An Efficient Methodology for 3D Tracking and Pointing Localization of Humans for Robotic Guidance

BY

SHANKARANAND JAGADEESAN
B.E. Anna University, India 2007

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2011

Chicago, Illinois

Dedicated to my parents

for their encouragement, support and endless love

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Jezekiel Ben-Arie, for his invaluable advice, creative ideas, constant words of motivation and encouragement during my research. His endless energy and enthusiasm in research is an inspiration to all his students, including myself. I would also like to thank my thesis defense committee members, Dr. Daniel Graupe and Dr. Milos Zefran for their support and assistance.

My special thanks to my colleagues at the Machine Vision Laboratory Kai Ma, Simone Franzini, Jayanth Kolar and Gokul Ganesan for the numerous discussions and constant support during my research.

I would like to thank my friends, especially Arthi Vijayakumar and Vickram Subramanian who have always been very helpful in completing my masters with thesis.

SJ

TABLE OF CONTENTS

CHAPTER PAGE

1. INTRODUCTION 1
   1.1. Scope of the work 1
   1.2. Contributions 4
   1.3. Outline of the thesis 6

2. BASICS AND BACKGROUND 8
   2.1 Image Acquisition in cameras 8
   2.2 Transformation matrices 11
   2.3 Depth Extraction and Reconstruction 14
   2.4 Pointing Gestures 17
   2.5 Previous Work 19

3. OUTLINE OF HUMAN POINTING TARGET ESTIMATION FRAMEWORK 23
   3.1 Robotic Assistants for Elderly: Integrating Speech, Vision and Haptics 23
   3.2 Estimation of Human Pointing targets 25
      3.2.1 Self-Calibration of Room 27
      3.2.2 Detecting Pointing Region 28
      3.2.3 Finding Human Pointing location 30

4. CAMERA CALIBRATION AND STEREO RECONSTRUCTION 32
   4.1 Single Camera Calibration 32
      4.1.1 Projective Geometry 32
      4.1.2 Intrinsic and Extrinsic Parameters 34
      4.1.3 Zhang's Chessboard Calibration Method 37
      4.1.4 Undistortion 39
   4.2 Stereo Camera Calibration and Depth Extraction 41
      4.2.1 Triangulation 41
      4.2.2 Epipolar Geometry 43
      4.2.3 Essential and Fundamental Matrices 44
      4.2.4 Stereo Rectification 45
      4.2.5 Disparity Images and 3D Reconstruction 46

5. SELF CALIBRATION OF MULTI CAMERA ENVIRONMENT 48
   5.1 Multi view reconstruction 48
      5.1.1. Projective reconstruction by bundle adjustment 49
      5.1.2. Projective Reconstruction by factorization 49
   5.2 Self-Calibration Theory 51
   5.3 Implementing Svoboda's Multi-camera Self-Calibration 52
      5.3.1. Data Acquisition and detecting Corresponding points 52
      5.3.2. RANSAC Analysis and Estimation of projective Depth 55

TABLE OF CONTENTS (Continued)

CHAPTER PAGE

5.3.3. Rank 4 factorization and Euclidean Stratification 57
5.3.4. Calibration Output and Alignment with world Coordinate System..... 58
5.4 Validation of existing calibration..... 64

6. HUMAN POINTING GESTURE DETECTION AND 3D TRACKING..... 65
6.1. Detection of Human Pointing gesture..... 65
6.2. Real-time Human detection and tracking in Scene..... 67
6.2.1 Motion detection and background subtraction..... 69
6.2.2 Morphological Operations and Human Silhouette extraction 72
6.2.3 Head detection and 3D Tracking 76
6.3. Region Analysis for detecting Pointing Region..... 80
6.3.1. Skin based hand detection..... 81
6.3.2 Marker based finger detection 85

7. POINITNG LOCALIZATION - METHODS & CHARECTERISTICS 89
7.1 Accurate Pointing Vs Approximate Pointing 89
7.2. Pointing target estimation by machines 90
7.2.1. Eye-Fingertip line approach..... 91
7.2.2. Finger and Forearm approach 92
7.3 Pointing Localization and Pointed Object detection..... 97
7.4. Results and Statistics..... 103

8. CONCLUSION AND FUTURE WORK 108
8.1. Overall contribution in the project..... 108
8.2 Future Work 109

REFERENCES 111

VITA 117

LIST OF TABLES

TABLE PAGE

I. OBJECT DATABASE 99

II. ANGULAR ERROR IN FINGER AND FOREARM POINTING 106

LIST OF FIGURES

FIGURE PAGE

1 : Multi camera setup....3

2: CCD camera imaging a vase....9

3: Pinhole camera model with perspective projection [6] 10

4: Stereo Camera System....15

5: Architecture of multimodal user interface....24

6: Human pointing target determination framework 27

7: Forearm detection on pointing frame....29

8: Finger detection on pointing frame....30

9: Target Bounded by Red Box 31

10: Projection of 3D point into 2D Image plane....32

11: Different Coordinate frames in Camera calibration 34

12: Relation between pixel coordinate system and Image plane coordinates [47]....35

13: Summary of transformations from 3D to 2D.....36

14: Zhang's chessboard calibration method....37

15: Radial distortions plot for camera lens 39

16: Undistortion of images 40

17: Triangulation in stereo camera setup....41

18: Epipolar Geometry....43

19: Rectification of Images [52] 46

20: Stereo images and Disparity map. [53]....47

21: LED is the only bright spot in the frame: Data acquisition for calibration 53

22: Standard deviation image. White spots indicate movement of LED pointer....55

LIST OF FIGURES (Continued)

FIGURE PAGE

23: Multi camera Geometry 57

24: Calibrated camera - Detected Vs Reprojected points 59

25: Distortion Model of camera 60

26: 3D Plot of unaligned calibrated environment 62

27: 3D plot of aligned calibrated environment 63

28: Time correlation between start of gesture and start of deictic words [11] 66

29: Flowchart of Pointing estimation and 3D Tracking 68

30: Histogram of edged scene with no movement (top) and movement (bottom) 70

31: Original scene and Background subtracted image 71

32: Thresholded binary image (left) and thickened binary image (right) 73

33: Closed Binary Image (left), Regions bounded by yellow box (Right) 75

34: Extracted Human Silhouette 76

35: Head templates 78

36: Head detection in a single camera 79

37: Trajectory of Human movement in the room 80

38: Skin color distribution in Cb and Cr channels 82

39: Orientation of a Region with x-axis 83

40: Forearm detection in side and semi-side views 85

41: Region analysis to detect pointing finger 87

42: Marker based finger detection 88

43: Accurate pointing (left) and Approximate pointing (Right) 90

44: Misdetection of pointing location by Eye-Fingertip approach 92

LIST OF FIGURES (Continued)

FIGURE PAGE

45: Extreme points of a region.....93

46: Finger endpoint detection 96

47: Forearm endpoint detection 97

48: Line of Pointing 98

49: Object locations bounded by a rectangle 100

50: Pointed Object found. 101

51: Vacant pointing.....102

52: Comparison of Errors in Finger and Forearm Pointing 104

53: Comparison of error in azimuth angles in reconstructing LOP using 2 Vs 3 Images 105

LIST OF ABBREVATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
ADL	Activities of Daily Living
LOP	Line of Pointing
LOS	Line of Sight
RGB	Red Green Blue
CCD	Charge-coupled device
EM	Expectation maximization
RISq	Recognition by Indexing and Sequencing
NLP	Natural Language Processing
SVD	Singular Value Decomposition
RANSAC	RANdom SAmple Consensus
LMedS	Least Median of Squares
ML	Maximum Likelihood
USB	Universal Serial Bus
LED	Light Emitting Diode
FG	Foreground
BG	Background
HSV	Hue Saturation Value
MVL	Machine Vision Lab

SUMMARY

Human gestures are one of the primary ways of communication in Human-Machine Interactions. In our study of “Effective communication with robotic assistants for the elderly: Integrating Speech, Vision and Haptics”, we concentrate on interpreting the human pointing gestures. My thesis study is about the post-detection stage of a human pointing gesture which is essentially determining the human pointing targets.

In this work, we investigate the calibration of a multi-camera environment for a perfect 3D reconstruction of an ADL (Activities of Daily Living) room. This calibration process was inspired from the work of Tomas Svoboda’s Self-Calibration of virtual environment. The calibration process involves synchronized data capture of the calibration object, projective reconstruction by factorization, projective depth estimation, Euclidean stratification, estimation of non-linear distortion and finally alignment with the world coordinate system. The aligned coordinate system will provide the location of each object in the room with precise 3D coordinates in metric units.

Our work is a part of the NSF project that deals with Integrating Speech, Vision and Haptics. Thus we propose a multi-modal method to detect human pointing gestures and implement a vision based method that interprets the pointing and finds the pointed target location accurately. Two approaches of determining the pointing direction are used in this work; one is based on the

forearm and the other one is based on index finger. In our experiments with the elderly in ADL Room we found that only these two pointing gestures are predominantly used. In our research we develop methods to find the pointed objects by using a 3D database of object locations in the room. We also compare the results of finger and forearm based pointing. The proposed method would also list the closest objects if the computed pointing location is vacant.

We have also proposed and implemented a method for detecting human head in upright pose and tracking the location of the person in 3D inside a calibrated environment. We obtain the trajectories of the person as function of time. The trajectory information along with the pointing information can be used in conjunction with speech to disambiguate communication with the robotic assistant.

1. INTRODUCTION

The research on obtaining the 3D information using the data from images and various devices has been one of the most active areas of computer vision, image processing and computer graphics. With existing growth of virtualization concepts, computer gaming, gesture based device communication; this area has been explored in various ways. Even though there are various ways of obtaining 3D information, the computers are quite far away from matching human capabilities. Human vision is much superior even when compared with the state-of-art cameras and 3D reconstructing devices. Our aim is not replicate human vision using computers but to get accurate 3D information of a room that includes persons and objects. We make use of this 3D information to analyze human pointing gestures and obtain locations of objects as part of our study.

1.1. Scope of the work

If you consider any 3D reconstruction system it will have its own limitations and constraints. Every system is built to meet its requirements and customized for task performed in a particular environment. Starting from 3D glasses to X-Box Kinect, there are a lot of such systems available to get the real 3D Experience. Different devices are used to gather data and build a 3D model out of it. The common ones are Polarized glasses, Structured Light, Time of Flight Cameras, Depth cameras using a Stereo Rig and array of RGB Cameras. The Challenge is to pick the correct way of approach to build a 3D system for accomplishing the requirements of your task.

My thesis study is part of NSF funded project titled “Effective Communication with Robotic

Assistants for Elderly: Integrating Speech, Vision and Haptics" [1]. The objective of the project is to design a communication interface between the elderly patients and an assistive robot. One of the most trivial forms of communication between humans is gestures. Hence we are trying to interpret human gestures in a controlled environment such as ADL (Activities of Daily Living) Room. A 3D model of this room was built and monitored the activities in three dimensions. As mentioned earlier, the real challenge in 3D Reconstruction is choosing the best system for the required task and we chose the multi-camera approach. The reason for choosing this approach is because the activity of an elderly has to be monitored continuously all over the room. A well designed multi-camera system is capable of covering the whole room with enough overlaps, so that every point is available in at least two cameras.

Figure 1 : Multi camera setup

Given above is an example of a room with multi-camera setup. Since we are using a multi modal communication framework the robots can always use combination of the modes to interpret the human communication. This will be discussed in section 3.1. The scope of the work is restricted to 3D reconstruction using a multi camera approach.

Among the gestures studied by conducting experiments with elderly patients, my thesis concentrates on pointing gestures. Why only pointing gesture? A lot of research has been done in finding the best ways of non-verbal communication and results indicate pointing gestures has been the most important form [2]. A pointing gesture involves at least two participants and creates a shared space between them to communicate effectively. It has a body performing the act of pointing, a speech which expresses a request by pointing and receive more accurate definition by the characteristics of the target space which is being pointed to. [3].

1.2. Contributions

This thesis study concentrates on three main tasks. First, the self-calibration of a real world environment (ADL Room) to get the every point in the room in 3D Coordinates (X,Y,Z) with a fixed world center. The calibration problem which we are dealing here is much more complex than the normal calibration of stereo cameras where the field of view is limited to a certain extent. The second task is to detect the human pointing gesture and analyze the images for detecting the pointing region in the image. A pointing gesture can be accomplished in various ways. In real world scenario, Humans use their hand for pointing but for the study of localization of pointing we use markers, gloves and different types of pointing gestures. Hence in this part we need to find the region in the image which depicts the pointing hand precisely. This task is initiated by detection of human head in the video and followed by detection of the hand pointing. The final task is to derive the pointed location coordinates to identify objects which are being pointed to and calculate the error in terms of angle between estimated pointed location and the real pointed location. The main contributions of this thesis are as follows:

- We designed a robust, clean and long-running data collection system using 6 High Definition USB cameras to capture synchronized video data with time-stamps for calibration, testing and 3D reconstruction of the room. Matlab Image acquisition toolbox and I-spy were used for the above purposes.
- Implemented a Self-Calibration technique [4] proposed by Tomas Svoboda for Virtual Environments successfully to calibrate the ADL and Machine Vision Lab with 6 cameras.
- Successful Transformation of self-calibrated environment to real world coordinates in terms of metrics (cms/inches) using the concept of Similarity transformation. The room dimensions were measured and obtained transformed coordinate system in terms of metrics has a measurement error of less than 2cm.
- We proposed a method for detecting human silhouette and head tracking of the person in the image using region analysis and shape template matching. The position of the person is tracked successfully in the 3D world coordinates by using above methods.
- We propose an efficient method using multi-modal communication which includes speech and vision to detect pointing gestures during Human-machine communication.
- We developed a pointing detection and estimation system which detects the forearm and pointing finger in several images. This data is then used to estimate the LOP in 3D. The estimate of the LOP direction with the 3D location of the LOP enables to find the 3D locations which are being pointed to.
- We also developed a method to find several objects in the same line-of-pointing (One behind another) by creating a database of objects in the room and storing its location. If there are multiple objects in the line of pointing (LOP) each object is returned with the distance from the user to identify the correct pointed object.

1.3. Outline of the thesis

This chapter gave a brief description about the problem we want to solve and the approaches that are used in this study. The overall goals and contributions are also listed in the above section. The remaining chapters are organized as follows:

- Chapter 2 describes the fundamental concepts of camera models, 2D to 3D transformation. We discuss the basic mathematical theories behind transformation matrices, Depth extraction and 3D reconstruction. The final parts of this chapter involve a description about human pointing gestures and other previous works in this research.
- Chapter 3 outlines the overall human pointing target estimation framework. The initial part gives an outlook of the whole project which is based on multi modal communication interface. Then we move on to estimation of human pointing targets and the sub divisions talks about each stage in the pointing determination framework
- Chapter 4 presents camera calibration and stereo reconstruction techniques. We talk about Zhang's chessboard calibration and detailed descriptions of camera parameters. The second part of the chapter involves theory behind stereo calibration, stereo rectification and correspondence problem.
- Chapter 5 is concentrated on the process of self-calibration of a multi-camera environment which is inspired from Svoboda's "Self-Calibration of Virtual Environments". This is the method followed in this thesis for calibrating two different environments for recovering 3D information of the experiment space. It explains each stage in the process separately.
- Chapter 6 explains the pre-pointing stages required for detecting the pointing gesture and analysis of objects used for pointing. Then it talks about Motion detection, Human

silhouette tracking in scene and region analysis of the tracked object. We also discuss about the different ways of pointing object detection.

- Chapter 7 provides the core of our pointing target estimation method. We talk about different types of pointing, the approaches used in identifying those types in our study. We tabulate the results and perform statistics to give a clear picture of human pointing target estimation.
- Chapter 8 briefs about the overall contribution of this thesis to the project and scope of future work based on this thesis.

2. BASICS AND BACKGROUND

2.1 Image Acquisition in cameras

There are different devices that produce images and the image we consider here are digital images. Each device uses a different type of sensors and the type of sensors we use here are CCD. CCD's are more like old film based cameras but instead of chemical reacting to light it has solid state cells that convert light energy to electrical charge. By default all the cells have 0 and then the value of each cell gets changed according to the amount of light falls on each cell. Hence the imaging plane is a collection of cells which can be read as a matrix. The figure shown below is taken from the book Computer Vision by Shapiro and Stockman [5] explains how the CCD captures the image of a Vase.

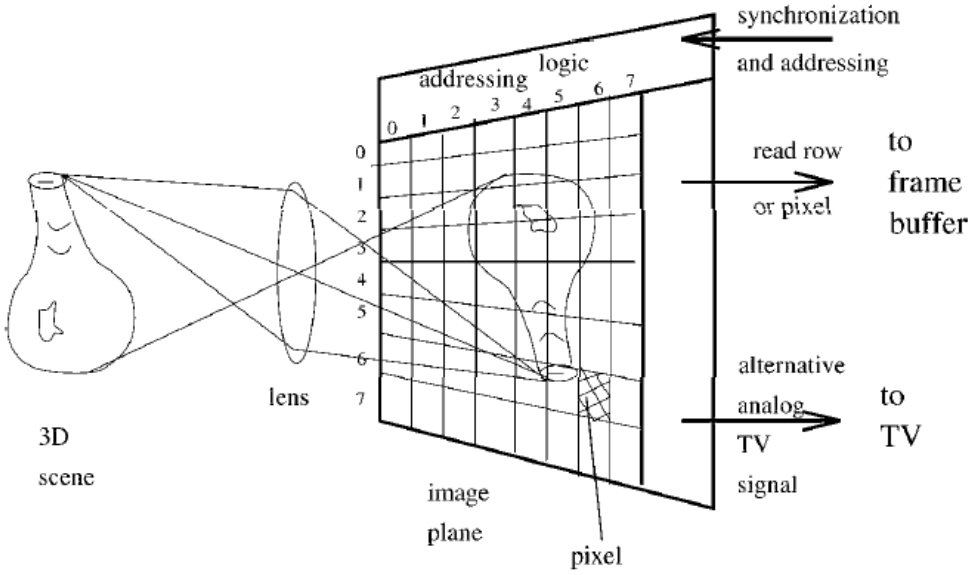


Figure 2: CCD camera imaging a vase

Each cell is referred to as a pixel meaning picture element.

A geometrical model is designed to visualize the process of image formation in a CCD camera. This is done by considering projection of each point in the real world through the center of projection or a lens center onto the imaging plane. This projection model is much similar to the pin-hole camera model explained in physics. Look at the diagram given below.

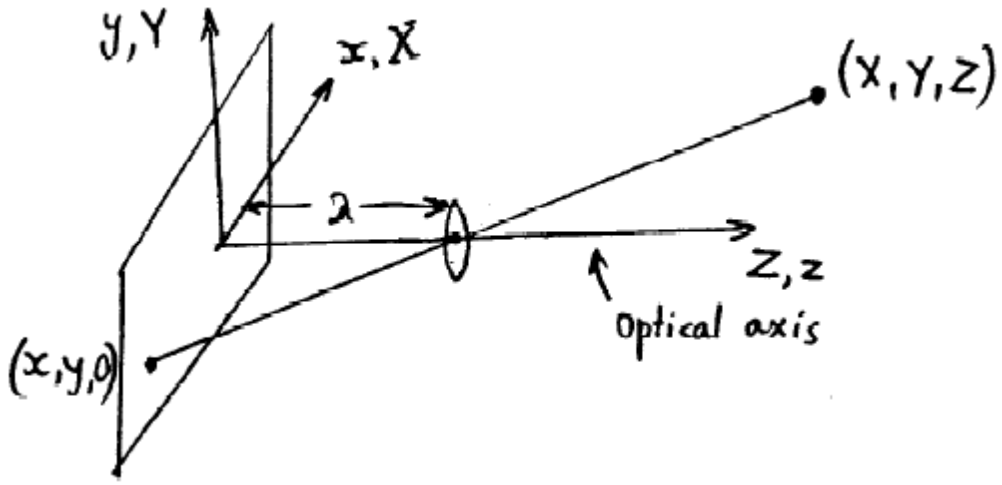


Figure 3: Pinhole camera model with perspective projection [6]

The Line joining the points (X,Y,Z) and (x,y,0) is the projection of 3D world point onto an imaging plane with its axis center in the center of imaging plane. The optical axis passes through the center of imaging plane and lens center. This type of projection is called perspective projection and its assumed that camera coordinate system [x,y,z] and the world coordinate system [X,Y,Z] are coincidental [6]. From the ideal lens equation and using the mathematical concept of similar triangles,

$$x = \frac{\lambda X}{\lambda - Z} \tag{2.1}$$

where x is the projection of X coordinate of the world point onto imaging plane. λ is the focal length of the lens used in the camera. Similarly

$$y = \frac{\lambda Y}{\lambda - Z} \tag{2.2}$$

Since these are non-linear equations, to linearize the above equations homogeneous coordinates are introduced which are of the form

$$\underline{w} = [X, Y, Z]^T \xrightarrow{yields} \underline{w_h} = [kX, kY, kZ, k]^T \tag{2.3}$$

Homogeneous coordinate system can be converted into Cartesian coordinate system by dividing the elements by 4th component. The perspective transformation is denoted by a matrix P and the following section explains about the transformation matrix.

2.2 Transformation matrices

The importance of transformation matrices is to find the relation between the projective image plane P^2 on the CCD and the Projective 3D world space P^3. Transformations in the images are represented by homographies of $P^2 \rightarrow P^2$. Such a homography is represented by 3 X 3 matrix H. We can have homographies in the form of H or λH where λ is a non-zero scalar value. Similarly the transformation in P^3 to P^2 happens in the form of transformation matrices and the equation describing the transformation is

$$\underline{c_h} = P\underline{w_h} \tag{2.4}$$

where $\underline{c_h}$ is the camera homogenous coordinates [ax,ay,az,a], $\underline{w_h}$ is the world homogenous coordinates [X,Y,Z,W] and P is the transformation.

P matrices are of the form $\{[I_{3*3}\ 0_{3*1}],[0,0,1/\lambda,1]\}$ where I is the identity matrix and 0 will be a column vector of zeros. In order to use the perspective model and derive the relation between

two coordinate systems we need to have coincident coordinate systems. This can be done in two ways, 1.) Move the camera coordinates to coincide with world coordinate system 2.) Move the world coordinate system to coincide with the camera coordinate system. It has various stages and the following procedure for transformation is taken from course notes of Image Analysis and Computer Vision II by Jezekiel Ben-arie [6].

- First [X,Y,Z] is translated by $\underline{w}_0$ where $\underline{w}_0$ is the distance between origin of world coordinate system and camera gimbal's center. Assume $\underline{w}_0$ to have positive values, then the translation matrix G is given by

$$G = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

- Then [X,Y,Z] is rotated to be aligned with the camera axes [x,y,z]. Counterclockwise rotations are considered positive and if the pan is θ and tilt is α, the combined rotation matrix is given by

$$R = R_\alpha R_\theta = \begin{bmatrix} cos\theta & sin\theta & 0 & 0 \\ -sin\theta\ cos\alpha & cos\theta\ cos\alpha & sin\alpha & 0 \\ sin\theta\ sin\alpha & -cos\ \theta\ sin\alpha & sin\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

- Next, it’s followed with a translation by r which is the distance between the gimbal's center and camera's center.

$$C = \begin{bmatrix} 1 & 0 & 0 & -r_1 \\ 0 & 1 & 0 & -r_2 \\ 0 & 0 & 1 & -r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

- Finally, when the two systems are aligned the world coordinates are multiplied by the Perspective projection P and the equation is

$$\underline{c}_h = PCRG\underline{w}_h \tag{2.8}$$

and the PCRG matrix combined together is denoted as T which is the transformation matrix.

