin objects such as fur or hair
- Noise effects such as film grain
- The quick and unorganized motion in a scene
- Small pieces such as snow, rain and explosions

1.4 **Scope of Thesis**

The scope of the thesis is limited to obtaining the structure from the motion of the camera (SFM), and on exploring the methods of conversion of two-dimensional images into three-dimensional models. The thesis reviews the conversion method in the general part, and with regard to the structure from motion, the thesis is based on SFM with two views using a single camera achieving acceptable results as expounded in Chapter VI.

1.5 **Structure of the Thesis**

The structure of the thesis is as follows:

In the second chapter, the fundamental concepts are clarified with illustrations. The third chapter demonstrates background information about the conversion algorithms. The fourth chapter discusses past work carried out in this area. The fifth chapter contains the proposed method and introduces a theoretical discussion about the method. The sixth chapter presents the experimental results and a discussion thereof. The seventh and final chapter presents the conclusion and any future work.

CHAPTER II

FUNDAMENTAL CONCEPTS

2.1 2-D and 3-D [5]

In order to make the later chapters more clear, we introduce the current chapter with definitions and explanations for the terms and concepts of the related thesis topics.

The process of transformation from 3D space to a 2D plane can be illustrated with a *pinhole model* (Figure 2.1), which consists of a plane *R*, called the *image plane* and a point *C*, the *optical centre,* which does not belong to the image plane.

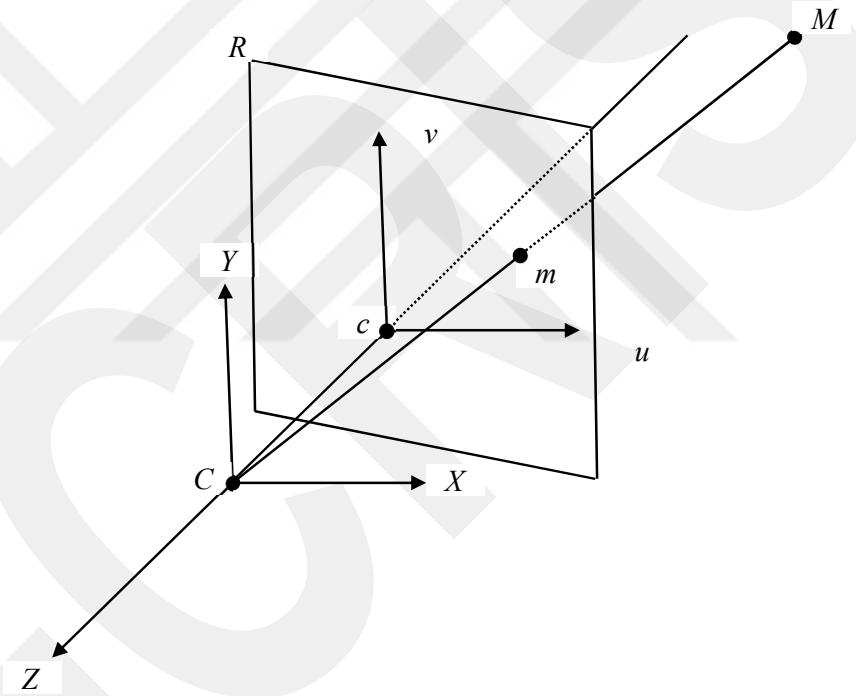


Figure 2.1 PinholeModel[1]

M object has a projection on the image plane R at the m point, and that projection represented by intersection of the optical ray (C, M) and the image plane R. The principal point c represents the center of the perpendicular of the optical axis on the image plane. The camera coordinate system (CCS) could be carried out with the center C and two axes (X and Y) which are parallel to the image plane (*u, v*) and the third axis Z corresponds the optical axis. The distance between the center C and the image plane represent the focal length *f*.

[1] *Figure Source:Reference* [5]

Thales theorem defines the relationship between the coordinates of M, $[X, Y, Z]^T$ and those of its projection m, $[u, v]^T$ as shown below:

$$u = \frac{X}{Z} \qquad v = \frac{Y}{Z} \qquad (2.1)$$

The aim of computer vision is to infer features of the world from images. The main problem of 3D vision is the inversion of the projection due to the transformation from a poorer representation of 2D to the richer representation of 3D.

2.2 The Relationship between the Camera and the Real World

In general, all images that we have represent the reflection of any object in our world, so those images represent the results of the relationship between cameras and the real world, and each point in the image has a corresponding point in the real world. Clearly, the position of any object in an image depends on its position in the real world. In fact, after the camera captures any scene, we obtain a 2D image coordinate $P(u, v)$ from 3D points (scene coordinates) $P(X, Y, Z)$, as shown in Figure 2.2 [6].

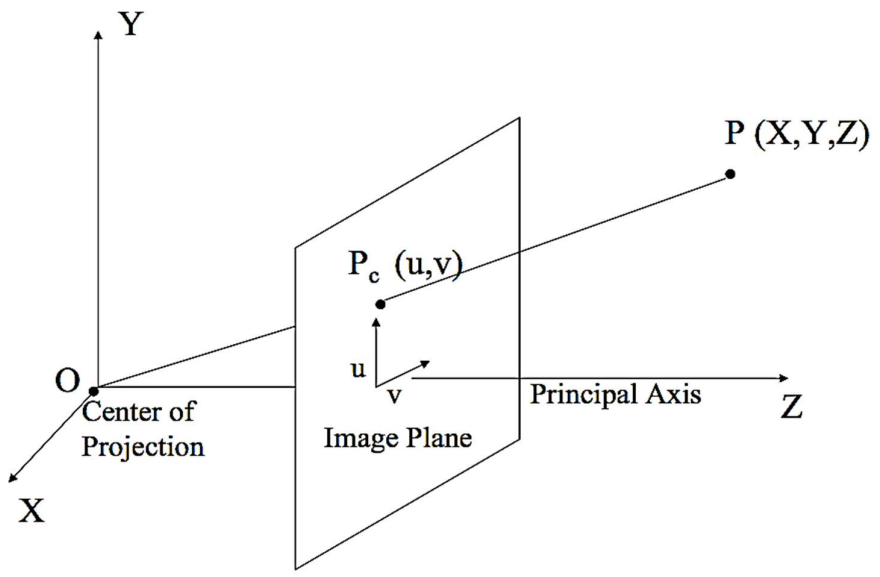


Figure 2.2 Image coordinate and world coordinate[2]

[2] *figure source:reference* [6]

2.3 Camera Calibration

Camera calibration is the process of estimating the internal camera parameter (intrinsic parameter) that relates the direction of rays through the optical centre to coordinates on the image plane. The importance of the internal camera parameter lies in the need for building 3D models of the world using a camera with a known intrinsic parameter [6].

2.4 Components of a 3D Point and its Corresponding 2D Image Point (Camera Works) [7]

1. **Internal camera parameter: (Intrinsic parameter)**: inherent from the camera regardless of the physical location of the camera in the world. Mathematically the intrinsic parameter is represented by the following matrix and is known as the camera calibration matrix:

$$K = \begin{bmatrix} \alpha_x & 0 & X_0 \\ 0 & \alpha_y & Y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Where

$$\alpha_x = \frac{f}{d_x} \quad And \quad \alpha_y = \frac{f}{d_y} \quad (2.3)$$

f : $Focal\ Length$

d_x , d_y: Scale x, y by physical dimension of a pixel

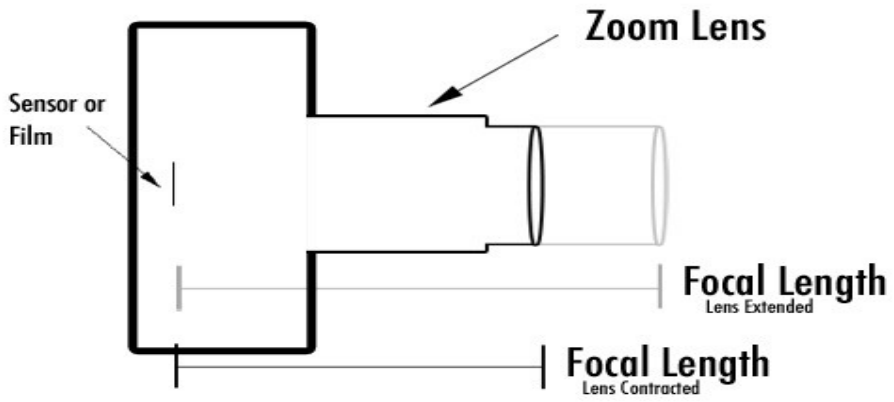


Figure 2.3 Focal Length[3]

[3] *Figure Source: http://photographycour .net/blog/focal−leng*

$X_{0\,\text{and}}\,Y_0$ represent the camera shift (Center of the image) or the principle point.
The camera converts 3D points (scene coordinates) from the real world to 2D (image coordinates) by using the following equation:

$$homogenous\ coordinates \rightarrow \begin{bmatrix} \chi \\ \gamma \\ 1 \end{bmatrix} \backsim K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (2.4)$$

$\backsim$ means 'proportional to' or 'equal up to scale.'
Therefore, by inserting more detail into the above equation, we obtain the following:

$$\begin{bmatrix} \frac{f}{d_x} & 0 & X_0 \\ 0 & \frac{f}{d_y} & Y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \frac{f}{d_x} X_c + X_0 Z_c \\ \frac{f}{d_y} Y_c + Y_0 Z_c \\ Z_c \end{bmatrix}$$

Then, we need to divide the result by Z_c to acquire the 2D image coordinates (pinhole projection equation):

$$\begin{bmatrix} \frac{f}{d_x} & 0 & X_0 \\ 0 & \frac{f}{d_y} & Y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \frac{f}{d_x} X_c + X_0 Z_c \\ \frac{f}{d_y} Y_c + Y_0 Z_c \\ Z_c \end{bmatrix} \sim \begin{bmatrix} \frac{f}{d_x}\frac{X_c}{Z_c} + X_0 \\ \frac{f}{d_y}\frac{Y_c}{Z_c} + Y_0 \\ 1 \end{bmatrix} \quad (2.5)$$

2. External camera parameter (camera extrinsic parameter**):** describes the camera pose [R, T] or the location of the camera in the world. We convert from the WCS (world coordinate system) to the CCS (camera coordinate system) with a rotation and translation [R, T].

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (2.6)$$

where $\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$ is the CCS, $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ is the WCS, $R(3 \times 3)$ is the rotation matrix, and t is the translation vector.

Here, we put the internal and external parameters into one equation:

$$\begin{bmatrix} \chi \\ \gamma \\ 1 \end{bmatrix} \backsim \mathrm{K} \begin{bmatrix} \mathrm{X_c} \\ \mathrm{Y_c} \\ \mathrm{Z_c} \end{bmatrix} = \mathrm{K} \left(\mathrm{R} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{bmatrix} + \mathrm{t} \right)$$

$$= K \, [R \mid t \,] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \text{ where } [R \mid t] \text{ is a } (3 \times 4) \text{ matrix.}$$

$$= \mathrm{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mathrm{P} \text{ is the camera matrix (2.7)}$$

We can encapsulate all the above equations as follows under the name of the image formation process:

$$\mathbf{x} \sim \mathrm{P}\mathbf{X} \qquad (2.8)$$

where

x represents the image coordinates and **X** the scene coordinates.

2.5 Two Views, Stereopsis

Stereo vision has great importance to the human due to the research into vision systems with two inputs. Stereo vision uses the information of their own relative geometry to infer depth information from the two views they receive, and use this information in the three-dimensional (3D) display which is not exist in the conventional two-dimensional content [8].

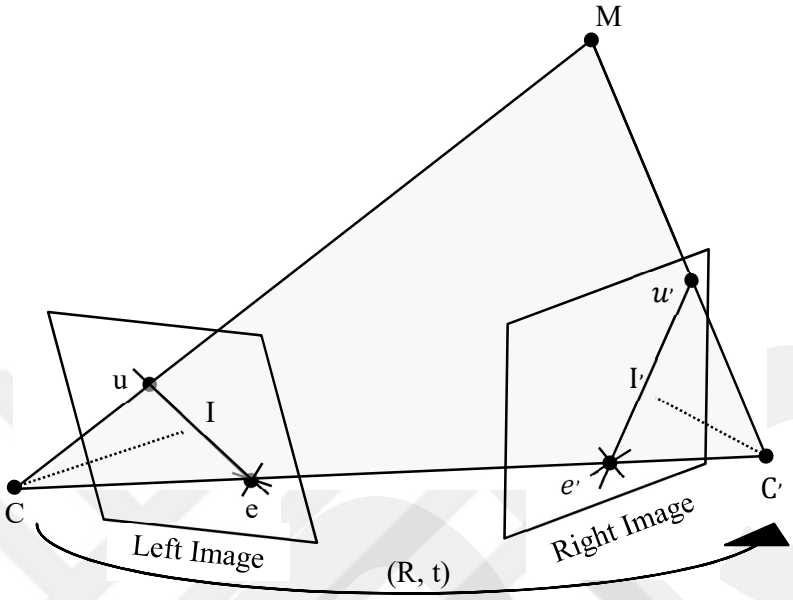


Figure 2.4 Epipolar Geometry[4]

Figure 2.4 shows the geometry of the system with two views in which the line between the optical centres **C** and **C′** is called the ***baseline***. When the optical centres **C** and **C′** intersect at the same point ***M*** by the corresponding rays in the scene, they create the ***epipolar plane***. The lines that intersect the image planes define the ***epipolar lines*** **I, I′**. The intersections of the baseline with the respective image planes represent ***epipoles*** **e, e′** which represents the points though which all epipolar lines pass when the scene point ***M*** moves in space [9] [10].

The projections of the scene point ***M*** in both images consecutively are ***u,u′.*** The ray ***CM*** is projected onto the epipolar line I′ in the right image which also represents every possible position of point ***M*** for the left image. The corresponding points ***u, u′*** in the right and left images must thus lie on the same epipolar line I′ in the right image. This geometry supplies a powerful **epipolar constraint** that minimizes the dimensionality of the search space for a correspondence between ***u*** and ***u′*** in the right image from the two-dimensional to the one dimensional [9] [10].

[4] *Figure Source:Reference* [8]

2.6 **Essential Matrix (E) and Relative Motion of the Camera**

The ***relative motion*** of the camera is the movement of a single camera in space with known calibration. The role of ***essential matrix E*** is to capture all the information about the relative motion between the two positions of the calibrated camera. The essential matrix is denoted by the following equation [6][8]:

$$E = [t]_x R \tag{2.9}$$

where *t* is the translation vector and R is the rotation

The properties of the essential matrix *E* are as follows [10] [11]:

- Rank 2
- Matrix of 3×3
- The first two singular values are always identical and the third is zero
- Depends only on the rotation and translation of the camera
- Usually considered to have five degrees of freedom
- Epipolar lines are retrieved from *E*.

$$\mathrm{I} = x_2^T E \ , \ \mathrm{I}' = x_1^T E^T \tag{2.10}$$

- Epipoles can be extracted from E.

$$e = null\ [E] \ , e' = null\ [E]^T \tag{2.11}$$

2.7 **Fundamental Matrix F**

The role of the fundamental matrix *F* is to capture all information that can be retrieved from two images in cases where the correspondence problem is solved. Moreover, the fundamental matrix *F* plays the role of essential matrix *E* from a camera with an arbitrary internal matrix. The fundamental matrix is denoted by the equation below and includes essential matrix equation 2.9 [6] [8] [10] [11] [12]:

$$F = K_1^{-T} E K_0^{-1} \tag{2.12}$$

where

K_1, K_0 are the calibration matrices

The properties of the fundamental matrix *F* are as follows:

- Rank 2
- Fundamental matrix has a relationship with the epipoles

$$e^T F = 0 \text{ and } F\, e' = 0 \tag{2.13}$$

- They have seven degrees of freedom
- Fundamental matrix *F* has the possibility of recovering the essential matrix

$$E = K_1^T F K_0 \tag{2.14}$$

2.8 **Motion**

The term ***motion*** appears when we deal with a sequence of images taken during different periods of time. According to the term *motion*, the position of objects changes between multiple images and the motion of those objects is called optical flow, which may be detected. In addition, motion can be used to generate a 3D description of objects from more than one view [8].

