

**3D MODELLING USING REAL IMAGES**

by

**MUHAMMAD MUAZ KHAN FARRUKH AHMED KHAN**

051- 11 – 115672 051- 11 – 115678

Under the supervision of:

**Mr.ZulfiqarAli Bangash**

A Final Year Project Report is

Submitted in Partial Fulfilment of the

Requirements for the Degree of

**Bachelor of Science in Computer Sciences.**

**Department of Computing & Technology**

**Iqra University, Islamabad Campus**

November, 2015

**Approval Sheet**

Date: 10/11/2015

**Title of Final Year Project:** 3D MODELLING USING REAL IMAGES

**Program:** BSCS

**Student’s Name:**

MUHAMMAD MUAZ KHAN

FARRUKH AHMED KHAN

**Registration No:**

051 11 115672

051 11 115678

The evaluation jury hereby approves this final year project in partial fulfilment of the requirements for the degree of Bachelors of Sciences in Computer Sciences.

Date: \_\_ \_\_\\_\_ \_\_\\_\_ \_\_ \_\_ \_\_

**Mr.ZulfiqarAli Bangash**, *Project Supervisor*

Date: \_\_ \_\_\\_\_ \_\_\\_\_ \_\_ \_\_ \_\_

**Mr.**,*Internal Examiner*

Date: \_\_ \_\_\\_\_ \_\_\\_\_ \_\_ \_\_

**Mr.**, *External Examiner*

Date: \_\_ \_\_\\_\_ \_\_\\_\_ \_\_ \_\_

**Ms.Tooba**,*Project Coordinator*

Date: \_\_ \_\_\\_\_ \_\_\\_\_ \_\_ \_\_

**Dr.KhawajaTauseefTasneem**, *HOD-CS*

# Declaration

We, hereby, declare that this work, neither whole nor in part, has been copied from any source. It is further declared that we have prepared this report entirely on the basis of our personal efforts made under the sincere guidance of our teachers especially our supervisor Mr. Zulfiqar Ali Bangash. If any part of this thesis is proved to be copied out from any source or found to be a reproduction of others, we will stand by the consequences. No potion of the work presented has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Names of students:

Muhammad Muaz Khan

Farrukh Ahmed Khan

# Dedication

We dedicate this project to Allah (SWT), The Almighty,

Unto Whom belongs whatsoever is in the earth

and whatsoever is in the heavens.

And to the greatest benefactor and teacher of the humanity

Hazrat Muhammad (SAW).

Also, to our ever-loving and beloved parents

whose benedictions are a part of our beings.

# 

# Acknowledgement

The Prophet (PBUH) said, “**He has not thanked Allah who has not thanked people**.” No doubt, we, as Muslims are always thankful to Allah (SWT) and our fellow people.

We feel honored to present our humble gratitude to our parents who have never stopped praying for our success since we stepped into this world.

We bow our heads in the approbation of all our esteemed teachers who conferred us with the light of knowledge when we sought it, especially our favourite teacher Mr. Zulfiqar Ali Bangash. He has always been very kind to guide us from day one. He benefited us with his valuable advice and has inculcated in us the real taste for CS studies. He taught us how to accomplish our task with a high sense of responsibility. His guidance paved our path and we felt harmony in our project work.

We would also like to thank our senior university mates who drew our attention to some useful technical issues regarding the project.

**Abstract**

Today’s life has become more logical and hence understandable due to the advancement in computing technology.

The field of image processing reciprocated the approach of camera because it has introduced the advanced features of camera modelling. It has also comprised three dimensional surface of an object and formed a product called as 3D model.

Nowadays, it is really very difficult to buy a new car or furniture for your home. Almost all the companies attract you by presenting their 3D Models. In many cases, these models are rather animated not real which are prepared by highly paid experts in the field. When a customer chooses a car or a piece of furniture on a website and places an order to buy it, many a time he gets disappointed because there is a huge difference between the real item and the one shown on the website.

We have tried our best to solve this problem as we have developed a real 3D model which shows the real image without animations. We will prove its originality with our results.

**TABLE OF CONTENTS**

**Approval Sheet** …………………………………………………………………………….…ii

**Declaration** ………………………………………………………………………………….....iii

**Dedication** …………………………………………………………………………………......iv

**Acknowledgement** ……………………………………………………………………………..v

**Abstract**………………………………………………………………………………………...vi

**List of figures** …………………………………………………………………………………..ix

**Chapter 1: Introduction**………………………………………………………………………..1

1.1 ProjectOverview......................................................................................................................2

1.2 Purpose....................................................................................................................................2

1.3 Current 3D Models..................................................................................................................3

1.4 User’s Description...................................................................................................................3

1.5Project Goal.....................................................................................................................….....3

1.6 Design and Implementation Challenges..................................................................................4

**Chapter 2: Requirements Analysis**………………………………………………………….....5

2.1 Software requirements: ...........................................................................................................6

2.2 Hardware requirements: ..........................................................................................................6

2.3 Tools and Technologies: .........................................................................................................6

2.4 Functional Requirements: .......................................................................................................6

2.4.1 Pair Saving GUI: ..................................................................................................................6

2.4.2 Disparity Track Bar Window: ..............................................................................................7

2.4.3 3D View of Object: ..............................................................................................................7

2.4.4 Chessboard detection………………………………………………………………………7

2.4.5 Stereo cameras……………………………………………………………………………..8

2.5 Non-Functional Requirements.................................................................................................8

2.5.1 Calibrated rectified image………………………………...………………………………..8

**Chapter 3: Literature Review** …………………………………………………………………9

3.13D Modeling Theories...................................................................................................10

**Chapter 4: System Design** ……………………………………………………………………..12

4.1 Use Case Diagram: .................................................................................................................13

4.1.1 Overview: ............................................................................................................................13

4.1.2 Purpose: ...............................................................................................................................13

4.2 State Transition Diagram: .......................................................................................................14

4.2.1 Overview: ……………………………………………………………………………….14

**Chapter5: Implementation** …………………………………………………………………….15

5.1 Pair Saving: ……………………………………………………………………………..........16

5.1.1 “Addweighted” Function: …………………………………………………………….........16

5.1.2 Parameters: …………………………………………………………………………...…….16

5.2 Calibration: …………………………………………………………………...……………....17

5.2.1 Findchessboardcorners Function: ………………….………………….................................17

5.2.2 Parameters: ………………………………………………………………………………....18

5.3 Stereo rectification and Disparity: …………………...…………………………………….....18

5.3.1 Remapping: …………………………………………………………………..……….....….18

5.3.2 Parameters: ……………………………………………………………………..……….…..18

5.3.3 Stereo Block Matching Algorithm: ………………………………..……………………......18

Parameters: ………………………………………………………………..………………...….…18

5.3.4 Stereo Semi Block Matching Algorithm: …………………….………………………..........19

Parameters: ……………………………………………..…………………………...…………….19

5.4 Reprojection: ………………………………………………...………………………………..19

Parameters: ………………………………………………………...…….......................................19

5.5 Point Cloud Library: …………………………………….……………………………...……..20

5.5.1 Visualization Library: …………………………………………...…..……………………... 20

5.5.2 3D Viewer: …………………………………………………………...…………………..….20

**Chapter 6: TestingProcess** …………………………………………………………………....….21

6.1 Pair Saving of Images: …………………………………………………………..…...……...…22

6.2 Calibrated and Rectified: ……………………………………………………..………………...22

6.2.1 Stereo Left and Right Images: ………………………………….………………...………..... 23

6.2.1.1 Left Images:………………………………………………….…………..………………….23

6.2.1.2 Right Images: ……………………………………………………………..……………...…24

6.2.2 Rectified: ……………………………………………………………………………………...25

6.2.2.1 Good Rectified Image with Disparity…………………………................................……….25

6.2.2.2 Disparity with RMS Error: ……………………….…………………………………………27

6.3Stereo Rectification and Live Disparity…………………………………………….…………....28

6.3.1Object Live Disparity: …………………………………………………………….…..……….29

6.4My Point Cloud and 3D Result…………………………………………………………..............29

**Chapter 7: Conclusion and Future Work**……………………………………………………...31

7.1 Conclusion ………………………………………………………………….………..………32

7.2 Future Work ……………………………………………………………………………...…..32

7.2.1 Product form…………………………………………………………………………..…....32

**References**………………………………………………………………………...……….……...33

**LIST OF FIGURES**

Figure 2.1: Pair Saving GUI................................................................................................6

Figure 2.2: Disparity Track Bar Window............................................................................7

Figure 2.3: 3D View of Object ............................................................................................7

Figure 2.4: Detected chessboard ……………………………………..................................7

Figure 2.5: Calibrated Rectified Image ................................................................................8

Figure 2.6: Live disparity of object.......................................................................................8

Figure 4.1: Use Case Diagram ……….........................................................................13