Suppose given a set of points in the image $\underline{c}_h$ and corresponding set of points in the 3D world$\underline{w}_h$, then the transformation equation is of the form

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_M \\ y_1 & y_2 & \cdots & y_M \\ 1 & 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{bmatrix} \begin{bmatrix} X_1 & X_2 & \cdots & x_M \\ Y_1 & Y_2 & \cdots & y_M \\ Z_1 & Z_2 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \end{bmatrix} \tag{2.9}$$

$$B = T.A$$

where transformation matrix T is apriori unknown. It is called as projective depth. Mathematically T is estimated and given as $T = BA^T (AA^T)^{-1}$ with an estimated total squared error ε.

2.3 Depth Extraction and Reconstruction

Depth extraction is the process of recovering the length of Z axis from the center of the imaging plane to the object location. In a perspective projection model all the points along the projected line falls under a single point in the imaging plane and therefore there is loss of depth information. The ultimate aim of the depth extraction and reconstruction is to find the distance of each point in the projected line from the world coordinates. This problem is solved either by using two cameras or by obtaining two views of the same object each acquired from a different viewpoint in space. The images can be obtained using two cameras or one moving camera. The following image explains the terminologies associated with a stereo camera system.

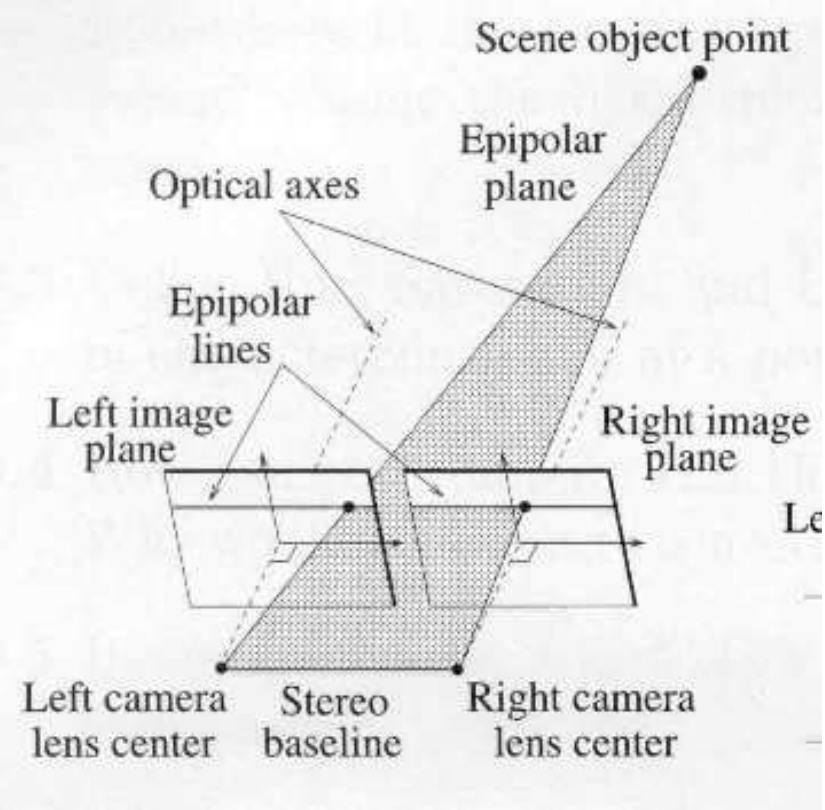


Figure 4: Stereo Camera System

Some of the terminologies involved with a stereo camera system are explained below:

- Fixation point: The point of intersection of the optical axis.
- Baseline: The distance between the centers of projection.
- Epipolar plane: The plane passing through the centers of projection and the point in the scene.
- Epipolar line: The intersection of the Epipolar plane with the image plane.

- Conjugate pair: Any point in the scene that is visible in both cameras will be projected to a pair of image points in the two images.
- Disparity: The distance between corresponding points when the two images are superimposed.
- Disparity map: The disparities of all points from the disparity map (can be displayed as an image).

The mathematical concept that is used to recover the depth using a stereo based system is called triangulation. This will be explained in Chapter 4 under section 4.2. The triangulation is directly associated with problem of stereo correspondence and a bad correspondence between points in two images will result in ambiguous interpretation of the scene.

The Reconstruction stage is the final step of a building a 3D model of the acquired image scene. Given the corresponding points, the disparity map of the scene is formed. This disparity map is converted into a 3D map of the scene if the stereo geometry is known. The 3D Image reconstruction will be explained in detail in Chapter 4 under section 4.2.

2.4 Pointing Gestures

Human gestures are one of the main modes of communication in man-to-man interactions. The same applies to human machine interactions. A lot of computer vision based products embed gesture recognition technology because it gives the opportunity for humans to stay in their comfort zone while communicating with the machines. Gestures are many, starting from waving, thumbs up, calling, pointing, signs and the list goes on. In this thesis study we concentrate on a particular gesture called Pointing Gestures. Pointing gestures can be of various types and here we show some of the instances of pointing recorded during our experiments with elderly people.

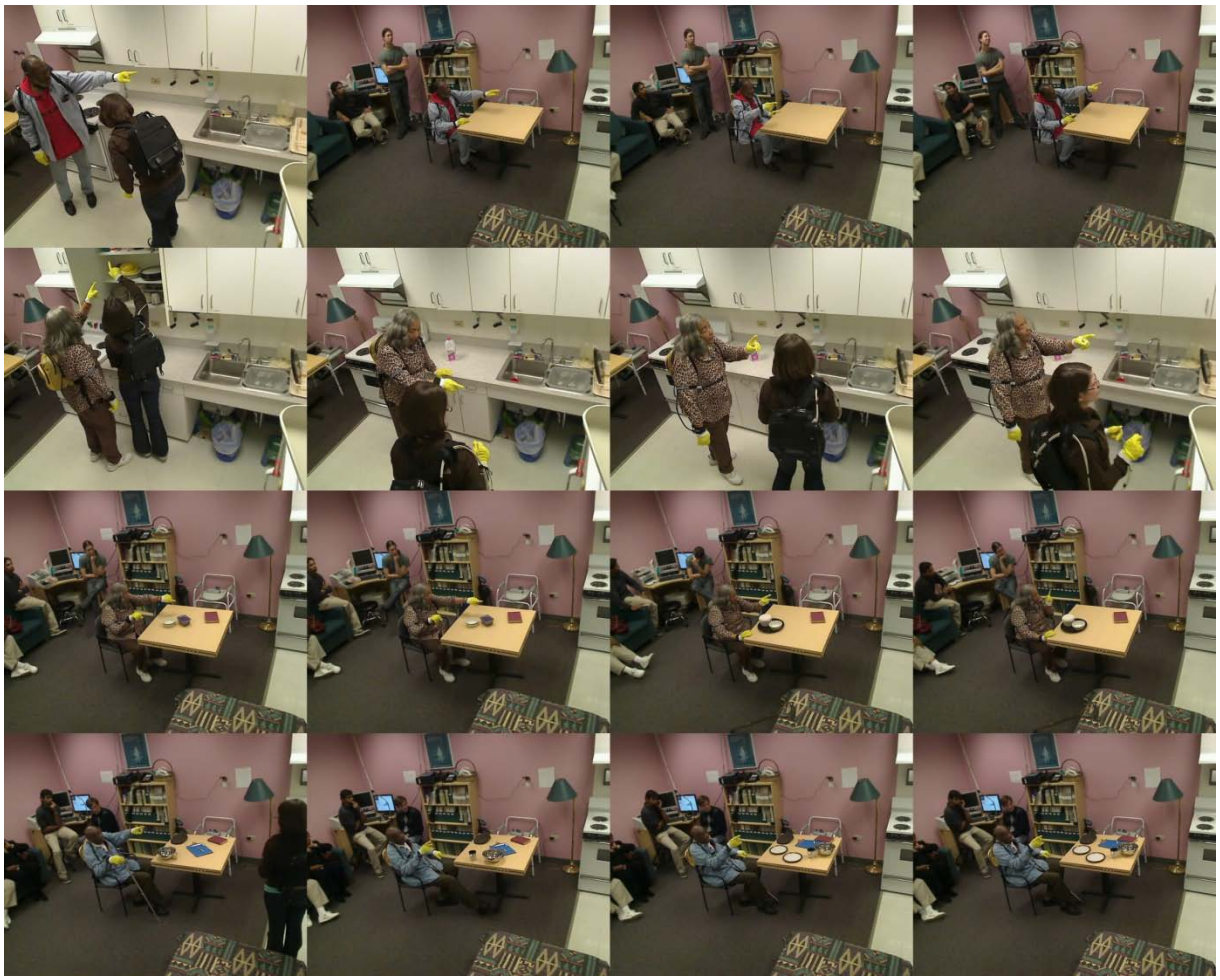


Figure 5: Pointing Gestures of Elderly Patients

As shown above the users point using different hand positions namely index finger pointed, all fingers pointed, forearm aligned with eyes, elbow bent, forearm rested, and wrist position bent.etc. Pointing gestures have the advantage that they express an accurate location and can act

alone as a form of communication. Pointing also can be mixed with speech to express more explicit request. Usually Speech and Pointing go hand-in-hand. Some of the examples are "Take that bottle", "Go there", "Bring it to me" and all these are phrases accompanied by a pointing gesture. From a computer vision point of view the pointing gesture has more visual clues than others because the head of the person is oriented towards the pointed object. Hence when we perform pointing gesture detection or recognition, concepts of head pose detection, eye gaze tracking can also be used. In Chapter 7 we shall elaborate about Accurate Pointing Vs Other types of pointing and assess their importance in finding the pointed location.

2.5 Previous Work

Several studies have been performed in the field of “Vision based pointing gesture recognition”. Pointing gestures are higher in the hierarchy of gesture based recognition and they are subdivided in to different research areas. Some of the work concentrates on just detecting the pointing gesture in an image and not the locations. Only few works compare the result of a pointing gesture by identifying the location pointed by the user by different pointing approaches [16]. The research studies which use 3D information in identifying targets compare their accuracy of their method by calculating the angular errors. In this section previous works in the area of pointing gestures are described and compared to the methods used in our approach.

In the paper [7] by Yamamoto & Yoda, the detection of intended pointing gesture is done to distinguish from unconscious arm movements by defining the intended pointing gesture as single straight movement of the arm. Then the target location is found using the concept of eye-finger tip approach which considers the target is located on a extension of line between the eye and tip

of the finger. The algorithm first detects the direction of hand and then adjusts itself to direction of eye-finger tip to find the target location. The system consists of four stereo cameras placed in four corners of the room and the targets are 20 panels in the wall. The results are tabulated for sitting and standing posture with recognition rates for each arm with different illuminations.

Another paper which describes this kind of pointing target estimation is [8] where the camera which sees the frontal pose of the face is identified and the cameras to right and left of this camera form a stereo pair to identify the locations of face and hand in 3D space. They achieve a mean error of 1.94^{o} with variance of 4.37^{o}. The face and hand regions are identified using skin region discriminant analysis in LUV Color space. Jojic et al. [9] use the dense disparity maps instead of skin color to control a cursor in the screen with a pointing gesture. The images are captured using a single stereo camera system and here the pointing direction is estimated as extension of forearm instead of LOS approach. The target is a flat screen which is near to the finger tip of the user. In a similar application using large displays Cabrini et al. [14] track the face and finger using skin color and EM algorithm but use eye-tip of the finger approach to get the accurate pointing location on the screen. The results are provided in terms of Accuracy of Pointing Vs Pointing time.

A very good approach is explained by Nickel and Stiefelhagen in [16] for detecting 3D pointing gestures. First color and range based tracking of head and hands are done, then the pointing gesture is detected by training a HMM with sample pointing gestures and applying it to real time scenario. The gesture recognition rate achieved is 88%. Then two approaches of head-hand line, 3D-forearm direction is used to determine the pointing direction. The average error angle and

percentage of targets identified are tabulated. The Head-hand line beats the forearm line method in both the parameters. The overall correct pointing target was found in 90% of the pointing gestures. The stereo system on the robot is used to recover 3D information of the scene.

Ching-Yu Chein et al. [17] uses particle filter for tracking the forearm in the scene and estimate the pointing locations. One advantage of this method is three cameras are fixed in the ceiling and user need not be restricted to be in the view of a single camera. The results are evaluated based on hit rates of each pre-designated target with a spherical tolerant space. Average hit rate of all targets is 85% and it's increased to 95% by refinement process. In Refinement process the corresponding points are adjusted to meet the constraints of Epipolar geometry and hence the major axes of pointing arm is refined.

In another approach by Nickel and Stiefelhagen [18] multi modal communication interface is being used to determine pointing targets. The acoustic model helps in detecting pointing gestures and a combination of speech recognition with head pose estimation for pointing gives the best results for estimating the pointed target. Sakuri [20] uses color markers in RGB color space to identify pointed targets using a stereo system fitted in the ceiling. The user is being asked to wear a colored glove and colored cap to estimate the 2D positions of head and hand in the video. The errors are tabulated for targets in terms of centimeters.

Most of the approached used head-hand line, eye-tip of the finger, forearm-line methods to identify the pointed targets. There are other methods too like [21] where Hyung-O Kim et al. uses shoulder-to-hand line approach to determine the pointing direction. The pointing gestures

are limited to extended arm pointing which are called as arm-pointing gestures and their approach describes the way to solve the problem of finding targets even if the user is not looking at the object. There are other pointing target detection papers which include use of other optical devices such as structured light systems, Z-cams and we are not discussing about those since they are out of the scope of this project.

Hence most of the previous works indicate that the pointing gesture detection and target estimation revolves around human head, hands, forearm and finger tips. This suggests that in order to determine pointing targets we can concentrate only on the top part of the human body. This also suggests that the task of tracking the human location in the room is also important for pointing. For our study of "Robotic assistants for elderly" we are using a multi-camera system which will give us an advantage over other stereo based system by covering the whole room instead of just a part of the room. We use the forearm and straight arm approach based on the study of pointing by elderly patients during our experiments. A lot of elderly patients don't use the accurate pointing mechanism of having the extended fore-arm so that the tip of the finger, eye and target fall on the same LOP.

We calculate the errors using forearm technique and rectify the constant error in elevation angle to identify the pointing locations accurately. We also use multi modal approach to get a robust pointing detection mechanism and which will be combined with head pose estimation in the near future

3. OUTLINE OF HUMAN POINTING TARGET ESTIMATION FRAMEWORK

3.1 Robotic Assistants for Elderly: Integrating Speech, Vision and Haptics

This thesis study is closely tied to the overall aim of the project. It focuses on the area of Human-Robot interaction with the emphasis on a multimodal communication interface that fits the needs of elderly persons or patients. One of the uniqueness of this study is haptics is considered also as a mode of communication with speech and vision. The proposed interface will also learn and adapt the communication to each user. To recognize these speeches, gestures and haptics signal we use a novel, reliable recognition methodology called RISq (Recognition by Indexing and Sequencing). RISq [25] was invented by Prof. Jezekiel Ben-arie in Machine vision lab at University of Illinois Chicago. RISq achieved 100% recognition rate with human activity recognition [26] for activities such as walking, running, sitting, jumping etc. RISq can be adapted any vector sequence hence even pointing gestures can be identified effectively with it.

There are various instances in ADL where multimodal communication plays a key role. Consider the following communication "let's put water in this one", here "this one" has deictic reference and needs vision and haptic modes to describe it. When the human uses a pointing gesture to elucidate the above phrase the context of communication is found easily. In this scenario the pointing gesture helps the speech and it can act vice-versa too. Pointing gestures can be detected by monitoring the speech signal of the human and identifying instance where the human uses the words like "this one", "that one", "get that", "over there", "here" etc. In the paper [28] Lin chen, Di Eugino talks about the correlation between the occurrence of pointing gestures and certain

deictic phrases. The following figure is taken from the abstract paper by Di Eugino, Zefran, Ben-Arie et al [27] which explains the architecture of the proposed multimodal user interface.

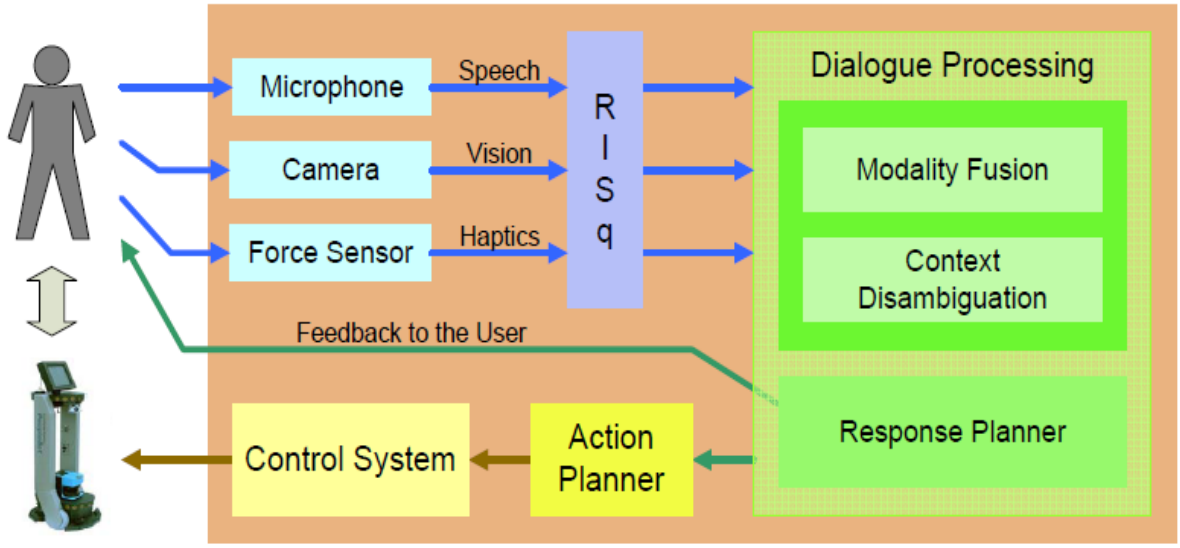


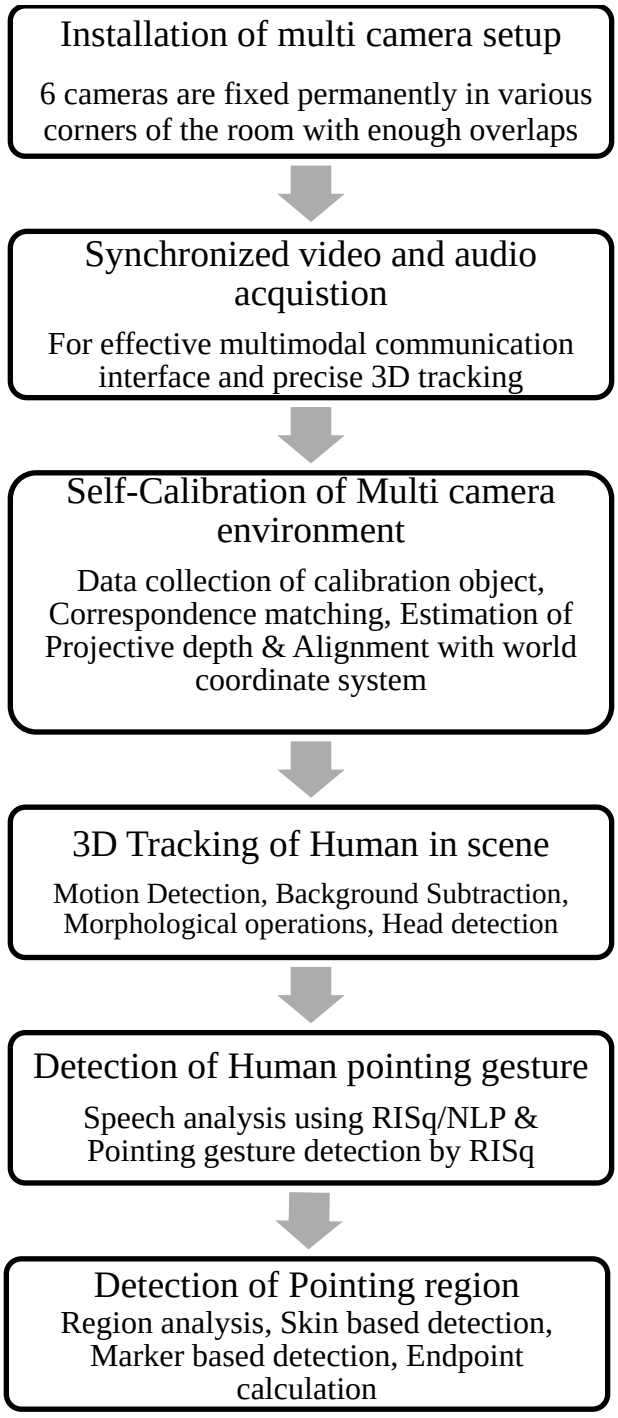
Figure 6: Architecture of multimodal user interface

Since our study is part of this project, other modes will be used to identify pointing gestures with the vision based methods. This is one of the main reasons why we concentrate mainly on the post-pointing gesture stage of determining pointed targets. It contributes effectively to the robotic guidance in our study. The Robotic assistants which will be Humanoid robots will need an accurate output from this system to aid the elderly. There will be other gestures that will be

needed to be recognized but might not need 3D information. The pointing gestures will need the 3D information hence it needs an extra stage of calibrating the environment accurately.

3.2 Estimation of Human Pointing targets

The flow chart below describes the process involved in the human pointing target estimation.



Human pointing target determination flow chart (Continued)

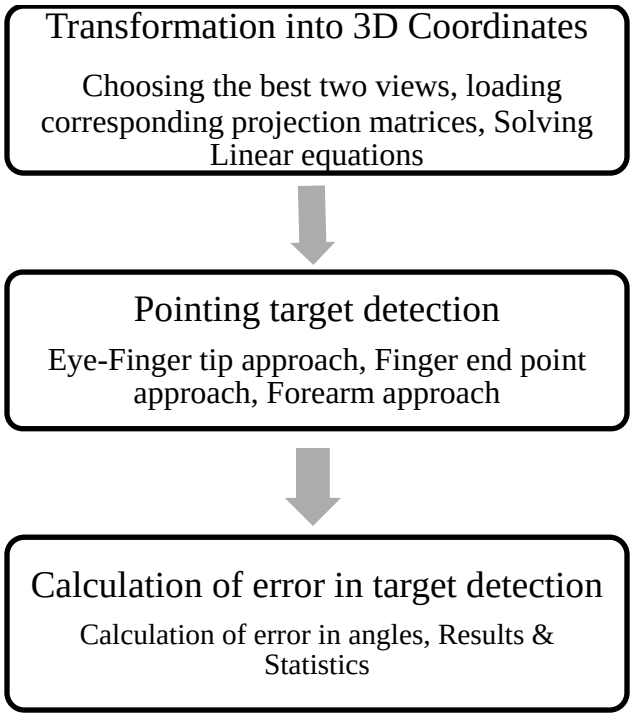


Figure 7: Human pointing target determination framework

3.2.1 Self-Calibration of Room

Self calibration of a real world environment is a tedious and lengthy process. There are various ways to self-calibrate a room. We wanted to make this process less time consuming and repeatable to adapt the changes to a multi camera setup. After reviewing the self-calibration literature [29] we decided to implement the method [30] that was used by Tomas Svoboda et al, for self calibrating virtual environments. The great advantage of this method is there is no need of any huge calibration object; instead we just need a small light source. The method can

calibrate any environment irrespective of the size of the environment and can even handle occlusions in data collection. All the cameras need not see the calibration object all the time and the parameters of self-calibration can be controlled and customized according to the environment, camera being calibrated. The whole process takes only 30-45 minutes and depends on the quality of data collected for self-calibration process. This will be explained later in chapter 5 in detail. The first 3 steps in the above flowchart constitute the self-calibration process.