Generally, from a practical point of view, there are three types of motion-related problems. The first, known as ***motion detection***, represents the detection of any motion and used for security purposes and mostly uses a single static camera. The second is known as ***moving objects detection***, which poses another problem: a camera with a static position and objects which are moving in the scene, or vice versa. The second situation is considered to be more difficult in comparison with the first. The solution to the moving objects detection may depend on motion-based segmentation techniques. This problem becomes more complex when it includes object moving detection, and the detection of the path of its motion in the present and future. Image matching methods are mostly used to solve this problem. The last problem is the ***derivation of 3D objects properties*** from a group of two-dimensional projections acquired at varying time moments of object motion [13] [14].

2.9 **Motion Parallax**

Motion parallax is the phenomenon that provides the moving observer with the information about the depth to on object even when static objects appears to be moving relative to each other, so closer objects move faster than the distant ones [15].

2.10 **Disparity**

The first use of the term ***disparity*** was to describe the difference in position of the corresponding features seen by human eyes. In computer vision, this term refers to the difference in the image location of the same point in the three-dimensional scene when projected under perspective to two different views [11] [15].

2.11 **Binocular Stereo Vision**

The term ***binocular disparity*** denotes the procedure of deriving a three-dimensional structure from two images of a scene captured from multiple but slightly different standpoints. The variance of location gives rise to proportional displacements or variances of corresponding points in the images, and these variances allow the depth to be computed by triangulation [16].

2.12 **Structure from Motion SFM**

The technique of building three-dimensional models from two-dimensional images taken by a single moving camera around a static scene is not straightforward due to the formation process of the image not being invertible. To build a 3D model, we need to establish the properties of the camera and its position in each frame simultaneously. This technique is known as ***structure from motion*** **(SFM)** although this is somewhat a misnomer as both *motion* and *structure* are recovered simultaneously [6] [14] [17].

The use of structure from motion techniques are found in a wide range of applications such as:

- Photogrammetric surveys;
- Automatic reconstruction of virtual reality models from a video sequences; and
- The determination of camera motion.

2.13 **Blocks of Structure from Motion** SFM [11] [14] [18]

The trend of this section is to make the understanding of structure from motion easier and clearer. Thus, we describe how SFM works as steps that will create links between those steps, and how it makes it easier to adapt to the camera model.

The first three steps in the blocks are for a *calibrated* camera are as follows:

- ***Pose estimation***
- ***Motion estimation***
- ***Triangulation of points***

The fourth step demonstrates the role of ***bundle adjustment*** in both structures from motion and calibration.

2.13.**1 Pose estimation**

Pose estimation, or extrinsic calibration, is the opposite of the intrinsic calibration of the camera parameters, such as focal length. The pose estimation problem is one of the classical problems in computer vision. The computation of the object position and orientation is usually carried out by using points or lines corresponding between the object and the image. The minimal number of correspondence points necessary is three correspondences, which is known as the ***perspective three points problem*** (P3P), and extends to contain as large a number of points as PnP.

In order to solve the pose estimation problems, there are a number of techniques that can be used, such as ***direct linear transform***, ***linear algorithms***, and ***iterative algorithms***. All of these techniques have been developed to solve pose estimation problems.

2.13.1.**1 Linear algorithms**

The pose of the camera can be recovered by forming a set of linear equations similar to those used for two-dimensional motion valuation from a camera matrix form of the perspective projection.

$$x_i = \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}} \tag{2.15}$$

$$y_i = \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}} \tag{2.16}$$

(x_i , y_i) are the computed two-dimensional coordinates, and (X_i , Y_i , Z_i) are the three-dimensional coordinates (Figure 2.5). The camera matrix **P** is unknown and can be solved in a linear fashion by multiplying the denominator on both sides of the equation. The algorithm that is the result of this process is called a *direct linear transform* (DLT). The minimum known correspondences between the three-dimensional and two-dimensional coordinates are six correspondences that are needed to compute the 12 (or 11) unknowns in **P**. The intrinsic calibration matrix **K** and the rigid transformation (**R, t**) can be recovered after the entries in **P** have been recovered.

$$P = K[R|t] \tag{2.17}$$

When the camera is calibrated, the matrix **K** is known and the pose estimation can be used with as few as three points. In the *linear perspective n point* (PNP), the main notation is the visual angle between any pair $(\hat{x}_i, \hat{x}_j)$ of two-dimensional points that must be the same angle in the corresponding three-dimensional points (P_i, P_j) (Figure 2.5).

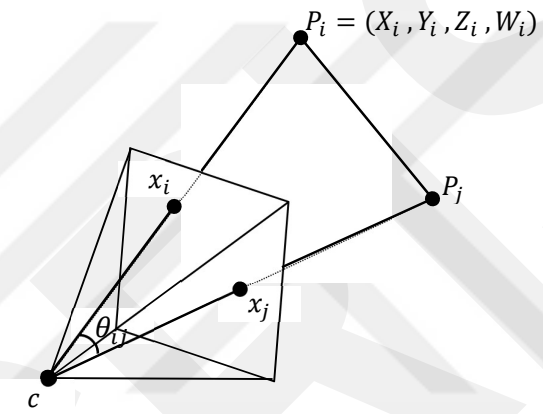


Figure 2.5 Pose Estimation[5]

2.13.1.**2 Iterative Algorithms**

Pose estimation can be recovered more accurately and flexibly by minimizing the squared re-projection error for the two-dimensional points of the unknown pose parameters (**R, t**), and optionally **K** by using nonlinear least squares. The projection equation can be written as:

$$x_i = (p_i, R, t, K) \tag{2.18}$$

5 *Figure Source:Referenc* [11]

We minimize the squared linearized re-projection error iteratively:

$$E_{NLP} = \sum_i p \left(\frac{\partial f}{\partial R} \Delta R + \frac{\partial f}{\partial t} \Delta t + \frac{\partial f}{\partial K} \Delta K - r_i \right) \quad (2.19)$$

where

$r_i = \tilde{x}_i - \hat{x}_i$ is the current two-dimensional error in the predicted position, and the partial derivatives are with respect to the unknown parameters such as the translation, rotation, and calibration parameters.

2.13.**2 Motion Estimation** [8] [18]

Motion or optical flow computation is dependent upon two propositions:

- that the spotted brightness of any point be steady over time; and
- that contiguous points in the image plane move in an identical style (*smoothness velocity constraint*).

The motion estimation then depends on a Gauss-Seidel iteration method of two dynamic images. If the number of images exceeds two, the computation will be more accurate by using the results of one of the iterations in the previous method to launch the current two images in sequence. These algorithms are parallel, and the iterations potentially are slow with computational intricacy.

According to the above propositions, the optical flow computation will be recovered with the algorithms mentioned above. Unfortunately, if those propositions are broken, error will occur in the results. Typically, the motion changes significantly in extremely textured zones, around moving edges, and at depth discontinuities. For these situations, *global and local optical flow computation,* and *global relaxation methods* of motion estimation are employed to determine the smoothest velocity area consistent with the image data. Relaxation methods have the property to reproduce topical constraints globally.

As an outcome, not only constraint information but also motion estimation errors are reproduced across the solution. Therefore, even problems in the small area in the motion estimation area potentially cause prevalent errors and poor motion estimates.

The global error reproduction is the most problematic of the global motion estimation scheme. Local motion estimation shows a good solution to this obstacle. The local estimation is dependent upon the same above propositions with the concept of the local estimate splitting the image into small areas where the propositions hold. This solves the error reproduction problem; however, another one appears in areas where the locative gradients change bit by bit, the motion estimation becomes poorly conditioned due to the absence of motion information. If a global approach is applied to the same area, the information from contiguous image pieces reproduces and represents a ground for motion estimation even if the local information was insufficient by itself.
The conclusion is that global sharing of data is useful in constraint sharing but bad with respect to error reproduction. One way to deal with the smoothness violation problem is to detect areas in which the smoothness holds. A pair of heuristics for specifying contiguous constraint equations that vary basically in their flow value are introduced. The main problem of the reproduction error is still unsolved. However, an estimation or rough guess is used with each flow vector that is dependent upon the heuristic rule of correctness, and the local average flow is estimated as a measured average. Consequently, the reproduction of error-free estimates holds.

2.13.**3 Triangulation of points** [11] [17] [18]

The meaning of the term ***triangulation*** represents the problem of locating a three-dimensional point from a group of corresponding image positions with known camera locations. Triangulation is considered to be the converse of the pose estimation discussed in 2.13.1.
The reconstruction algorithm reduces the result of squared errors between the weighted and the forecasted image locations of the three-dimensional point in the whole views in which it is apparent.

$$X = \arg min_x \sum_i \|u_i - \hat{u}_i (P_i, X)\|^2 \qquad (2.20)$$

where $(u_i , \hat{u}_i)$ and (P_i , X) respectively represent the weighted and forecasted image locations in the view. So far, the triangulation represents the process of determining the three-dimensional points as the intersection of two projection rays when two images are available (Figure 2.6).

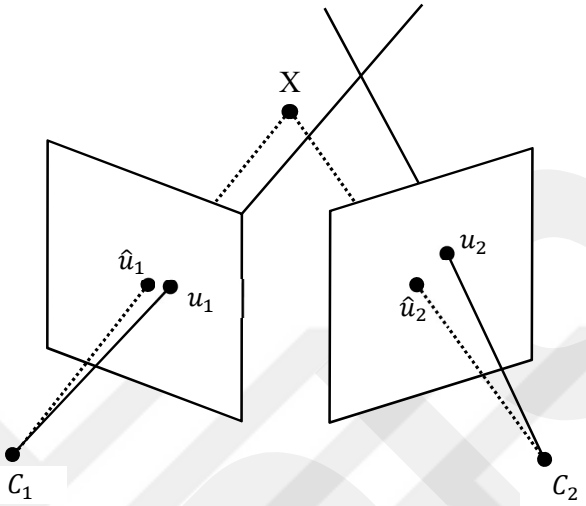


Figure 2.6 Triangulation[6]

2.13.**4 Bundle Adjustment** [10] [11] [17] [18]

The term *bundle adjustment* comes from the field of photogrammetry. Bundle adjustment is considered to be the final step in most algorithms. The goal of bundle adjustment is to purify the structure and motion parameters by using it repeatedly to reach the most probable estimate, which is carried out by reducing the suitable cost function where the suitable cost function represents the result of the total squared errors.

Purifying the structure and motion can be carried out by using nonlinear smallest squares to reduce the error measure:

$$E = \frac{1}{mn} \sum_{ij} \left[(u_{ij} - \frac{m_{i1}.P_j}{m_{i3}.P_j})^2 + (v_{ij} - \frac{m_{i2}.P_j}{m_{i3}.P_j})^2 \right] \quad (2.21)$$

Although the bundle adjustment is potentially costly, it provides the upper hand of merging all computations to reduce the important error measure, that is, the mean squared error between the current image point locations and those forecasted using the estimated scene structure and camera motion.

6 *Figure Source:Reference* [11]

CHAPTER III

BACKGROUND INFORMATION

3.1 **Introduction**

Generally, human vision can perceive the three-dimensional real world, including the depth in the shape of multi-ocular disparity. Human vision system represents the reason of observe various sights of the world. This phenomenon offers an advantage for the three-dimensional present by producing two somewhat differing images of each scene and then displaying them to each eye individually. The correct perception of three dimensions can be perceived by offering a suitable disparity and calibration of parameters.

The realm of computer vision has evolved a variety of algorithms to convert two-dimensional images into three-dimensional models. Each of these algorithms has its own advantages and disadvantages. Most algorithms have the advantage of using specific depth cues to produce a depth map [19].

3.2 **Two-Dimensional to Three-Dimensional Conversion Algorithms**

In general, conversion algorithms can be classified into two groups depending on the number of input images. The first group contains algorithms which deal with two or more images and the second group contains algorithms dealing with single images. In the first group, the inputs can be obtained either by using more than one fixed camera located in different positions or by using only one camera with a moving object in the scene. The depth cues used in the first group are known as multi-ocular depth cues. In the second group, the depth cues work on only one image and the depth cues are known as monocular depth cues [20].

Figure 3.1 shows the types of conversion algorithms mentioned above based on the number of input images.

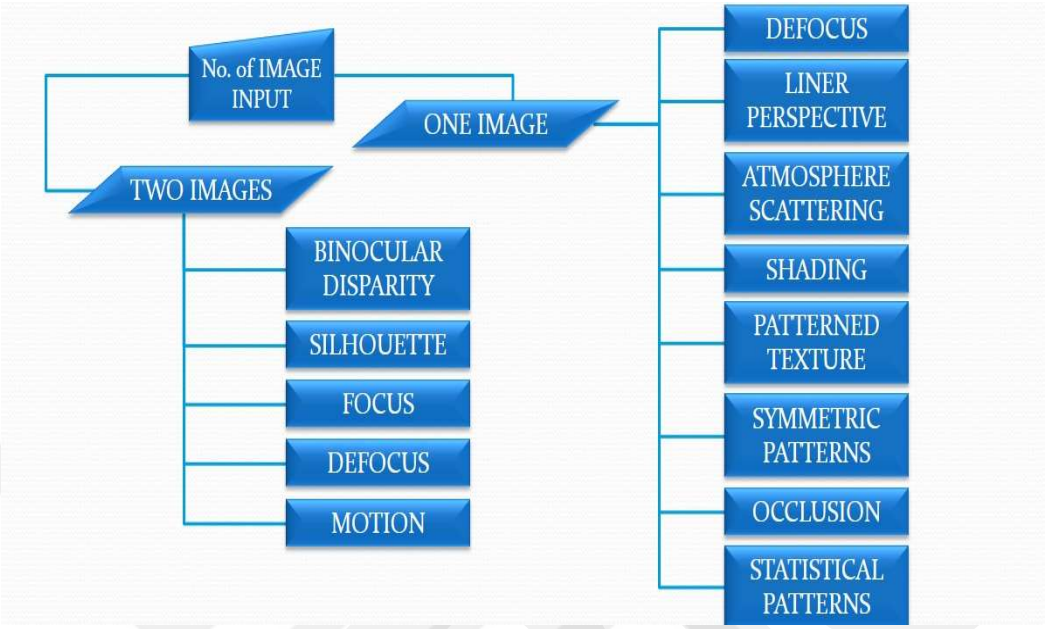


Figure 3.1 Conversion Algorithms [7]

The thesis deal with two images only, so later we will be demonstrating only the multi-ocular algorithms.

3.2.1 **Binocular Disparity**

The technique for obtaining the three-dimensional structure from a pair of two-dimensional images, which represents the same scene from two different views, is known as *binocular stereo vision algorithm*. The steps of this technique are, starting with detecting the corresponding points between a pair of images, then matching them and finally calculating the depth, by using triangulation. This algorithm is based on the restriction of disparity gradients in the matched image [LIoyd S.].

The binocular stereo vision solutions were obtained by imposing additional restrictions in order to solve the correspondence problem. These solutions were produced to cope with issues such as the occlusion or transparent surfaces, due to the violation of the constraints. Determining the binocular disparity, which must be a unique disparity, by using the motion parallax that was obtained from a moving monocular camera, represents an alternative solution instead of adding additional constraints [Nishikawa].

[7] *Figure Source:Reference* [20]

The quality of the matching and the execution time represents important issues in most algorithms. In order to achieve these issues, the square and gradients will be the base of the binocular vision algorithm. The gradient of the points of the image lines will be compared, and the image lines used as a series of areas. While the role of the sum of squares differences will be the basis for determining the best matching in each area. This algorithm is characterized by being quality derived from the gradients algorithm, and cope with the additive noise [Hu T.].

3.2.2 **Silhouette**

Silhouette estimates of the perspective projection have some difficulties in order to obtaining an efficient result. Silhouette estimates, based on a sequence of viewpoints and the parallel projection, is the proposal method. This algorithm using the sequential frames of the polyhedral model, and tracking the ongoing changes in the silhouette between them. Then the computation is carried out based on the point-plane duality in 3D [Pop M.].

The objects have a silhouette images, some algorithms are using this feature in order to construct a three-dimensional model for these objects. The shape from silhouette (SFS) is one of those algorithms that used the silhouette images to create a 3D model which known as visual hull (VH). Shape from silhouette carry out on static and dynamic objects, the moving objects in the case of videos. The steps of the proposal algorithm are VH alignment and VH refinement. The algorithm extended from the rigid objects to the articulated objects which have unknown motion [Cheung K.].