Figure 4.2: State Transition Diagram............................................................................14

Figure 6.1: Left and Right view of the Chessboard taken by stereo cameras................22

Figure 6.2: Mix view of the chessboard image.............................................................22

Figure 6.3: Left Image of chessboard (Left 01)...........................................................23

Figure 6.4: Left Image of chessboard (Left 02)...........................................................23

Figure 6.5: Left Image of chessboard (Left 03)...........................................................23

Figure 6.6: Left Image of chessboard (Left 04)...........................................................24

Figure 6.7: Right Image of chessboard (Right 01)........................................................24

Figure 6.8: Right Image of chessboard (Right 02)........................................................24

Figure 6.9: Right Image of chessboard (Right 03)........................................................24

Figure 6.10: Right Image of chessboard (Right 04).......................................................25

Figure 6.11: Good Rectified result of image (01)..........................................................25

Figure 6.12: Disparity of the rectified image (01).........................................................26

Figure 6.13: Good Rectified result of image (02)..........................................................26

Figure 6.14: Disparity of the rectified image (02)........................................................26

Figure 6.15: Good Rectified result of image (03)..........................................................27

Figure 6.16: Disparity of the rectified image (03).........................................................27

Figure 6.17: RMS error and disparity............................................................................28

Figure 6.18: Left image of the chair..............................................................................28

Figure 6.19: Right image of the chair............................................................................28

Figure 6.20: Object Live Disparity................................................................................29

Figure 6.21: 3D view of object (01)..............................................................................29

Figure 6.22: 3D view of object (02)...............................................................................30

Figure 6.23: 3D view of object (03)...............................................................................30

Figure 6.24: 3D view of object (04)...............................................................................30

# 

# Chapter 1

# Introduction

* 1. **Project Overview**

3D modelling belongs to the field of image processing in computer science in which images are processed for any alterations. An image is always defined in a 2Dimensional function. It is always processed in the form of a signal with a different set of attributes involved in it. The concept of 3D modelling applies to converting the 2Dimensional image into a 3D model. For this purpose, we use several 2Dimensional images to form a 3D model. While constructing a 3D model, there is a need for a set of images of a particular object which are joined or attached at accurate points to construct a model of it. Here the information of each image plays an important role. Each 2Dimensional image is captured from a level in which the data and figures are most appropriate, clear and are nearby the upcoming figures and values. Thus the images are interconnected with each other to form a complete 3D model.

* 1. **Purpose**

Our project is the second module of an existing project named “**There or Not - Smart Augmented Reality Using Image Processing**” by the seniors of our CS department. In that project, a system was developed in which users can take pictures of different pieces of furniture they like and paste them in front of a selected background image. We have decided to transfer those images to develop a 3D model.

The main aim of this project is to develop a 3D model which is applicable and beneficial for processing real images. Through this model, each and every real image can easily be processed, displayed and presented in a 3D form which is more attractive for users. Our model can be used by any individual to attain a 3D impact of the real images more efficiently.

Nowadays, there are many smart phone devices available in the market with touch screens and outstanding front and rear cameras which produce excellent results. Our model can easily be installed and operated on these smart phones. Any picture of a desired object can very quickly be converted into 3D form. This will enable any individual to check whether or not the desired object would be suitable for his/her available space.

* 1. **Current 3D Models**

During the past few years, 3D modelling has developed into a prominent branch of computer technology and has allowed us to understand new concepts of 3D modelling theory and new developments within it. Presently, 3D models are more associated with users because they give them outstanding control of different features and thus captivate the user for hours.

* 1. **User’s Description**

3D modelling mechanics are designed to facilitate an easy modelling experience. Our model suits all age groups and caters to a large number of consumers. 3D modelling is also beneficial for the large automobile companies as they can represent their new and upcoming vehicle models in real images. When a user visits his desired car on the website of some automobile company, it looks attractive, charming and beautiful, but one should keep in mind that the model of the car on the website is animated. When the user visits the showroom to see and buy the same car that he/she has selected from the website, he/she might be surprised by the differences between the animated and the real car. Our model would provide the user the ability to observe the car in its real 3D form.

* 1. **Project Goal**

The struggle with our model is to provide our users a fascinating 3D model which they can compare with animated models .They can have the real 3D models of the various objects very easily.

We aim to implement the concepts of image processing in a way to help users regarding their different points about modelling theory and techniques. The model performs and achieves better results than what the users used before.

Our major goal while constructing our model was to provide a realistic appearance. The model provides complete and remarkable access to the user. Thus, he enjoys the new modelling features.

* 1. **Design and Implementation Challenges**

It was a challenge for us to ensure a great user-experience. We made sure every aspect would be enjoyable for the user so that they may enjoy the model and desire to use its features again and again. We also tried our best to implement the latest concepts of computing and the topics which we have studied in our different set of courses.

The main challenge was to be in accord with new and upcoming ideas from our friends in the 3D modelling world. No doubt, we found different features of 3D modelling in other 3D models and we tried to include those features in our model with even more details and efficiency. As we observed other models where the modelling concepts are applied to build an animated model, we decided that our model should be fully applicable to work on and operate real images for making a 3D model. In the area of digital image processing and computer vision, our knowledge was good with very clear concepts. Therefore, we finally applied our knowledge to our project in order to build real 3D models.

**Chapter 2**

Requirements Analysis

**2.1 Software Requirements**

* Windows XP or a higher version (operating system).
* Microsoft Visual Studio 2010 or a higher version.

**2.2 Hardware Requirements**

* Pentium dual core or new updated hardware.
* Graphics card with 256 MB VRAM and Pixel Model 3.0 of hardware support.
* 1 GB RAM or higher.
* Peripheral devices like cameras or smart phone cameras, mice, keyboards etc.

**2.3 Tool and Technologies**

* Microsoft Visual Studio 2010 or a higher version.

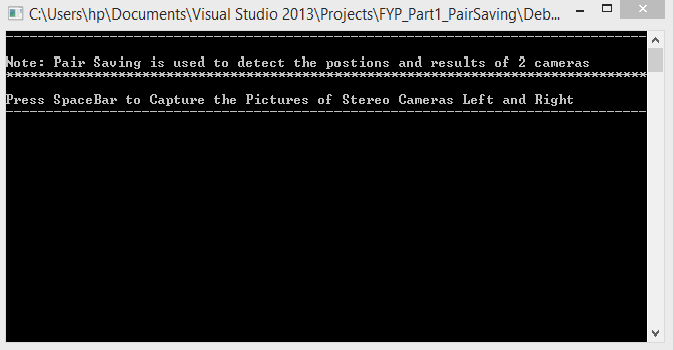
(Microsoft Visual Studio is the integrated development environment in which different libraries are added.)

* OPEN CV with C++ interface configured in Microsoft Visual Studio.

(OPEN CV is a tool to develop 3D Models and is configured in Microsoft Visual Studio.)

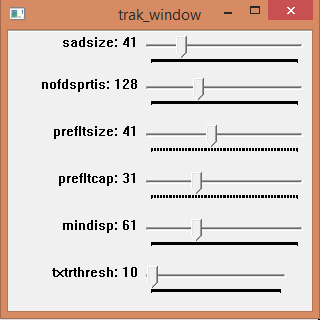
**2.4 Functional Requirements**

2.4.1 **Pair Saving GUI:**Pair saving technique is used to detect the positions and capture the results of two cameras. The images can be captured by pressing the space bar.



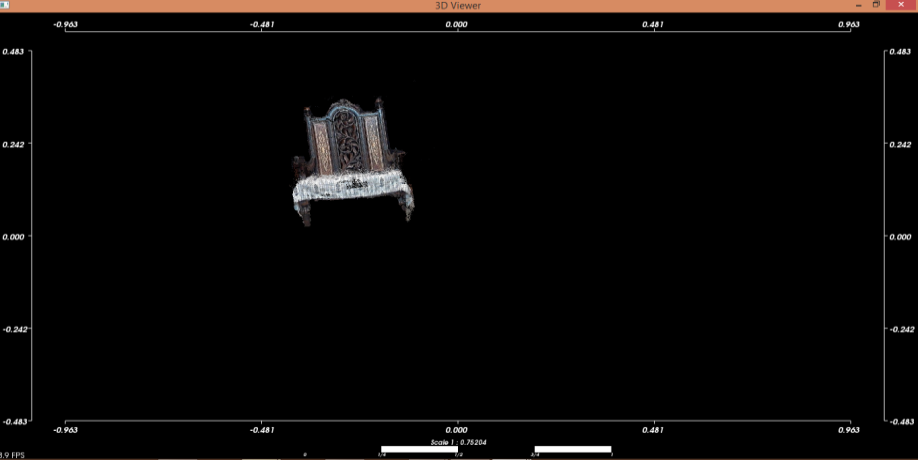
**Figure (2.1): GUI for pair saving**

2.4.2 **Disparity Track Bar Window:** Disparity track bar is used here to alter the disparity of the object manually for obtaining good results. The values of described parameters can be adjusted to get the suitable disparity.