3.2.2 Detecting Pointing Region

The second major stage in the human pointing target estimation is detecting the desired pointing region. Before we start detecting the pointing region, it must be well defined in the study. As explained earlier there are various segments in the human body that can be detected and tracked for accurately estimating the local targets. The region which is being used to locate the targets are called pointing region. In our study we concentrate on forearm and marker based finger detection. The various stages in the pointing region detection are as followed and will be explained in detail in Chapter 6.

- Motion Detection & Background Subtraction
- Morphological operations and Binary image analysis
- Human Silhouette extraction and tracking
- Head detection
- Skin color segmentation or marker color segmentation
- Classification of Pointing regions from false positives caused by previous steps

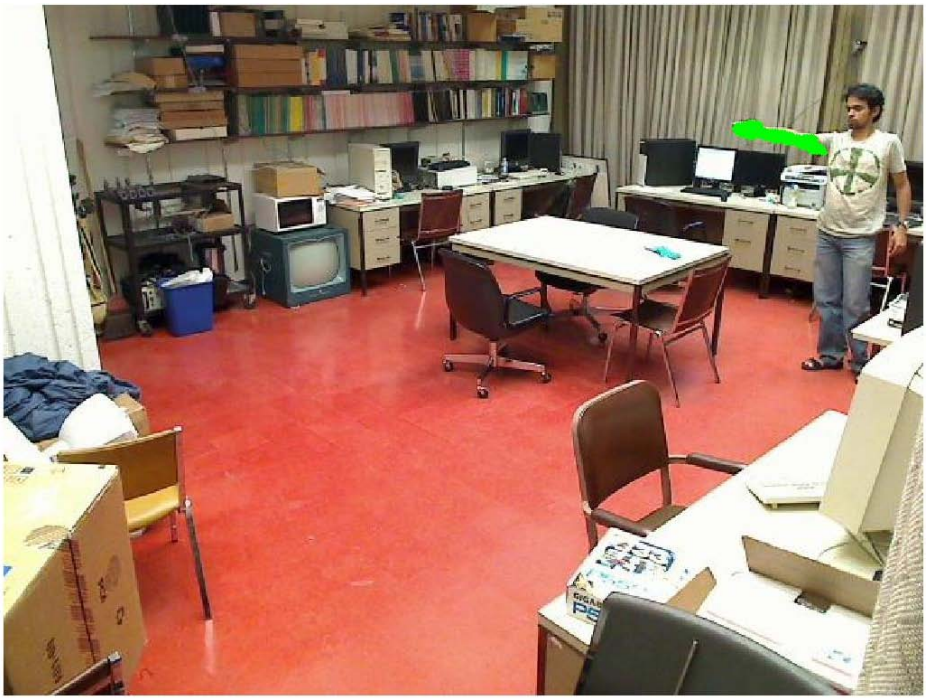


Figure 8: Forearm detection on pointing frame

Figure 9: Finger detection on pointing frame

3.2.3 Finding Human Pointing location

The third and final stage of the target determination framework is finding the human pointing location. The output of the pointing region detection stage is taken and the endpoints of the region are found. These endpoints are then transformed in to 3D coordinates by collecting the same set of end points in another view. Once the 3D coordinates of the pointed region are found the LOP method is used to determine the exact location of the target. We created a 3D database

of objects in the room with their 3D coordinates to improve our target determination. By using such a database, the human can be informed if there are multiple objects in line of pointing. If the user points to unknown location then the 3D coordinates of the unknown location with the closest objects to the unknown location are found. The detected target is communicated back to the user both by speech output and visuals.

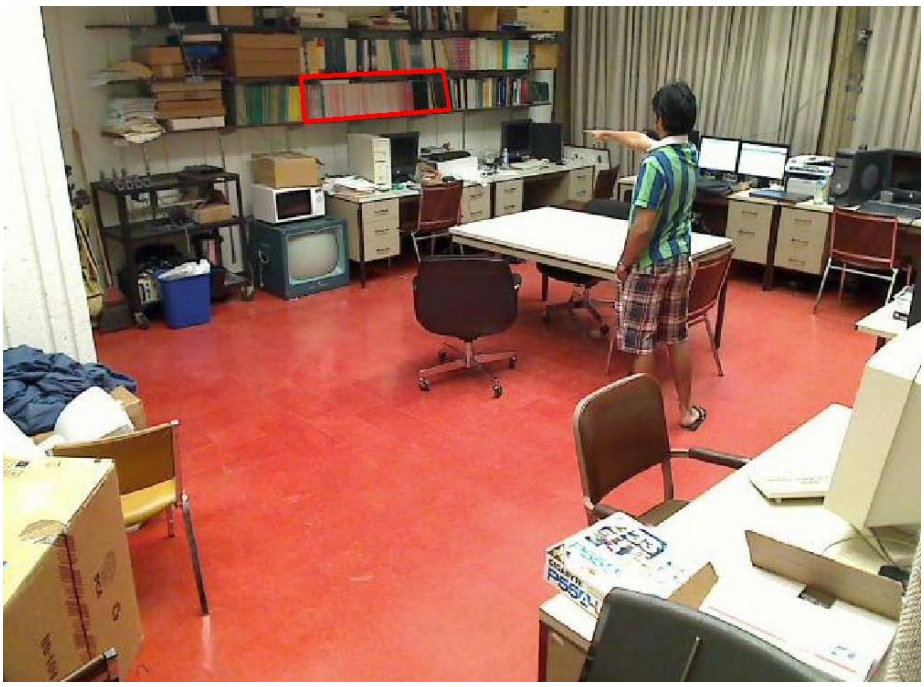


Figure 10: Target Bounded by Red Box

4. CAMERA CALIBRATION AND STEREO RECONSTRUCTION

4.1 Single Camera Calibration

Camera calibration is one of the essential tasks in computer vision and robotics when dealing with 3D systems. Geometric camera calibration is about the problem of determining the accurate mapping between the 3D coordinates of viewed points and 2D coordinates of viewed points in the image sensor. A camera model may be defined with two sets of parameters namely extrinsic and intrinsic parameters. The former depends on the pose of the camera as a whole and the latter depends upon the internal nature of the lens and sensor.

4.1.1 Projective Geometry

Have a look at the following diagram [39]

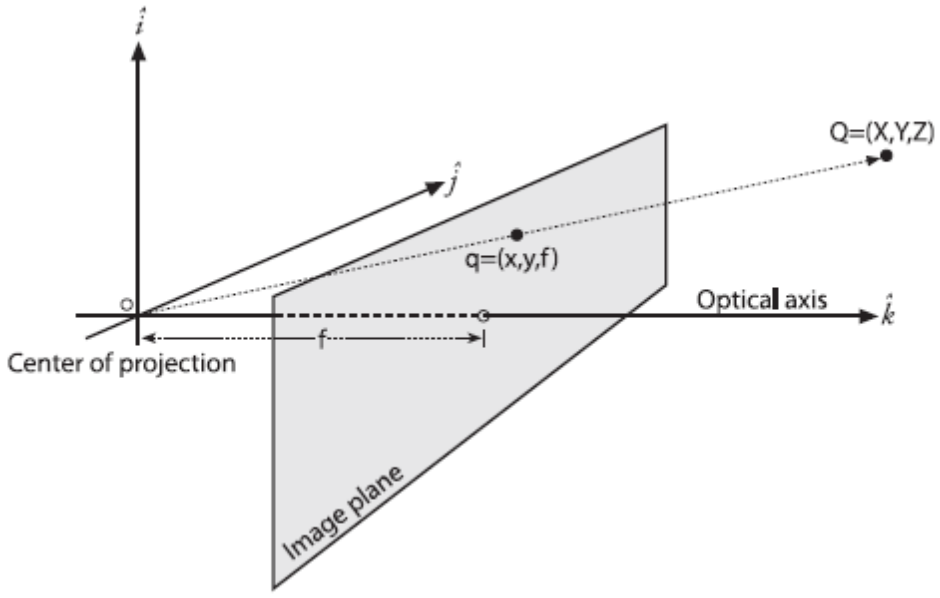


Figure 11: Projection of 3D point into 2D Image plane

A point Q = (X,Y,Z) is projected onto image plane by the ray passing through the center of projection and the resulting point on image plane is q = (x,y,f) or (x,y). The camera performs a linear transformation from the 3D projective space to the 2D projective space. The projective transformation was explained in Chapter 2 under section 2.2. The transformation matrix $T = [T_{ij}]$ has 12 elements arranged in the form of 3 by 4 matrix and it has 11 degrees of freedom since only the ratios of T_{ij} are important. When transformations are used, homogeneous coordinates are being used. So both the coordinates will be represented as n+1 vectors where n is the number of coordinates in each space. A *collineation* is a mapping between projective spaces, which preserves collinearity. A collineation from P^m to P^n is mathematically represented as (m+1) x (n+1) matrix and that's the reason we have the T matrix as 3 by 4. In previous sections the transformation till image coordinates are explained. In next section, the transformation from Image plane to pixel coordinates is described

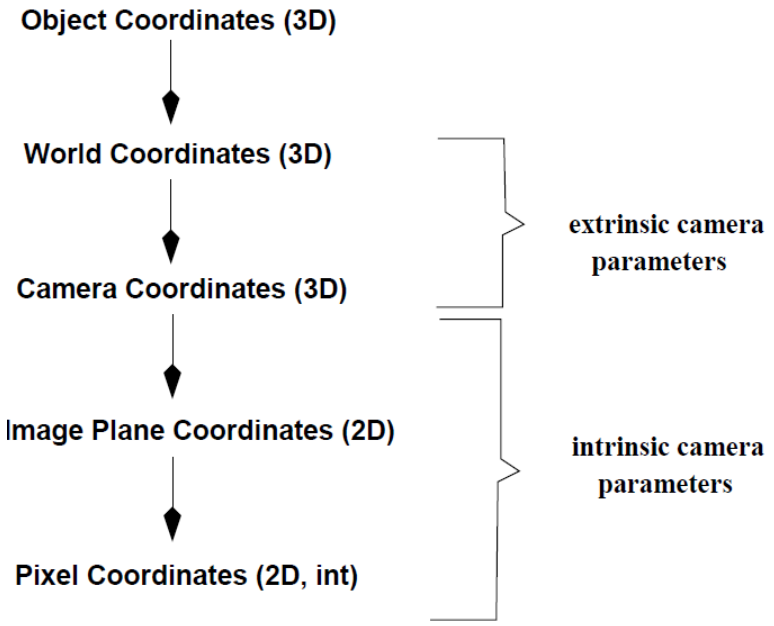


Figure 12: Different Coordinate frames in Camera calibration

4.1.2 Intrinsic and Extrinsic Parameters

As shown in the above flowchart a pixel coordinate system is different from the Image plane coordinate system. A pixel coordinate system is defined as (u,v) [row,column] array with its [0,0] either at top left or bottom right of the image. The picture below will give you an idea about these statements.

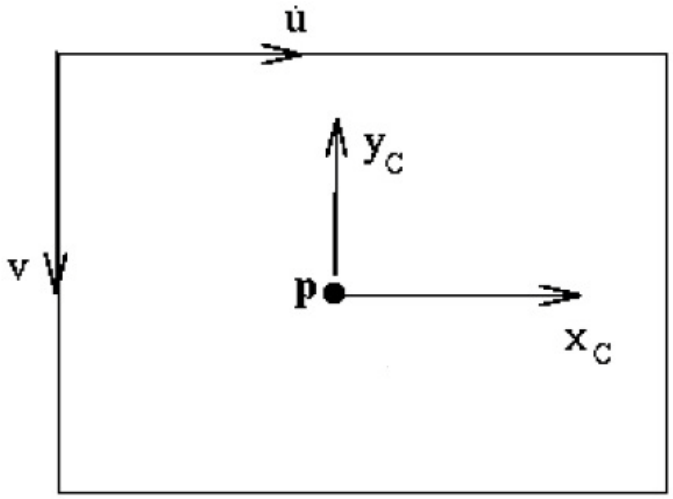


Figure 13: Relation between pixel coordinate system and Image plane coordinates [47]

The concept intrinsic parameters and camera calibration matrix are used for this. Normally in most of the CCD cameras the principle point is not equivalent to the center of the imager because the center of the chip is usually not on the optical axis. To model this displacement, two parameters namely c_x and c_y are introduced. On a low cost imager the individual pixels might be rectangular instead of square. To model this error another two parameters are introduced namely f_x and f_y. Intrinsic parameters describe the conversion from unit focal length metric to pixel coordinates and vice-versa. From [48] the intrinsic camera matrix is of the form $C = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$. The parameters derived in Chapter 2 under section 2.2 by transforming from

world coordinates to camera coordinates are called as extrinsic parameters. Hence the Euclidean transformation between the camera and world coordinate is expressed as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = C\ [R \mid t] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = C\ [T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{4.1}$$

where C and T are matrices with intrinsic and extrinsic parameters. R and t are the rotation and translation matrices which form the extrinsic matrix. Multiplying C and T, we obtain a 3 by 4 matrix which will be called as Projection Matrix P_E for a camera.

$$P_E = C\ [R \mid t] \tag{4.2}$$

The following picture [49] will summarize the steps involved in projective transformation.

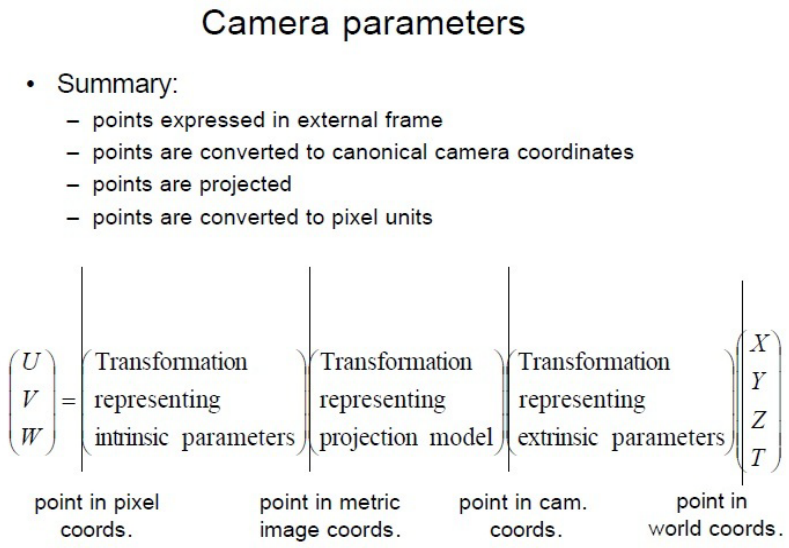


Figure 14: Summary of transformations from 3D to 2D.

4.1.3 Zhang's Chessboard Calibration Method

One of the biggest advancements in camera calibration techniques came from Zhengyou Zhang [50] when he proposed a simple calibration method. It uses chessboard as a calibration object causes it uses regular patterns of black and white. The endpoints of each grid can be easily found using Harris corner detection [51]. Each corner is found to sub pixel localization. The process needs frames of a person holding the chessboard in different planar views as the input. The more the numbers of views better the calibration. Some of the chessboard images captured in the lab during calibration of Sony security cameras are shown below.

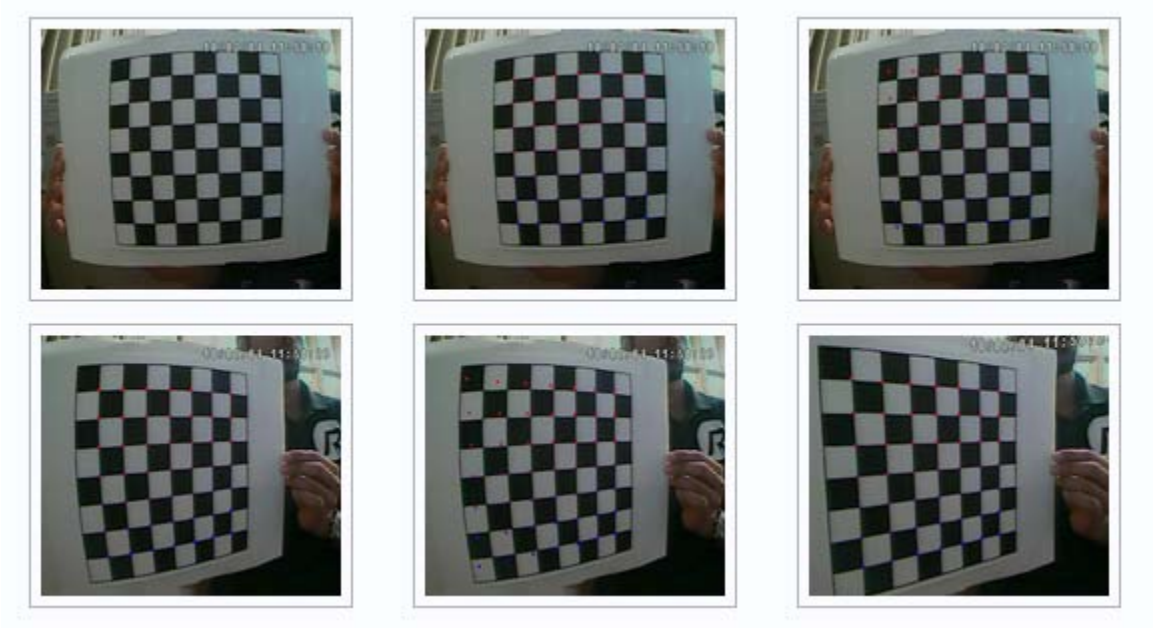


Figure 15: Zhang's chessboard calibration method

The pattern size must be of standard metric units such as 1 inch or 1cm for easy calculations. During this calibration, Open CV's [39] chessboard calibration functions were used.

All the interior corners of the chessboard are found and the pixel locations are stored in matrix format which forms the input. The next step is to find the 3 x 3 homography matrix which is nothing but a projective mapping from one plane to another. Multiple images of same object are used here to find the individual translations, rotations and intrinsics as well. Rotation and translation are totally six unknown parameters and each new frame (with a planar object of chessboard) gives eight equations. Hence if enough images are obtained, these parameters can be found and homography can be calculated without knowing the intrinsics too. To find the intrinsic parameters and distortion parameters which are three radial (k1,k2,k3) and two tangential (p1,p2), just one view is needed because these parameters are not tied to 3D world. The radial distortions are mainly caused because of the shape of the lens and tangential distortions arise from assembly process. Real cameras don't have tangential but do have radial distortions.

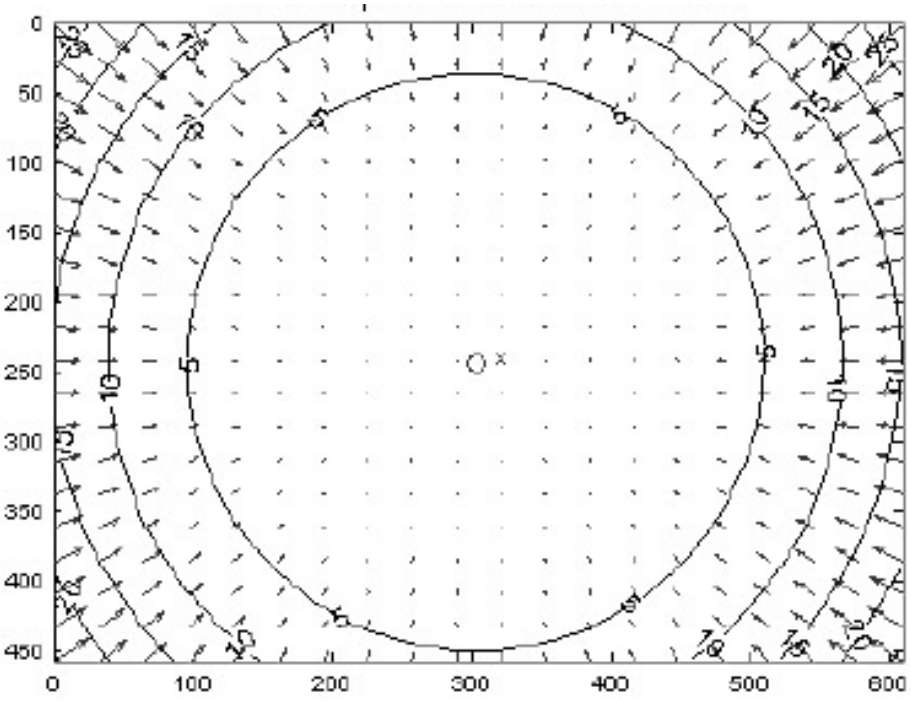


Figure 16: Radial distortions plot for camera lens

A large list of linear equations are collected and solved to find the distortion parameters. Then the intrinsic and extrinsic parameters are re-estimated.

4.1.4 Undistortion

Undistortion is the process of removing radial and tangential distortions in a camera. This is very important stage in camera calibration because the pixels get displaced from their original pixel locations in the 2D image cause of these distortions. Undistortion puts it back to the right

locations. Even the displacement of 2 or 3 pixels might cause huge measurement errors in 3D. To perform Undistortion first we need the distortion map of the cameras. The distortion map is then converted into two single arrays of x and y values and the operation is done on each pixel. Below is the stereo images of original and undistorted, Undistortion was done using Open CV.

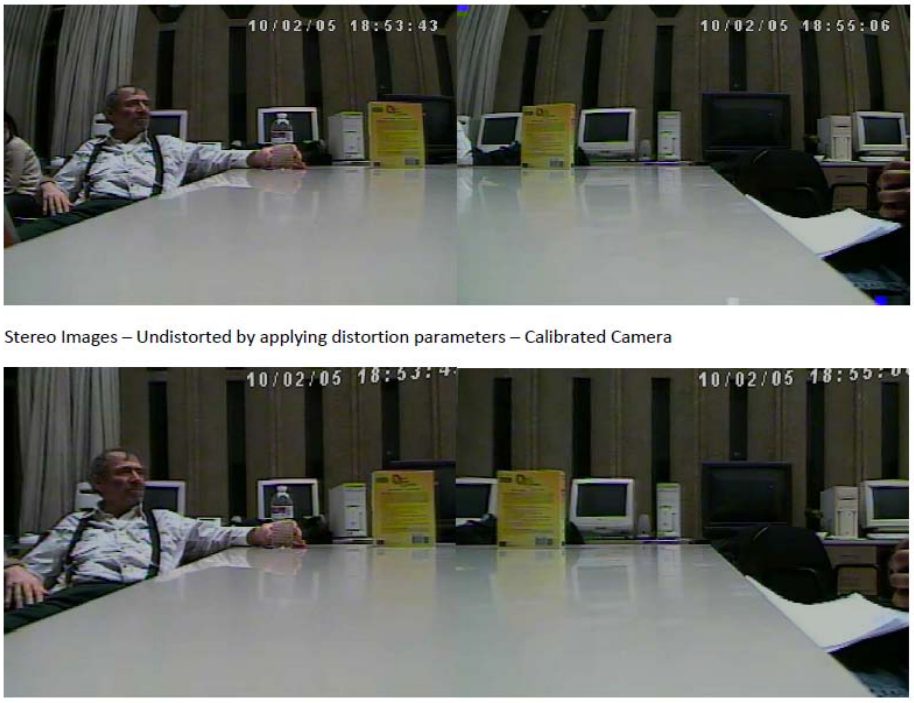


Figure 17: Undistortion of images

4.2 Stereo Camera Calibration and Depth Extraction

Stereo cameras are pairs of same models of cameras separated by a baseline looking at the same object in a scene. Stereo cameras are the basic building blocks of Depth extraction and 3D reconstruction techniques. Refer to Figure 16 to see the images of a stereo camera. These are images of a book taken from two separate cameras with their optical axis parallel to each other. The following sections will describe how the 3D information is extracted using stereo camera setup.