The occlusion, noise, and the errors in the background all these are considering as a reasons for generation inconsistent silhouette. The proposal algorithm is dealing with all these reasons and construct a robust silhouette model. This algorithm is proposing to continue to reduce the energy problem which is known as the error between the shape and the silhouette. Also, the algorithm is using the visible feature of the surface in order to construct the shape. The results of the algorithm are obtained by using the graphic card processor with parallel computing, this method will reduce the computation time. Finally, the algorithm introduced an assistant function that construct simultaneously the visible surface and empty visual cone [Haro G.].

For more methods, [Hartner A.] contain a comparison and explanations of the object space silhouette algorithms.

3.2.3 **Focus**

The shape from focus (SFF) algorithm proposed as a method which is using different levels of focus to generate a sequence of images. This algorithm is used the Sum-Modified-Laplacian (SML) operator which is applying on the image sequences in order to measures the quality of the image focus. The local depth estimates are computed by using the set of focus in each image point. The depth estimation is figure out by using two algorithms, the first one is tracking the focus levels, and the second one is used the SML focus measure differences at each point as a Gaussian distribution. Both of algorithms can be used with smooth and non-smooth textured surfaces by using specific illumination methods [*Nayar S.*].

The auto focusing algorithm which proposed is based on paraxial geometric optics of the image origination. Due to the adoption of the algorithm, the focus measures which is based on the energy of the image gradients have some negative side effects. The proposed method solution is based on divided the auto focus algorithm into two steps. The first stage result is obtained by using the image disparity to find the vicinity of focus, then the second stage result represents the optimum focused image with the focus measures. The proposed algorithm is designed especially to the digital cameras [*Lee J.*].

Depth estimation by using more than on image with different focuses, and by using only the spatial image gradients as the focus measure is the proposed method. The algorithm is used two types of the decisions, the corroborates and soft decision. Those decisions add more accurate to the algorithm in order to deal with the sensor noise and optics-related effects [*Eltoukhy H.*].

The algorithm based on a one-dimensional Fourier transform and the Pearson correlation is the proposed method. The algorithm process is done by using a specific vector pattern which is used to search in each image. Then, the Fourier transform is carried out and extract the frequency content of the vector pattern. Finally, the frequency vector is comparing with a reference image which is detected by using the Pearson correlation. This algorithm is suitable to cope with different environments with consideration of the illumination [*Bueno M.*].

3.2.4 **Defocus using more than two images**

The technique of estimate the depth from two images with different amount of defocus of the same scene without correspondence problem is the proposed algorithm. The algorithm consists from two stages, first one is the calibration stage and the second one is the depth recovery stage. The defocus process in this algorithm is addressed as a Gaussian point spread function (PSF) [*Hwang T.*].

The shape from defocus (SFD) is the process of obtaining a 3D geometry which required a set of defocused images. The typical method is required a deplurring for the focused images and approximation of the scene which known as equivocal assumption. The proposed algorithm is introduced a method of obtaining the three-dimensional geometry without a strong assumption for the scene in order to avoid the deblurring. The solving of the defocus problem requires forming the interaction between the light and the optics, this interaction known as point spread function. The algorithm introduces two solutions; each solution is suitable for specific situation. These situations are defined by the known and the unknown form of the point spread function. The proposed solutions have only one simple matrix-victor multiplication, and based in general on the minimize of the Euclidean norm of the difference between the observed image and the estimated image [*Favaro P.*].

The projection defocus analysis, which is modelling by using the linear system, is the base of the proposed method. The projector's model is used to estimate the depth at each camera pixel through computing of the parameters of the projection defocus in frequency domain. In order to ensure that the recovered depth is covering all the camera pixel, the algorithm is used the coaxial projector camera system.

This algorithm effectively contributed to the increases the depth of field of the projector without needed to justify the projector optics. Also, the algorithm is get rid of the strong pixelation artifacts which is caused by the digital projectors with consideration to the quality of the projected image [*Zhang L.*].

In the past the defocus could be obtained through using multiple images exposures focused at variant depths, and the correspondence cues is required multiple cameras or multiple exposures at variant viewpoints. Nowadays, the light-field cameras become available in the market, so in a single capture those cameras are offering the depth information from defocus and correspondence at the same time. The proposed algorithm is combining the focused and the correspondence depth cue, which are obtaining from light-field cameras, in order to calculate the dense depth estimation [*Tao M.*].

3.2.5 **Motion**

The technique of obtaining the structure and motion information from multiple images without needed to the correspondence information is the proposed algorithm. This algorithm is based on the probability distribution which is iteratively refines on the set of correspondence. At each iterative, structure from motion problem is solved. The Markov Chain Monte Carlo technique is used to obtain the probability distribution [*Dellaert F.*].

The two-frames motion estimation is the proposed algorithm. This algorithm is consisting from two stages; the first stage is carried out by using the quadratic polynomials in order to estimate each neighbourhood of the frames. The second stage is done by observing the polynomial transform under translation in order to estimate the displacement fields from the polynomial expansion coefficients [*Farneback G.*].

The structure from motion technique is used to reconstruct three-dimensional model by using multiple two-dimensional images. The proposed algorithm is based on the incremental of the SFM by using unordered 2D images, and the accuracy and the efficiency are considering as a purposes of this algorithm [*Schonberger J.*].

CHAPTER IV

STATE-OF-THE-ART

4.1 **Conventional Techniques to Structure from Motion**

The procedure of obtaining structure from a set of images began in the 1980s [21-24]. Normally, structure from motion is initially approached by placing a set of obvious characteristics that are found in two image structures. This is commonly denoted as *the correspondence problem solution*. Then, the proportional motion of these characteristic correspondences is given the structure of the environment [25]. By computing of the optic flow within the given image sequence, there is the possibility of estimating the structure from motion without directly placing the correspondence points [25].

The conventional estimation of structure from motion mostly uses two images obtained from a single camera to slant the field of view of 45 to 60° [34-36]. However, there are advantages to raising the number of images in the estimation process [29] and also raising the field of view [30]. Additional refinement in accuracy can be achieved by assuming further constraints, which can be varied, such as the restriction of the objects' speed in linear motion, breaking down the two-dimensional photo into two one-dimensional photos, and so on [31].

Batch processing means several images being processed at once, which causes a significant delay if the calculation was wanted in real time. Instead, it is suitable for real-time executions to produce a structural computation of a recursive nature, permitting recurring refined calculations to be usable after each new image is scanned [25].

4.2 **Related Works**

Using the structure and motion together under the name of *structure from motion* to reconstruct the three-dimensional model from multiple images is considered to be a significant topic in computer vision research. The pioneers in the field of computer vision have proposed many techniques to fill the lacunae in the structure from motion approach.

Zhengyou Zhang [32] used structure and motion from two perspective views based on the essential parameters, a fundamental matrix and Euclidean motion. The typical technique consists of two steps:

- Calculate the nine essential parameters by using the 8-point algorithm (considered a linear calculation problem).
- Rectifying the motion calculation depending on statistically optimum measures (considered a non-linear calculation problem in 5-dimensional space).

The problem with this technique is that the results mostly are not good enough due to the sensitivity of the second step to the incipient guess and the difficulty of obtaining an accurate incipient estimate from the first step. In order overcome this problem, Zhengyou Zhang proposed an approach by imposing the fundamental matrix (*zero-determinant constraint*). The process of this technique is carried out gradually through project parameters calculated in a higher-dimensional space onto a lower-dimensional space, which means moving from 8 dimensions to 7 and finally reaching 5 dimensions.
Unlike [32], *Frank et al* [33] introduced another technique by using the *structure from motion without correspondence.* This method exceeded the traditional techniques that require the presence of a known correspondence point [34] or calibrated images from a known camera viewpoint [35] or known shape [36]. Furthermore, this method deals with non-sequential images which are taken from vastly different viewpoints.
Masahiro [37] introduced a method of using the structure from motion in map reconstruction. This method was a system of three-dimensional simultaneous localization and mapping (*SLAM*), which is based on the SFM scheme. The steps of this method are as follows:

- Basic Framework
- Feature Tracking
- Initial Estimation

The first step considers the three-dimensional SLAM as a set of images obtained from a monocular camera. The three-dimensional map is represented as three-dimensional points from the feature points tracked through the set of images. The second step occurs based on KANADE-LUCAS-TOMASI [38]. The third step occurs by using the factorization method [39].
The precision and robustness of this method is based on the selection of the baseline distance, so the proper baseline selection depends on standards for object shape

reconstruction and the camera pose estimation. Figure 4.1 clarifies the procedure of this method.

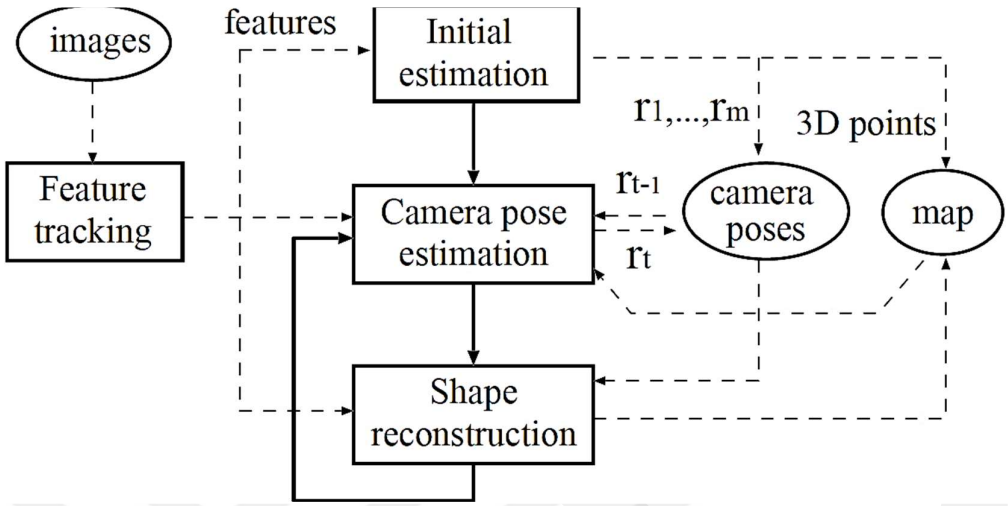


Figure 4.1 3D SLAM Basic Procedure[9]

Zach et al [40] discovered and used three-dimensional symmetries based on image cues in SFM. The aim of this technique is to retrieve the symmetry connections in conditions under which the initial structure from motion veers due to drift and can be imprecise. In order to cope with this problem, this approach should discover symmetry restraints within uncertain three-dimensional structures, and enforce them during structure from motion. Thus, reconstruction will be more accurate where the derived structural restraints are observed. In this method, Zach et al proposed a bundle adjustment equation in the case of the structural restraints being imposed between different subsets of three-dimensional point sets linked by propinquity transforms. Additionally, the symmetry knowledge offers a natural coordinate for the structure to be selected during the bundle adjustment. To this end, the underlying symmetries allow us to complete the three-dimensional model.

Klingner et al [41] uses the structure from motion to model the street view images by extending the SFM technique in order to repair the pose of those images.

[9] *Figure Source:Reference 37*

This method presents two challenges: the planet-wide scale and the rolling camera shutter. In order to overcome these problems, Klingner et al used a good initial approximation of the local vehicle route. Through the incorporation of techniques, such as SFM, GPS, and INS (Inertial Sensors), the approach corrects the distortions in the street view image, as shown in Figure 4.2:

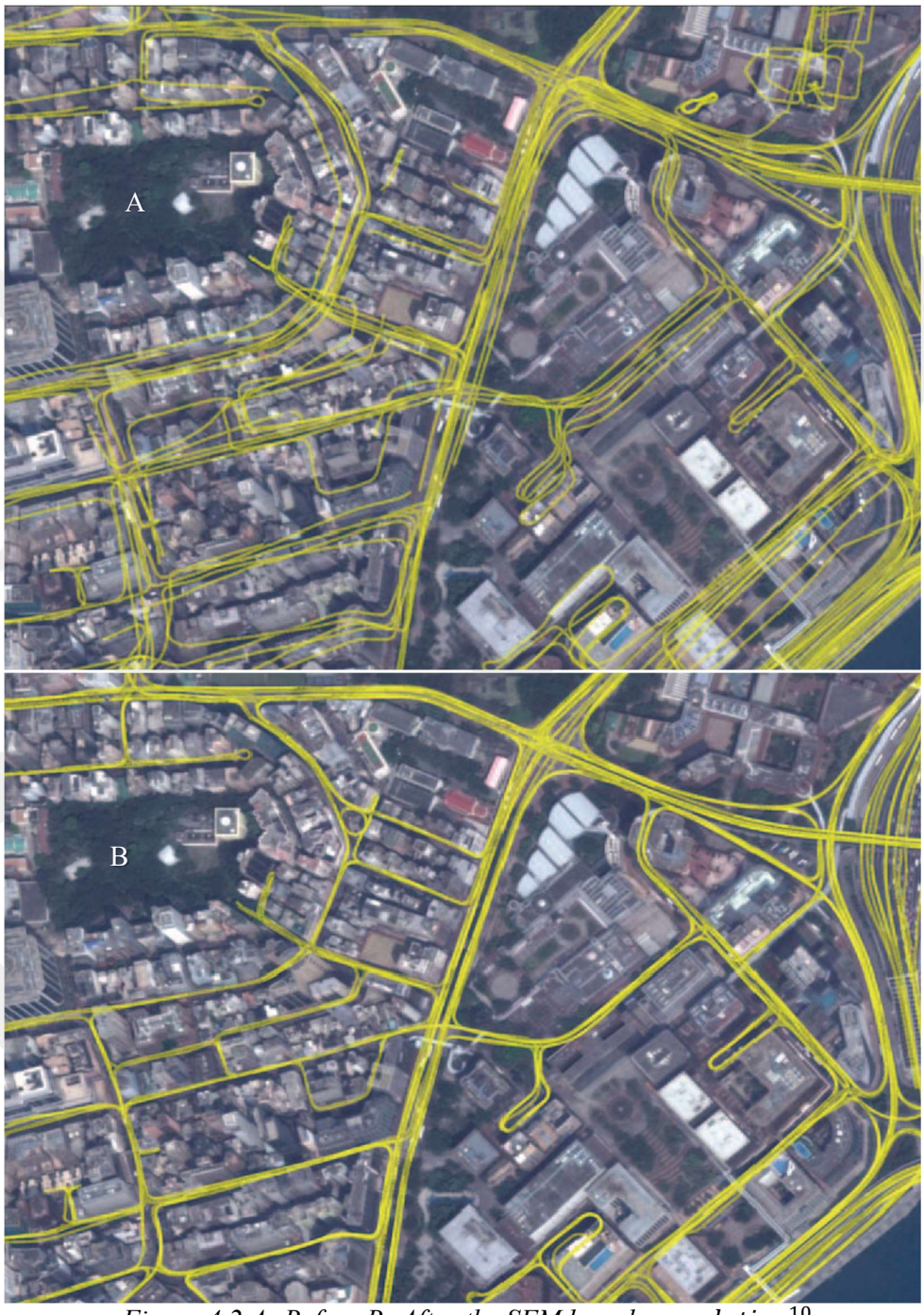


Figure 4.2 A: Before B: After the SFM based-correlation[10]

10 *Figure Source:Reference* 41

In spite of the advantages that are offered by the previous method of reconstructing the city-scale, it is still expensive due to the use of the GPS/INS systems. Yongjun Zhang et al [42] introduced an SFM method of producing the city-scale reconstruction based on images obtained with a driving recorder without any information from the GPS/INS systems in order to decrease the cost of reconstruction. Figure 4.3 shows the steps of this method.