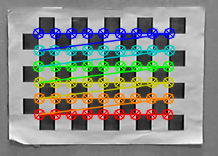
**Figure (2.2): Modifying the disparity while using track bar**

2.4.3 **3D View of Object:**3D viewer is used here to view the object in the 3D plane. It includes the scaling method which scales the object in the 3D space. The object can also be dragged and rotated with the help of the mouse to monitor it from different angles.



**Figure (2.3): Object in 3D space**

2.4.4 **Chessboard detection:** To detect a chessboard, we have to provide some parameters as an input. First, we have to define the size of the chessboard which we are using. Secondly, we have to describe the size of the square box in the chessboard as the detection is based on the black square box which is attached with another square box and finally the chessboard must be printed well.

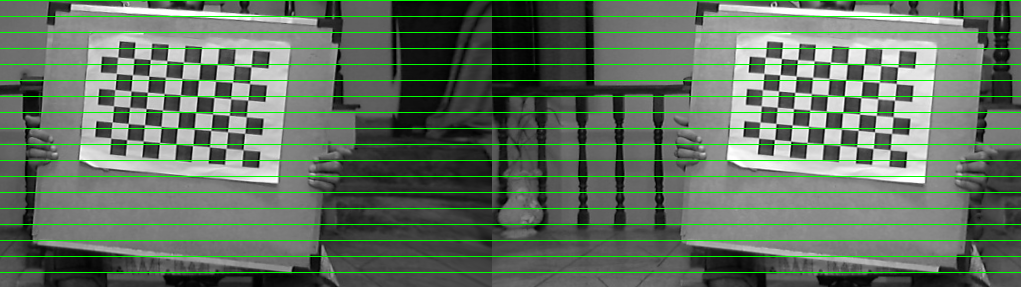


**Figure (2.4): Detected chessboard**

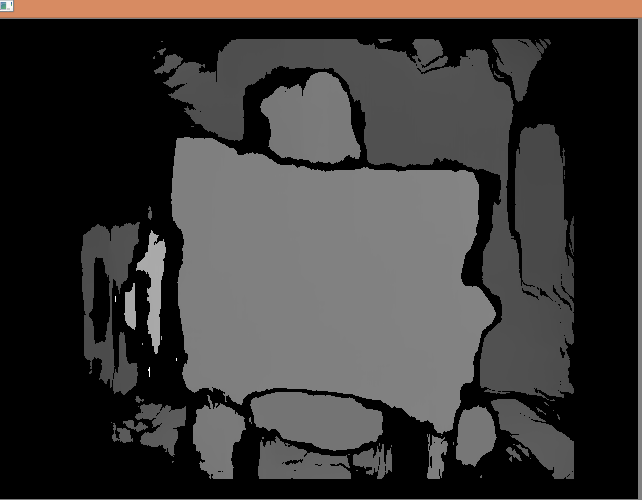
2.4.5 **Stereo cameras:** The cameras which are being used for the calibration are stereo cameras. The stereo cameras must be of same focal length, megapixel and resolution because during the process of calibration this is important that our cameras should capture the object with same set of parameters. If we use different types of cameras, then it would be difficult to calibrate them. Hence they will give us two different results which are not acceptable. In that case, we have to expect the distortion problem which is another task to be handled. To achieve good results, we have to calibrate and alter the position of our stereo cameras in an equivalent order.

**2.5Non-Functional Requirements**

2.5.1**Calibrated Rectified Image:** When the calibration process is applied on the stereo images, we get a rectified image. As we can see in the screen shot that the rectified image is in grey scale which is obtained from the calibration process. The two images from two cameras are joined into one single image which is called a rectified image. It also includes the live disparity of the object.



**Figure (2.5): Calibrated rectified image of chessboard**



**Figure (2.6): Live disparity of object**

**Chapter 3**

Literature Review

**3.1 3D Modelling Theories**

3D modelling concepts provide an excellent execution of perspective relationship patterns that are acknowledged by way of using experience to learn and improve. In layman terms, the assumptions used brains and experience in the world to organize the 3D Model and perceive it through the eyes. Since the brain searches for patterns, we can do it with a pattern followed by challenge. One way to create a pattern is the use of basic primitives. These basic shapes are what the brain already perceived: triangles, squares, rectangles and circles.

Now we have a natural attraction to facial features arranged with a nose and mouth below because we wave human interaction.

We can also use different terms to manipulate what the viewer sees. We bring the shape closer by replacing a larger area. We can make the prospect of a virtual environment using a technique called “Virtual Perspective”. This method creates the illusion of depth perception by using the Vanishing Point [1]. Vanishing Points are points where the object disappears visually. In vanishing point the images are processed to make a 3D model. In above technique, first of all, we try to achieve vanishing point. It is obtained where the information get vanished .On that point, we apply the equation of vanishing point which gives us the perspective image. Input points are given to the perspective image found before .When the points are given the image is stretched to make a 3D model of it. The image can be stretched because it doesn’t contain the third value. The model is developed on the basis of x and y values of a plane.Computer Graphics are just another means that the developer uses to create works on 3DModelling based on measurement [2].

There are several software’s which convert a 2D image plan into 3D model. They are usually comprised of a full range of 3D manipulations. They usually include different transformations like scaling, rotation, translation; perspective and no rigid deformations [3].There are several different proposed algorithms from 2D to 3D conversion. As we know, OPEN CV also played an important role in these conversions [4]. Algorithms are based on depth cue techniques. Different surveys explained and analysed that these algorithms have various cues in them [5].They include line perspective; pattern texture and simple transforms [6].Transforms include horizontal coordinates of different points.

Developing a model of real images is also done by placing multiple cameras. The cameras are attached with numerous PC’s .In this scenario the information of the desired object is in rich amount. Cameras can be placed in a pair of two on different sides of the object .While using the multiple cameras, there is a need of several PC’s .We can condense the availability of several PC’s because it is not necessary to attach multiple cameras. We just need several PC’s [7].

Nowadays animated 3D models of different items are being used. The animated models have replaced the physical objects. It contains a lot of animations effects in it .It doesn’t contain any real effect of an object which is to be modeled. The animated 3D model contains different materials properties, texture and enhances colors which makes the model more attractive. The animated models are completed in different set of tools where the model acquires additional features in it which fluctuate it from the real time model. Animation technique takes a 3D model and labels it as a 3D grid and a quantity of distortions of the grid. This process constructs a texture and achieve significant benefits to follow the indicators in the object selected.

The animated models are diverse from the real models because the real models don’t contain any animation effect. They contain complete real effects which are very attractive and lively. Whereas the animated models are full of animation and are designed by the professionals.

Models can also be prepared by combining images which are in close range and have the closest set of values. These images are stitched together to form a model. These types of models are not 3D they contain a little effect of 3D model because these models are formed by taking the x and y values which are converted to form a model.

While developing a 3d model, calibration is an important part. The cameras are calibrated. While refining the noise, they make an effective model. The cameras show a pattern in different directions so that the vision of camera gets vast. The pattern which is used to calibrate cameras is of two types: (**a.** using the chessboard **b.** making of circles on the page and using them as a pattern to calibrate the cameras.) Cameras comprise distortion factor in it which destroy the calibration process and the final results. This factor can be upgraded by improving the resolution of cameras used in calibration method [8].

Displaying the model in a 3D space is also done by Open GL and point cloud library. It works as a visualizer. It displays the model in the form of clouds. Different clouds are combined to form a model. The model can be filtered and segmented [9].

# Chapter 4

# System Design

* 1. **Use Case Diagram**
     1. **Overview:**

Use case diagram plays an important role in developing new structures. It develops the relationship among different entities and describes their behaviour. In UML, figures terminals and interpreters are included, and a UML diagram is used for the purpose of interaction. It also helps with the classification process of different structures.

* + 1. **Purpose:**

The UML diagram is the structure of collected requirements for an internal and external purpose. It revises the concepts of requirement engineering. It includes the design needed for the structure. When any organization has to be examined, the UML is available to describe each and every aspect. It is used for specifying and constructing the artifacts of any system. It facilitates different teams that are working on projects and help them to communicate which each other.

**USE CASE DIAGRAM**

{Take the images of chessboard}

<<extend>>

<<extend>> <<extend>>

{Calibrated cameras}

USER

{Get image}

{Get the live disparity}

**{**3D viewer} {Installed library}

**Figure (4.1): Use case diagram of a complete system**

**4.2 State Transition Diagram**

4.2.1**Overview:**

The state transition diagram is a structure that describes the performance of an organization. All conditions of a body, such as the actions that occur, must be labelled. Each figure shows the general elements of a specific class.