4.2.1 Triangulation

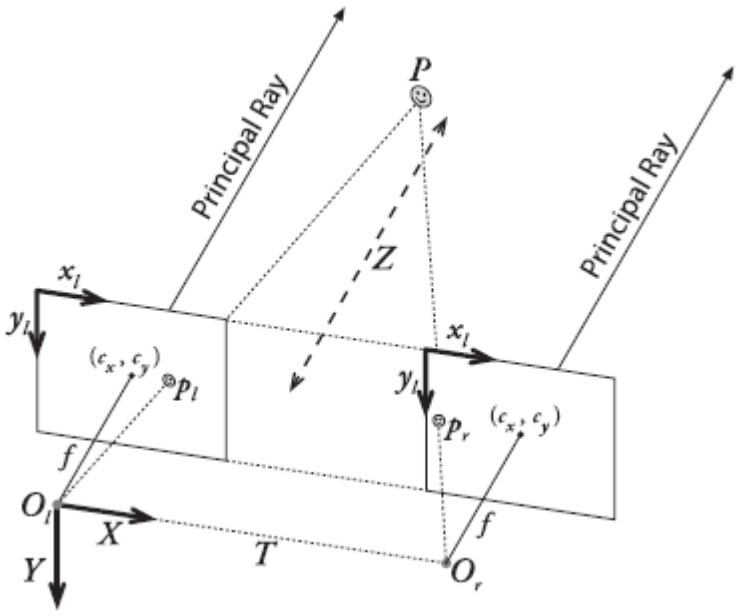


Figure 18: Triangulation in stereo camera setup

Triangulation is the basic geometrical concept used in a stereo camera setup to find the distance between the 3D world point and the cameras. In the above diagram [39] P is the world point being seen in two cameras and the projected points are namely p_l and p_r. The line O_lO_r is the base line of the camera and this gives the distance T between two camera's optical axes. The above camera setup is a basic stereo setup where the images are row-aligned and there is no disparity in y axis of the image plane. Hence the disparity in the above image is$d = x^l - x^r$, where x^l is x coordinate of pixel in left image and x^r is the corresponding x coordinate in the right image. The depth Z of the 3D point can be easily derived using the concept of similar triangles and it is given by

$$Z = \frac{fT}{x^l - x^r} = \frac{fT}{d} \tag{4.3}$$

4.2.2 Epipolar Geometry

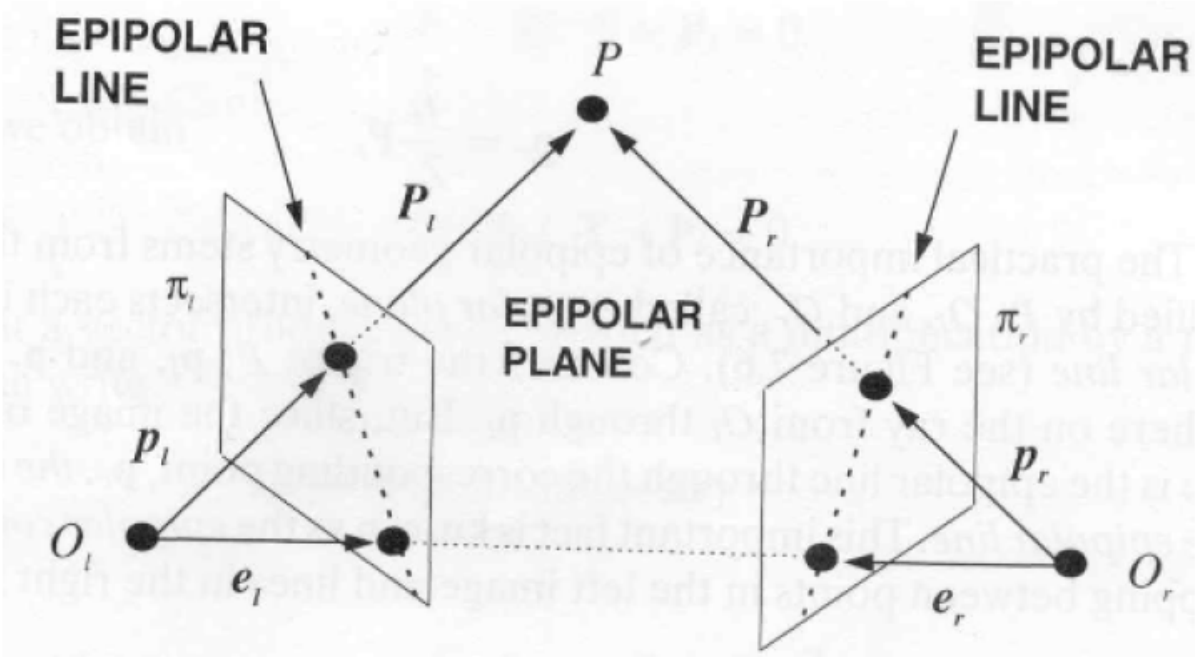


Figure 19: Epipolar Geometry

Another important geometrical concept that deals with stereo cameras is Epipolar geometrical concepts. Epipoles are images of projection centers of one camera on the other. In the above figure left epipole will be the projection of O_r on the left image plane and right epipole will be the projection of O_l on the right image plane. The Epipolar plane in the above image is defined by points P, O_l and O_r. An Epipolar line is the intersection of Epipolar plane with the image plane. If the line through the centers of projection is parallel to one of the image planes then the

corresponding epipole will be at infinity. One of the major advantages of using Epipolar geometry is in the search of corresponding points using epipolar constraint. According to it the correct match must lie on the Epipolar line. Now the search for the correspondences is reduced to 1D problem. This is very effective in rejecting false matches due to occlusions during self-calibration process of cameras.

4.2.3 Essential and Fundamental Matrices

How we do find the Epipolar lines in the images? The answer is two matrices namely Essential and Fundamental matrices. The essential matrix E contains information about the translation and rotation that relate the two cameras in world space. The fundamental matrix F is similar to essential matrix but it goes a step further by including the information about camera intrinsics too. Since F has the information of intrinsic parameters it will relate the camera in pixel coordinates. The essential matrix and fundamental matrix satisfies the following equations

$$p_r^T E p_l = 0 \tag{4.4}$$

$$\overline{p_r^T} F \overline{p_l} = 0$$

where p_l and p_r are corresponding points in image coordinates, $\overline{p_l}$ and $\overline{p_r}$ are corresponding points in pixel coordinates. $\overline{p_l} = M_l p_l$ and $\overline{p_r} = M_r p_r$, where M_l and M_r are camera intrinsic matrix for left and right cameras.

There are various ways to compute essential and fundamental matrices. The most used approaches are the 7-point algorithm, 8-point algorithm, RANSAC algorithm and LMedS algorithm. We are not going to discuss these algorithms because these are basic mathematical approaches for correspondence matching and out of the scope of this study.

4.2.4 Stereo Rectification

It is the transformation of each image such that conjugate Epipolar lines become collinear and parallel to the horizontal axis. Stereo rectification is done when there is disparity in both the axis and the correspondence matching becomes little complex. When the image is rectified the search for corresponding points becomes simpler cause the disparities are reduced to x axis only. The algorithm for stereo rectification is

- Rotate the left camera so that the Epipolar lines become parallel to the horizontal axis
- Apply the same rotation to right camera to recover original geometry
- Rotate the right camera by Rotation matrix R.

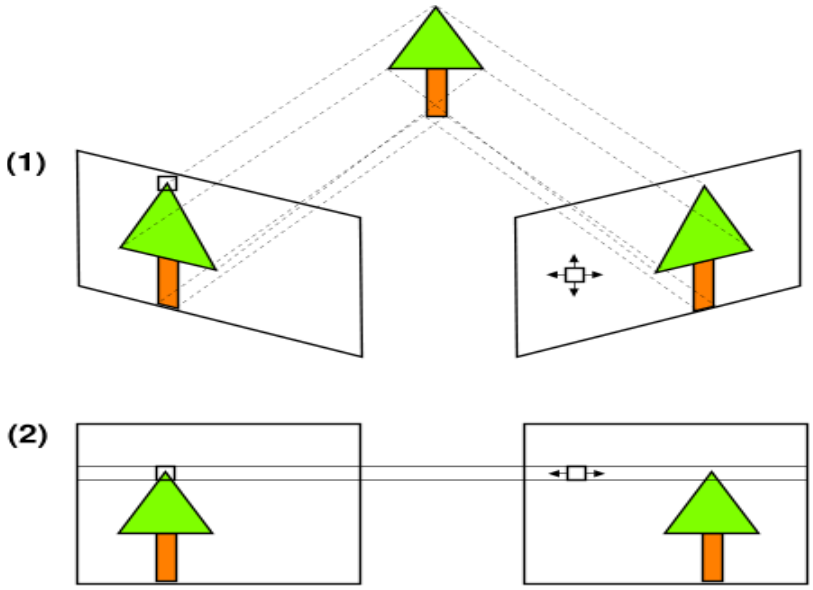


Figure 20: Rectification of Images [52]

4.2.5 Disparity Images and 3D Reconstruction

Stereo correspondences play a vital role in reconstructing the 3D image of the scene. Once the images are stereo rectified we can employ any correspondence finding technique such as block matching, template matching, etc. The corresponding points will be searched along epipolar lines for removing false errors and increase speed in calculating disparity. Once the correspondence is found then the pixel location are derived to calculate the disparity. A disparity map (shown below) is a grey level or RGB intensity map which visually shows the disparity in from an

image. When the intensity of pixels is high then the object is close to the camera and objects away from the camera have less intensity. 3D reconstruction can be done from disparity map by projecting it back into 3D space with depth information of each pixel. Sometimes there won't be disparity values in particular pixel locations and those places would have holes in the 3D model.

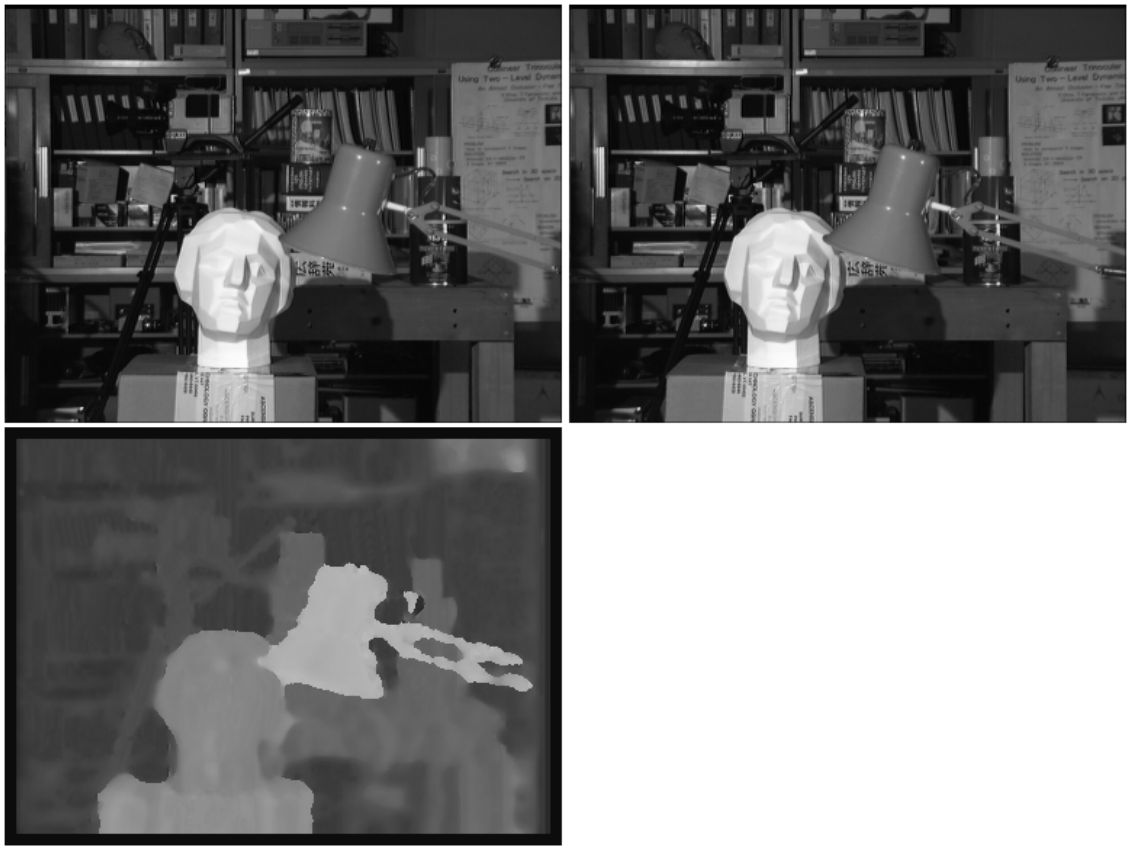


Figure 21: Stereo images and Disparity map. [53]

5. SELF CALIBRATION OF MULTI CAMERA ENVIRONMENT

There are three types of calibration methods namely Photogrammetric Calibration, Self Calibration and Multi plane Calibration. Photogrammetric calibration uses a big imaging pattern with multiple orthogonal planes with known 3D values having high precision. The Efficient calibration methods are Self-calibration and Multi plane calibration. The term self-calibration or auto-calibration is used to refer to calibration with no object or a small object. The metric properties of the camera and of the images scene are recovered from a set of un-calibrated images using the constraints on the camera parameters or images scene. Self-calibration is mostly used in 3D modeling to upgrade a projective reconstruction to a metric reconstruction. Most of the methods have constraints on intrinsic parameters of camera [29].

5.1 Multi view reconstruction

A minimum of two views are required to perform 3D reconstruction. As the number of views or cameras increases it doesn't bring any negative effect to the recovered 3D information unless a n^{th} camera is badly calibrated which will bring the accuracy down. The lower limit is 2 and there is no upper limit in number of views for a 3D reconstruction. The computational methods used in 3D reconstruction using a stereo system were discussed in the last chapter and for more number of views the number of steps in the reconstruction process increases. The following section will describe some of the important computational methods [45] used in projective reconstruction from a set of images.

5.1.1. Projective reconstruction by bundle adjustment

A set of 3D points X_j is viewed from a set of cameras with projection matrices P^i and x_j^i is the j^{th} point viewed by the i^{th} camera. Now the reconstruction problem is defined as: Given a set of x_j^i points, find the camera matrices P^i and 3D points X_j such that

$$P^i X_j = x_j^i \tag{5.1}$$

When noisy data is collected, the above equation will not hold true. In that case we use the Maximum Likelihood solution assuming the measurement noise is Gaussian. Using this method the projection matrices are to be estimated as $\widehat{P^i}$ and 3D points as $\widehat{X_j}$ which project back to image as $\widehat{x_j^i}$ and it's aimed towards minimizing the image distance between reprojected and detected point for each view

$$\min \sum_{ij} d\ (\widehat{P^i}\widehat{X_j}, \widehat{x_j^i})^2 \tag{5.2}$$

where d is the geometric image distance to be calculated. The process of minimizing the reprojection error is called as bundle adjustment as it involves adjusting the bundle of rays between each camera center and the 3D points. Bundle adjustment is normally used in the last stages of reconstruction and it is tolerant of missing data but still provides a good ML estimate. The only disadvantages are it needs a good initialization and can become a large minimization problem because of the large number of parameters. Bundle adjustment will be used in our self-calibration process depending on the reprojection error obtained.

5.1.2. Projective Reconstruction by factorization

The equation 5.1 which represents the projective mapping can be interpreted as true only up to a constant factor. Writing these explicitly we have

$$P^i X_j = \lambda_j^i x_j^i \tag{5.3}$$

where x_j^i is the homogenous image point. If each point is available in all the views the complete set of equations can be written in matrix form

$$\begin{bmatrix} \lambda_1^1 x_1^1 & \lambda_2^1 x_2^1 & \cdots & \lambda_n^1 x_n^1 \\ \lambda_1^2 x_1^2 & \lambda_2^2 x_2^2 & \cdots & \lambda_n^m x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^m x_1^m & \lambda_2^m x_2^m & \cdots & \lambda_n^m x_n^m \end{bmatrix} = \begin{bmatrix} P^1 \\ P^2 \\ \vdots \\ P^m \end{bmatrix} [X_1 \quad X_2 \quad \cdots \quad X_n] \tag{5.4}$$

The above equation holds good only if the correct weights λ_j^i are applied to each 2D point. The process of obtaining these weights will be described in section 5.4.2. For now let us assume these weights are known. The matrix on the left hand side in Equation 5.4 is denoted as W and transformed in to a matrix of rank 4 using SVD. If $W = UDV^T$, all but the first four diagonal entries of D are set to zero resulting in $\widehat{D}$. Hence the corrected measurement matrix will be $W = U\widehat{D}V^T$. Now the camera matrices are retrieved from $U\widehat{D}$ and the points from V^T. This process is called projective factorization. The algorithm is provided below

1. Normalize the image data using isotropic scaling [45].
2. Get the initial estimate projective depths λ_j^i using initial projective reconstruction or set it to 1.
3. Normalize the depths λ_j^i by multiplying rows and columns by constant factors.
4. Form the 3m x n measurement matrix on left side of Equation 5.4. Find its nearest rank-4 approximation using the SVD and decompose it to find the camera matrices and 3D points.
5. Reproject the points into each image to obtain new estimates of the depths and repeat from step 2 (Optional).

5.2 Self-Calibration Theory

In many cases the specific values of the intrinsic or extrinsic camera parameters are not known. Often there are, however, some restrictions on these parameters. The usage of these restrictions to achieve a metric calibration is called self-calibration or auto-calibration. The traditional self-calibration problem is more restricted. In this case it is assumed that all intrinsic camera parameters are unknown but constant and that the motion of the camera is unrestricted. This corresponds to an unknown camera which is freely moved around (e.g. hand-held). This problem has been addressed by many researchers in the past. In some practically important cases, however, the motion of the camera is restricted. This knowledge can often be exploited to design simpler algorithms. But these are not always able to retrieve all the desired parameters, since restricted motion sequences do not always contain enough information to uniquely determine the metric stratum of the reconstruction. Many methods exist in self-calibration but can be classified into few classes. The self-calibration method used in our study is a unique method where projection matrices are factorized and a condition that intrinsic camera parameters are constant is being imposed. This will be explained in section 5.3.3. Before that we have to get familiar with the concept of absolute conic.

The absolute conic is a conic on the infinite plane and is part of the 3D Euclidean geometry. In a metric frame $\pi_\infty = (0,0,0,1)^T$ and points on the conic (Ω_∞) satisfy

$$\left.\begin{matrix} X_1^2 + X_2^2 + X_3^2 \\ X_4 \end{matrix}\right\} = 0 \tag{5.5}$$

For more details about absolute conic refer Multiple View Geometry by Hartley and Zisserman [45]. It is invariant under Euclidean transformations; its relative position to a moving camera is

constant. For constant intrinsic camera parameters its image will, therefore, also be constant. This is similar to a person who has the impression that the moon is following him when driving on straight road [40]. Note that the absolute conic is more general, because it is not only invariant to translations but also to rotations and reflections.

5.3 Implementing Svoboda's Multi-camera Self-Calibration

We adopt Tomas Svoboda's multi-camera self calibration technique [42] to calibrate our environment. The minimum number of cameras is 3 and there is no maximum limit. The method is totally automatic and the only work from a user's perspective is waving a bright spot throughout the working volume. The cameras need not see all the points and only reasonable overlap between subgroups are needed.

5.3.1. Data Acquisition and detecting Corresponding points

A perfect data acquisition system is required to calibrate the environment. In calibration of the Machine Vision Lab, we use 6 High Definition USB cameras to record the videos in the room. Matlab Image Acquisition toolbox and i-Spy server software [54] are used to acquire video frames into our system. One important condition for the video acquisition is that the frames of each camera must be highly synchronized else the calibration will fail. For our experiments we use green LED light as the calibration object. This green LED light can be moved slowly or switched on and off across the whole volume of the room. It's better to cover as much volume as possible because more the volume of 2D points better the calibration. The lights in the room are switched off for easy detection of the bright spot and any reflective surface should be covered so

that the reflected bright spot might cause false positives in data collection. A sample frame which is used as input data to calibration process is shown in the next page.

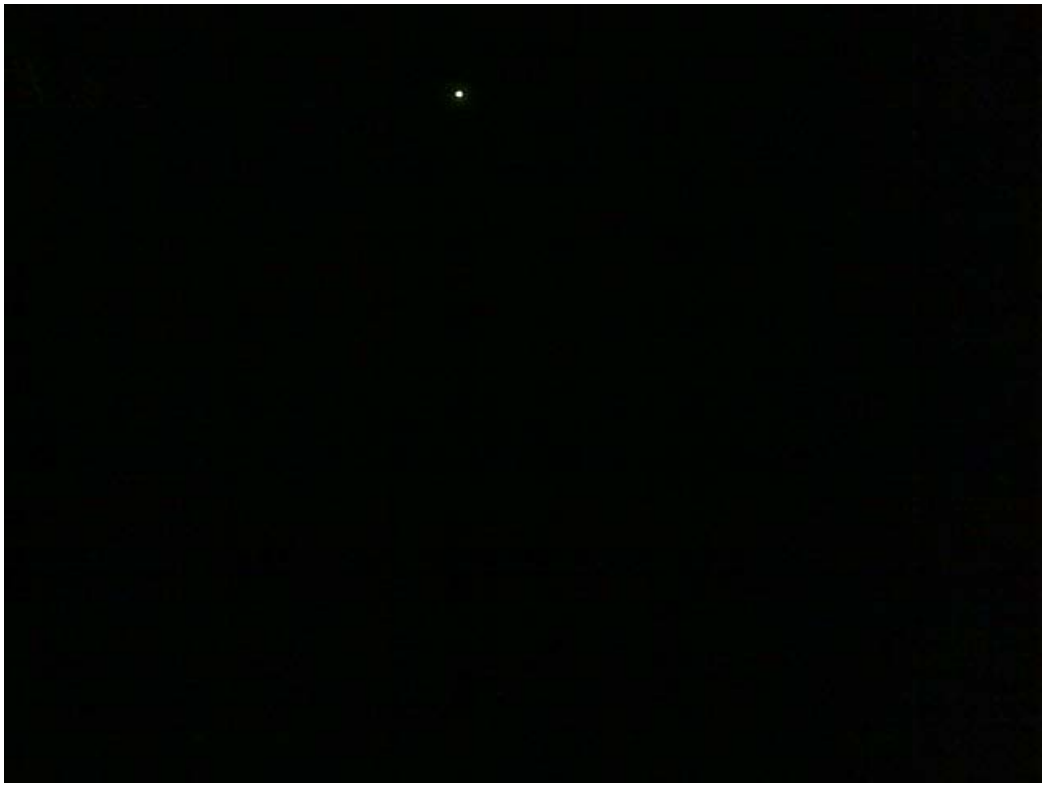


Figure 22: LED is the only bright spot in the frame: Data acquisition for calibration

For the detection of the bright spots which will be given as corresponding points to the calibration process, a robust image processing algorithm is used. The steps in the algorithm are:

1. The mean image and the image of standard deviation are computed for each camera. These two images represent the static scene and the projections of the laser pointer are found by comparing the actual image with these two.

2. The differential image is computed by using the appropriate color channel depending on the color of the laser pointer. A threshold is set to 4/5 of the maximum of the differential image. The image is discarded if any of the following conditions hold:

a) The number of pixels in the thresholded differential image is much higher than the expected LED size.

b) The maximum of the differential image is less than 5 times the standard deviation in this pixel.

c) The thresholded pixels are not connected, i.e. they compose more than one blob.

d) The eccentricity of the detected blob exceeds a pre-defined threshold. This condition is against motion blur.

3. The neighborhood of the detected blob is resampled to a higher resolution by using bicubic interpolation in order to reach sub-pixel accuracy and robustness against irregular blob shapes.

4. A 2D Gaussian is fitted to this interpolated sub-image by 2D correlation to get the final position of the LED projection.

The LED size and color of the LED needs to be altered according to the calibration object used.