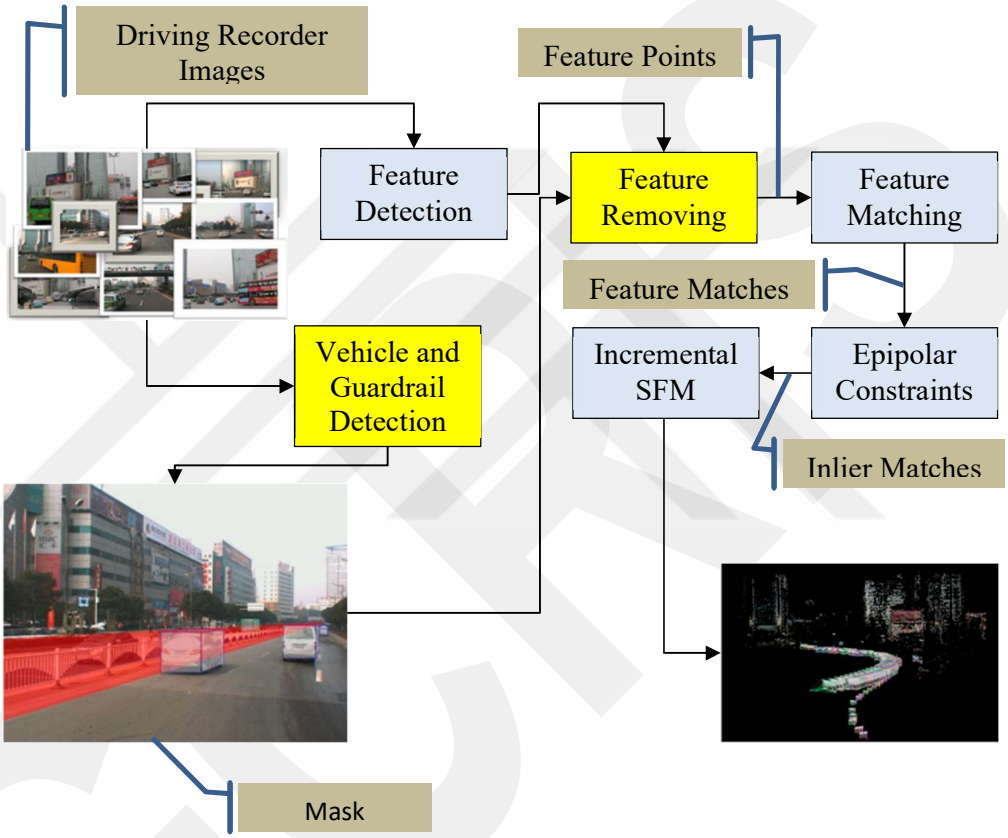


Figure 4.3 the Steps of three-dimensional reconstruction[11] : The blue frames represents the SFM process, while the yellow one represents the main betterment steps proposed by this method

[11] *Figure Source:Reference 42*

CHAPTER V

THE PROPOSED METHOD

5.1 **Methodology**

This chapter describes the method based on achieving the goal of this thesis. According to the title of the thesis, the technique of reconstructing a three-dimensional model from a pair of two-dimensional images depends on structure and motion. In order to obtain this information, there are a number of steps to follow. First, we need a static scene with an object of known size (in our scene, the object is a ball of size 10 cm), and a calibrated camera to obtain two views. After obtaining the real data in two images, the work of the algorithm begins at this step. The workings of this algorithm are presented in the following sections.

5.1.1 **Detection of The Correspondence Points**

In order to continue to the others step, it is necessary to find the correspondence points. Therefore, the best features need to be detected in order to track from image to image. This process is carried out by using the *minimum eigenvalue algorithm* as proposed by C. TOMASI & J. SHI [48], and as the below equation shows:

$$R = \min(\lambda_1, \lambda_2) \qquad (5.1)$$

where (λ_1, λ_2) represents the eigenvalues and the window (corner) is accepted if those eigenvalues are greater than the predefined threshold value (λ) as shown below:

$$\min(\lambda_1, \lambda_2) > \lambda \qquad (5.2)$$

According to the C. Tomasi & J. Shi method, the strongest corners will be found in the image, which is a grayscale image.

5.1.2 **Features Tracking**

This step begins after finding the strongest corners (best features) from the first image. The role of this process is to track those features in the second image. This process is carried out by using the KLT algorithm (*KANADE-LUCAS-TOMASI*) [43]. The goal of this algorithm is to find the specific location of a specific point in the second image according to the first image. This is achieved with the following equation:

$$\bar{V}_{opt} = G^{-1}\bar{b} \qquad (5.3)$$

5.1.3 **Computing The Fundamental Matrix**

The computation of the fundamental matrix from the correspondence points which are detected is the first step. The fundamental matrix was explained briefly in Chapter 2 (2.7).

5.1.4 **Camera Motion Calculation**

In this section, we will estimate the position and orientation of a calibrated camera. Normally, there are two views, hence there are two poses. Both poses are relative to each other as denoted by the fundamental matrix *F*. The camera poses are computed up to scale and the position denoted a unit vector. The second chapter (Section 2.4) also mentions camera pose.

5.1.5 **Triangulation**

The three-dimensional positions of the matched points can be determined by triangulating. (This term is explained in detail in Chapter II (2.13.3)).

5.1.6 **Detect an Object with Known Size**

This process is carried out by using the MSAC algorithm (*M-estimator sample consensus*). The fitting of a sphere to an inlier point cloud using an object with known size is here a ball of size 10 cm.

CHAPTER VI

EXPERIMENTAL RESULTS AND DISCUSSION

6.1 **Experimental Results**

The experiments were carried out on an ordinary PC equipped with the following specifications:

- System Type: 64-bit operating system, x64-based processor.
- Edition: Windows 10 Home.
- Processor: Intel (R) Core (TM) i3-2310M CPU @ 2.10 GHz.
- RAM: 4.00 GB.

The input images were obtained from a digital camera (NX3000) equipped with:

- 20.3 MP APS-C CMOS Sensor.
- 16-50 mm Power Zoom Lens.
- 1/4000 sec Shutter Speed.

All experiments were carried out using the MATLAB R2015b software package. The methodology of the thesis was based on the technique of 'structure from motion', but by using a single calibrated camera with the *camera calibration application* in MATLAB and by obtaining two views of the scene with a little motion for the second view. The algorithm that will create the three-dimensional model of the scene, from a pair of two-dimensional images following a number of steps, as the next section shows.

1. The first step is carried out by loading a pair of images of the scene obtained by using the above camera.
2. Next, the camera parameters are obtained by loading the camera calibration. In order to understand the mean reprojection error, which represent the difference in distance between the actual scene and the estimated one, we show below the equation of mean projection error:

$$\sum d\,(x_i, \hat{x}_i)^2 + d\,(x_i', \hat{x}_i')^2$$

The unit of the reprojection error in pixel, so less than one it will be acceptable rate as shown in figure 6.1.

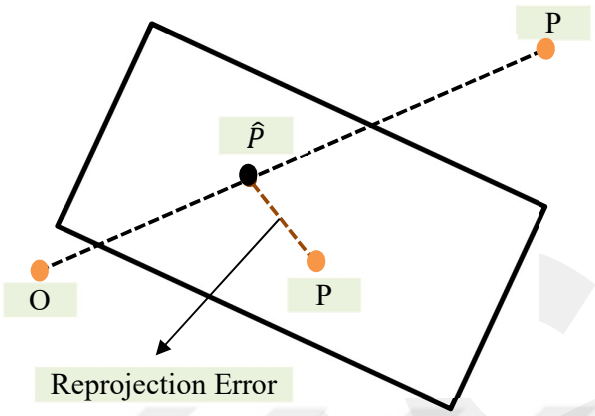


Figure 6.1: The reprojection error

- Steps of the camera calibration:

Firstly, there are some tools that should be available to implement the calibration such as the camera, checkboard, and camera calibrator application which are found in MATLAB. Next, obtain a group of images of the checkboard using the mentioned camera, and inserts these images into the camera calibrator application. For best results, load or acquire between (10) and (20) images. The following figures show the above steps of calibration:

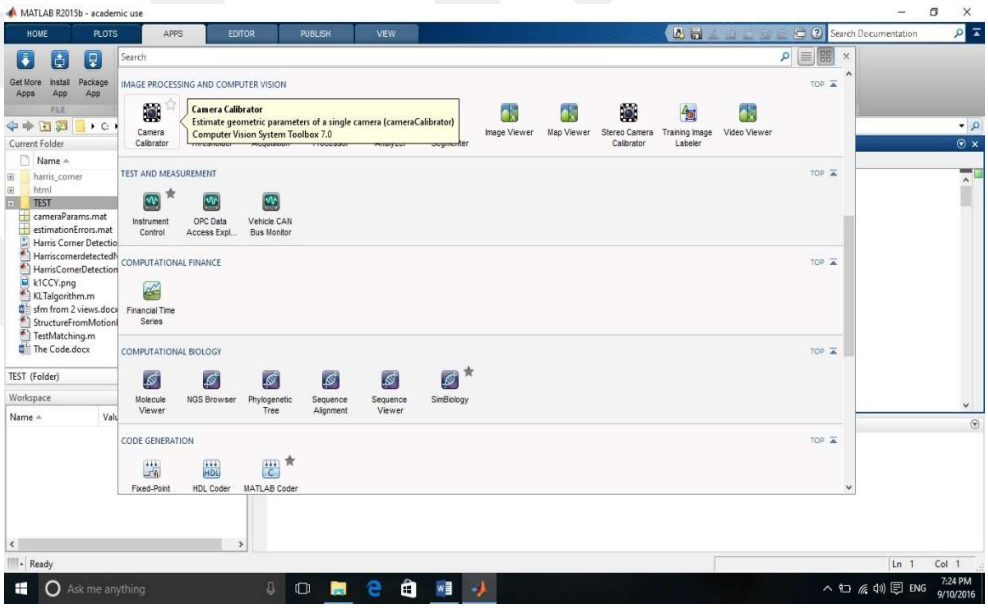


Figure 6.2: Select the camera calibrator application

After opening the calibrator application, add the images and specify the size of checkboard as shown in figure 6.3:

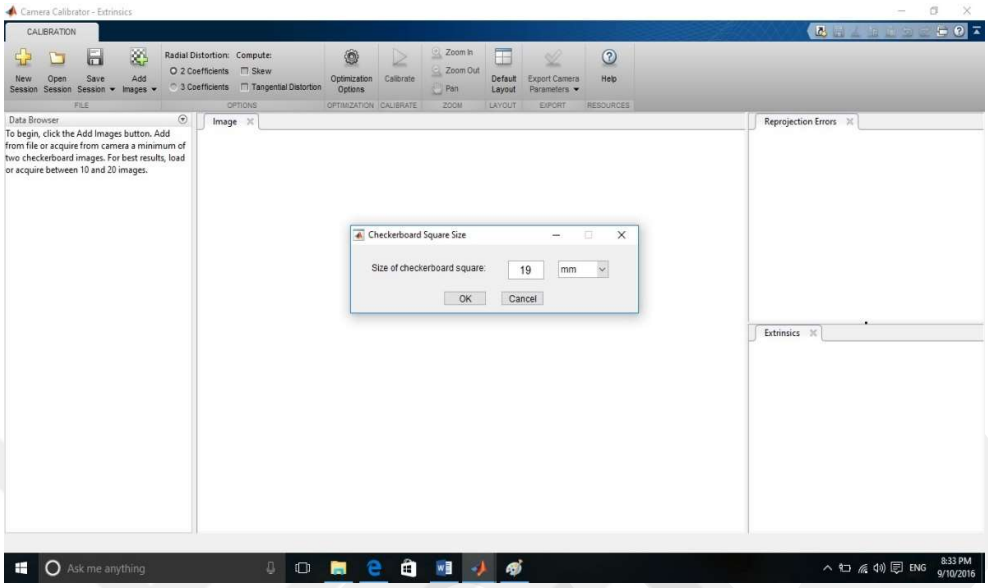


Figure 6.3: Adding and specify the size of checkboard

Here, the total images processed was 32 and the added images were 28 while the rejected were 4 images (Figure 6.4).

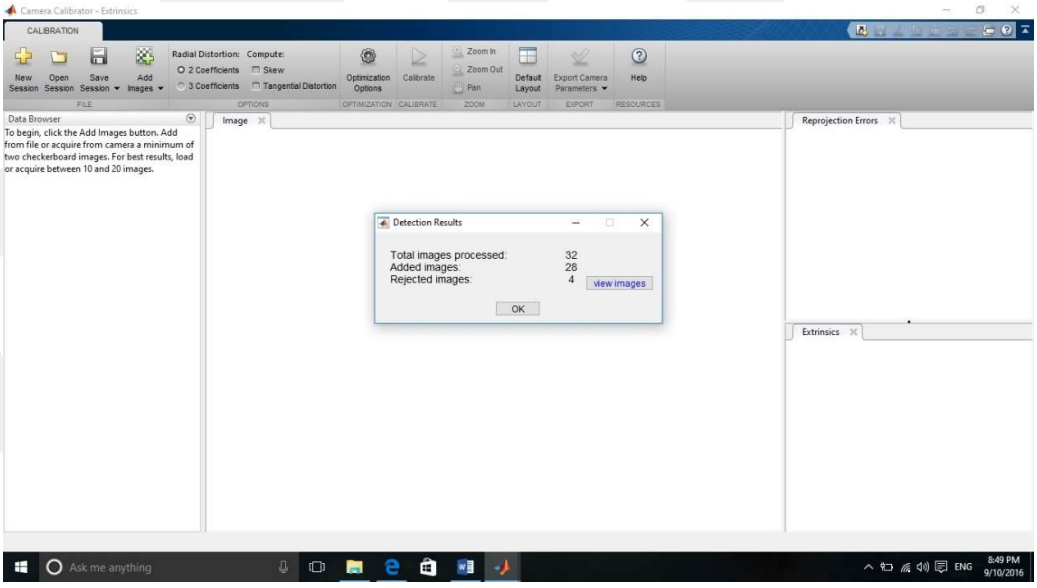


Figure 6.4: The added images

Later, the camera is calibrated, shown in figure 6.5.

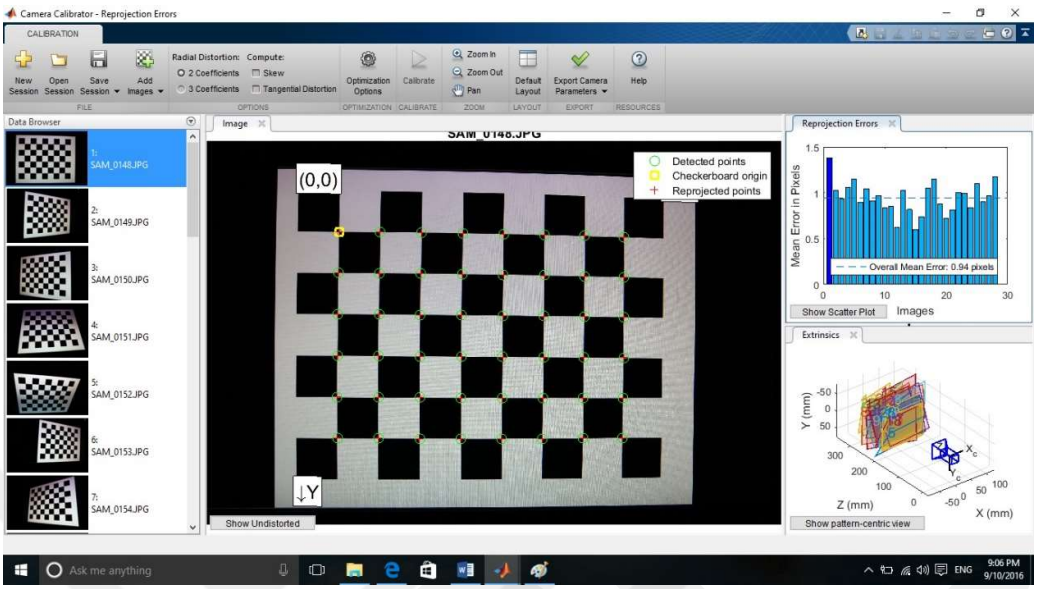


Figure 6.5: The Result of calibrator

The mean error in pixel which is 0.94 shown in the figure 6.6, the mean error could be minimizing by removing any images which have the biggest error.