**STATE TRANSITION DIAGRAM**

Stereo rectification and Disparity checker

Pair Saving

Point cloud

Calibrate

Reproject image to 3D

Left and Right Pairs

Rectified image and Disparity

3D viewer

Re projection

Projection matrix Map x, Map y

**Figure (4.2): Structure and design of system**

# Chapter 5

# Implementation

**5.1 Pair Saving**

In pair saving, stereo cameras are placed in particular positions. This method helps to get the mixed results of both cameras. “Addweighted” function of Open CV is used to get the mixed result.

**Note:** For taking the images, we use “space bar”.

5.1.1 **“Addweighted” function:**

The “addweighted” function is used to determine the sum of the two arrays. It is used to check the position level of the cameras C++ format: void adwghtd (ip array sr1,d.alp,ip array sr2,d.bet, d.gma,op.arraydst,int m typ 2)

5.1.2 **Parameters:**

Following are the seven parameters of the “addweighted” function:

**a.** Input Source Array 1: It is the first input of the camera.

**b.** Alpha: It is used for the weight of first array elements.

**c.** Input Source Array 2: It is the second input of the camera.

**d.** Beta: It is used for the weight of second array elements.

**e.** Gamma: It is a scalar which is added to each and every sum.

**f.**dst: It is an output array which has the same size and the number of channels.

**g.**(d type):When both the arrays have the same depth then it is an optional method. Addweighted calculates the sum of the two arrays by using the following equation:

d (m) = sat ( sr1 (m) \* alp+ sr2 (m)\* bet+ gam)

Where:

**d** is destination.

**sat** is saturate.

**sr** is source.

**alp** is alpha.

**bet** is beta.

**gam** is gamma.

**5.2 Calibration**

After taking the left and right pictures in the pair saving, the step of calibration is to cross-check the intrinsic and extrinsic values of the camera. The extrinsic and intrinsic values of cameras give us the following XML and YML files.

1. Distortion coefficient of camera 1
2. Distortion coefficient of camera 2
3. Camera Matrix 1
4. Camera Matrix 2
5. Map of X and Y of camera 1
6. Map of X and Y of camera 2
7. Rotation of camera 1
8. Rotation of camera 2
9. Translation
10. Q Matrix

* **Distortion Coefficient:**The distortion coefficient is same as the camera resolution. These resolutions are scaled next to the existing resolutionfrom the resolutionwhich is calibrated. Distortion Coefficients = (a1 a2 b1 b2a3)
* **Camera Matrix:** If we want additional information about intrinsic and extrinsicparameters, camera matrix is desired. It includes focal length (*f*s,*f*e) and the optical centers (Cz,Cr).
* **Rotation:**Rotation is linear transformation .It is used to rotate the image. We can also generate the rotation matrix. It includes two things.

1. The center where the image rotates.
2. The angle which have to be rotated.

* **Translation:**Translation is a vector addition. It moves a vector to a certain distancein a certain direction.

(xyz) Translation by (5, 6, 7) (x+5, y+6, z+7)

* **Q - Matrix:**It is mainly known as the projection matrix. It is used to generate the3D points. Then the generated points are used in the function “Reprojectimage to 3D”.

The following pictures of the different angels of the chess board are taken to prove the good results. For this purpose Open CV function is used.

5.2.1**Findchessboardcorners function:**

We use the above function to find the internal cameras position of thechessboard.C++ format: bool fcbc (ip array imgsze,patternsze,oparray,intflgs=calib\_cb\_adp\_thres\_calib\_cb\_norml\_img)

5.2.2 **Parameters:**

The parameters of the chessboard corners are as follow:

**a. Input Array Image:**Thereis a number of images of chessboard taken

from the stereo cameras.

**b.Inner Corner Size:**It is used to delete the pattern recognition i.e., the

innercorners of the chessboard in its height and width.

**c. Output Array Corners:**It is used to detect the number of corners

found in the chessboard pictures.

**5.3 Stereo rectification and Disparity**

In this part, different pictures of an object are taken from right to left with stereo cameras. The x and y values, calculated from calibration cameras, are used here in both cameras. To remap the left and right pictures of the object in a suitable manner we use “remap” function of the Open CV. Its arguments are as follows:

5.3.1**Remapping:**

Remapping is a process which is used to take different pixels from one place and it helps them to locate it in other position.C++ format: void remp (iparrysr,oparrydt,iparrympx,iparrympy,intb.m

bor\_const,contscalr&bv=scalr() )

5.3.2**Parameters:**

The parameters of the Remapping are as follow:

**a.Array of Source:**It is required to process the same image.

**b.Array of destination:**It is used for the output image determination.

**c.Array of MapX:**It is provided by the calibration of camera to map

the function in X direction.

**d.Array of MapY:**It is also provided by the calibration of camera, to

map the function in Y direction.

5.3.3 **Stereo Block Matching Algorithm:**

It is known as a class for stereo correspondence which uses the

algorithm called as Block Matching.

C++ format: sterbm (intprs,intdisp=0,int sws= 5 to 255)

**Parameters:**

**a.SAD WINDOW SIZE:**The size for SAD WINDOW must be

between5 and 255 and it must be odd. The block which is larger in

size,implies smoother.

**b.Number of Disparities:**Disparity plays a role of search range.

There are number of pixels. So, for each pixel the algorithm will find

the best disparity. In our situation, the disparity ranges from 1 to 8.

**c.Pre-filterSize:**Pre-filter size value must be odd and lies between 5

and 255.

**d.MinimumDisparity:**The minimum disparity required in our case

is -64.

**e.UniquenessRatio:**The value for uniqueness ratio is “0”(zero)in

our scenario.

**f.Speckle Window Size:**The Speckle window size value is “0”(zero)

in our case.

**g.SpeckleRange:**There are several ranges of Speckle but in our case

value 8 is a desired one.

5.3.4**Stereo Semi Block Matching Algorithm:**

It is another class which is also used for stereo correspondence while using the semi-global block matching algorithm.C++ format: stersgbm(int min disp,intnumdisp ,intsws,int p1=0,p2=0,int dis 12 mx diff=0, bool full dp=false)

**Parameters:**

**a.** SAD WINDOW SIZE:-SAD WINDOW SIZE is used for matching of block. It should be an odd number.

**b.** P1: It is the first parameter which controls the disparity smoothness.

**c.**P2: It is known as the second parameter which also aims to control the smoothness of disparity.

**d.**Pre-filter Cap:The image pixels are truncated through pre- filtered cap.

**e.**dis 12 Max dif:This is the maximum difference in integer pixel

to check the left and right image disparity to disable this check

set the value to negative.

**5.4: Reprojection**

The above function is used here to reproject the image into 3D plan. It includes the disparity of the image and the projection matrix which is obtained from the calibration step. The disparity of the image is reprojected into a 3D space.

**Parameters:**

The parameters of the above function are as follows:

**a. Disparity of image:**The disparity image should be in 32 bit floating

point channel. The channel should be signed.

**b. Projection matrix:**The projection matrix is also required here which

can obtained from the function named as stereoRectify().

**c. 3D image:**The 3D image is the output image. The channel which we are

using in the disparity is also same for 3D image.

**5.5 Point Cloud Library**

Point cloud library is a set of library which is used for processing point clouds. It includes different algorithms like segmentation, fitting model and reconstruct the surface. The following algorithms are used to filter the data which contains some noise. It also helps to catch the 3D points of clouds and then stitch them up together. It also contains the segmentation technique which is applied on the desired parts which are to be segmented. The point cloud library is Open source software.

5.5.1 **Visualization Library:**

The above library is designed and used to process the visualize results getting from different algorithms as explained above which are working on 3D points.

5.5.2 **3D Viewer:**

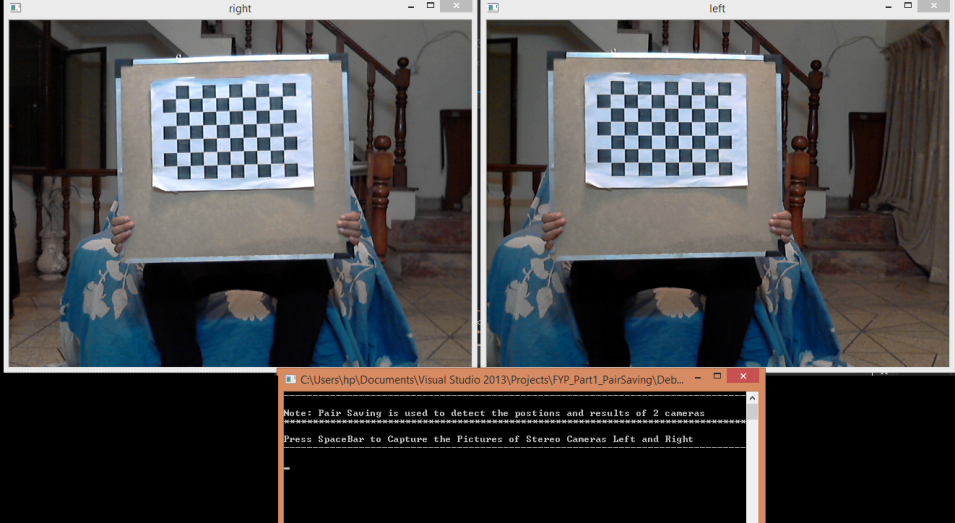
The 3D viewer is the part of Point cloud library which is used to create and display the object. It collects the range of values of a picture while going during the visualization process. After receiving the values it then obtains the information from the reprojection and point cloud library steps about the object .When it has the complete information, it then displays the object to sight with certain clouds in it which help to view the object more resourcefully.