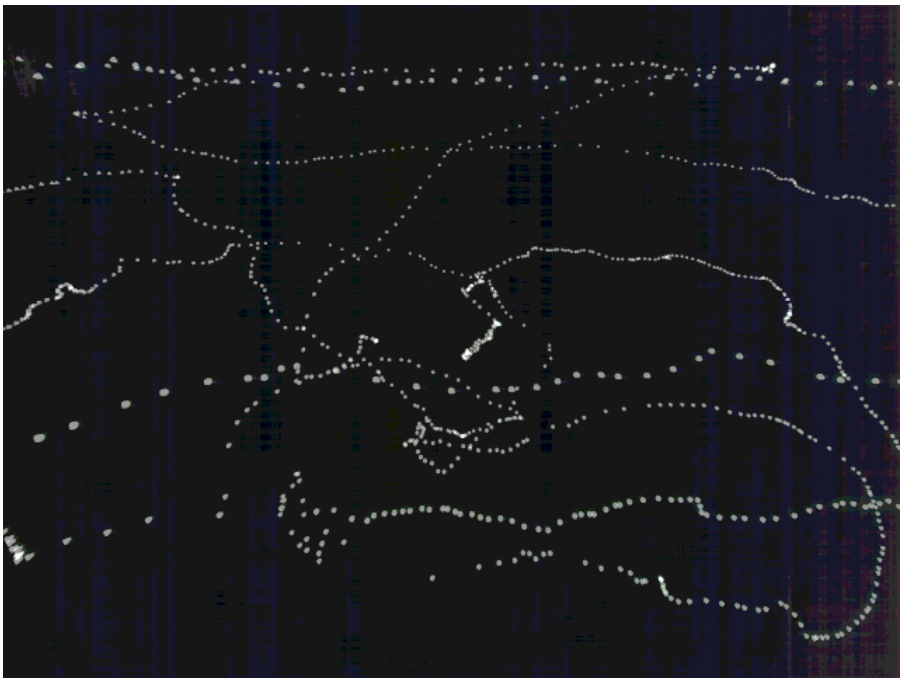


Figure 23: Standard deviation image. White spots indicate movement of LED pointer

5.3.2. RANSAC Analysis and Estimation of projective Depth

The next step is correspondence matching. Even though the above mentioned steps are highly robust there might be false positives because of glass walls, glossy surfaces, etc. Two discarding steps are implemented to stabilize corresponding matching. They are:

- First step is a robust pair-wise computation of Epipolar geometry and removing points that lie too far from Epipolar lines. This step cleans the data at the very beginning of the whole process.
- The second step is an iterative loop which removes outliers by analyzing 2D reprojection error.

The image pairs are iteratively re-selected according to the number of visible corresponding pairs. The points that were already detected as outliers are removed from the list of points found in these two cameras. The Epipolar geometry is robustly computed via the RANSAC 7-point algorithm [55]. The validation step based on the Epipolar geometry may fail to discard a misdetected projection if it lies along the Epipolar lines. However, such a point can often be correctly reconstructed in 3D space from other (good) projections. If projected back to the cameras where it was misdetected, it exhibits large 2D re-projection errors. Such problematic points are discarded from further computation.

Estimation of projective depth is based on the paper by Strum and Triggs [44]. The formula for calculating the projective depth is given below in Equation 5.6

$$\lambda_p^i = \frac{(e_{ij} \wedge q_{ip}) . (F_{ij}\, q_{jp})}{\|e_{ij} \wedge q_{ip}\|^2}\ \lambda_p^j \tag{5.6}$$

where i and j are images from successive cameras. e_{ij} is the epipole between views i and j, q_{ip} and q_{ip} are measured homogenous coordinate vectors of image points and F_{ij} is the fundamental

matrix corresponding to views from i and j. Initially λ_p^j is set to 1 and then its recursively chained together to give estimates of complete set of depths for all the points.

5.3.3. Rank 4 factorization and Euclidean Stratification

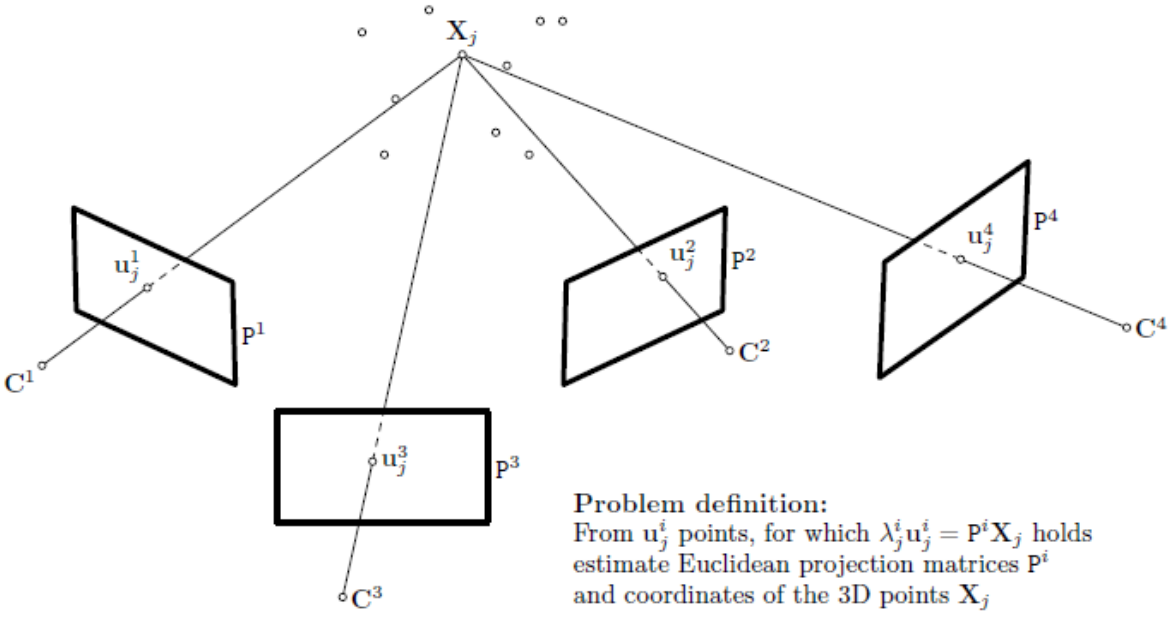


Figure 24: Multi camera Geometry

Have a look at the above figure and Recall Equation 5.4. The matrix on the left hand side is denoted as W_S which is called as scaled measurement matrix. Rewriting the equation in this form $W_S = PX$, where $P = [P^1 \ldots P^m]^T$ and $X = [X_1 \ldots X_n]$ are referred as projective motion and

projective shape respectively. If we collect enough 2D points and the scales λ_j^i are known, then W_S has rank 4 and can be factored into P and X as shown in section 5.1.2. [44]. The factorization recovers the motion and shape to a 4 x 4 projective transformation H:

$$W_S = PX = PHH^{-1}X = \hat{P}\hat{X} \tag{5.7}$$

where $P = PH$ and $\hat{X} = H^{-1}X$. Any non-singular 4 x 4 matrix may be inserted between P and X to get another compatible motion and shape pair$\hat{P}, \hat{X}$. The self-calibration process computes such a matrix H such that $\hat{P}$ and $\hat{X}$ become Euclidean. This process is called as Euclidean Stratification. The task of finding the appropriate H can be solved by imposing certain geometrical constraints. The most general constraint is the assumption that rows and columns of camera chips are orthogonal. Alternatively, we can assume that some internal parameters of the cameras are the same, which is more useful for a monocular camera sequence. The minimal number of cameras for a successful self-calibration depends on the number of known camera parameters, or on the number of parameters that are unknown but are the same for all cameras. For instance, 8 cameras are needed when the orthogonality of rows and columns is the only constraint and 3 cameras are sufficient if all principal points are known or if the internal camera parameters are completely unknown but are the same for all cameras [45]. The detailed description and derivation of the Euclidean stratification is available in Section 2.2 of [30].

5.3.4. Calibration Output and Alignment with world Coordinate System

Once the calibration process is completed, the images of each camera with detected points and reconstructed points are obtained. This image will clearly show if there are any bad reprojections in the camera. The distortion model of the camera is also presented to the user to verify the lens

distortion in the camera. Rectification of distortion is left to the user based on the values of distortion. A 3D plot of the calibrated environment with camera positions and orientation are displayed. Some of the calibration outputs are given in the next page.

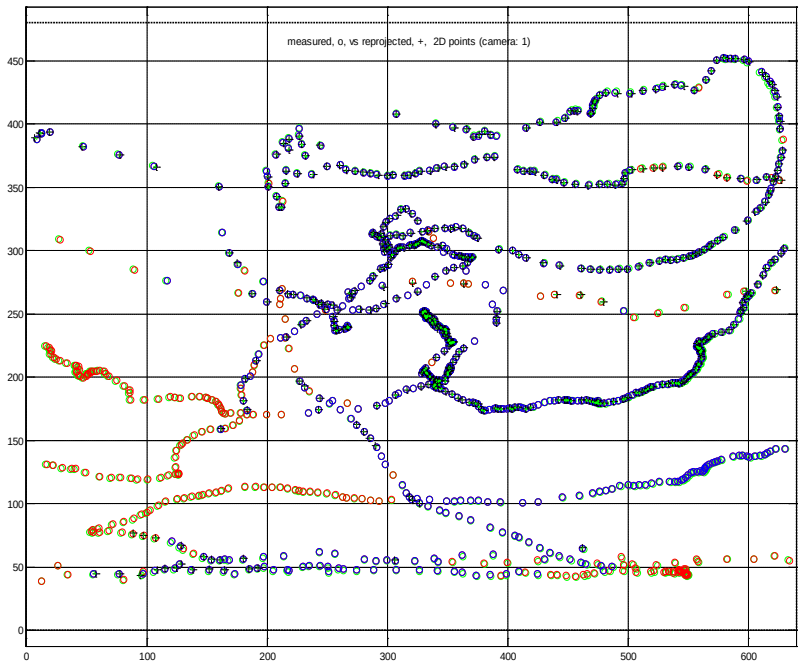


Figure 25: Calibrated camera - Detected Vs Reprojected points

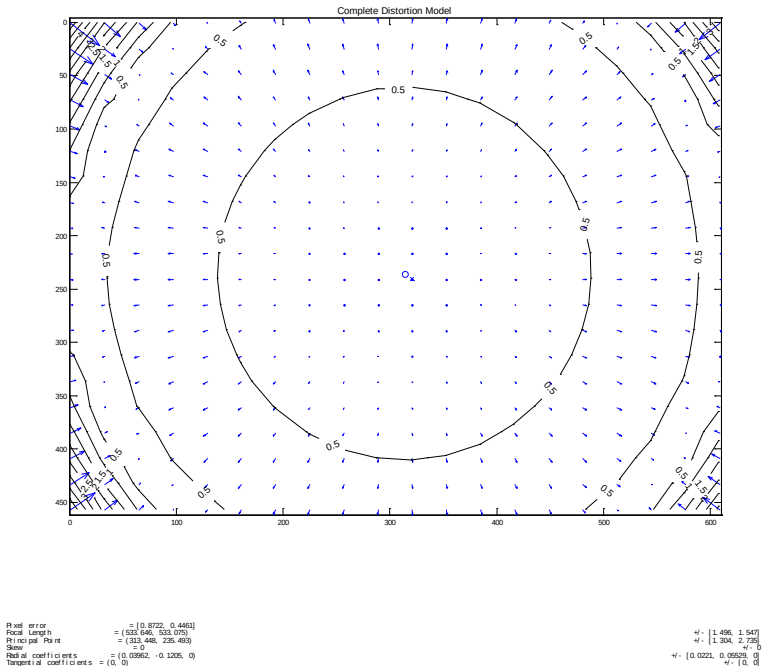


Figure 26: Distortion Model of camera

The next main task is to align the calibrated data to a real world coordinates. The calibration mentioned above will consider the 3D axis origin to be at the center of the 3D points used for calibration. The output of the calibration is a raw one and doesn't relate to any metric coordinates. To align this coordinate system to the real world coordinate system we use the concept of similarity transformation. Before the similarity transform is used, it is necessary to fix a origin in the 3D real world say a corner of the room and calculate a minimum of 6 points in metric coordinates. For simplicity, the camera locations were measured because the unaligned

camera locations are already obtained from the calibrated output. Now how do we relate these two coordinate systems? Consider matrices A and B with the following relation

$$B = P^{-1}AP \tag{5.8}$$

A and B are called similar matrices if it satisfies the above relation and P is called as similarity transformation. Similar matrices represent the same linear transformation under two different bases, with P being the change of basis matrix [56]. After we find the similarity transformation matrix we can apply it to the camera center matrix, Projection matrices, Rotation matrices and Data matrices to get everything in terms of a known coordinate system fixed by us. The great advantage of using a well aligned coordinate system is the measurements of objects in the real world gets simplified by a great deal and once similarity transformation is done, we have each point in the room with its accurate 3D coordinates. The 3D Matlab plots shown in the next page shows the difference between the un-aligned coordinate system and the aligned coordinate system. The final stage of the calibration is the verification of the calibration.

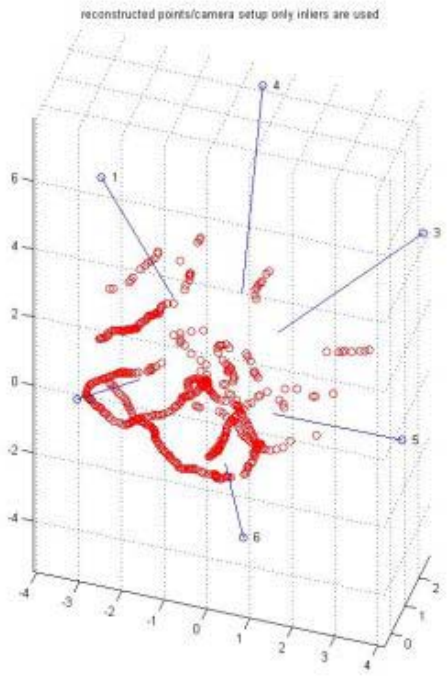


Figure 27: 3D Plot of unaligned calibrated environment

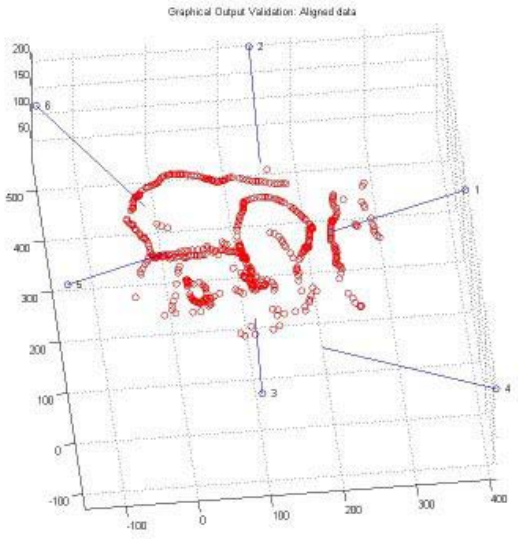


Figure 28: 3D plot of aligned calibrated environment

Summarization of the self-calibration process:

1. Find the projections of the laser pointer in the images [30].

2. Discard misdetected points by pair wise RANSAC analysis [55].

3. Estimate projective depths and fill the missing points by the method described in [44].

4. Optimize the projective structure by using the Bundle Adjustment [57], if applicable.

5. Perform the rank 4 factorization of the matrix Ws to get projective shape and motion [45].

6. Upgrade the projective structures to Euclidean ones by the method described in [30].

7. Detect the remaining outliers by evaluating the 2D reprojection error. Remove them and repeat steps 3–6 until no outlier remains.
8. Estimate the parameters of the non-linear distortion repeat the steps 2–7. Stop if the reprojection error is below the required threshold or if the number of iteration exceeds the maximum allowed.
9. Optionally, if some true 3D information is known, align the computed Euclidean structures with a world system.

5.4 Validation of existing calibration

The 3D reconstruction can be verified by performing certain tests. A common object such as table or an edge of an object visible in at-least two scenes is used and their 2D locations in image coordinates are obtained. The length of the object is measured using the measurement tape. For every 2D point correspondence, an equation $Ax_i = 0$ is formed. The first 2 rows of each P matrices are considered and x_i value is determined. The values of the unknown are determined and are converted from homogeneous to Cartesian coordinates. The distance between the points are calculated and is compared with the measured value.

6. HUMAN POINTING GESTURE DETECTION AND 3D TRACKING

6.1. Detection of Human Pointing gesture

The detection of pointing gesture is the first step in determining human pointing targets. Our research involves study of human-machine multi modal communication framework. When a human interacts with the robot he/she can use speech, gestures and haptics to communicate with the robot. These 3 modes can occur individually or together depending on the topic of communication. Detection of pointing gestures has been done using various computer vision methods in the past. These methods have no big disadvantages, except for the fact that it involves a lot of processing time, power and is computationally expensive during a real time scenario. The activities of the person must be constantly looked upon and a robust tracking method must be employed to track hands and heads continuously. With this information, it can be classified whether the actions of the person is a pointing gesture or not.

Instead of using vision methods, we propose a method which can detect pointing gestures in the videos using the speech information. Only few people [18] in the past have tried to use this approach even though the pointing gesture, almost everytime is accompanied with a highly useful speech input. Phrases like "this", "that", "go there", "come here" mostly occur with pointing gestures and if this information can be processed successfully then pointing gestures can be identified with very much less computation power. These sets of words or adverbs which refer to time and place are called deictics.

The key aspect in finding whether information from different modalities make a contribution for effective communication is by monitoring their temporal correlation. Speech and deictic gestures have shown to be highly correlated in time [11]. This is explained in the following bar chart.

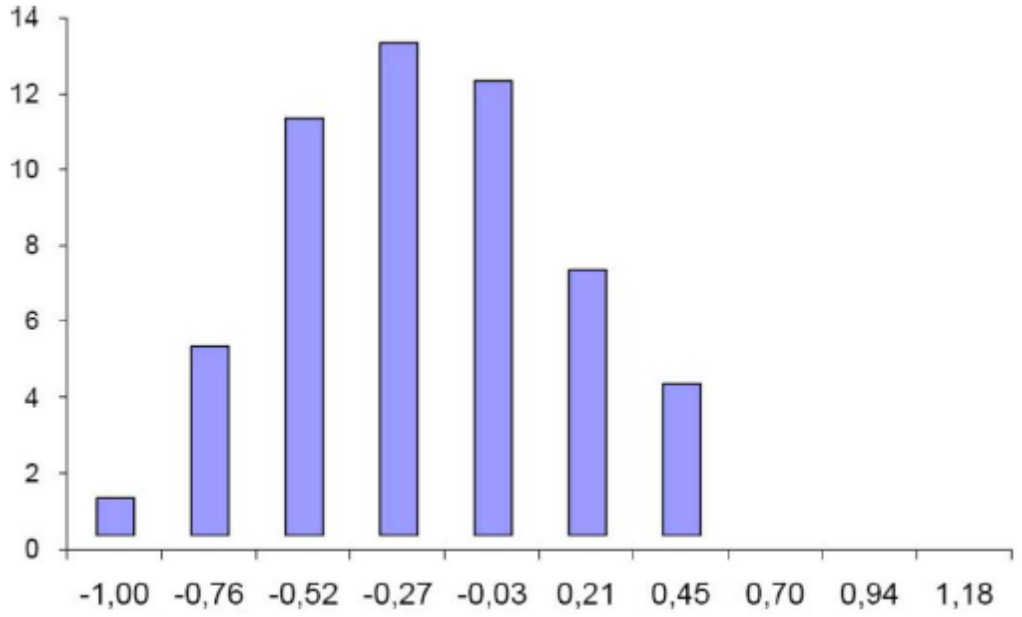


Figure 29: Time correlation between start of gesture and start of deictic words [11]

The values on x axis describe the difference in time between a deictic speech and a pointing gesture. The y axis depicts the value of number of gestures with that interval of starting points. The chart shows that the majority of starting points are between -0.52s to 0.45s. This means the speech and pointing occurs almost simultaneously because their difference in starting periods is less than a second. We have a speech recognition technique being developed in our lab based on

RISq which will be used for identifying deictic words. Currently, the frames of pointing gestures are manually labelled to estimate pointing locations robustly.

6.2. Real-time Human detection and tracking in Scene

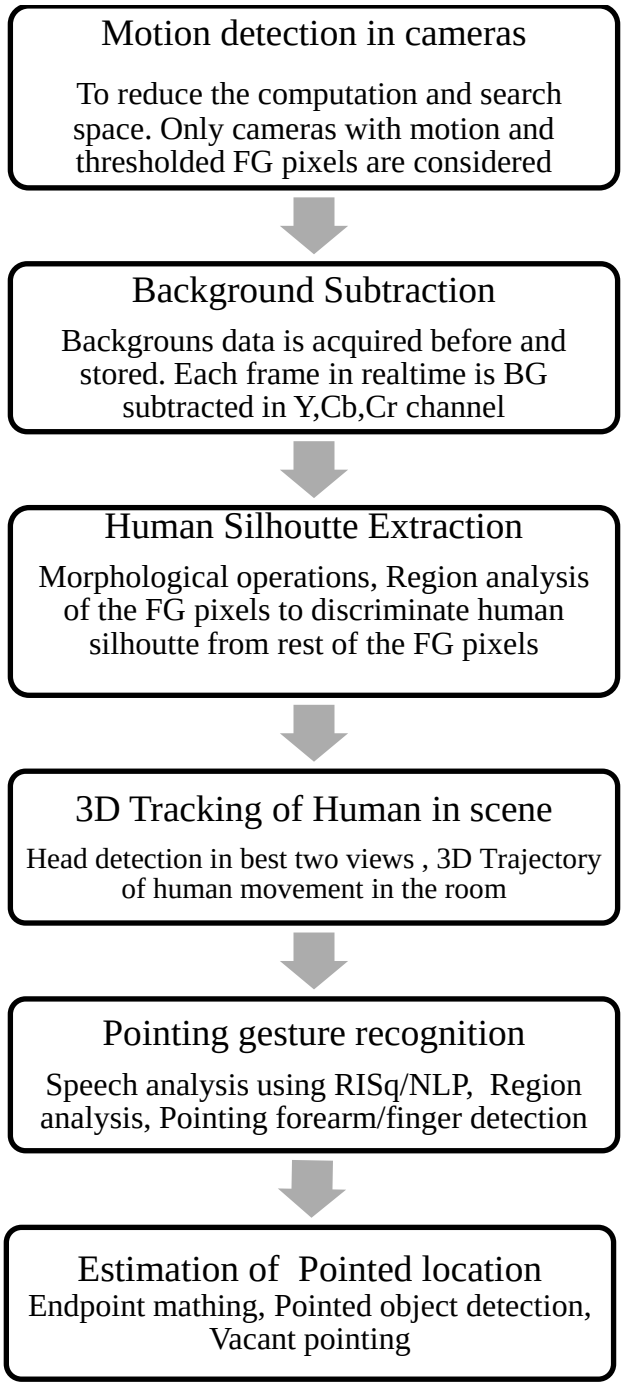


Figure 30: Flowchart of Pointing estimation and 3D Tracking

In our research a real time human detection technique is implemented to identify pointing gestures and to obtain the trajectory of the person in the room. This step is very important as identification of pointing locations is based upon this. This method is very robust and detects pointing regions effectively because of stable human detection in scene. The flowchart in Fig.29 explains our methodology in determining pointing regions in the image namely forearm, finger, glove etc.