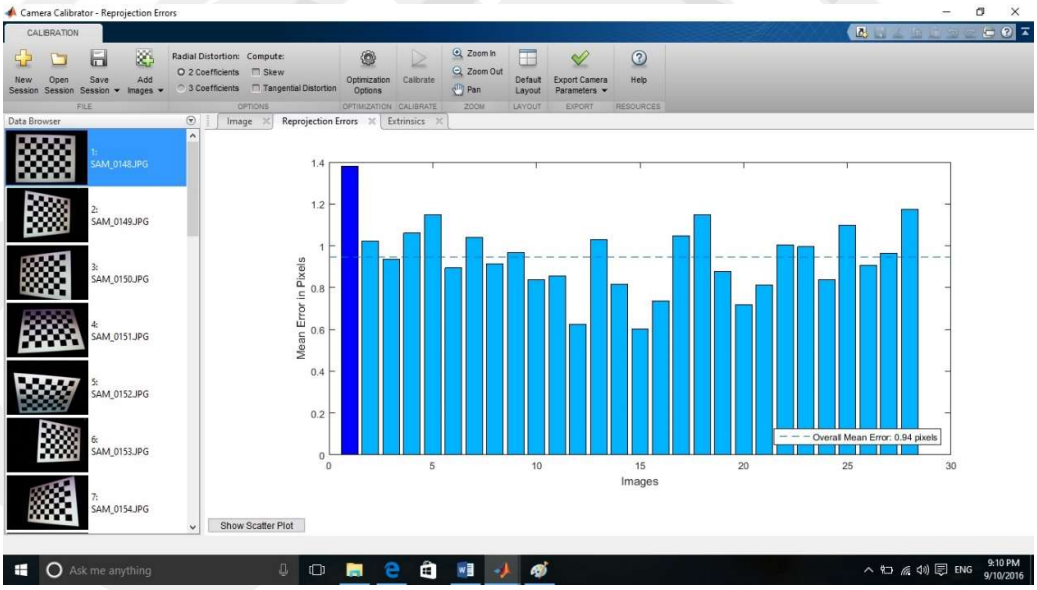


Figure 6.6: The Reprojection errors (0.94)

After removing the images which have the biggest error, which was only 18 images, we decrease the mean error from 0.94 to 0.77 as shown in the figure 6.7. The reason of stopping remove more images is the calibrator application require at least 10 images to give an accurate result.

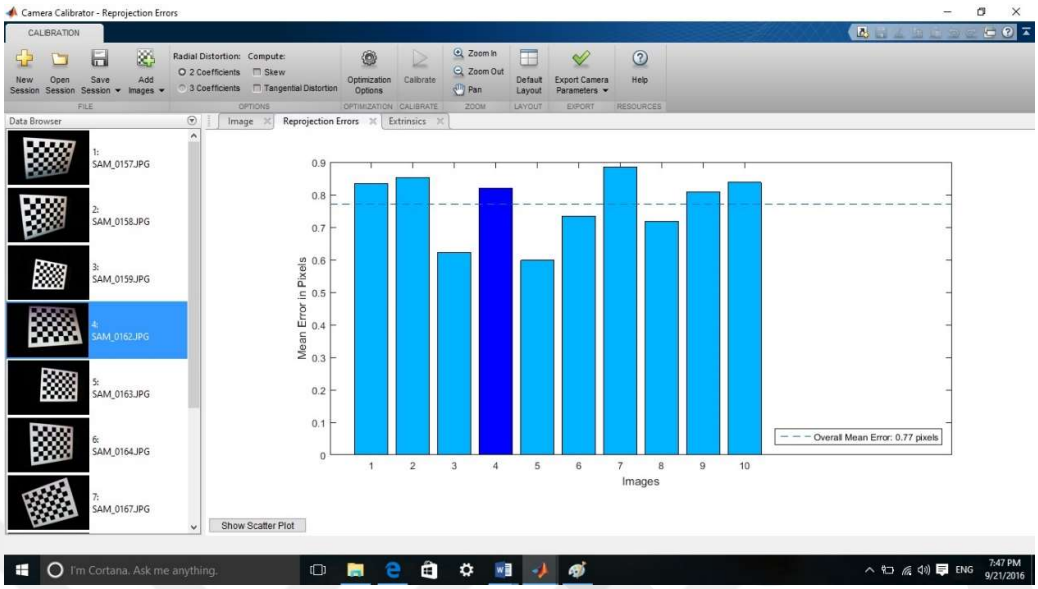


Figure 6.7: The Reprojection errors (0.77)

The extrinsic parameter of the camera for 28 images with mean error 0.94 is shown in figure 6.8.

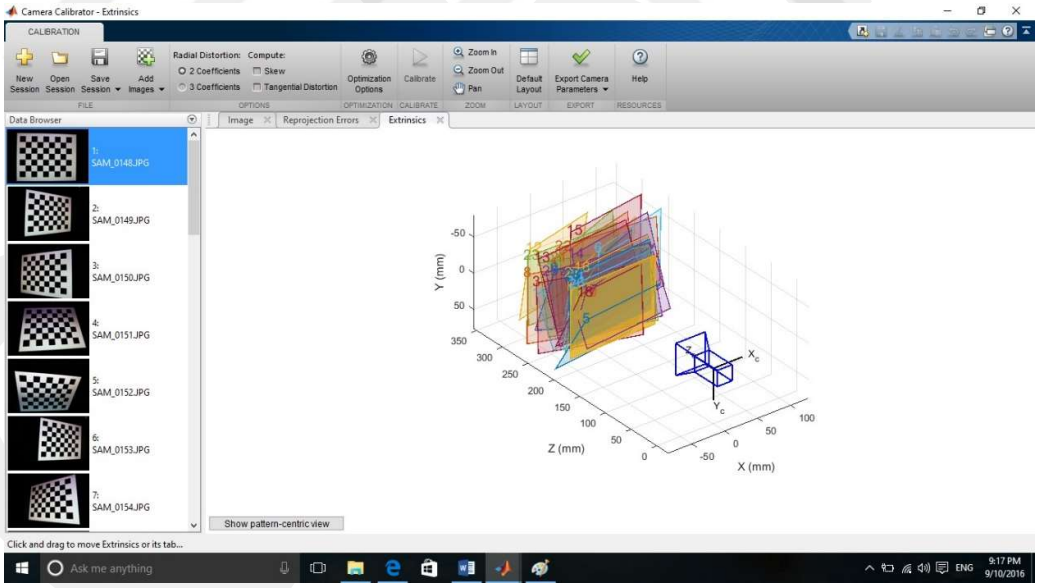


Figure 6.8: Extrinsic (Pattern – centric view 28 images)

The extrinsic parameter of the camera for 10 images with mean error 0.77 shown in the figure 6.9.

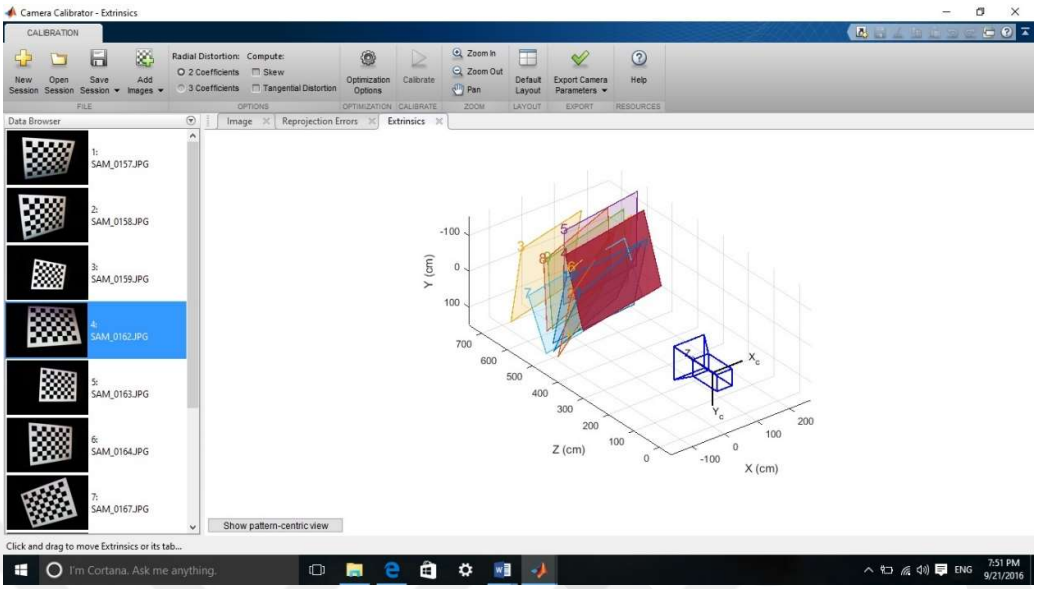


Figure 6.9: Extrinsic (Pattern – centric view 10 images)

Finally, we export the camera parameters in order to use them in the code later as shown in figure 6.10.

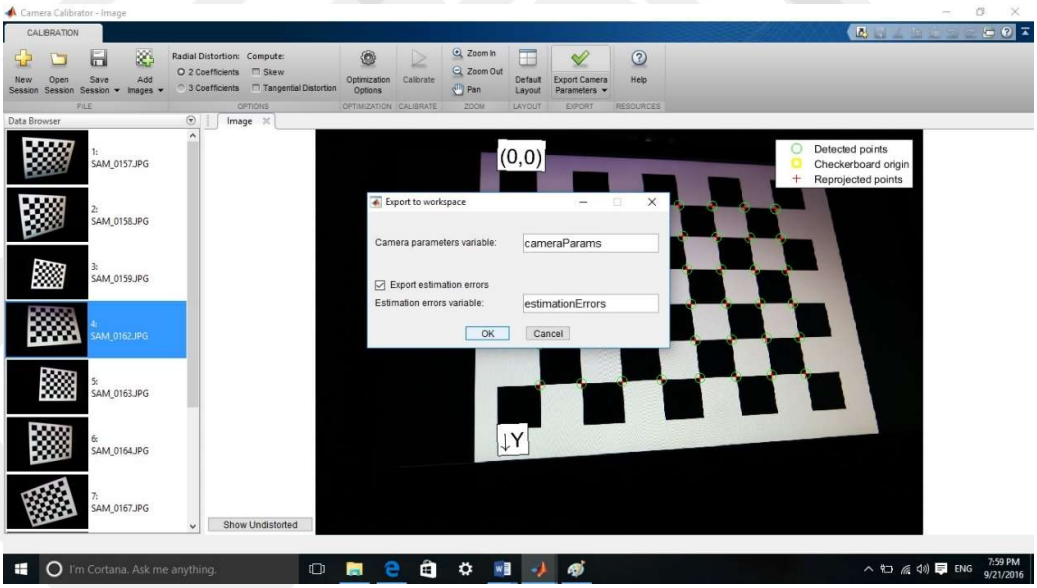


Figure 6.10: Export camera parameters

3. In order to avoid any lens distortion effects on the accuracy of the final reconstruction, MATLAB offers a simple function for this purpose which straightens any lines that may deform due to the radial distortion of the lens.

4. At this step, the algorithm detects the corresponding points between the two images. This process can be carried out in a number of ways; however, here, the motion occurs not too far from the first position, so the KLT algorithm (KANADE–LUCAS–TOMASI) is suitable to create the point correspondences.
5. Computing the fundamental matrix is carried out at this point and according to the results, the inlier points are obtained and those points match the epipolar constraints.
6. The computation of the camera position, which consists of the translation and rotation, is carried out by using the *CameraPose* function in MATLAB.
7. The three-dimensional locations of the matched points found in the fourth step are reconstructed using the triangulation function.
8. The *Plot Camera* and the *PcShow* functions are used to display the three-dimensional point cloud.
9. In order to detect the actual scale factor, the algorithm uses an object with known size, so the scene contains a ball with a known radius (of 10 cm). The *PcFitSphere* function fits a sphere to the point cloud to detect the ball.
10. The final step is the metric reconstruction, which mean the coordinates of the three-dimensional points will be in centimetre due to the actual radius of the ball which was 10 cm.

The following images show the results of the above steps with multiple different scenes, and each image has a title to clarify its identity. The time consumed by the algorithm to reach the results was different in each test, where the 1st test consumed 102 seconds, the 2nd test consumed 280 seconds, and the 3rd one consumed 131seconds. The results are shown below:

Figure 6.11: The original images

Figure 6.12: The Undistorting images

Figure 6.13: The Strongest corners from the first image

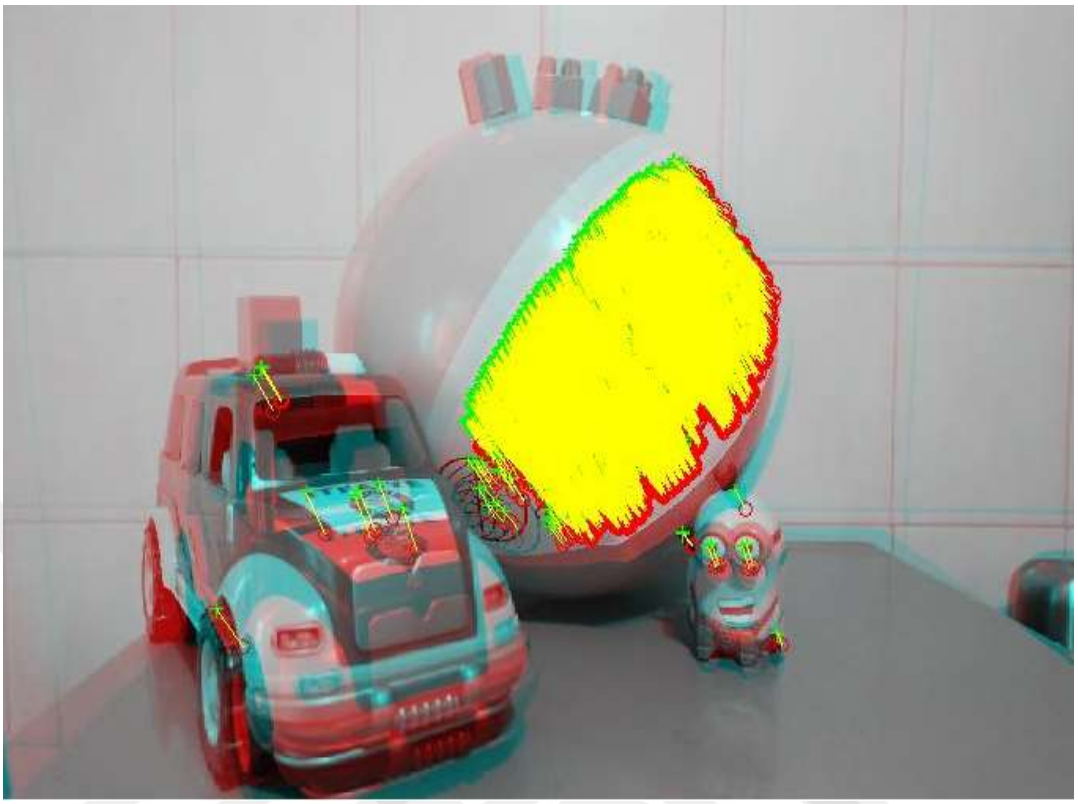


Figure 6.14: The Tracked features

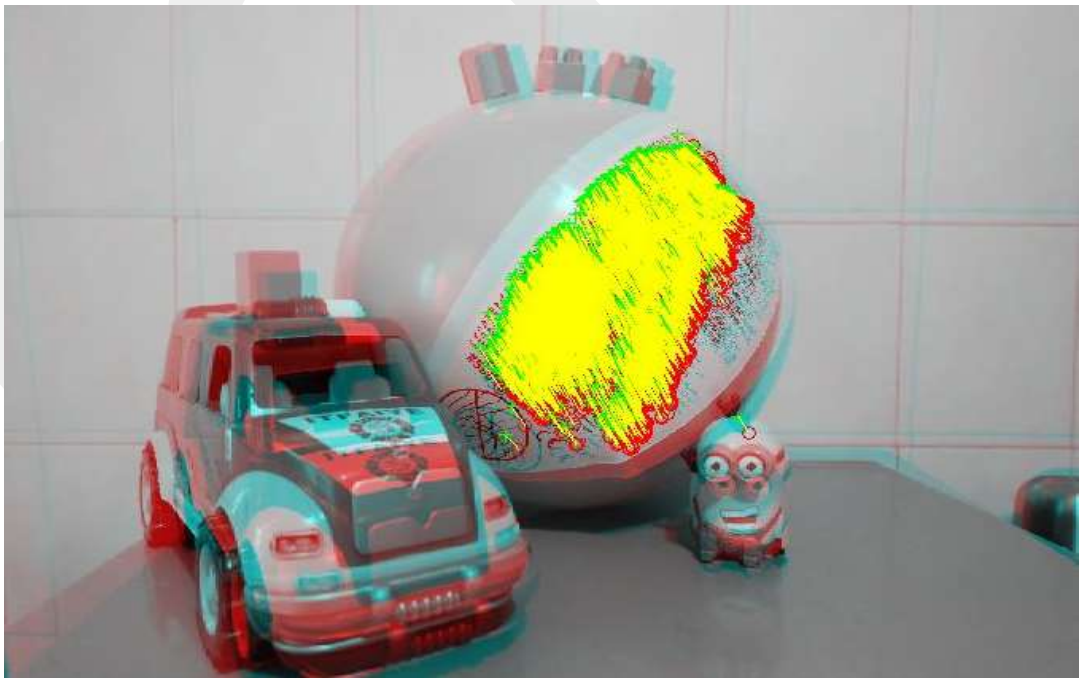


Figure 6.15: The Epipolar inlier

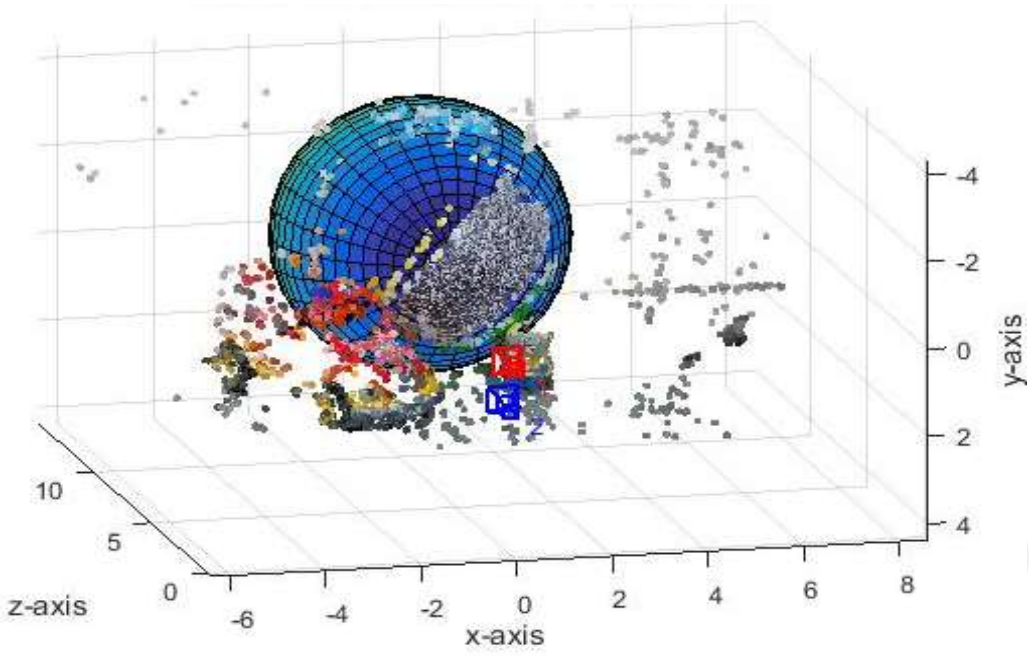


Figure 6.16: The estimated size and location of the ball

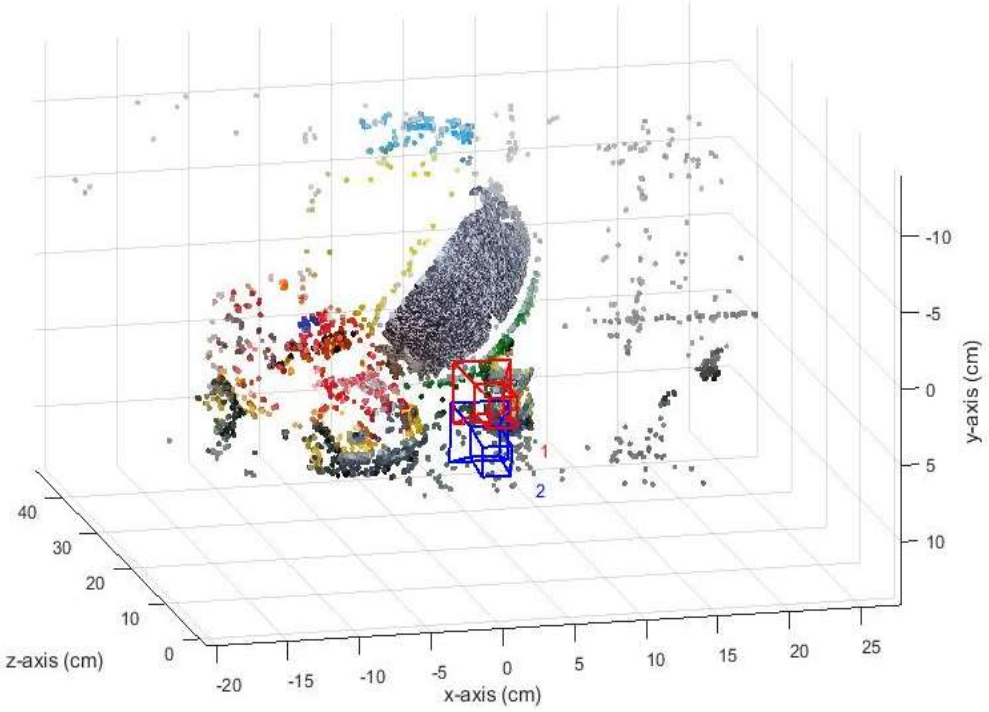


Figure 6.17 A: The metric reconstruction of the scene

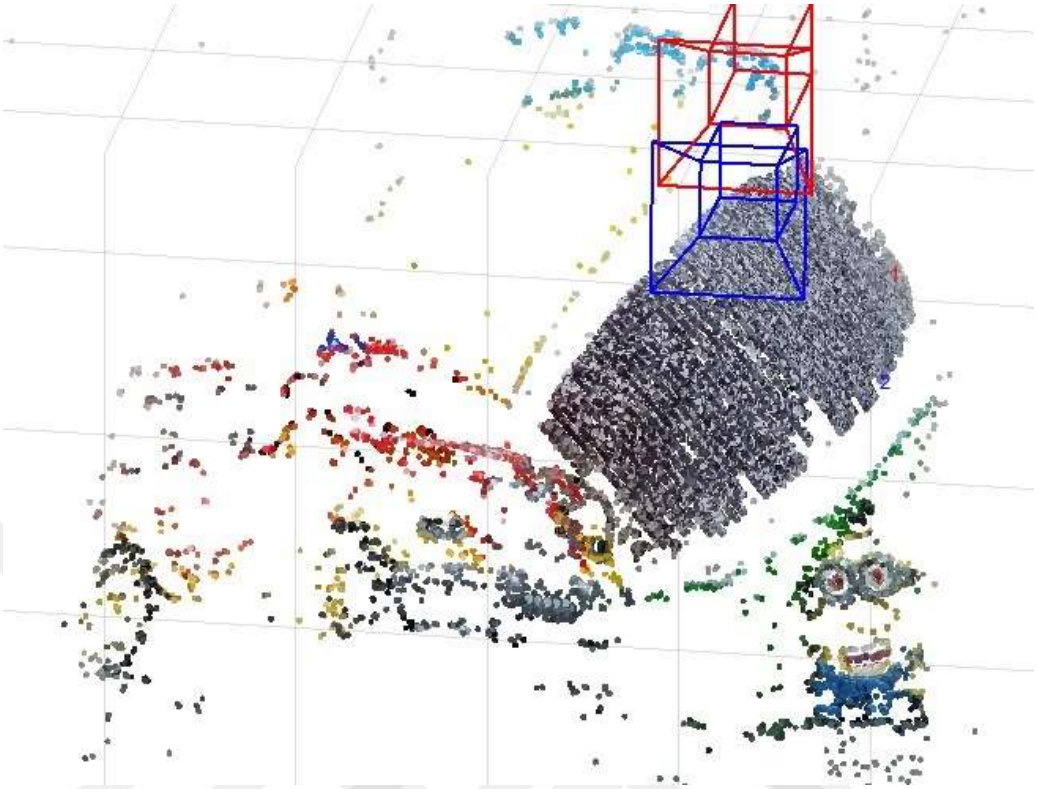


Figure 6.17 B: The metric reconstruction of the scene with another position

In order to test the algorithm with another scene, which consisted of different ball with rugged surface and objects with more details, we repeat the execution of the code and the results were as shown:

Figure 6.18: The original images (second test)

Figure 6.19: The undistorted images (second test)

Figure 6.20: The Strongest corners from the first image (second test)

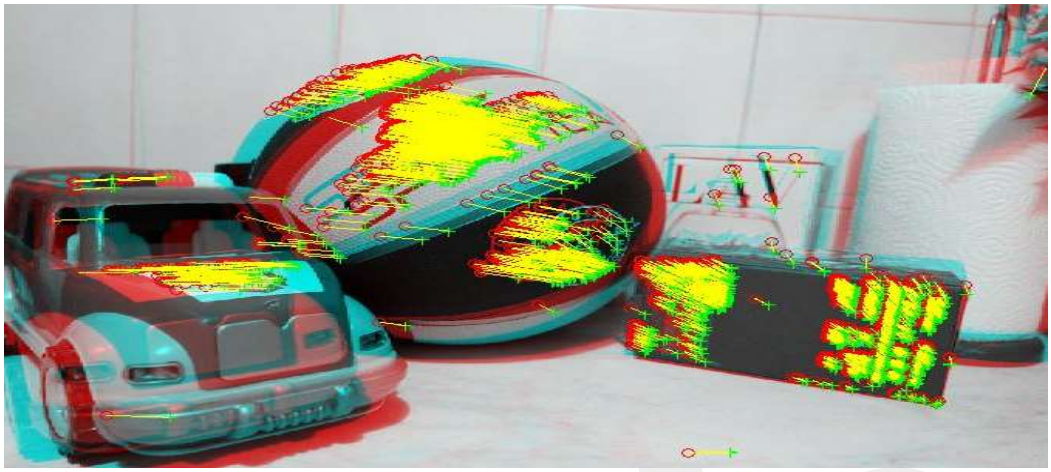


Figure 6.21: The Tracked features (second test)

Figure 6.22: The Epipolar inlier (second test)

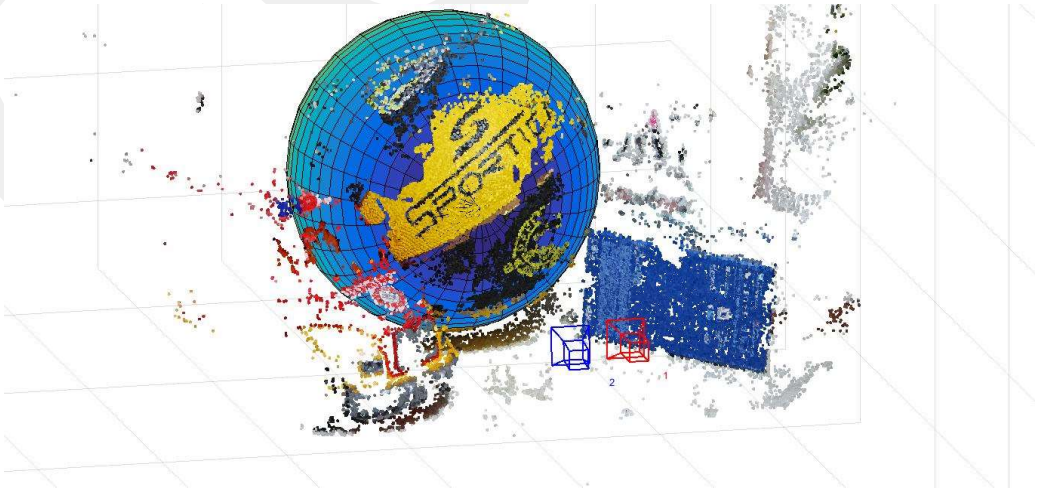


Figure 6.23: The estimated size and location of the ball (second test)

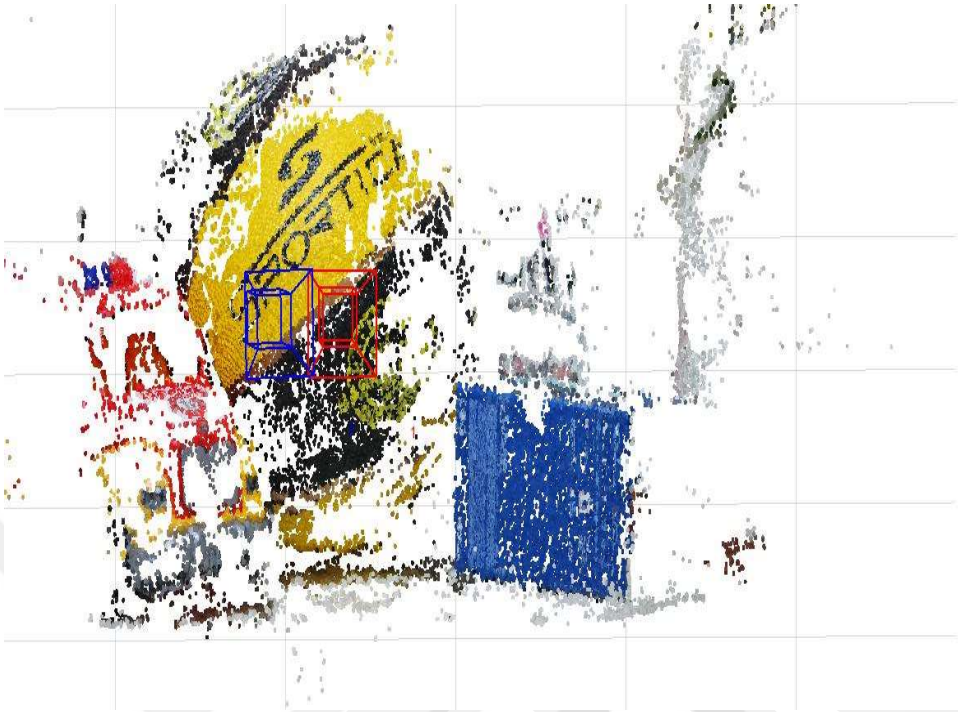


Figure 6.24 A: The metric reconstruction of the scene (second test)

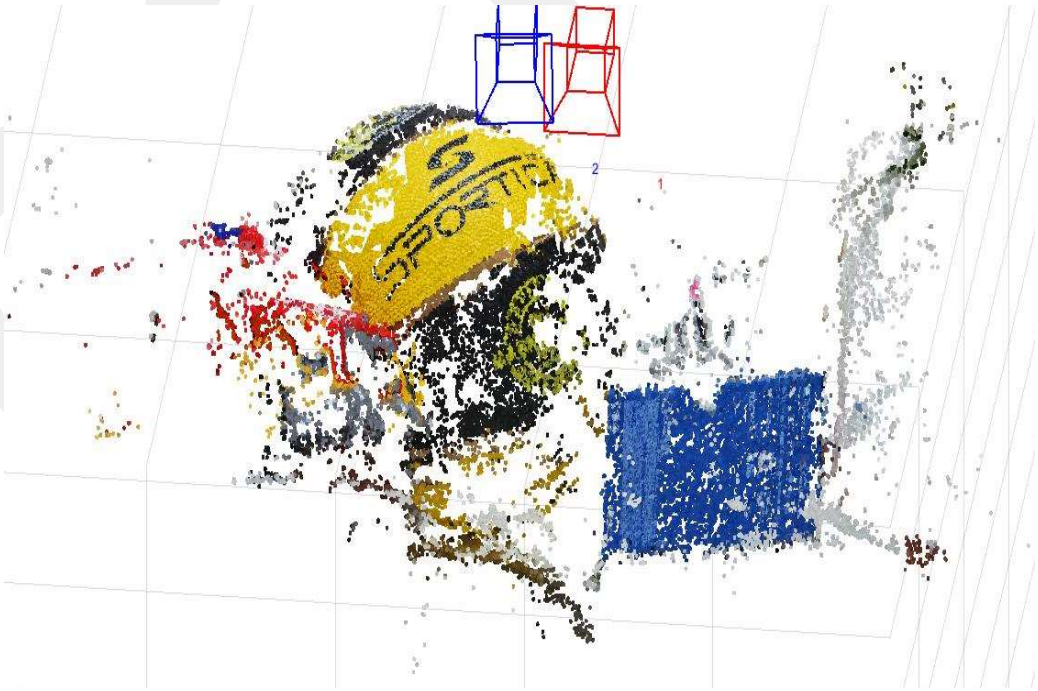


Figure 6.24 B: The metric reconstruction of the scene with another position (second test)