# Chapter 6

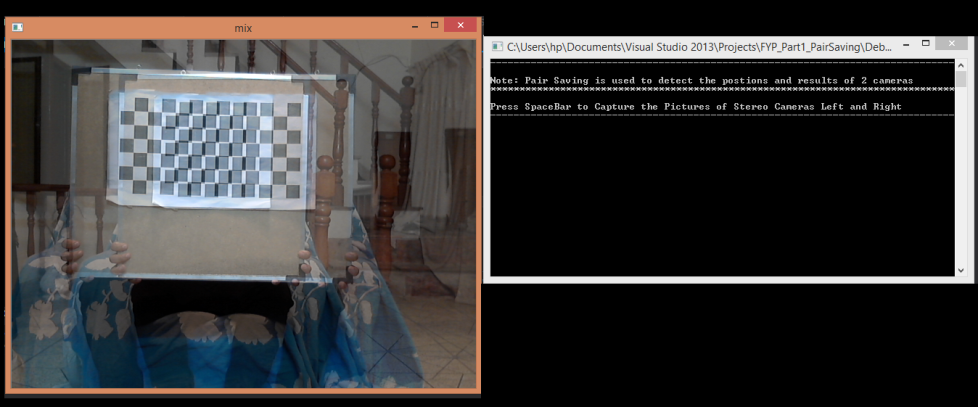
# Testing Process

**6.1 Pair Saving of Images**

We have described in chapter 5 that the pair saving method is used to locate the stereo cameras .It is used here to save the left and right images of chessboard captured by the cameras. It also contains the mixed view of the images. The “addweighted” function is used here to detect the positions of cameras. The images are captured simply by pressing the space bar button as we can see in the screen shot.



**Figure (6.1): Left and Right view of the Chessboard taken by stereo cameras**

**

**Figure (6.2): Mix view of the chessboard image**

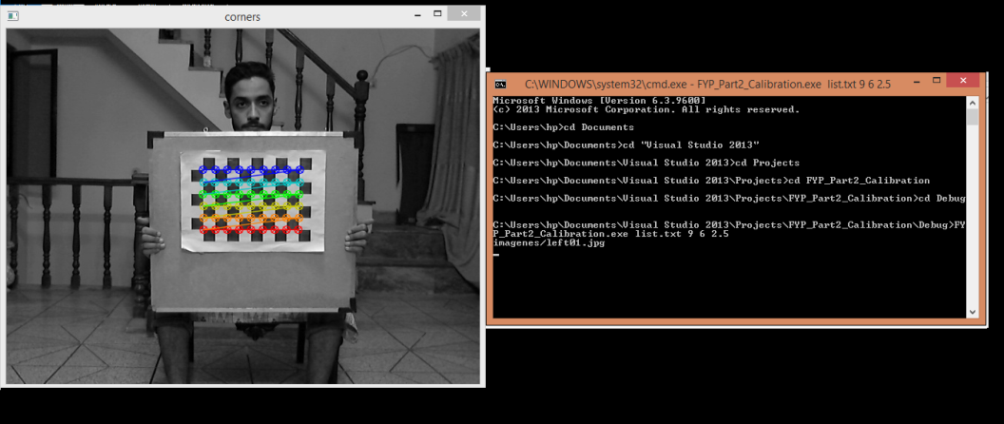
**6.2 Calibrated and Rectified**

Calibration is the second step where the stereo cameras are calibrated. Now the intrinsic and extrinsic values of the camera are ensured here because they give the xml and yml files. The function named “Findchessboardcorners” is implemented here to detect the corners of the chessboard. When this function is applied there is a mapping on the chessboard which shows that the corners are detected. The size of chessboard we used here is 9\*6.

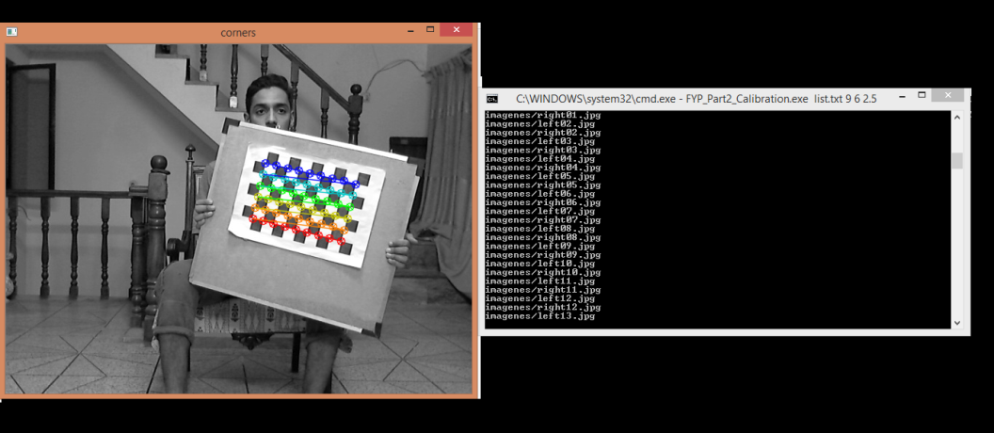
6.2.1 **Stereo left and right Images:**

As we know that we are using stereo cameras so it contains the left and right set of images when it is calibrated. The following screen shots will show the detection.

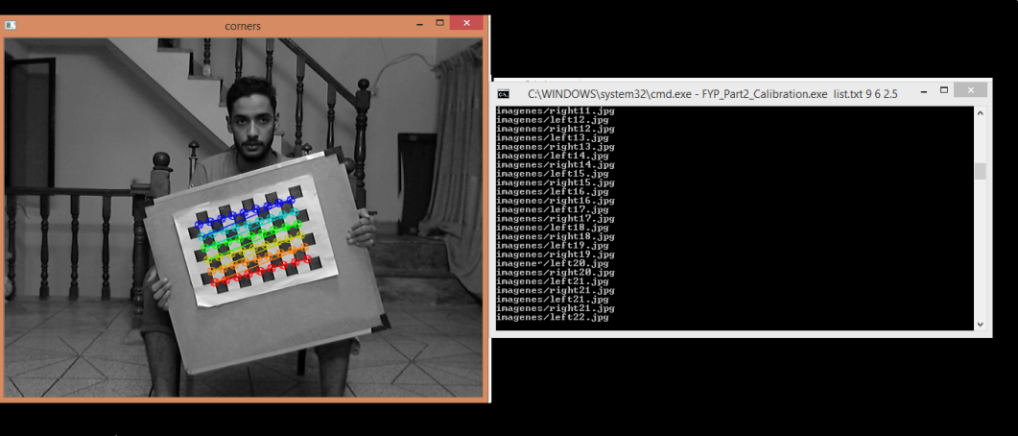
6.2.1.1 **Left Images:**



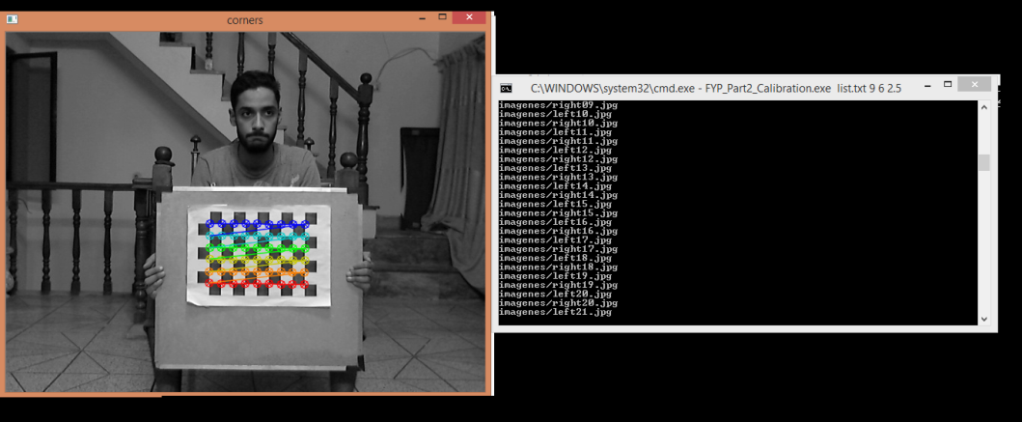
**Figure (6.3): “Left image of chessboard01”**