6.2.1 Motion detection and background subtraction

In multi-camera setups the main issues will arise during real-time implementation. In our research, six cameras are used to acquire images of the environment. Motion detection is one of the easiest ways to reduce the processing time of the application. As mentioned earlier, two environments are used to study our approach. In the Machine Vision lab environment we have cameras on each corner facing the center of the room and most of the time there are movements in most of the cameras. In the Rush ADL room we have cameras fixed in certain places to monitor the activity in a limited space. For example, there is a camera which only monitors the space of the kitchen and another camera monitors a table alone. In such environments the activities are performed for only certain amount of time in the limited space. By implementing a motion detection algorithm we can discard the sections of videos where there is no activity performed. Our motion detection algorithm is based on frame differencing and uses a canny edge detector [58] and histogram of edge images. The absolute difference between the previous frame and current frame is calculated in three channels namely Y, Cb and Cr. These are added to get a difference image and edge detection is performed using a Canny edge detector. Then a histogram of this image is drawn to get the grey level values and their intensity bins. The maximum

intensity bin for each camera's background image is calculated and stored. If the number of histogram intensity bins is larger than the background threshold value then a motion is detected. This works method for varying illuminations and its highly adaptive cause the difference of images is done using two successive frames and not a static background. The histogram of background scene and a scene with movement is shown below.

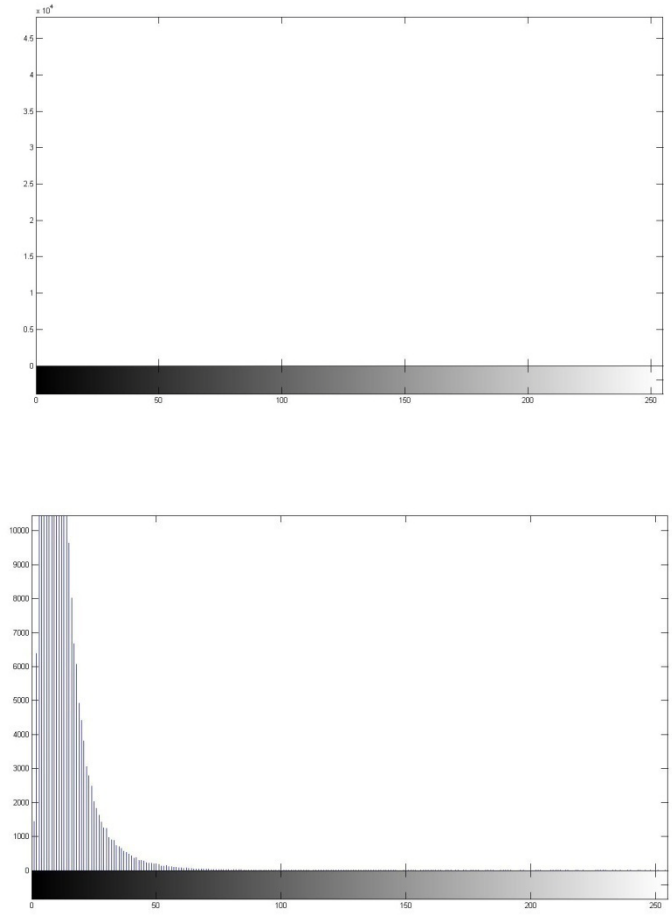


Figure 31: Histogram of edged scene with no movement (top) and movement (bottom)

Once the movement is confirmed in a camera then the background subtraction using the above frame difference method is done. But this time we are not performing the absolute difference on successive frames but with a static background image. The main reason is we need to detect the human even though there is no motion. So this would give all the new objects that have changed from the original background scene. Now the next task is to remove all these unwanted moved objects and to detect only the human in the scene. A background subtracted image and a new object (bounded by red box) with static background is given below

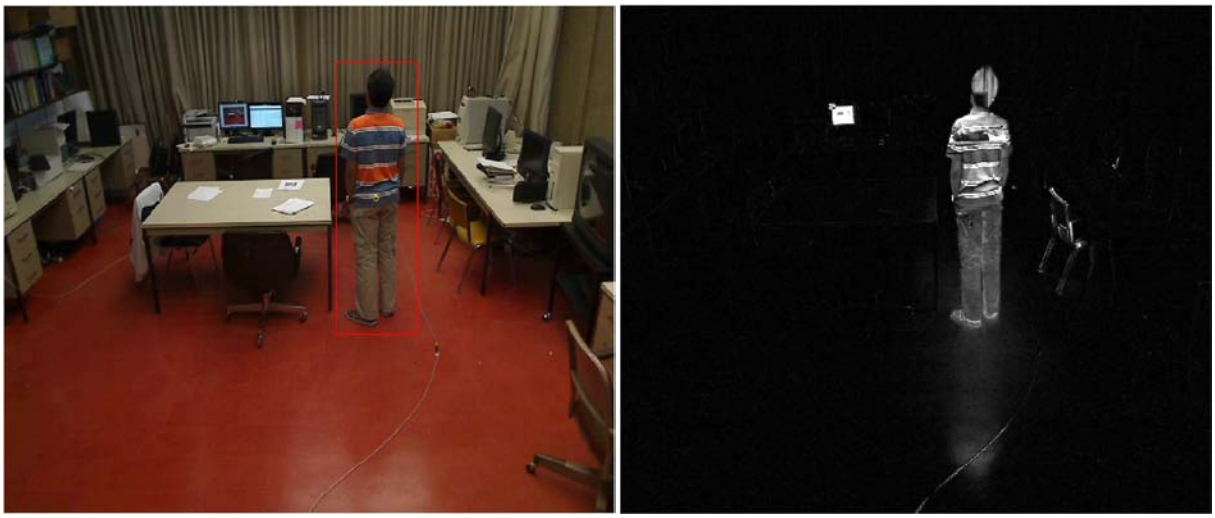


Figure 32: Original scene and Background subtracted image

6.2.2 Morphological Operations and Human Silhouette extraction

The previous stage is effective in discarding the background pixels of the scene. As shown in the above figure there is still unwanted information along with the human being detected in the image. These are caused due to small movements of objects in the room, reflection, shadows, etc. To discard those non-human pixels we use binary image analysis and concept of region properties. First the difference image is converted into a binary image by thresholding. A binary image is a logical image; it has 0 for background pixels and 1 for segmented foreground pixels. The next step is to apply the thickening operation on the binary image. Thickening is used to grow selected regions of foreground pixels and is closely related to dilation or closing. It works on the basis of hit-and-miss transform. The structuring element is passed over the image and if the background and foreground pixel in the structuring element matches, then pixel underneath origin of the structuring element is set to 1. We use this to close the open regions and it also performs the operation of thinning the background. The following images show how thickening works on a binary image.

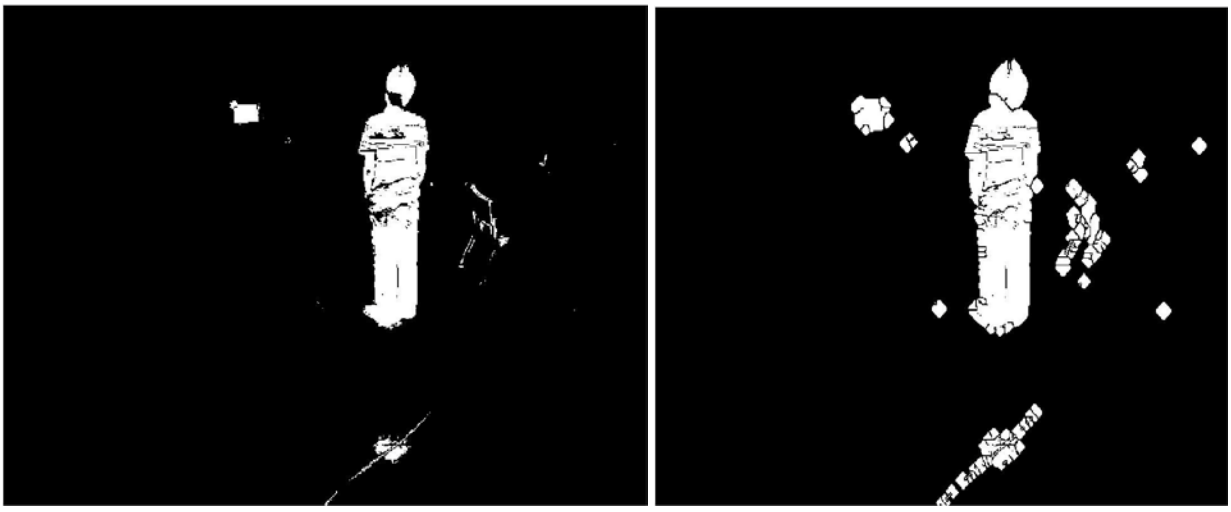


Figure 33: Thresholded binary image (left) and thickened binary image (right)

The final morphological operation performed is the close operator. Close operator is the combination of dilation followed by erosion. Closing tends to smooth sections of contours, fuses narrow breaks, eliminates small holes and fills gaps in the contour. Consider you have a set A and structuring element B then the closing operation [59] is defined as

$$A \circ B = (A \oplus B) \ominus B \tag{6.1}$$

which indicates that closing is simply dilation of A by B, followed by erosion of result by B. The Figure 33 gives the pictorial idea of the closing operation when compared with Figure 32. Now a clean set of closed regions in the image is obtained. The final task in the human silhouette

extraction will be to analyze these regions and to pick the region which belongs to human in the scene. This stage is simplified because we need to perform this operation on each frame. It is based on the theory that the largest region in the binary image is the region of human silhouette. It works perfectly because in normal ADL there is no bigger change to the foreground apart from the region covered by the human. Hence the region with maximum area is selected by calculating region properties. Region properties calculation includes the method of analyzing individual sets of pixels which are disconnected and finding the geometrical properties of those regions. The Area property is denoted by the following equation [5]

$$A = \sum_{(r,c)\in R} 1 \tag{6.2}$$

which means area is just the count of pixels in the region. The above theory is based on the assumption that only one human is present in the room. In real time application the silhouettes of both the human and assistive robot can be extracted if needed and the only change in the algorithm will be to find the second biggest region. The region extraction is shown below.

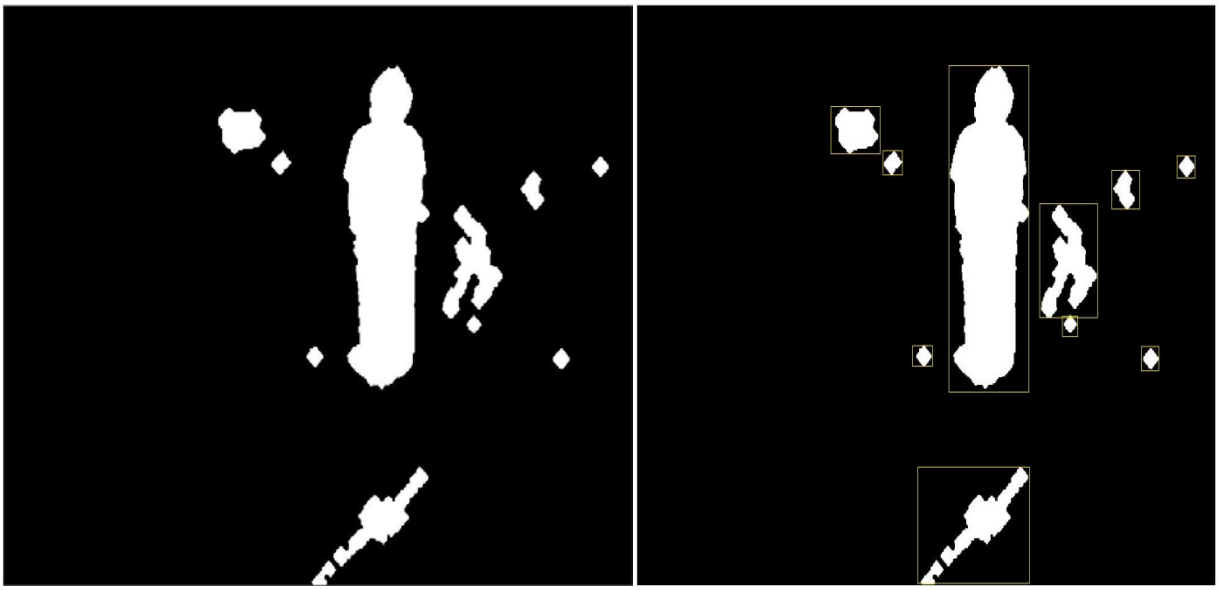


Figure 34: Closed Binary Image (left), Regions bounded by yellow box (Right)

Once the human silhouette is extracted from the background and other moved objects, we can always search for parts like human head, hand and finger inside this region.

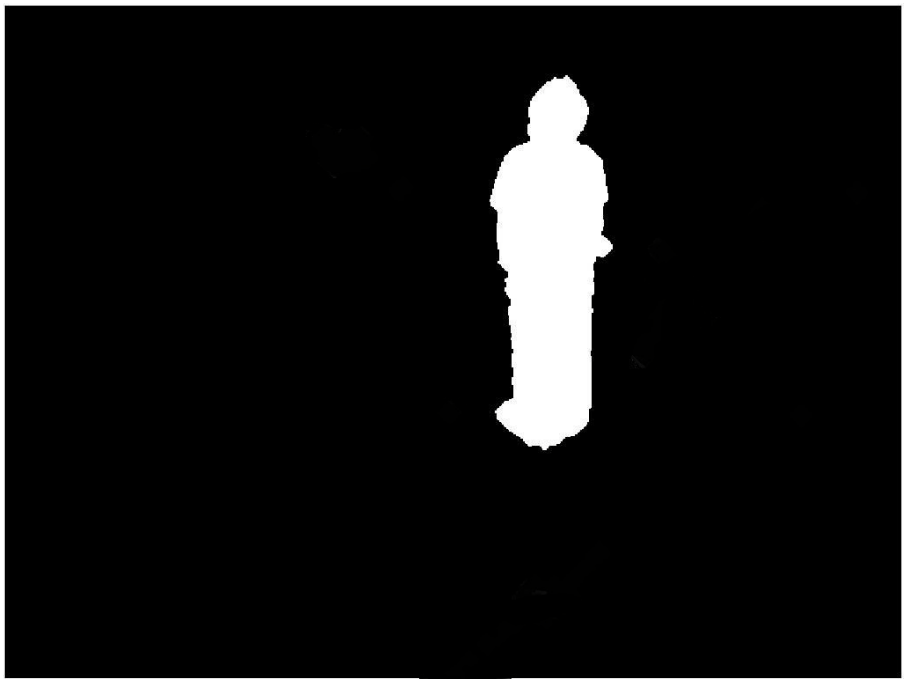


Figure 35: Extracted Human Silhouette

6.2.3 Head detection and 3D Tracking

Head is one of the important features in the human body. Almost all of the computer vision methods that deal with person detection have robust head detection in their algorithm. Head will also provide the information of a person's location in the image. In our study, we have developed a method to detect heads in the images of each camera irrespective of the orientation of head. The only limitation in our head detection algorithm is the requirement that it relates only to

upright poses like sitting, standing, walking, etc. The idea behind head detection is to determine the person's location and draw the trajectory of movement in the room in 3D coordinates. When the person is moving in the room or pointing, the head is always on the top of the body and we have an upright pose.

Our head detection algorithm is based on normalized 2-D cross correlation. The algorithm [60] used for normalized cross correlation is as follows:

- The cross-correlation in the spatial or the frequency domain is calculated, depending on size of images.
- The local sums are calculated by pre-computing running sums [61]
- The local sums are used to normalize the cross-correlation to get correlation coefficients.

  The Matlab Implementation is based on the following formula

$$\gamma(u,v) = \frac{\sum_{x,y}\left[f(x,y) - \bar{f}_{u,v}\right]\left[t(x-u,y-v) - \bar{t}\right]}{\left\{\sum_{x,y}\left[f(x,y) - \bar{f}_{u,v}\right]^2 \sum_{x,y}\left[t(x-u,y-v) - \bar{t}\right]^2\right\}^{0.5}}$$

where f is the image, $\bar{t}$ is the mean of the template and $\bar{f}_{u,v}$ is the mean of $f(x,y)$ in the region under the template

We have set of head templates which are binary images of ellipses of varying sizes. The minimum and maximum sizes of ellipse are determined by measuring the dimensions of the head in pixels when the user is closest and farthest from the camera in the room. The figure 35 shown in the next page gives the varying size of ellipses. This template is matched on the upper region (1/2 of the full silhouette) of the human silhouette. The template that matches with highest

correlation value is selected and an ellipse with the same parameters of the selected template is drawn to verify the detection. Our head detection works for any orientation of head as far as the person is in upright pose. Once the head is detected in two images, the 2D centroid points of the ellipse is recovered in both the images and determine 3D location of the person by solving the linear equations of the two projection matrices with 2D points as described in 5.4.

Figure 36: Head templates

Figure 37: Head detection in a single camera

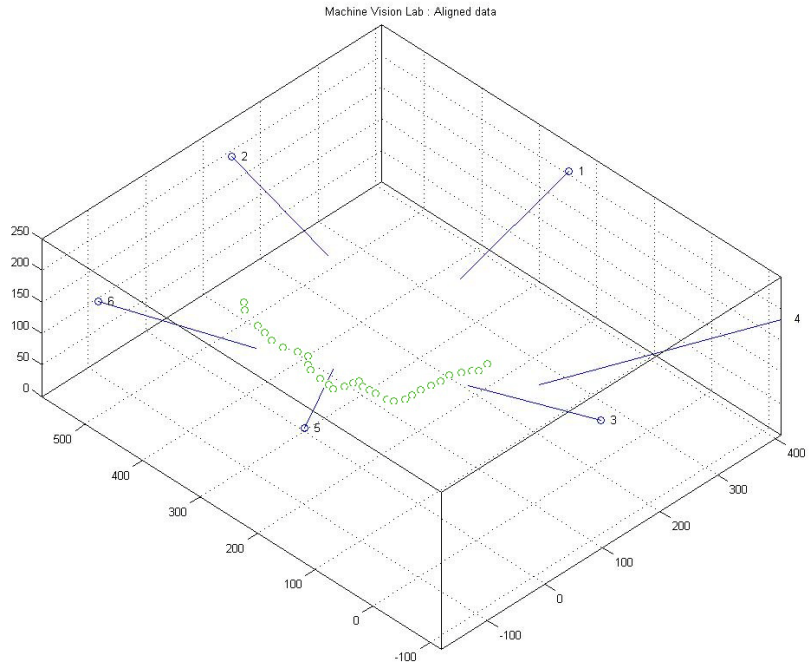


Figure 38: Trajectory of Human movement in the room

6.3. Region Analysis for detecting Pointing Region

The final step in successfully detecting a pointing gesture is to detect the pointing region in the image. A pointing region is defined as the region of the human body which is used to point the objects in the room. A pointing region can be anything starting from index finger, other fingers, multiple fingers, palm and fingers together, forearm, whole arm. Based on the study of

experiments with elderly patients, we are only considering forearm based pointing and finger based pointing. The elderly patients use these two types of pointing in most of the situations. When they use forearm based pointing, they point with a single finger or multiple fingers. Hence we propose two methods namely skin based detection and marker based detection to solve this problem of detecting pointing regions.

6.3.1. Skin based hand detection

In the previous sections we have successfully found the human silhouette in the image which reduces our search space for the pointing region by a great deal. However still the human silhouette has lot of pixels in its region and the next task is to find the pointing region inside the human silhouette. We propose an efficient method to detect the pointing region inside the extracted silhouette. The first task is to use a skin detection to filter out the non-skin pixels in the human silhouette. The skin detection algorithm comprises of search of a pixel in a particular color space which has a wide range of skin color. The color space that we use in our skin detection algorithm is combination of Hue and Cb, Cr values. The algorithm that is used here is a modified version of [62] because we don't use RGB information in our skin extraction process.

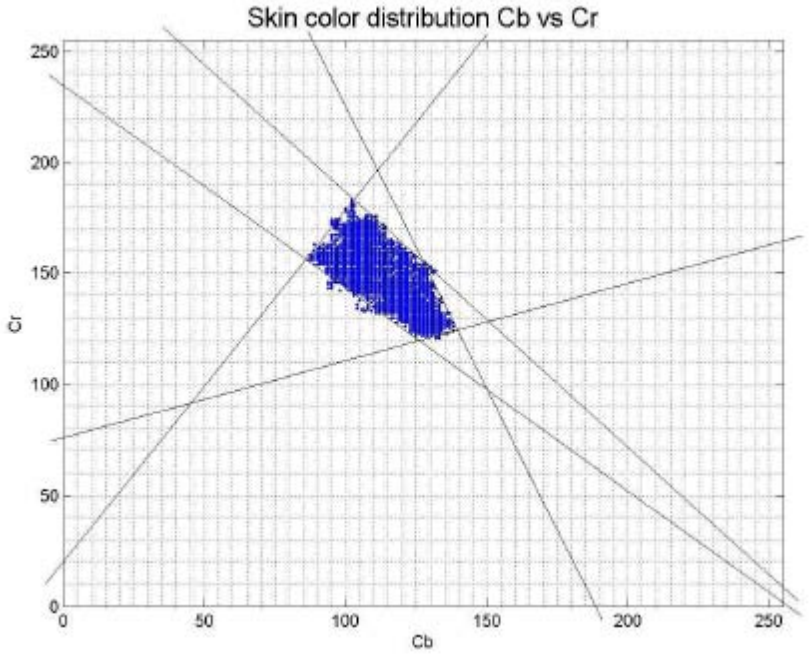


Figure 39: Skin color distribution in Cb and Cr channels

First the RGB image is converted to HSV color map and just the Hue information is extracted. A pixel is marked as skin if it has hue value between 0.01 and 0.1 with Cb,Cr values falling under the region showed in Figure 38. Once the skin pixels are detected it limits our search space to face, hands and legs assuming that the person's other body parts are covered by non-skin colored clothes. The next step is to use the region properties to detect the forearm in the image. We approximately know the shape of the forearm in the image; hence we can use the geometrical

constraints to filter non-pointing regions in the image. Each region that is marked positive for skin color is checked for the following geometrical conditions.

1. Size of the region is greater than 100 pixels

$$A = \sum_{(r,c)\in R} 1 > 100 \tag{6.4}$$

2. Eccentricity is greater than 0.75. Eccentricity gives the measure of a circle and straight line. If the region is a line then Eccentricity is 1 and is 0 for a perfect circle

$$E = \frac{\sqrt{\left(\frac{x}{2}\right)^2 - \left(\frac{y}{2}\right)^2}}{x} \tag{6.5}$$

where x and y are length of major axis and minor axis respectively

3. Absolute value of the Orientation of the region must be less than55°. Orientation is nothing but the angle between the x-axis of the image and major axis of the region [63].

Figure 40: Orientation of a Region with x-axis

4. Ratio of Major axis to Minor axis must lie between 11 and 2.
5. If more than one region satisfies the above conditions, then the region with maximum 2D Euclidean distance between its centroid and centroid of human silhouette gets the best vote to be selected as forearm region.

The above values are obtained by detecting hand regions in various poses and from different views of each camera and calculating their geometrical region values. As we can clearly see from condition 3 we are primarily considering side and semi-side views of the hand. Even though the hand regions can be detected if the user points towards the camera, a lot of hand pixels are lost and it doesn’t which doesn't provide enough information about the end points of hand regions. By considering only side and semi-side views accurate 3D reconstruction is obtained from just 2 views.

Figure 41: Forearm detection in side and semi-side views

6.3.2 Marker based finger detection

Our second method to detect a pointing region in the image is by using a color marker for pointing index finger. The index finger of the hand is covered by a colored cloth material and the human uses this finger to point at objects in the room. Again we search for the particular color marker region inside the human silhouette. The color detection algorithm is different from the skin detection algorithm. The color map of the marker is converted from RGB to HSV space. Then we take average values of Hue, Saturation and Value separately for each component by

selecting pixels in different regions of the marker. We store this color model and this is more like the training stage. During the testing stage we compare the Hue values of the current pixels in the image with the hue of the color model with a given threshold. If this condition is satisfied then we test for Saturation in a similar way. Only if these two conditions are satisfied then we perform the comparison in Value channel. Still there might be regions in the silhouette which can match the exact color of the marker. We filter these false positives again using the region properties. Most of the conditions remain the same as skin region detection except for the boundary values since size of the finger is much smaller than the hand. The conditions are:

1. Area of the region must be greater than 7 pixels
2. Eccentricity must be greater than 0.75
3. Absolute value of the orientation must be less than 60°
4. Ratio of Major axis to Minor axis must lie between 8 and 3.
5. If multiple regions satisfy the above condition the region with its farthest distance from the center of human silhouette gets the highest vote.