The first scene (Figure 6.11) contained a ball with a soft surface which had some parts with only one colour. We added some details to this ball in order to induce the algorithm to detect more matching points, and the results were as shown below:

Figure 6.25: The original images (3rd test)

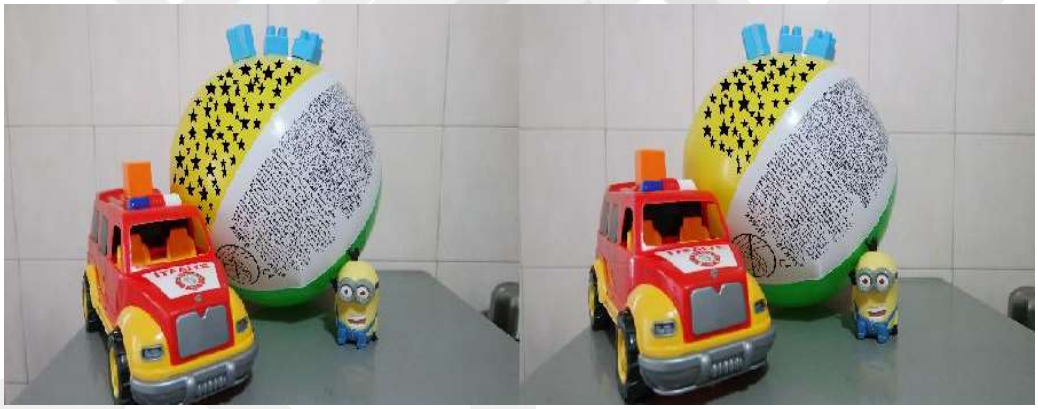


Figure 6.26: The undistorted images (3rd test)

Figure 6.27: The Strongest corners from the first image (3rd test)

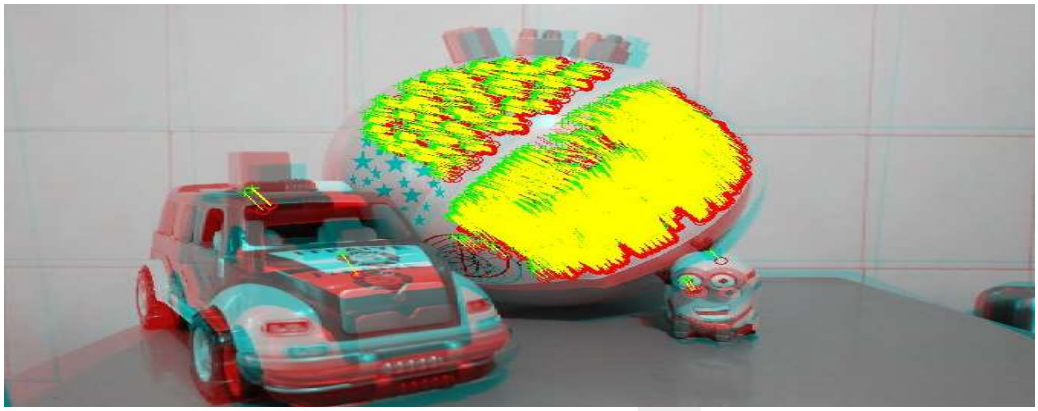


Figure 6.28: The Tracked features (3rd test)

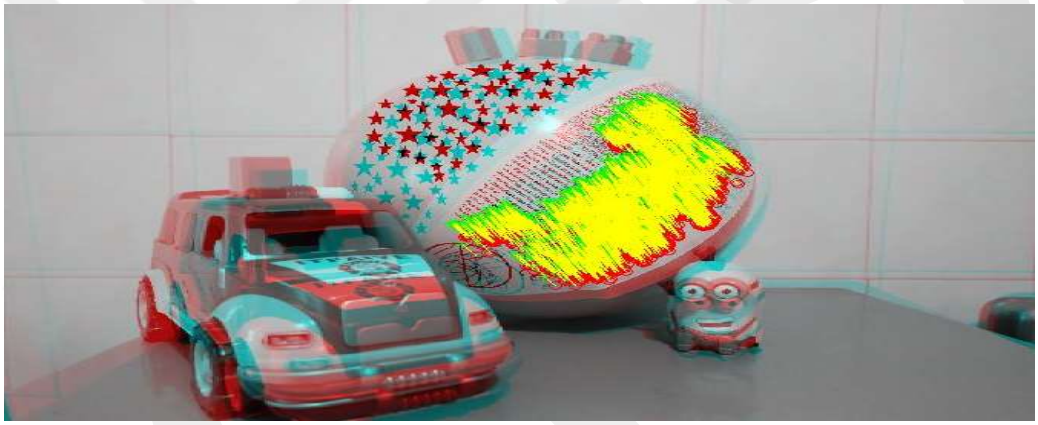


Figure 6.29: The Epipolar inlier (3rd test)

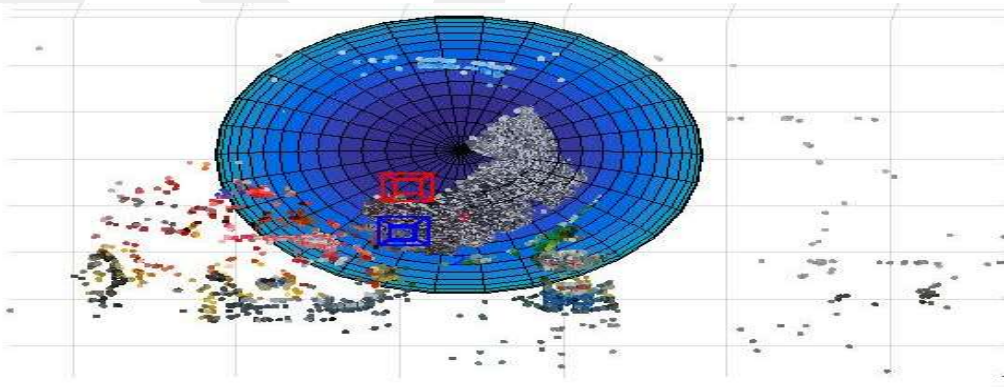


Figure 6.30: The estimated size and location of the ball (3rd test)

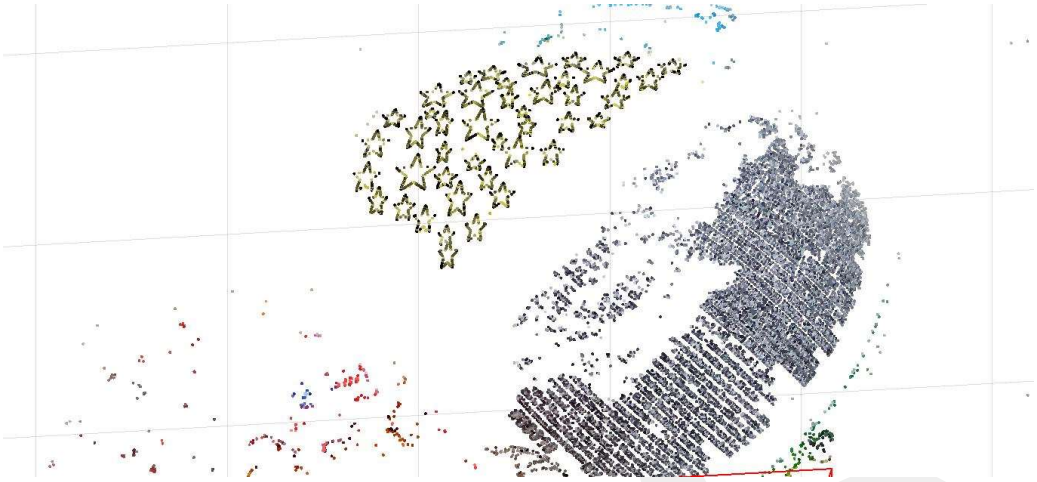


Figure 6.31 A: The metric reconstruction of the scene (3rd test)

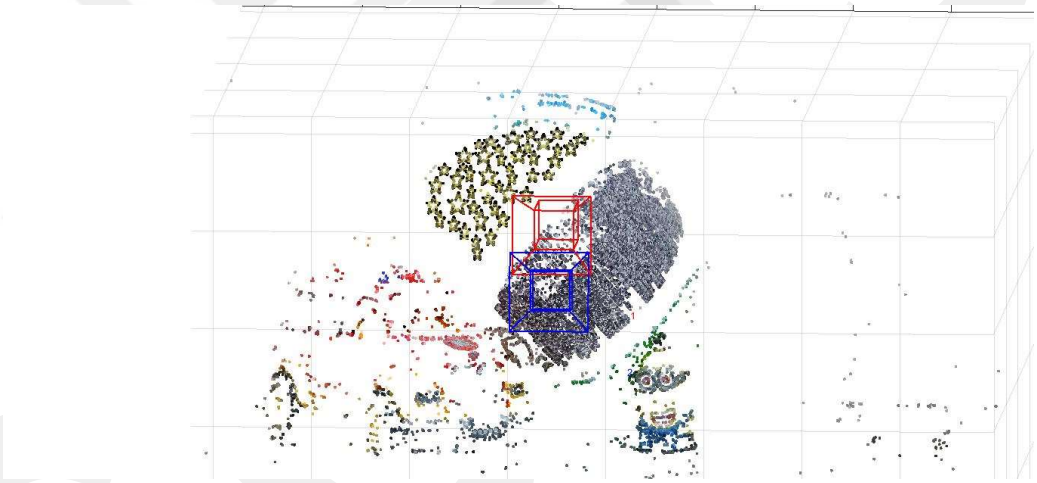


Figure 6.31 B: The metric reconstruction of the scene with another position (3rd test)

As we said in section 6.1 step 4 the distance between the images is not too far, so the KLT algorithm will work properly, but when the distance become more than 5 cm (distance between the camera and the scene was 50 cm) the algorithm fails to match the points between the images as shown in the figure 6.32.

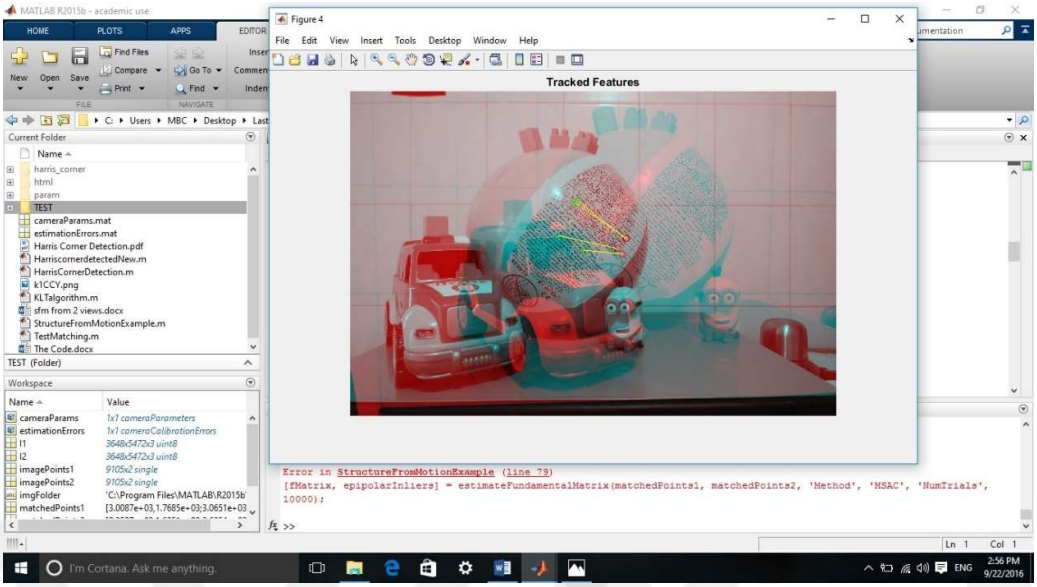


Figure 6.32: KLT algorithm error

6.2 **Real Data and Numerical Results**

- First Test

In Figure 6.11, we show two real images of a constructed composite scene. This scene represents a difficult set of data due to its soft surface. We have covered the images with matched points using the KLT algorithm technique. When the sixth step of the algorithm is applied to the matched points of the real data, the motion estimate is a single matrix (1×3) for translation and a double matrix (3×3) for rotation, as shown below:

$$t = \begin{bmatrix} -0.27 \\ 3.16 \\ 0.3 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.99 & 0.005 & -0.034 \\ -0.002 & 0.99 & 0.096 \\ 0.035 & -0.095 & 0.99 \end{bmatrix}$$

As *Zhengyou Zhang* [32] used the same technique that we followed in our method and according to the available numerical data from his method, the translation and rotation data was as shown below:

$$t = [\,-9.6, 1.85, -1.75]^T$$

$$R = [-2.1, 4.2, 1.04]^T$$

The remaining data obtained from the experimental results are as follows:

		Mean projection error 0.94	Mean projection error 0.77
All colours		19961856x3 uint8	19961856x3 uint8
Ball Properties	**Parameters**	[0.57, -0.91,10.6,3.13]	[0.57, -0.91,10.6,3.13]
	Center	[0.57, -0.91,10.6]	[0.57, -0.91,10.6]
	Radius	3.13	3.13
Camera parameters	**Radial Distortion**	[-0.097,0.1]	[-0.099,0.12]
	Tangential Distortion	[0, 0]	[0, 0]
	Estimate Skew	0	0
	Intrinsic Matrix	[3.9,0,0;0,3.9,0;2.7,1.85,1]	[3.9,0,0;0,3.9,0;2.7,1.85,1]
	Focal length	[3.9,3.9]	[3.9,3.9]
	Principal Point	[2.7,1.85]	[2.7,1.85]
Fundamental Matrıx		[1.16, -2.55,0.01;1.9, -2.2,9.5; -0.01, -2.4,0.9]	[1.3, -3.07,0.01; -7.28, -2.6,0.001; -0.01, -2.1,0.9]
Scale Factor		3.18	3.18

Table 6.1: The Numerical Result Data (First Test)

- Second Test

The second test carried out by using another scene as shown in the figure 6.18, and the numerical results data as shown below:

$$t = \begin{bmatrix} -3.3 \\ 0.14 \\ -0{,}47 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.99 & -0.02 & -0.08 \\ 0{,}02 & 0.99 & 0.015 \\ 0.087 & -0.018 & 0.99 \end{bmatrix}$$

		Mean projection error 0.77
All colours		19961856x3 uint8
Ball Properties	**Parameters**	[-0.99, -0.68,9.7,2,92]
	Center	[-0.99, -0.68,9.7]
	Radius	2.92
Camera parameters	**Radial Distortion**	[-0.099,0.12]
	Tangential Distortion	[0, 0]
	Estimate Skew	0
	Intrinsic Matrix	[3.9,0,0;0,3.9,0;2.7,1.85,1]
	Focal length	[3.9,3.9]
	Principal Point	[2.7,1.85]
Fundamental Matrıx		[2.02, 6.49, -0.001; -4.02, -3.88,0.01; 2.33, -0.01, 0,99]
Scale Factor		3.41

Table 6.2: The Numerical Result Data (2nd Test)

- Third Test

The third test carried out by using the first scene but with adding some more details as shown in the figure 6.25, and the numerical results data as shown below:

$$t = \begin{bmatrix} -0.25 \\ 3.34 \\ -0,01 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.99 & 0.002 & -0.031 \\ 0,001 & 0.99 & 0.11 \\ 0.031 & -0.11 & 0.99 \end{bmatrix}$$

		Mean projection error 0.77
All colours		19961856x3 uint8
Ball Properties	**Parameters**	[0.69, -0.98,9.64,2,98]
	Center	[0.69, -0.98,9.64]
	Radius	2.98
Camera parameters	**Radial Distortion**	[-0.099,0.12]
	Tangential Distortion	[0, 0]
	Estimate Skew	0
	Intrinsic Matrix	[3.9,0,0;0,3.9,0;2.7,1.85,1]
	Focal length	[3.9,3.9]
	Principal Point	[2.7,1.85]
Fundamental Matrıx		[1.7, 3.06 ,0.01; -5.49, -4.03,0.003; -0.01, -0.00, 0,99]
Scale Factor		3.35

Table 6.3: The Numerical Result Data (3rd Test)

6.3 **Discussion**

The image resolution used in the algorithm was 5472×3648. Initially, the algorithm begins in the first test with loading a pair of images (Figure 6.11), followed by the camera calibration stored in the camera parameters object loaded, which included the camera intrinsic matrix, the radial distortion and the estimated skew. According to the value of the skew, which here is zero, there is no distortion in the lines of the lens. The next process aims to remove any bends in the lines of the lens, and as the skew is zero, there is no need for this step (Figure 6.12). Later, the feature points will have been detected in this step from the first image (Figure 6.13) and, as mentioned above, are carried out by using the KLT algorithm.