**

**Figure (6.4): “Left image of chessboard02”**

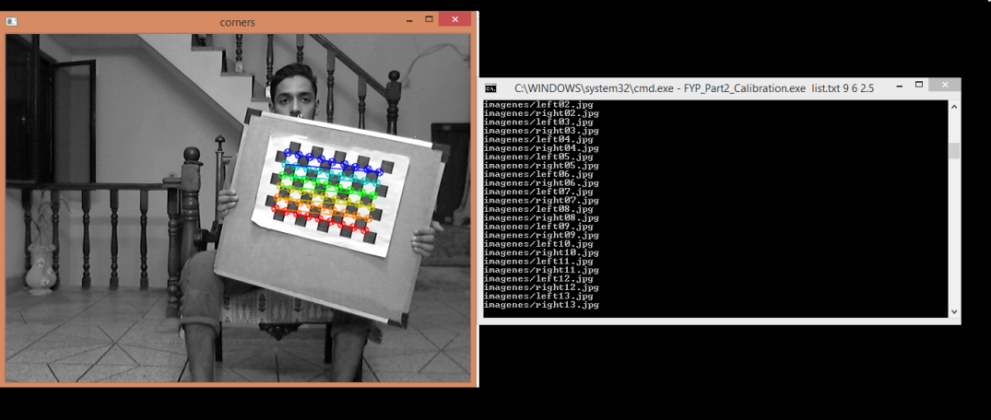
**

**Figure (6.5): “Left image of chessboard 03”**

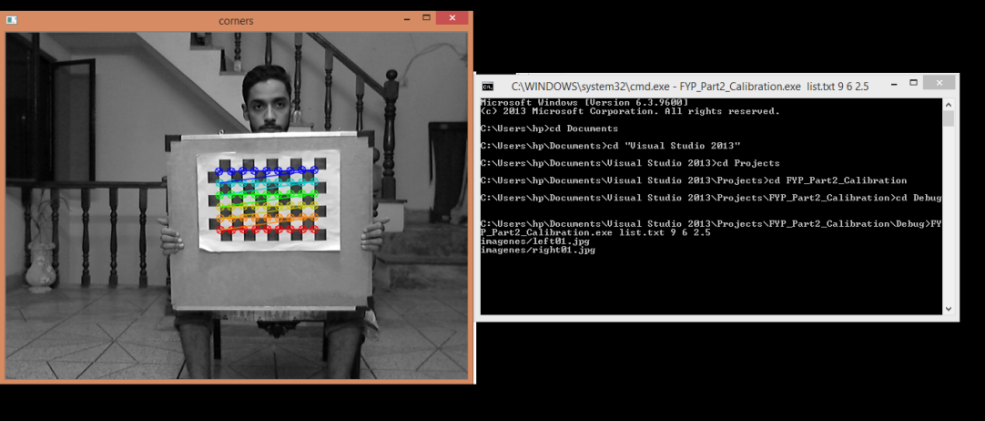
**

**Figure (6.6):“Left image of chessboard 04”**

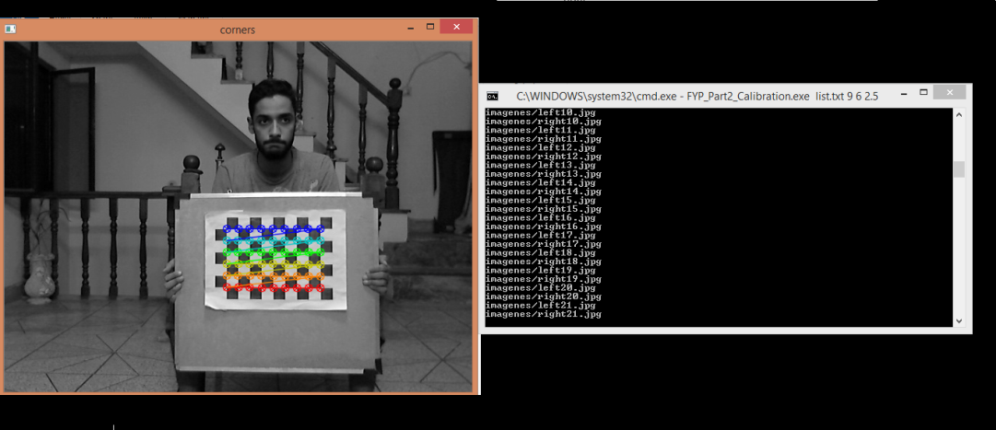
6.2.1.2 **Right Images:**



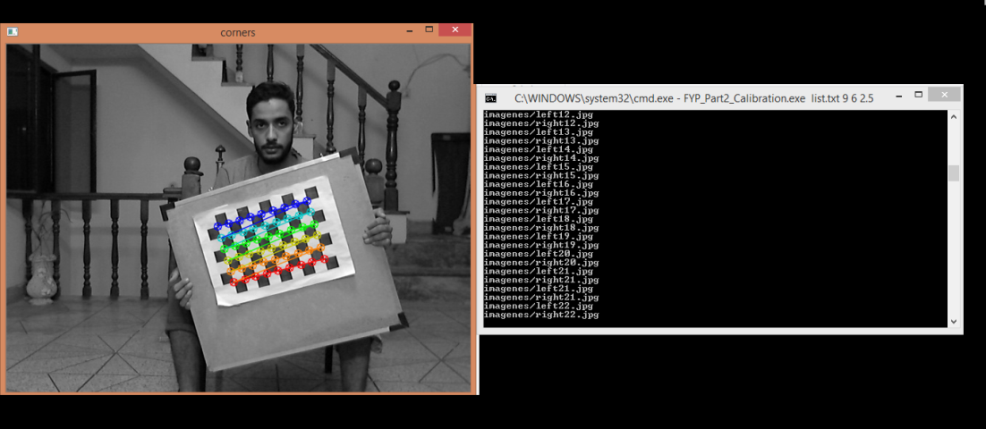
**Figure (6.7):“Right image of chessboard 01”**

**

**Figure (6.8):“Right image of chessboard 02”**

**

**Figure (6.9):“Right image of chessboard 03”**

**

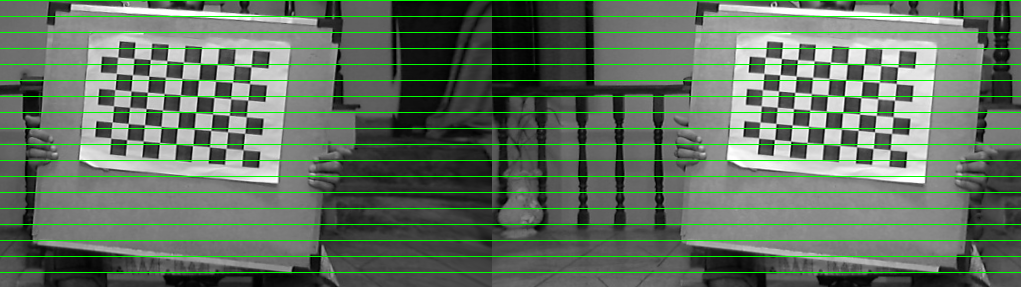
**Figure (6.10):“Right image of chessboard 04”**

6.2.2 **Rectified:**

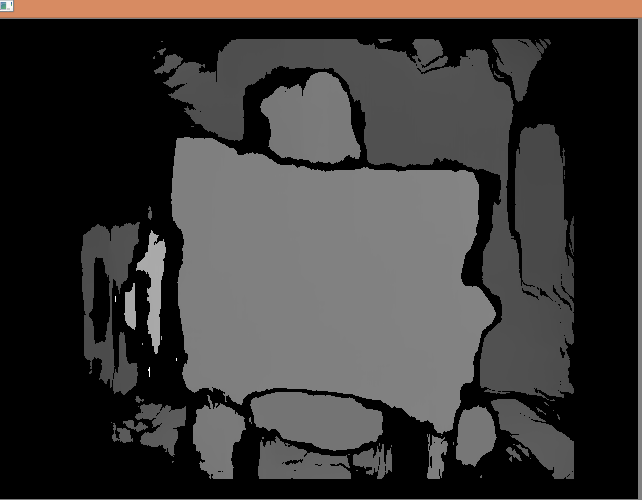
The rectified image is also a major element of calibration because when the cameras are calibrated and the images composed from stereo cameras are processed through the calibration process the rectified image is obtained. The rectified image is a combination of two images. The left and right image of chessboard is connected to make a single image. The rectified image is in grayscale .The rectified image also contains the disparity which shows the massive information of the object.