The region analysis and marker based finger detection results are shown in the Fig. 41 and 42 respectively.

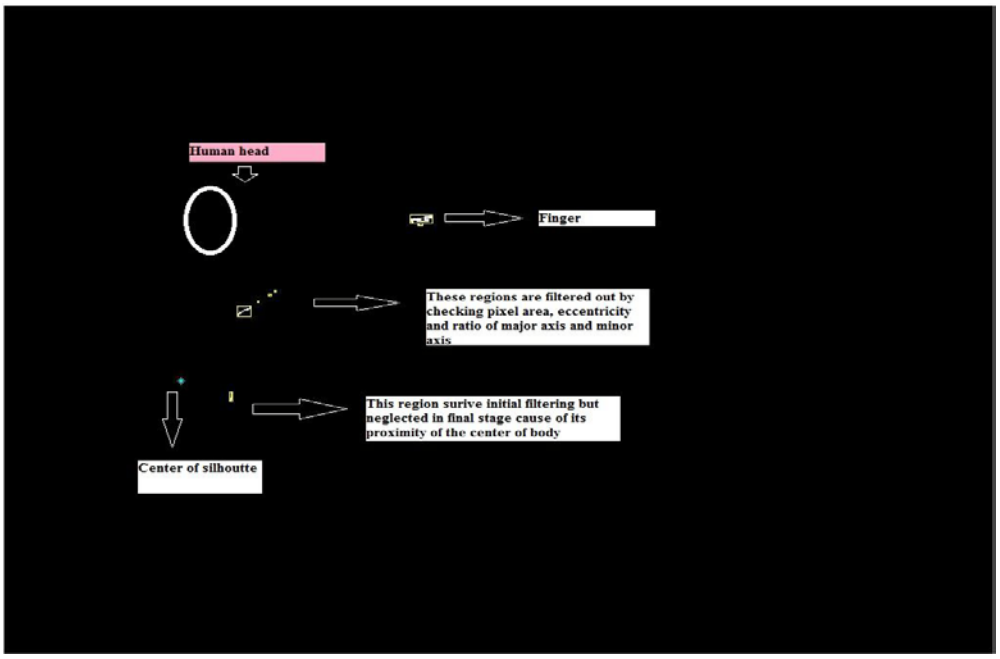


Figure 42: Region analysis to detect pointing finger

Figure 43: Marker based finger detection

7. POINITNG LOCALIZATION - METHODS & CHARECTERISTICS

7.1 Accurate Pointing Vs Approximate Pointing

In this chapter, the methods used for pointing localization is described i.e. methods to determine pointed locations during the pointing gesture. As we discussed in the earlier chapters pointing gestures vary. Human uses different type of pointing gestures during communication and pointing. We classify the pointing gestures into two types of pointing namely, accurate pointing and approximate pointing. Accurate pointing is done by humans when they need to precisely point to an exact location. It is performed by having your pointing arm, aligned with the eye and the object so that all three namely eye, finger-tip and the object are in a single straight line. This is more similar to techniques used by shooters and archers when they have aim for targets at long distances accurately. We don't use this type of pointing unless we have to aim exactly like a shooter or archer.

In ADL, humans mostly use approximate pointing to point objects in the room. During our study of elderly patients in ADL room we monitored the different types of pointing performed by the subjects. After analyzing the videos with speech to know where the user points, we found that they use an approximate pointing gesture. In approximate pointing, humans do not use the technique mentioned in accurate pointing. They look at the object using their eyes and point towards it. They make sure that their finger or forearm is pointing in the approximate direction and don't aim of the object as in the case of Accurate pointing. The following images demonstrate the difference between accurate and approximate pointing.

Figure 44: Accurate pointing (left) and Approximate pointing (Right)

7.2. Pointing target estimation by machines

In this section, the methods used for estimating pointed locations for an environment is described. In the previous section we described the types of pointing gestures being used by humans and how we estimate the pointed locations. Humans do not have difficulties in estimating pointed locations. When a computer system monitors the action of pointing through a camera which is away from the human head or eye line it is difficult to find the object or location being pointed. In the past, most of the research in pointing localization has been done using Eye-

Finger tip approach or forearm approach. In our study we examine these two along with the finger based pointing approach.

7.2.1. Eye-Fingertip line approach

The Eye-Fingertip approach is being used for accurate pointing gestures. We implemented the Eye-Fingertip line approach in our study with manual labeling of points in the 2D images. From the results it is observed that it worked very well with the accurate pointing gestures but failed in instances when approximate pointing gesture was used. The error between the pointed location and estimated location was huge. Thus, in general when pointing target estimation is done by machines using this approach it is assumed that the human points with an accurate pointing gesture. In Fig. 44 the elderly actually points towards the refrigerator which is on right end of this image while Eye-Fingertip line points to the floor.

Figure 45: Misdetection of pointing location by Eye-Fingertip approach

7.2.2. Finger and Forearm approach

In the finger and forearm pointing approaches we can estimate the pointed location by drawing a line between the two endpoints of the pointing region and extending it in 3D space to find the object being pointed. In our study we use these approaches to detect the pointed object. The difficult task in using this approach is to find stable endpoints and adjust the error in elevation and azimuth. We have developed a method to find stable endpoints of the pointing region and

also nullify the error caused in elevation. First let us see about stable endpoint detections. The previous section explained about detecting the region of pointing in the image. In this region our first task is to find the extreme points. According to [5] a region can have a maximum of extreme points irrespective of its shape. If the shape of the object is well defined like a rectangle or triangle then these extreme points overlap. Given below the 8 extreme points of a region and remember each extreme point lies on the bounding box of the region.

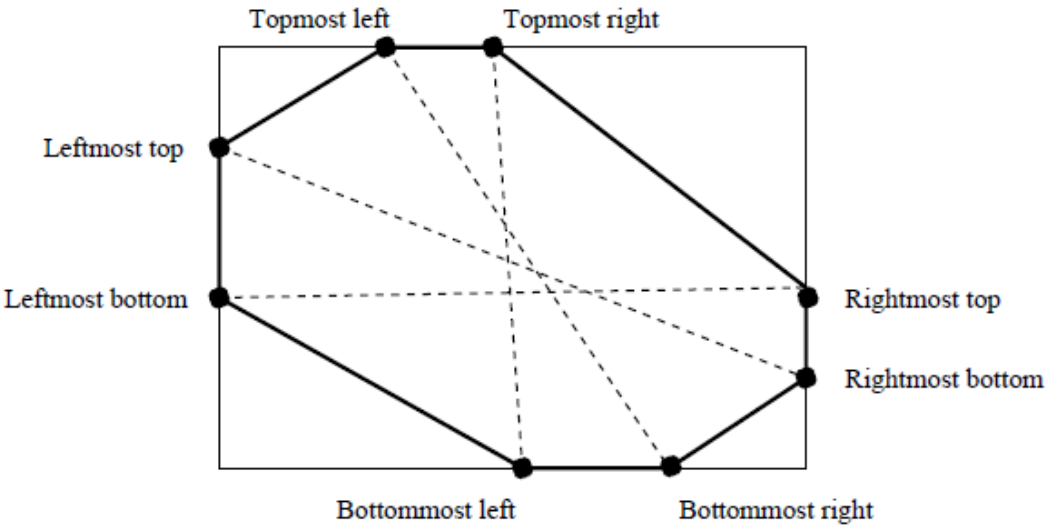


Figure 46: Extreme points of a region

The region properties function in MATLAB gives us the extreme points in the region and the method used is available here [64]. Since we consider only side views and semi-side views of the pointing region we look out for the left and right corner of the region. We find the topmost points in these corner regions to get the perfect endpoints. The algorithm is as follows:

1. The left-top extreme point in the region is found and its x and y coordinate are obtained.
2. The right-top extreme point in the region is found and its x and y coordinates are obtained.
3. If these two points are not located on top part of the finger due to bad color detection, then these points are shifted to the top part of the finger. This is done by changing the y coordinate of the endpoints to the y coordinate of the top most pixel of finger silhouette.
4. The Euclidean distance of these two points from the center of human silhouette is calculated. The farthest point will be the finger-tip.

For hand, the steps can be refined to give a more a stable algorithm since sometimes there are partial skin regions after the elbow towards the shoulder. Hence the algorithm of forearm approach is designed as follows:

1. The left-top extreme point in the region is found and its x and y coordinate are obtained
2. The right-top extreme point in the region is found and its x and y coordinates are obtained.
3. The x and y coordinate of the centroid in the hand region is calculated.

4. The Euclidean distance of these three points from the center of human silhouette is calculated. Only the two farthest points are considered. One will be a finger tip and other will be the centroid of the region
5. If these two points are not located on top part of the hand due to bad skin detection, then these points are shifted to the top part of the hand. This is done by changing the y coordinate of the endpoints to the y coordinate of top most pixel of forearm silhouette.

This way we will have both points in the forearm even if there are skin regions behind the elbow. Endpoint detections in finger and forearm are given below. The red marker indicates the tip of the finger or forearm and blue marker indicates the other end of the region.

Figure 47: Finger endpoint detection

Figure 48: Forearm endpoint detection

7.3 Pointing Localization and Pointed Object detection

The above finger and forearm detection algorithms have to be repeated for each camera to detect the pointing regions in each camera. The finger and forearm may not be detected clearly in all the views because of bad illuminations and occlusions. We also remove the views where the person is pointing towards the camera. In our tests the finger and forearm is detected in two cameras which enabled us to perform 3D reconstruction. If the finger or forearm is detected in

more than two images we choose the two best images by selecting the cameras where the finger or forearm region has highest eccentricity and maximum number of pixels. Now the 3D coordinates of these two end points are calculated by solving set of linear equations as mentioned in section 5.4. A line joining these two points is drawn.

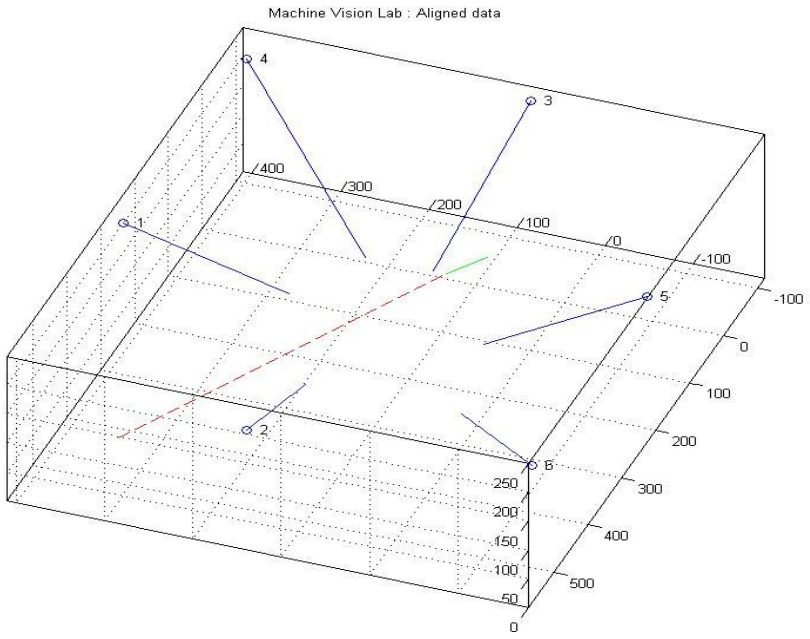


Figure 49: Line of Pointing

From the 3D point of the finger tip, we extend the line in that direction while maintaining the same slope of the forearm/finger line to get the pointed location. Figure 48 explains the line of pointing. The Green line is the pointing line formed by joining the end points of forearm and the red dotted line is the extension of the green line to get the LOP.

The above steps determine the human pointing direction. The next task is to find the object located in the direction of pointing. We provide an efficient methodology to find the pointed object and its distance from the person. We construct a database of all the static objects in the room with its 3D locations by using our 2D to 3D mapping. Each object is bounded by a rectangular 2D plane. The four corner 3D locations and its centroid information are stored in the 3D Object database with its name.

Objects		Ltop			Rtop			Lbottom
Printer 1	-101.79	43.3246	164.205	-101.79	4.98713	164.205	-101.79	43.3246
Box 1	-98.581	44.6152	146.986	-101.79	6.27777	146.986	-98.581	44.6152
White cabinet 1	-93.677	43.5636	130.61	-93.677	5.22618	130.61	-87.256	42.5287
Camera 6	-143.04	97.3041	225.077	-145.69	75.5879	225.077	-147.36	96.6285
Camera 4	-95.292	412.082	206.576	-152.86	402.127	208.505	-104.51	417.774
Brown Box 2	-102.12	308.634	94.4954	-146.87	309.794	95.383	-88.946	299.838
Motherboard Box	2.19983	351.821	84.4086	-34.959	319.045	83.0905	9.83002	343.697
WS Speaker	18.1831	371.081	100.629	3.26901	355.989	103.64	14.5352	374.312
PC Chair 1	73.5589	296.165	87.577	23.1319	285.481	89.9149	73.5589	296.165
WS Monitor	90.1602	357.711	127.293	42.1235	354.811	128.62	88.2555	357.895

Table 1: Object database

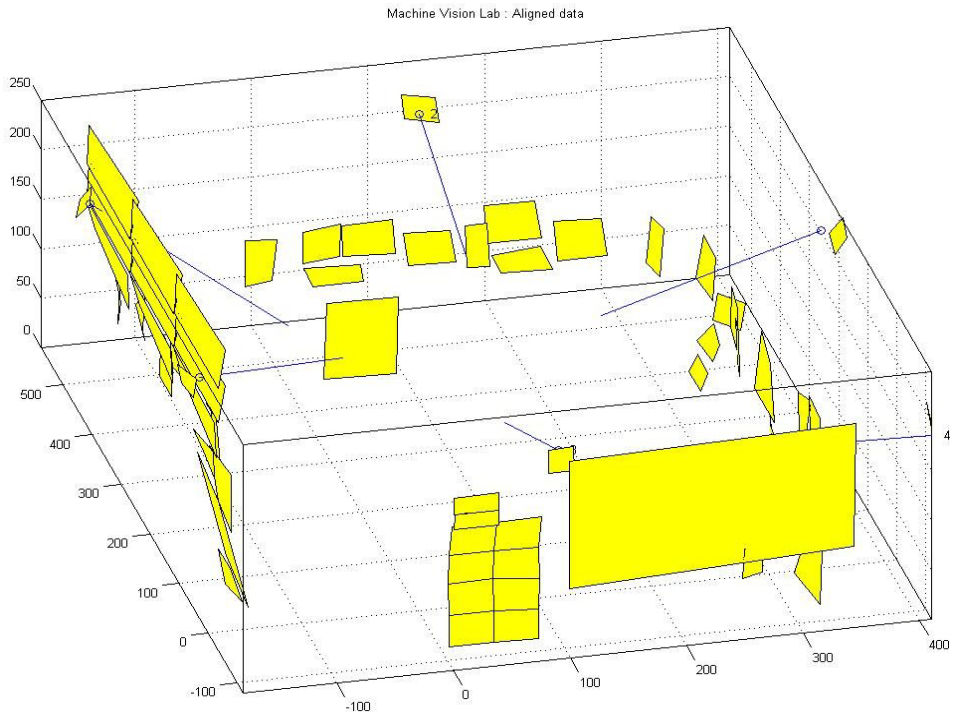


Figure 50: Object locations bounded by a rectangle

Figure 49 shows the bounding boxes of all the 65 objects in the Machine Vision Lab. If there are any new static objects in the environment the object information can be easily appended to the existing database. When the person points, we verify whether the LOP intersects with any of this 2D plane. The object name, 3D location and its distance from the user is provided to the assistive robot. We also let the user know about the pointed object by audio and images. The object is bounded by a Red box and displayed to the user along with object name spoken to the user. Figure 50 shows the objected detected with its bounding box in 3D coordinates. The next task is

to solve the problem of vacant pointing (Pointing to a location with no objects. e.g. Wall, Floor, etc)

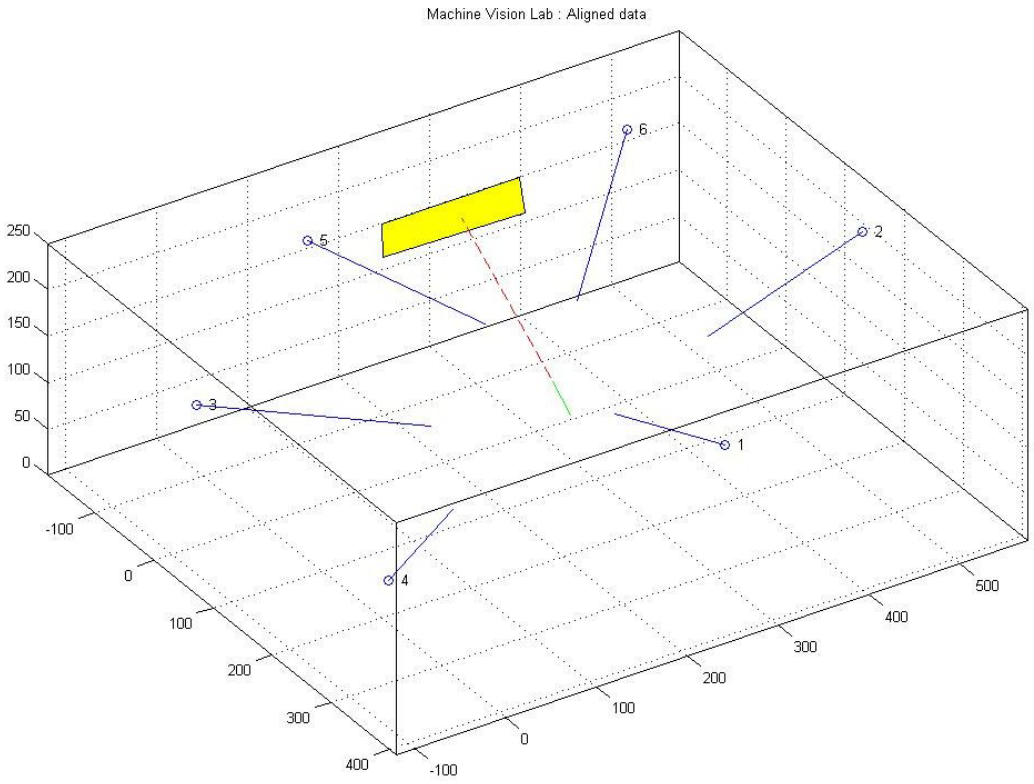


Figure 51: Pointed Object found.

When the user doesn't point to any object, we extend the LOP till it intersects the wall or floor planes which is the end of the room. Since the LOP doesn't intersect with any of the rectangular planes, the robot and the human are provided with the information of 3 closest objects near the LOP. The Robot can then interact with the person to verify whether the user pointed towards

any of these 3 objects. Figure 51 shown in the following page describes a scene where the user is pointing to vacant location in the wall.

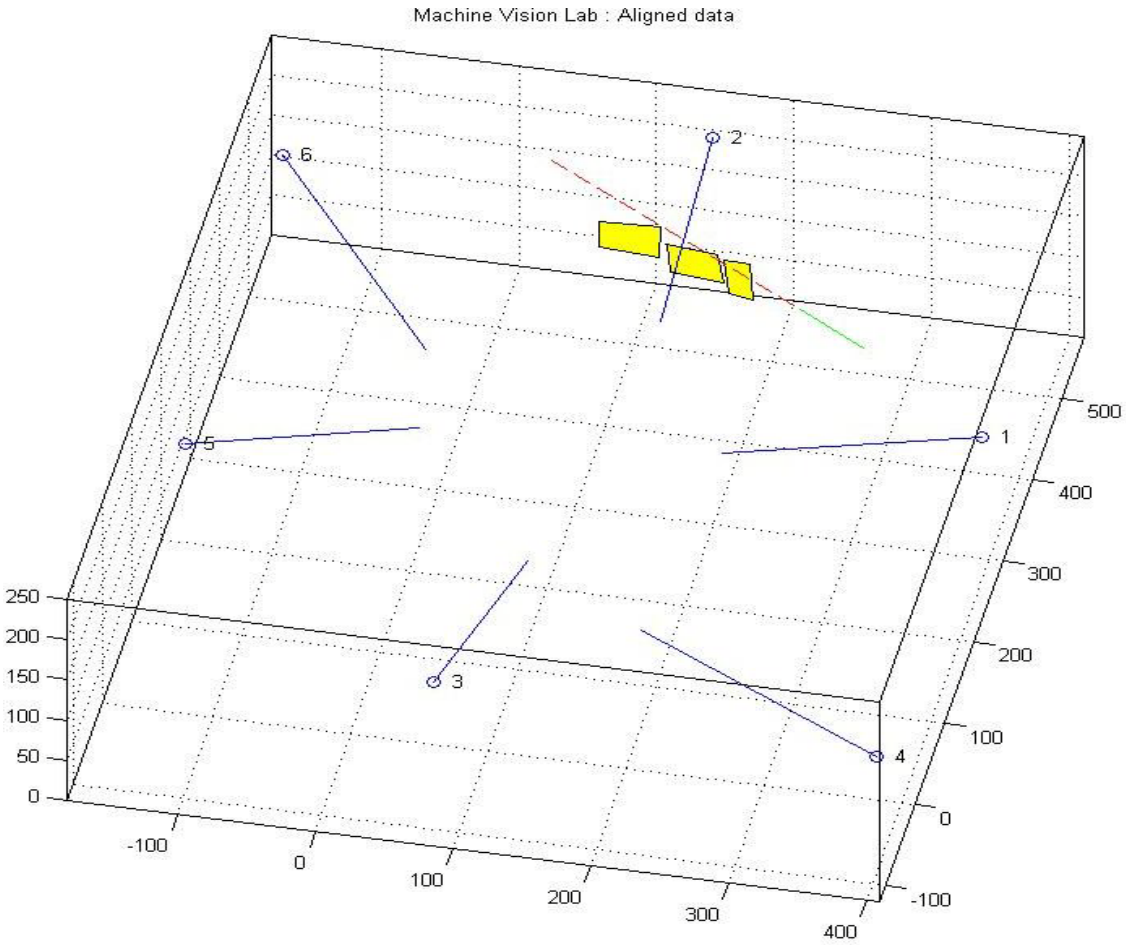


Figure 52: Vacant pointing

7.4. Results and Statistics

The performance of finger and forearm pointing localization were tested for the objects in the MVL Lab. The pointing was performed on a wide range of objects in the room which included objects in different planes. The number of pointing localizations that were considered for the comparison was 10 and only perfect forearm and finger detections were considered for the comparison. This is because inaccurate false finger or forearm detection will cause larger errors in pointing localization and hence the methods cannot be compared in those cases. For the comparison of finger and forearm pointing we consider 2D images of pointing from two cameras to obtain the 3D values of LOP and the user points to 10 pre-defined object corners.

First we compare the detection rates of finger and forearm in two images while performing the pointing gestures using these two methods. The detection rate is important because the 3D reconstruction can be performed only when images from at least two cameras are available.

$$Detection\ Rate = \frac{No.of\ Pointing\ gestures\ detected}{No.of\ pointing\ gestures\ performed}\ X\ 100$$

The detection rate of forearm pointing is about 91% and finger pointing is 32%. This clearly shows the forearm detection is much higher than finger detection in the images. The following plot shows the difference in errors in azimuth and elevation angles between the correct LOP and the estimated LOP.