The point tracker is created to find the correspondence points between the images (Figure 6.14). In order to specify the epipolar constraints, the fundamental matrix is estimated, and by computing the fundamental matrix, the inlier points will be established and matched to the epipolar constraints (Figure 6.15). Before the final step in the algorithm, the camera position (*R*, *t*), which represents the external parameters, are computed. Later, by using the sphere function to fit the point cloud in order to find the size and location of the ball in the scene (Figure 6.16).

Finally, the coordinates of the three-dimensional points in centimetres are determined according to the actual size of the ball (Figure 6.17 A and B). The final result of reconstruction of the three-dimensional model was not good due to the holes in the model; therefore, it was necessary to fill the uncovered areas

We used the SFM technique in computer vision to reconstruct the three-dimensional model from the two-dimensional images based on different methods as the previous chapter demonstrated, (such as Frank et al [33], Masahiro [37], and Zach et al [40]), and all of these methods have used the SFM technique based on a variety of approaches. However, these approaches do not meet the criteria as set out in this thesis, which introduces the use of structure from motion based on matching the correspondence points between the only two images as done by Zhengyou Zhang [32]. Due to the lack of the data from the mentioned method, it was not possible to compare with the results of this thesis.

Zach et al [40] in their methods using four different datasets, and by adding more points where are reduced the of error, except the third dataset where the error is increased, and this issue is left without explaining in their paper. Those results shown in the table 6.4.

Dataset	#Images	#3D points	Init. Image error	#Added points	Final image error
1	175	43553	2.17	1497	2.14
2	186	47756	6.18	5605	4.89
3	99	31876	1.77	5747	6.75
4	191	60997	3.3	1556	2.4

Table 6.4: The results of Zach et al method[12]

[12] *Table Source:Reference* 40

In our method we used different types of scenes in order to demonstrate the behaviour of the algorithm. The numbers of three-dimensional points, which are the algorithm obtained from the first scene (figure 6.11), are 19333 points. After adding more details to the first scene, and by using the same algorithm (figure 6.25), the numbers of three-dimensional points are increased from 19333 points to 22195 points. In the second test we using different scene (figure 6.18), which is have more colours and details, the result of using such a scene was obtaining more 3D points. Where the numbers of 3D points are increased from 22195 points to 59413 points. Table 6.5 is clarifying all those results which were carried out from the three tests.

The original scene (1st Test)			The original scene after modifying (3rd Test)			Different scene with more details (2nd Test)		
3D points	Image points	Matched Points	3D points	Image points	Matched Points	3D points	Image points	Matched Points
19333	30306	19333	22195	39402	22195	59413	247519	59413

Table 6.5: Numbers of points according to different scenes

Zach et al [40] in their method were added more points in order to reduce the rate of error, where the approach of the proposed method in this thesis is motivate the algorithm to obtain more matched points by using scenes rich in details.

The limitations of the previous algorithm were found in the fourth step of the feature detection, where the KLT algorithm will not work probably if the space between the obtained images is too great (Figure 6.18). Next, the tracker features had some difficulties detecting the soft surfaces in the scene (Figure 6.14), so we added some details to this surface in order to motivate the algorithm to detect more matching points (Figure 6.25). Then, the same steps which mentioned above were executed. As the final result of the third test shown in the figures 6.31 A, B, the algorithm detects more points and reconstruct new model with more points.

The second test was carried out by using a different scene (Figure 6.18), after executing the algorithm, the results were more accurate than the first test due to the details of the scene which was had more colours than the first scene (figure 6.11).

CHAPTER VII

CONCLUSION AND FUTURE WORK

7.1 **Conclusion**

In this thesis, we explored a variety of methods and techniques that aim to reconstruct the optimum three-dimensional model from two-dimensional data. This thesis concentrated on the structure from motion based on two views. The experimental results in the previous chapter have some limitations that need improvement and complementary solutions. Due to the limited time, there is no possibility to cope with the issues that are mentioned in the previous chapter section 6.3.

The results of the experiments show the insufficiency of KLT algorithm when the distance between images becomes more than 5 cm. Also, we figure out the possibility of reducing the rate of reprojection error by removing the images that have the biggest rate of error.

The experimental results are consisting from three stages. The first stage is done by using a scene with soft surfaces, the performance of the algorithm shows some deficiencies with the soft surfaces which are have few details. The second stage is done by using different scene with objects which have more details and rough surfaces, the algorithm results become more accurate than the first scene. The third stage is done by using the first scene of the first stage but after adding more details for surface of the ball in order to motivate the algorithm to detect more points, the results become more accurate than the results of the first stage. The experiments are showing the performance of the algorithm with different scenes and demonstrate the way of improving the algorithm.

In spite of the limitations mentioned above, the algorithm creates three-dimensional models that depend on only two views with the model being meaningful according to the original scene. Moreover, the work of the algorithm is quite good due to the rating of the *mean projection error,* which was 0.94 and decreased into 0.77, as shown in Table (6.1).

7.2 **Future Work**

Researchers in this field may use this thesis in investigations of two-dimensional to three-dimensional conversion algorithms. They can deal with the limitations mentioned herein by finding alternative algorithms instead of using the KLT algorithm so as to cope with widely-spaced images, or improve the 3D-model for greater accuracy. Also, they can estimate the depth information using the other algorithms that mentioned in the figure 3.1 in third chapter, and comparing the results with the current one in order to clarify the strengths and weaknesses of each algorithm.

BIBLOGRAPHY

LIoyd, S., *Binocular stereo algorithm based on the disparity-gradient limit and using optimization theory, Image and vision computing,* Hirst research center, Wembley,1985.

Nishikawa, A., and Miyazaki, F., *Active Detection of Binocular Disparities*, International Workshop on Intelligent Robots and Systems, IEEE/RSJ, P. 3-5., Osaka, Japan, 1991.

Hu, T., and Huang, M., *A New Stereo Matching Algorithm for Binocular Vision*, International Journal of Hybrid Information Technology, china, 2010.

Pop, M., et al, *Efficient Perspective-Accurate Silhouette Computation and Applications,* Medford, Massachusetts, 2001.

Cheung, K., et al, *Shape-From-Silhouette Across Time*, Carnegie Mellon University, 2005.

Haro, G., Shape from Silhouette Consensus, Barcelona, Spain, 2012.

Hartner, A., et al, *Object Space Silhouette Algorithms***,** University of Utah, 2003.

Nayar, S., *Shape from Focus*, The Robotics Institute, Pittsburgh, Pennsylvania, 1989.

Lee, J., et al, *Implementation of A Passive Automatic Focusing Algorithm for Digital Still Camera*, Korea Institute of Science and Technology, 1995.

Eltoukhy, H., and Kavusi, S., *A Computationally Efficient Algorithm for Multi-Focus Image Reconstruction,* Stanford University, Stanford, 2002.

Bueno, M., et al, *Fast autofocus algorithm for automated microscopes***,** optical engineering, 2005.

Hwang, T., et al, *A depth recovery algorithm using defocus information*, Harvard university, Cambridge, 1989.

Favaro, P., and Soatto, S., *A Geometric Approach to Shape from Defocus*, IEEE Transactions On Pattern Analysis and Machine Intelligence, Vol. 27, No. 3, 2000.

Zhang, L., and Nayar, S., *Projection Defocus Analysis for Scene Capture and Image Display,* Columbia University, 2007.

Tao, M., et al, *DepthfromCombiningDefocusandCorrespondenceUsingLight-FieldCameras*, University of California, Berkeley, 2012.

Dellaert, F., et al, *Structure from Motion without Correspondence*, Carnegie Mellon University, Pittsburgh, 1999.

Farneback, G., *Two-Frame Motion Estimation Based on Polynomial Expansion*, Sweden, 2002.

Schonberger, J., Frahm, J., *Structure-from-Motion Revisited*, University of North Carolina at Chapel Hill, 2016.

REFERENCES

[1] Ji Zhang et al, (2011), "Pose-Free Structure from Motion Using Depth from Motion Constraints", IEEE TRANS. ON IMAGE PROCESSING, VOL. 20, NO. 10, PP 2937-2953.

[2] Tony Jebara et al, (1999), "3D Structure from 2D Motion", IEEE SIGNAL PROCESSING MAGAZINE, VOL. 16, ISSUE: 3, PP 66 – 84.

[3] Scott Squires, (2011), "2D to 3D Conversion**",** ADAPT Web Site, effectscorner.blogspot.com.

[4] Jon Karafin, (2011), "State-Of-The-Art 2D to 3D Conversion and Stereo VFX", International 3D Society University, Presentation 3DU-Japan event in Tokyo.

[5] Olivier Faugeras and Quang-Tuan Luong, (2001), "The Geometry of Multiple Images".

[6] Simon J.D. Prince, (2012), "Computer Vision Models, Learning and Inference ".

[7] Rich Radke, (2014), "ECSE-6969 Computer Vision for Visual Effects", Rensselaer Polytechnic Institute.

[8] Milan Sonka et al, (1993), "Image Processing, Analysis and Machine Vision"**.**

[9] Lee Sang-Hyun et al, (2014), "Conversion 2D Image to 3D Based On Squeeze Function and Gradient Map ", ISSN: 1738-9984 IJSEIA, Vol.8, No.2, PP.27- 40.

[10] Richard Szeliski, (2010), "Computer Vision Algorithms and Applications".

[11] Forsyth and Ponce, (2003), "Computer Vision a Modern Approach**"**.

[12] Hai Tao, (2005), "Image Analysis and Computer Vision", Department of Computer Engineering University of California, Santa Cruz.

[13] Madhuri A. Joshi, (2010), "Digital Image Processing: An Algorithmic Approach".

[14] Mubark Shah, (1997), "Fundamental of Computer Vision", Department of Computer Science University of Central Florida.

[15] Shapiro and Stockman, (2000), "Computer Vision**"**.

[16] Sa Lioyd, (2003), "Binocular Stereo Algorithm Based on the Disparity-Gradient Limit and Using Optimization Theory" GEC Research Ltd. Research Centre, East Lane, WEMBLEY, MIDDX HA9 7PP, UK.

[17] Z. Zhang, (2000), "Flexible New Technique for Camera Calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL.22 No.1, PP.1330-1334.

[18] Peter Sturm et al, (2011), "Camera Models and Fundamental Concepts Used in Geometric Computer Vision", HAL Id: inria-00590269, VOL. 6, Nos. 1–2, DOI: 10.1561/0600000023, PP 1–183.

[19] Ijsselsteijn et al, (2002), "Human Factors and Quality Issues of Stereoscopic Broadcast Television", Deliverable ATTEST/WP5/01, Eindhoven University of Technology, Netherlands.

[20] Q. Wei, (2005), "Converting 2D to 3D: A Survey", Research Assignment for Master Program Media and Knowledge, Engineering of Delft University of Technology.

[21] LONGUET-HIGGINS, (1981), "Computer Algorithm for Reconstructing a Scene from Two Projections," Nature 293, PP:133–135.

[22] Prazdny, (1983), "On The Information in Optical Flows," Computer Vision, Graphics, and Image Processing VOL 22, No 2, PP:239 – 259.

[23] Tsai et al, (1981), "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch," Acoustics, Speech and Signal Processing, IEEE Trans on 29, PP:1147-1152.

[24] Soatto et al, (1998), "Reducing "structure from motion" a general framework for dynamic vision. Part 1: modelling.," International Journal of Computer Vision 20, PP:993-942.

[25] Joseph and J. Sean, (2013), "Structure from Motion in Computationally Constrained Systems", Micro- and Nanotechnology Sensors, Systems, and Applications V, edited by Thomas George. Saiph Islam, K. Dutta, Proc. of SPIE Vol. 8725, 87251G.

[26] Qian et al, (2004), "Structure from Motion Using Sequential Monte Carlo Methods," International Journal of Computer Vision 59, PP:5-31.

[27] Qian et al, (2005), "Bayesian Algorithms for Simultaneous Structure from Motion Estimation of Multiple Independently Moving Objects," IEEE Trans on Image Processing 14, PP:94-109.

[28] Roy-Chowdhury et al, (2005), "Statistical bias in 3-d reconstruction from a monocular video", IEEE Transactions on Image Processing 14, PP:1057-1062.

[29] Zucchelli, (2002), "Optical flow based structure from motion", Numerical Analysis and Computer Science. Stockholm: (Royal Institute of Technology).

[30] Astrom et al, (2000), "Solutions and Ambiguities of the Structure and Motion Problem for 1D Retinal Vision," J. Math. Imaging Vis. VOL 12, No 2, PP:121-135.

[31] Li et al, (2006), "Structure from Planar Motion", IEEE Trans on Image Processing 15, PP:3466-3477.

[32] Zhengyou Zhang, (1997), "Motion and Structure from Two Perspective Views: From Essential Parameters to Euclidean Motion Via Fundamental Matrix", Journal of the Optical Society of America, VOL.14, No.11, PP:2938-2950.

[33] Frank Dellaert et al, (2000), "Structure from Motion without Correspondence", Computer Science Department & Robotics Institute Carnegie Mellon University, Pittsburgh PA 15213, IEEE 1063-6919/00.

[34] M. Pollefeys, (1999), "Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Intrinsic Camera Parameters", *Int. J. of Computer Vision*, VOL.32, No.1, PP:7-25.

[35] K. Kutulakos, (1999), "Theory of Shape by Space Carving", In Proc. Seventh Int. Conf. on Computer Vision, PP:307-314.

[36] D. Lowe, (1991), "Fitting Parameterized Three-Dimensional Models to Images", IEEE Trans. on Pattern Analysis and Machine Intelligence, VOL.13, No.5, PP:441-450.

[37] Masahiro Tomono, (2005), "3D Localization and Mapping Using a Single Camera Based on Structure-from-Motion with Automatic Baseline Selection", IEEE PP: 3342-3347, DOI: 10.1109/ROBOT.1570626, International Conference on Robotics and Automation Barcelona, Spain.

[38] C. Tomasi & J. Shi, (1994), "Good Features to Track," Proc. of CVPR'94, PP: 593-600.

[39] T. Kanade & C. Tomasi, (1992), "Shape and Motion from Image Streams under Orthography: A Factorization Approach", International Journal of Computer Vision, VOL.9, No.2, PP:137-154.

[40] Christopher Zach et al, (2012), "Discovering and Exploiting 3D Symmetries in Structure from Motion", IEEE Conference, Computer Vision and Pattern Recognition, DOI: 10.1109/CVPR6247841, PP: 1514-1521.

[41] Klingner et al, (2013), "Street View Motion-from-Structure-from-Motion", IEEE Int. Conference on Computer Vision, DOI 10.1109/ICCV, PP: 953-960.

[42] Yongjun Zhang et al, (2015), "Optimized 3D Street Scene Reconstruction from Driving Recorder Images", Remote Sens. 7, PP: 9091-9121, ISSN 2072-4292 DOI: 10.3390/RS70709091.

[43] Derek Hoiem, 2012, "Feature Tracking and Optical Flow", Computer Vision, CS 543/ECE 549, Illinois Univ.

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Sure Name, Name: Iaswi, Muthana

Date and Place of Birth: 1985 Iraq

Martial State: Married

Phone: 05061067980 / +9647703742624

Email: Muthana2085@yahoo.com

EDUCATION

Degree	Institute	Year of Graduation
M.Sc.	Çankaya Univ., Information Technology	2016
B.Sc.	Kirkuk Univ., Computer Science	2007
High School	Tameem Preparatory School	2003

WORK EXPERIENCE

Year	Place	Enrolment
2010 – Present	Kirkuk University	Programmer
2008 – 2009	Ministry of Education	Assist Programmer
2007 – 2008	Almas Company	Assist Programmer

FOREIGN LANGUAGE

English, Beginner Turkish

HOBBIES

Football, Electronic Games, Traveling