6.2.2.1 **Good Rectified Image with Disparity:**

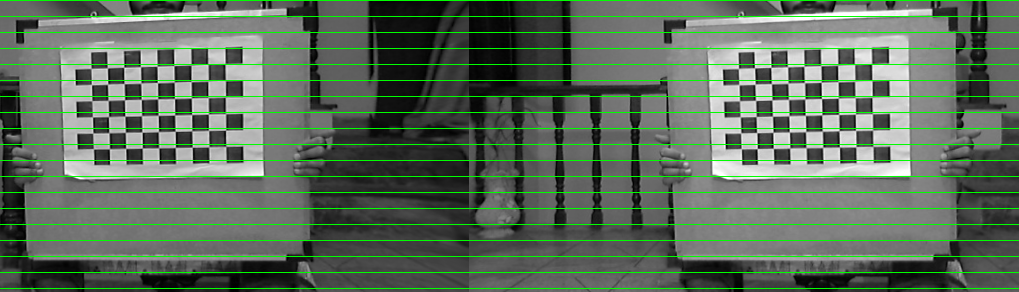
The excellent result of rectified image depends on the selection and tuning of the camera. It also depends on chessboard image. As you can see in the screen shot, green lines are horizontally passing through the image which means that the rectified image is good.



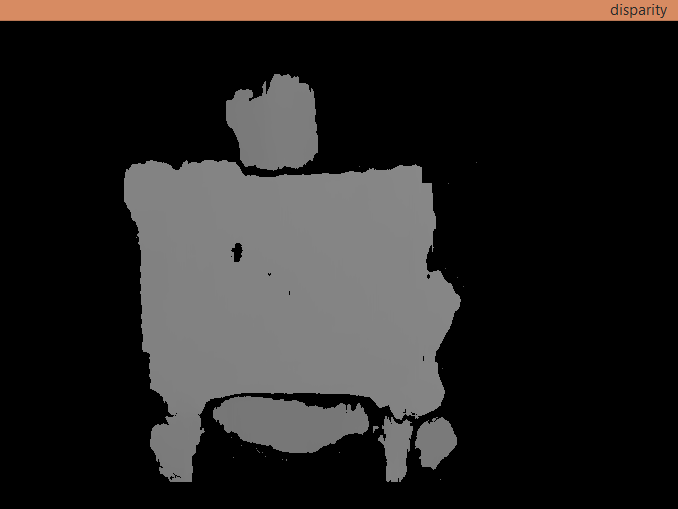
**Figure (6.11): “Good Rectified result of image 01”**

**

**Figure (6.12): “Disparity of the rectified image 01”**

**

**Figure (6.13): “Good Rectified result of image 02”**

**

**Figure (6.14): “Disparity of the rectified image 02”**

**

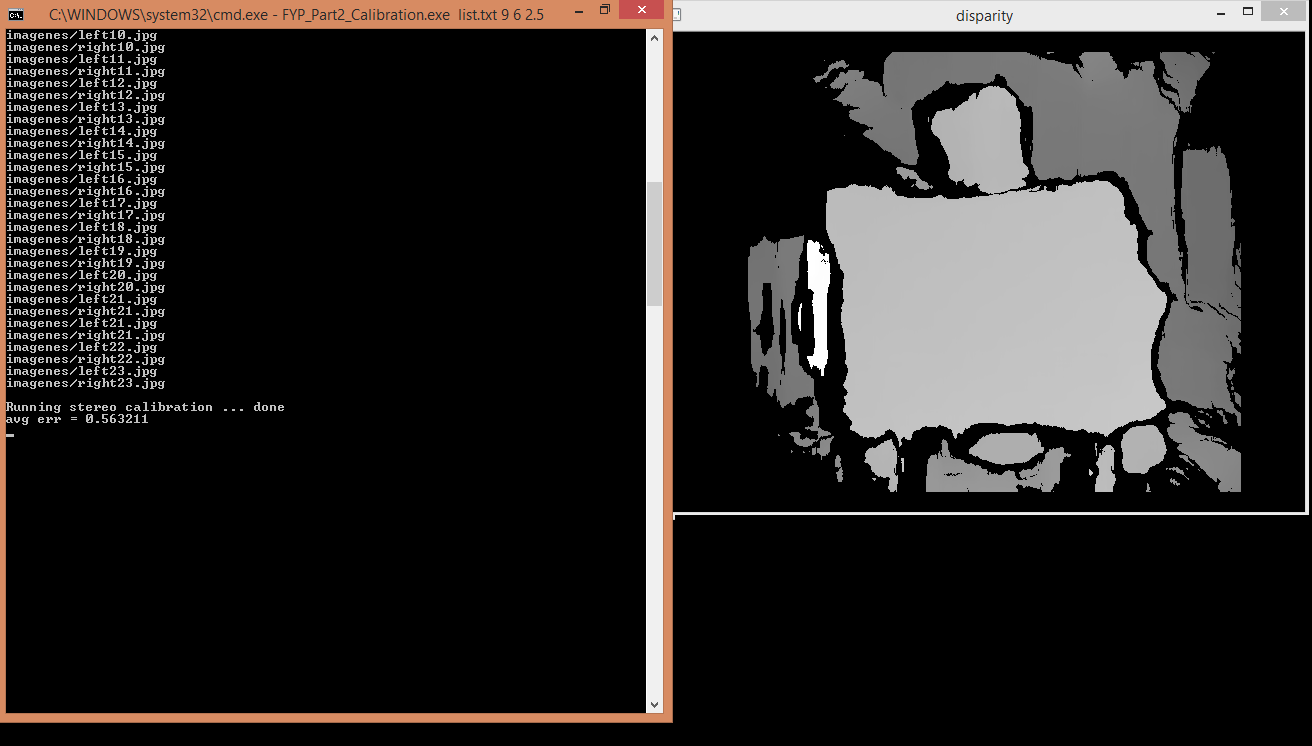
**Figure (6.15): “Good Rectified result of image 03”**

**

**Figure (6.16): “Disparity of the rectified image 03”**

6.2.2.2 **Disparity with RMS Error:**

When the images are processed at the end, there is a RMS (Root Mean Squared) error which also ensures the excellence of rectified image. It also includes the disparity .Here the RMS error is 0.563211.The images which we have processed are 23 in number.



**Figure (6.17):Rms error and disparity**

**6.3 Stereo Rectification and Live Disparity**

Stereo rectification can be computed from the results taken from the stereo cameras. As we discussed in chapter 5, the left and right pictures are remapped in an appropriate behavior and the live disparity of the desired object is ready to be examined. The following screen shots show the object which is selected along with its live disparity. It contains the left and right images of the object.

**Figure (6.18): Left image of the chair Figure (6.19): Right image of the chair**

6.3.1 **Object Live Disparity:**

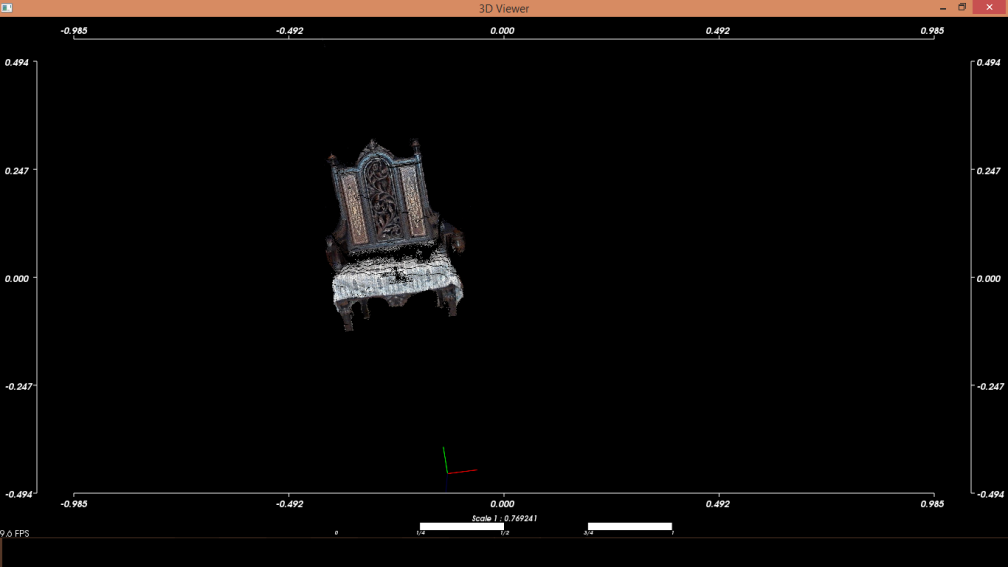
The live disparity of the object can be improved manually by adjusting the different values of parameters placed in a track bar window.