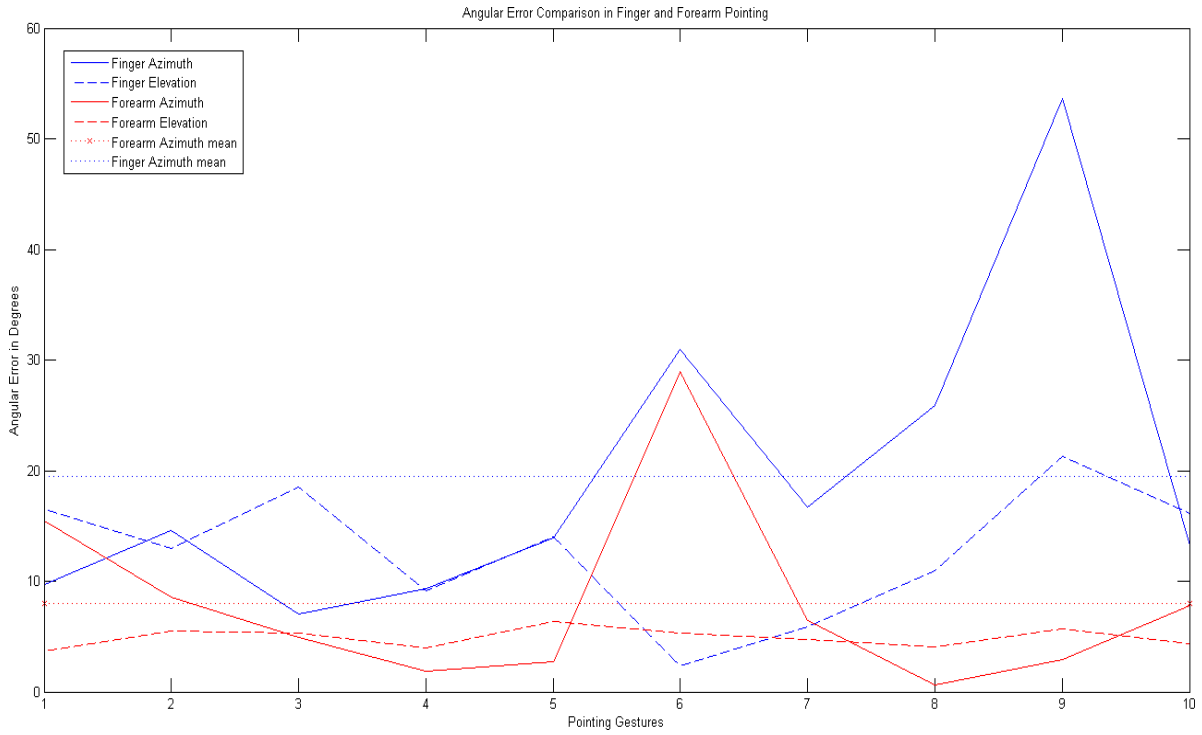


Figure 53: Comparison of Errors in Finger and Forearm Pointing

The plot in Fig. 52 describes the characteristics of the angular errors in both types of pointing gestures. The mean errors in azimuth and elevation are higher in finger pointing than forearm pointing. This proves the forearm pointing works much better than finger pointing in pointing localizations. The error in elevation angle of the forearm pointing gesture is almost constant. It has a mean of 4.885º and standard deviation is 0.8672 º. Hence this error can be minimized by adjusting the estimated LOP by the subtracting (error is in positive z axis) above mean value. Now to minimize the error in azimuth for forearm pointing the same pointing gestures were

considered but instead of two images we reconstruct LOP using three 2D images. We plot the results and compare their performance.

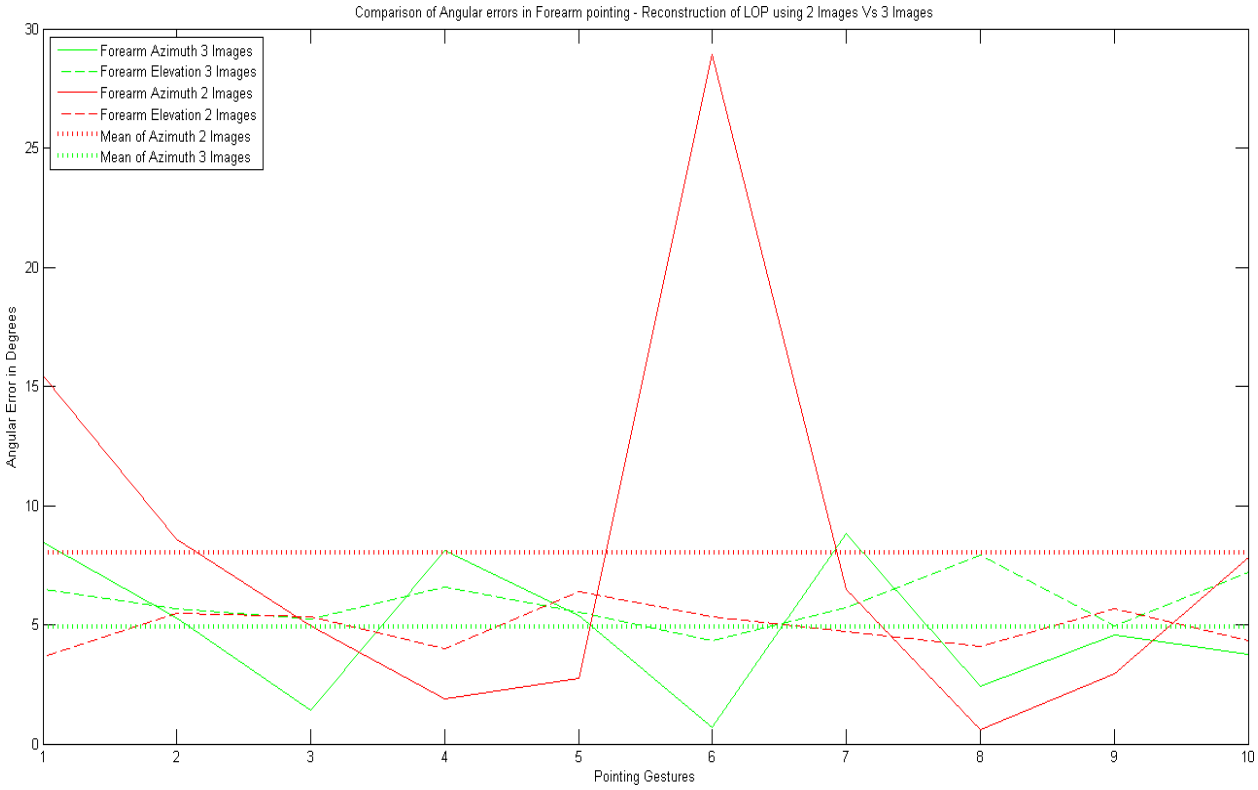


Figure 54: Comparison of error in azimuth angles in reconstructing LOP using 2 Vs 3 Images

The following table describes the mean and standard deviations of angular errors in reconstruction of LOP using finger detection in 2 images, forearm detection in 2 images and forearm detection in 3 images.

Pointing vs. Angular errors	Mean		Standard Deviation	
	Azimuth θº	Elevation φº	Azimuth θº	Elevation φº
Finger Pointing	19.4980	12.7720	14.1094	5.8165
Forearm Pointing – 2 Images	8.0373	4.8850	8.4933	0.8672
Forearm Pointing – 3 Images	4.8915	5.9539	2.9127	1.0825

Table 2: Angular error in Finger and Forearm Pointing

Finally after the testing of pointing localization by means of finger and forearm pointing we conclude that the forearm pointing estimation works better and has the following advantages over the finger pointing estimation.

- The person need not wear any kind of marker or glove because it's purely based on skin detection and makes the human more comfortable
- The person can use any type of pointing gesture and hand. Forearm method works with pointing with multiple fingers, objects in hand, arms stretched, elbow bent, etc.
- The error in pointing localization is very low when compared to finger approach. This is because we get a longer LOP with endpoints of the hand than we get in a finger approach.

The length of the line connecting end points in a finger is very small and therefore even a small error in the endpoints causes a large pointing localization error.

8. CONCLUSION AND FUTURE WORK

This thesis provides a complete methodology to detect pointing gestures and provides detailed information about pointing localization and 3D trajectory of a person in a room to the assistive robot. The multi modal approach proposed in this study is comparable to the state of the art methods used in Human-Computer Interaction [65].

8.1. Overall contribution in the project

- Self-Calibration of the ADL Environment:

Our main contribution to the overall project is in terms of setting up an efficient environment to monitor and record multi-modal communications during experiments. We successfully implemented the self-calibration in two different environments (ADL room and MVL). This enables us to test various scenarios and perform experiments with enough information in 3D world coordinates. It helps in the study of gestures, movements and activities of the elderly person to build a robust communication framework between the human and the assistive robot.

- 3D Tracking and Pointing Localization

In the second part of the thesis we make use of the calibrated environment to track the location of persons in the room. We studied the various pointing gestures of the elderly from experiment records obtained in ADL room. We propose a multi modal method to detect pointing gestures from video combined with speech and develop a method to detect pointing locations precisely. We also implement a method to store object locations in bounding boxes in a database. We use this database in conjunction with LOP's to inform patients of the exact object being pointed. We

also compared the pointing errors of forearm pointing vs. finger pointing approach. This investigation could help the research community in the study of pointing gestures.

8.2 Future Work

We mention few suggestions for future works which can be founded upon our method.

- Now, we use single patient tracking in the room against a static background scene. Currently, we are not updating the background model and it has limitations if a new person enters the scene. We suggest employing a tracking mechanism based on image features using probabilistic method like Kalman filter or Particle filter. This would track the elderly person continuously even if a new person enters the scene and occludes our patient's movements in the room

- We propose a multi modal method to detect pointing gestures using speech and video. For this thesis we manually label the frames of pointing in the video based upon some key words in the speech. We suggest combining our method with speech recognition to make it a completely automated pointing location finder.

- We have reviewed various research papers in the field of pointing to enhance the pointing localization. It was found that head orientation contributes hugely to the detection of pointed objects [15] [11] [65] because the pointed objects are always in the pointers field of view. We currently have a method being developed in our Machine Vision Lab (MVL)

which is based on RISq. RISq successfully identifies various head orientations in images. We would suggest combining the method discussed in this thesis with speech and head orientation which will improve results in the area of pointing gestures and pointed object localization.

REFERENCES

1. Barbara Di Eugenio, Jezekiel Ben-Arie, Milos Zefran, Marquis D. Foreman., *HCC: RI: Medium: Collaborative Research: Effective Communication with Robotic Assistants for the Elderly: Integrating Speech, Vision and Haptic,* 2009.

2. McNeill, D., *Hand and mind: What gestures reveal about thought, Chicago, The University of Chicago Press,* 1992.

3. Goodwin, C., *Pointing as situated practice, In: Kita, S., editors, Pointing: Where language, culture and cognition meet, Mahwah, NJ, Lawrence Erlbaum Associates,* 2003. p. 217-246.

4. Tomas Svoboda, Hanspeter Hug, and Luc Van Gool. *ViRoom -- low cost synchronized multicamera system and its self-calibration.* In Pattern Recognition, 24th *DAGM* Symposium, number 2449 in LNCS, Springer, September 2002. pages 515-522.

5. Shapiro, L., and Stockman, G. (2001). *Computer Vision*. Prentice Hall.

6. Jezekiel Ben-Arie, *The Perspective Projection,* In Image analysis and Computer Vision II -- Courese Notes, University of Illinois at Chicago, 2009: p.14-28

7. Yu Yamamoto, Ikushi Yoda, Katsuhiko Sakaue, *Arm-Pointing Gesture Interface Using Surrounded Stereo Cameras System*, 17th International Conference on Pattern Recognition (ICPR'04) - Volume 4, 2004: pp.965-970

8. Hiroki Watanabe, Hitoshi Hongo, Mamoru Yasumoto, Kazuhiko Yamamoto: *Detection and Estimation of Omni-Directional Pointing Gestures Using Multiple Cameras,* MVA 2000: p.345-348.

9. N. Jojic, B. Brumitt, B. Meyers, S. Harris, and, T. Huang, *Detection and estimation of pointing gestures in dense disparity maps* in, 4th IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France, March 2000: p.468-475.

10. N. Gourier, J. Maisonnasse, D. Hall and J. Crowley, *Head Pose Estimation on Low Resolution Images,* Multimodal Technologies for Perception of Humans: Proc. First Int'l Workshop Classification of Events, Activities and Relationships, , 2007: pp. 270-280

11. H. Holzapfel , K. Nickel and R. Stiefelhagen *Implementation and evaluation of a constraint based multimodal fusion system for speech and 3D pointing gestures*, Int. Conf. Multimodal Interfaces, 2004.

REFERENCES (Continued)

12. Gunther Heidemann, Holger Bekel, Ingo Bax, Axel Saalbach, *Hand Gesture Recognition: Self-Organising Maps as a Graphical User Interface for the Partitioning of Large Training Data Sets*, 17th International Conference on Pattern Recognition (ICPR'04) - Volume 4, 2004: pp.487-490.

13. R. Gherbi, A. Braffort: *Interpretation of pointing gestures: The PoG system*. In: Gesture-Based Communication in Human-Computer Interaction, LNAI 1739, Eds., Springer Pub., 1999.

14. S. Carbini, J.E. Viallet and O. Bernier, *Pointing gesture visual recognition for large display,* Pointing'04 International Conference on Pattern Recognition Workshop Cambridge, UK, 2004.

15. Stiefelhagen, R., Fugen, C., Gieselmann, R., Holzapfel, H., Nickel, K., Waibel, A., *Natural human-robot interaction using speech, head pose and gestures,* Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on , vol.3, 28 Sept.-2 Oct. 2004. pp. 2422- 2427

16. K. Nickel, R. Stiefelhagen, *Recognition of 3D-Pointing Gestures for Human-Robot-Interaction,* Proceedings of Humanoids 2003, Karlsruhe, Germany, 2003.

17. Ching-Yu Chien, Chung-Lin Huang, Chih-Ming Fu, A *Vision-Based Real-Time Pointing Arm Gesture Tracking and Recognition System,* Multimedia and Expo, 2007 IEEE International Conference on , 2-5 July 2007: pp.983-986

18. Stiefelhagen, R., Ekenel, H.K., Fugen, C., et al, *Enabling Multimodal Human–Robot Interaction for the Karlsruhe Humanoid Robot,* Robotics, IEEE Transactions on , vol.23, no.5, Oct. 2007: pp.840-851.

19. Guan, Ye-peng, *Robust video foreground segmentation and face recognition,* Journal of Shanghai University, Springer Pub, Aug 2009: p311-315.

20. Sakurai, S., Sato, E., Yamaguchi, T., *Recognizing pointing behavior using image processing for human-robot interaction,* Advanced intelligent mechatronics, *2007* IEEE/ASME international conference on , Sept. 2007: pp.1-6.

21. Hyung-O Kim, Soohwan Kim, Park, S.-K., *Pointing gesture-based unknown object extraction for learning objects with robot,* Control, Automation and Systems, 2008. ICCAS 2008. International Conference on , Oct. 2008: pp.2156-2161

REFERENCES (Continued)

22. Yepeng Guan, Mingen Zheng, *Real-time 3D pointing gesture recognition for natural HCI,* Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on , 25-27 June 2008: pp.2433-2436.

23. Axenbeck, T., Bennewitz, M., Behnke, S., Burgard, W., *Recognizing complex, parameterized gestures from monocular image sequences,* Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on , 1-3 Dec. 2008: pp.687-692.

24. Y.P. Guan, *Wavelet multi-scale transform based foreground segmentation and shadow elimination*, The Open Signal Processing Journal, 2008, 1(6): p1-6.

25. Ben-Arie, J. 2008. *Method for recognition of human motion, vector sequences and speech*. US patent 7,366,645 B2.

26. J. Ben-Arie, P. Pandit and S. Rajaram, "*View-Based Human Activity Recognition by Indexing & Sequencing*," appeared in IEEE Computer Society 2001 Conference on Computer Vision and Pattern Recognition, (CVPR'01), Kauai Marriott, HI, Vol. 2, December 2001: pp. 78 – 83.

27. Barbara Di Eugenio, Milos Zefran, Jezekiel Ben-Arie, Mark Foreman, Lin Chen, Simone Franzini, Shankaranand Jagadeesan, Maria Javaid and Kai Ma. *Towards effective communication with robotic assistants for the elderly: Integrating speech, vision and haptics*. Dialog with Robots, AAAI 2010 Fall Symposium, Arlington, VA, USA. November 2010.

28. Lin Chen, Anruo Wang and Barbara Di Eugenio. *Improving Pronominal and Deictic Co-Reference Resolution with Multi-Modal Features*. In SIGDIAL 2011, The 12th SIGDIAL Meeting on Discourse and Dialogue, Portland, Oregon, USA. June 2011.

29. Remondino, F. and Fraser, C., *Digital camera calibration methods: considerations and comparisons*. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, **36**(5), 2006: p266–272.

30. Tomás Svoboda , Daniel Martinec , Tomás Pajdla, *A convenient multicamera self-calibration for virtual environments*, Presence: Teleoperators and Virtual Environments, v.14 n.4, August 2005: p.407-422.

REFERENCES (Continued)

31. A.W. Senior, G. Potamianos, S. Chu, Z. Zhang, and A. Hampapur, *A comparison of multicamera person-tracking algorithms*, Proc. IEEE Int. Works. Visual Surveillance, May 2006.

32. Thomas B. Moeslund, Adrian Hilton, Volker Kruger, *A survey of advances in vision-based human motion capture and analysis, Computer Vision and Image Understanding*, Volume 104, Issues 2-3, November-December 2006: Pages 90-126.

33. Bouguet, J.Y., *Visual methods for three-dimensional modeling*, PhD thesis, Caltech, 1999

34. Murphy-Chutorian, E., Trivedi, M.M., *Head Pose Estimation in Computer Vision: A Survey*, Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.31, no.4, April 2009: pp.607-626.

35. Downton, A.C., Drouet, H., *Model-based image analysis for unconstrained human upper-body motion*, Image Processing and its Applications, 1992., International Conference on , 7-9 Apr 1992: pp.274-277.

36. Kastens, K. A. , Agrawal, S. and Liben, L. S., *Research in science education: The role of gestures in geoscience teaching and learning*. Journal of Geoscience Education, 2008: pp. 362-368

37. D. Martinec and T. Pajdla, *Structure from Many Perspective Images with Occlusion*, Proc. European Conf. Computer Vision, 2002: pp. 355-369.

38. Abhinav Gupta, Aniruddha Kembhavi, Larry S. Davis, *Observing Human-Object Interactions: Using Spatial and Functional Compatibility for Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 10, Apr. 2009: pp. 1775-1789.

39. Bradski, G., Kaehler, A., *Learning OpenCV: Computer Vision with the OpenCV Library*, 1st edn. O'Reilly (2008).

40. Pollefeys, M., 1999. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD Thesis, Center for Processing of Speech and Images (PSI), Dept. of Electrical Engineering, Faculty of Applied Sciences, K.U. Leuven, available at http://www.esat.kuleuven.ac.be/˜pollefey/publications/PhD.html 25 September 2000.

REFERENCES (Continued)

41. Liang Wang, Weiming Hu, Tieniu Tan, *Recent developments in human motion analysis*, Pattern Recognition, Volume 36, Issue 3, March 2003: Pages 585-601

42. T. Svoboda, D. Martinec, T. Pajdla, J.-Y. Bouguet, T. Werner, and O. Chum. *Multi-Camera Self-Calibration*. Czech Technical University, Prague, Czech Republic.\tt http://cmp.felk.cvut.cz/~svoboda/SelfCal/.

43. Thomas B. Moeslund, Erik Granum, *A Survey of Computer Vision-Based Human Motion Capture*, Computer Vision and Image Understanding, Volume 81, Issue 3, March 2001: Pages 231-268.

44. P. Sturm and W. Triggs, *A factorization based algorithm for multi-image projective structure and motion*, In *Proc. ECCV*, 1996: pages 709–720.

45. Hartley, R. and Zisserman, A., *Multiple view geometry in computer vision*, Cambridge University Press: Cambridge, UK, 2000.

46. Noskovičová Luci, Toolbox for camera – Self Calibration - http://www.posterus.sk/?p=2071, September 2009.

47. Subhashis Banerjee, *Anatomy of a projective camera*, In Projective geometry, camera models and calibration -- Courese Notes, Indian Institute of Technology, Delhi, 2008: p.19-20.

48. Heikkila, J., Silven, O., *A four-step camera calibration procedure with implicit image correction*, Computer Vision and Pattern Recognition, 1997. Proceedings., *1997 IEEE Computer Society Conference on* , 17-19 Jun 1997: pp.1106-1112.

49. Georgery D. Hager, *Projective Geometry and Calibration*, In Computer Vision -- Courese Notes, John Hopkins University, 2003: p.6-27.

50. Z. Zhang, *A flexible new technique for camera calibration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22, No.11, 2000: pages 1330–1334.

51. C. Harris and M. Stephens, *A combined corner and edge detector*. Proceedings of the 4th Alvey Vision Conference, 1988: pp. 147–151.

52. *Image rectification* - http://en.wikipedia.org/wiki/File:Image_rectification.png

REFERENCES (Continued)

53. *Stereo data sets with ground truth*, Middlebury College, UK - http://cat.middlebury.edu/stereo/data.html.

54. *i-Spy Free Camera Security Software* - http://www.ispyconnect.com/.

55. Martin A. Fischler and Robert C. Bolles, *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*, Comm. of the ACM 24**,** June 1981: 381–395.

56. *Similar matrix* - http://en.wikipedia.org/wiki/Similar_matrix

57. B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. *Bundle adjustment – A modern synthesis*. In W. Triggs, A. Zisserman, and R. Szeliski, editors, Vision Algorithms: Theory and Practice, LNCS, Springer Verlag, 1999: pages 298–375.

58. Canny, J., *A computational approach to edge detection.* Readings in computer vision: issues, problems, principles, and paradigms, 1987. 184.

59. Gonzalez, R. and R. Woods, *Digital image processing. 2002.* ISBN: 0-201-18075-8.

60. *Normalized 2-D cross-correlation* - Matlab Image processing toolbox http://www.mathworks.com/help/toolbox/images/ref/normxcorr2.html

61. Lewis, J. P., *Fast Normalized Cross-Correlation*, Industrial Light & Magic

62. Nusirwan Anwar bin Abdul Rahman, Kit Chong Wei, and John See, *RGB-H-CbCr Skin Colour Model for Human Face Detection*, MMU International Symposium on Information & Communications Technologies (M2USIC 2006), PJ, Malaysia, 2006.

63. *Measure properties of image regions* - Matlab Image processing toolbox http://www.mathworks.com/help/toolbox/images/ref/regionprops.html

64. Haralick and Shapiro, Computer and Robot Vision vol I, Addison-Wesley 1992: pp. 62-64

65. Karray, F., Alemzadeh, M., Saleh, J., Arab, M.: *Human-computer interaction: Overview on state of the art.* International Journal on Smart Sensing and Intelligent Systems 1, (2008): 137–159

VITA

NAME:	Shankaranand Jagadeesan
EDUCATION:	B.E., SSN College of Engineering, Anna University, India, 2007
	M.S., University of Illinois at Chicago, Illinois, 2011
EXPERIENCE:	Graduate Student Researcher, Machine Vision Laboratory, University of Illinois at Chicago, Chicago, USA 2009-2011
	Graduate Assistant, Liautaud Graduate School of Business, University of Illinois at Chicago, Chicago, USA 2010-2011
	Graduate Research Assistant, Health Systems and Science Department, University of Illinois at Chicago, Chicago, USA 2008-2009
	Programmer Analyst, Cognizant Technology Solutions, Chennai, India 2007-2008
PROFESSIONAL MEMBERSHIP:	Institute of Electrical and Electronics Engineers (IEEE)