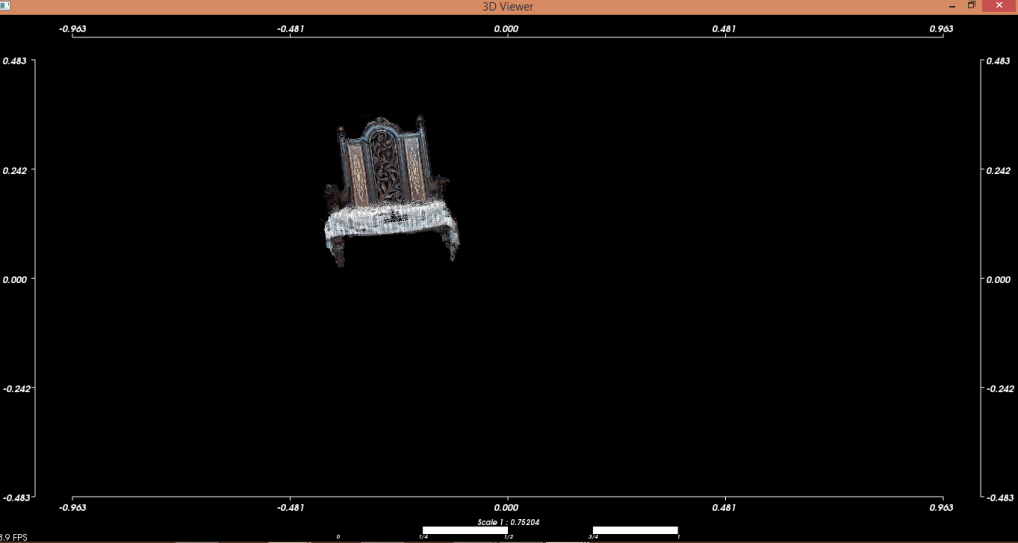
**Figure (6.20): Live disparity of the selected object**

**6.4 My Point Cloud and 3D Result**

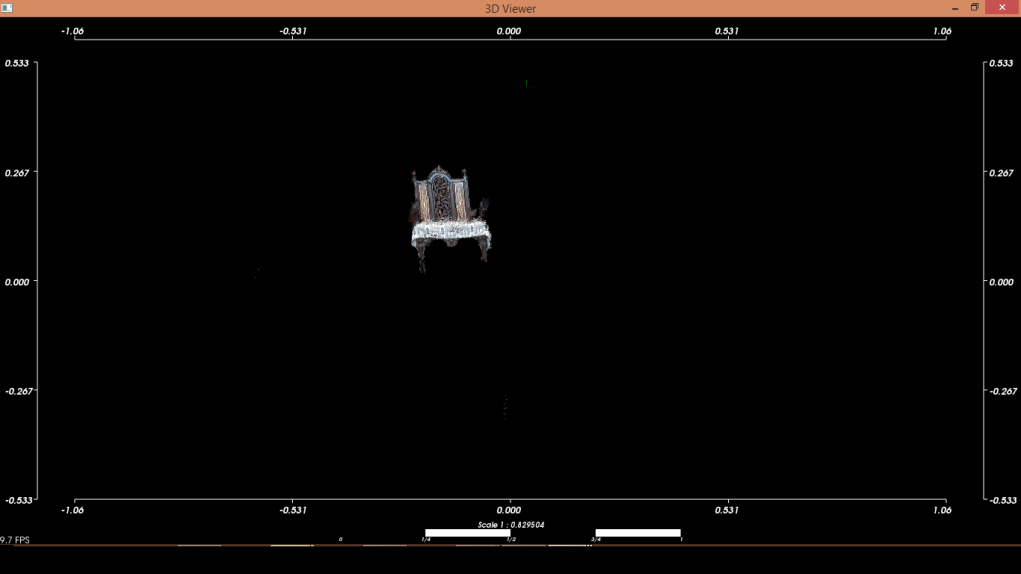
In chapter 5 the concept of point cloud library and 3D viewer is explained in detail. The 3D model of the object is displayed in 3D viewer .The object can be viewed from different directions by dragging the mouse. The object here is made up of different number of clouds by collecting and connecting them together. It also includes scaling technique which helps to scale the object. The below given screen shots contain the clear view of the object which is selected earlier.



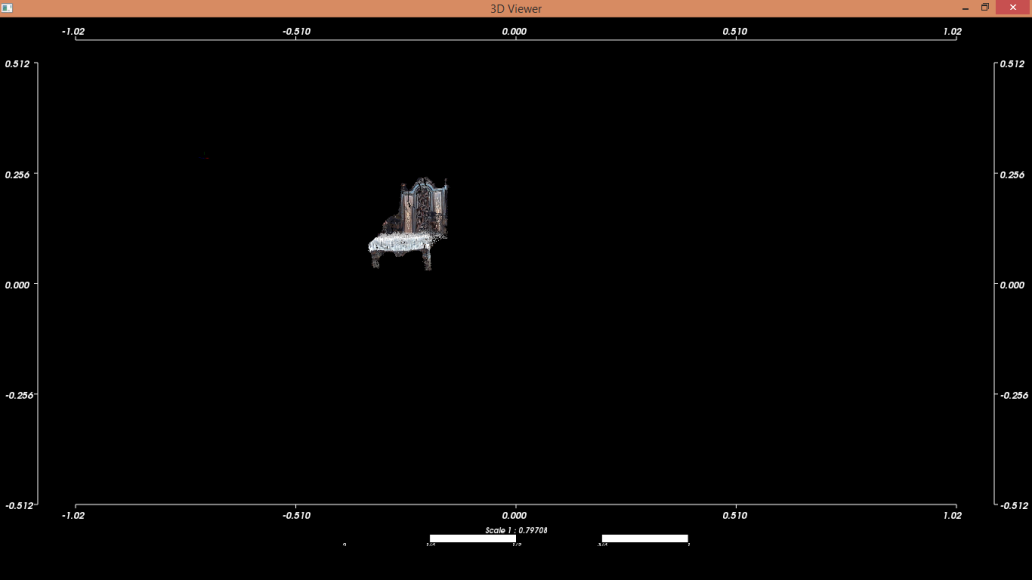
**Figure (6.21): “3D view of object 01”**

**

**Figure (6.22): “3D view of object 02”**



**Figure (6.23): “3D view of object 03”**

**

**Figure (6.24): “3D view of object 04”**

# Chapter 7

# Conclusion &Future work

# 7.1 Conclusion

In our project we implemented new modeling concepts and techniques by taking stereo images of an object. Camera Calibration, as s discussed in implementation chapter, is used to “find out the corners of chessboard” which helps to calibrate the stereo cameras. The camera calibration part is much required to attain best results.

After calibrating the cameras, we took different pictures of our given data and went through Stereo rectification process to check the disparity. When we obtained the best disparity results, we applied the “Remap” technique, as mentioned in implementation chapter, to collect different pixels from one position and then locate it on the desired position.

Since all the work is done with rectification and disparity techniques, we used the function named “Reproject image to 3D” which processed the disparity image into 3D.After applying the function, “Reproject image to 3D ” the results have been shifted into PCL visualization function to observe the resultant clouds in 3D space.PCL is Point Cloud Library which is used to process Point Cloud .PCL provides the 3D viewer where the object can easily be observed as a 3D model with the help of different clouds information.

The work we have completed here is a component of stereo images; there is further to be prepared. We have alerted ourselves to the one detailed scenario i.e.: taking the stereo images but in future we will refine it more by taking more images of an object from different angles to construct a more refined 3D model.

**7.2Future Work**

7.2.1 **Product form:**

In the future, we plan to refine our product into an application which can be operated effortlessly for different mobile phone devices. In addition, it will work on different operating systems like Android, iOS, and Windows. To make it more user-friendly, we will also introduce some amazing features. Moreover, we will increase the capability of the interface and the user will receive more benefit by having the ability to model different objects at the same time.

**References**

[1]<https://www.cs.cmu.edu/~om3d/papers/SIGGRAPH2014.pdf>

[2]<http://opencv.org/platforms.html>

[3]<https://en.wikipedia.org/wiki/3D_modeling>

[4]https://Stackoverflow.com/questions/16514762/point-cloud-library-on-visual-studio

[5][http://stackoverflow.com/questions/30275862/3d-reconstruction-from-multiple-images-with-one-camera](https://l.facebook.com/l.php?u=http%3A%2F%2Fstackoverflow.com%2Fquestions%2F30275862%2F3d-reconstruction-from-multiple-images-with-one-camera&h=0AQGGHFqs" \t "_blank)

[6]<http://Joelgrabados.com/2010/07/30/opencv-camera-extrinsics/>

[7]<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1384822&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1384822>

[8]<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=888718&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D888718>

[9] <http://pointclouds.org/documentation/>