Vision application of human robot interaction: Development of a ping pong playing robotic arm

Kalpesh Prakash Modi
VISION APPLICATION OF HUMAN ROBOT INTERACTION:

DEVELOPMENT OF A PING PONG PLAYING ROBOTIC ARM

By

KALPESH PRAKASH MODI

Thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

Approved by

Thesis Advisor
Dr. Ferat Sahin

Thesis Committee
Dr. Eli Saber

Thesis Committee
Dr. Wayne Walter

Department Head
Dr. Robert Bowman

Ferat Sahin
Eli Saber
Wayne Walter
Robert Bowman

DEPARTMENT OF ELECTRICAL ENGINEERING, KATE GLEASON COLLEGE OF ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK
MAY 2005
VISION APPLICATION OF HUMAN ROBOT INTERACTION:
DEVELOPMENT OF A PING PONG PLAYING ROBOTIC ARM

I, Kalpesh Prakash Modi hereby grant the permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Author: Kalpesh Modi
(Kalpesh Prakash Modi)

Date: 05/04/2005
Dedicated to

My parents, Mr. Prakash and Mrs. Meena Modi, who have always blessed me with their inspirational support for demonstrating persistent and sincere efforts throughout

and

My loving sister Bosky, who has always cheered and bolstered me through her optimism in my approaches.

-You'll always have the best of me.
Acknowledgments

I take this opportunity to express my utmost respect for Dr. Ferat Sahin, who has advised, guided and steered me enroute, through my Master’s degree with immense care. Under his supervision, I was able to develop the habit to prepare myself for a problem and then solve it thoroughly, by almost nullifying the pressure factor. I thank him earnestly for his faith in me as his graduate student and whilst including me as a member of his Robotics Lab.

I convey my sincere gratitude to Dr. Eli Saber for his keen counseling and salute his proficiency in Image and Video Processing. It was in his mentoring only, that I sought motivation to work hard independently. I thank him and Dr. Wayne Walter for assessing the work in this thesis.

A special vote of respectful appreciation to the Head, Dr. Robert Bowman for providing me with an opportunity to pursue my degree in Electrical Engineering. Many thanks to Mr. Kenneth Snyder and Mr. James Stefano for availing me with the privileges and technical assistance to operate the lab and system utilities. I am also thankful to Ms. Patty Vicari, Ms. Florence Layton and Ms. Jill Lewis for assisting me to keep track of my academic and registration status throughout my degree.

I am highly grateful to Mr. Mark Chast, Sr. Systems Administrator in the College of Engineering, for acquainting me about key issues pertaining to Linux and programming and providing a boost for applying them in this thesis. I am extremely thankful to Dr. Josef Torok, for bestowing upon me, a chance to serve the Dean’s office and earn my livelihood during my stay at RIT. With great honor, I thank him for his
recommendations on physics and math aspects. I am very thankful to Mr. Dave Hathaway and Mr. Steve Kosciol for their professional assistance in the mechanical interface of the ping pong paddle.

As an international student, I am obliged to Ms. Lilli Jensen at the International Student Office, for her utmost practical counseling towards my stay and services for the United States. I am also indebted to Dr. Howard Ward of Housing operations, for facilitating me to utilize the time and monetary benefits of living on campus.

My best regards to Mr. Lawrence Brown and Ms. Alfreda Brown for their intense familial support, love and care. I also express high esteem towards my family members back home for their assurance and confide in my pursuit for higher education.

Lastly, I thank HIM, for making things possible.
VISION APPLICATION OF HUMAN ROBOT INTERACTION: 
DEVELOPMENT OF A PING PONG PLAYING ROBOTIC ARM

By
KALPESH PRAKASH MODI
Master of Science in Electrical Engineering

Abstract

Robotics is a science that is implemented parallel to human behavior. This work describes and implements techniques to mathematically model the game of ping pong played by the humans, and utilization of these methods in the design and development of a ping pong playing robotic arm as an application of robotic vision. Displaced frame difference (DFD) is used to segment the ball motion from background motion and parametric calibration of single CCD camera is utilized to track the ball in three dimensions. This visual information is temporally updated and further applied to guide a robot arm to hit the ball at a specified location in time. The results signify the system development based on single camera tracking and also demonstrate its working with self-sufficiency for the color of the ball. System latency is measured as a function of the camera interface, processor architecture, and robot motion. Various hardware and software parameters that influence the real time system performance are also discussed.

Keywords: Real time machine vision, camera calibration, 3-D imaging, robot vision.
# Table of Contents

Thesis Release Permission Form ................................................................. ii
Acknowledgments .................................................................................... iv
Abstract ..................................................................................................... vi
Table of Contents ..................................................................................... vii
List of Figures .......................................................................................... x
List of Tables ............................................................................................ xiii

1. Introduction ............................................................................................. 1
   1.1 Motivation .......................................................................................... 1
   1.2 Objective .......................................................................................... 3
   1.3 Nomenclature .................................................................................... 5

2. System Theory ....................................................................................... 7
   2.1 Introduction ....................................................................................... 7
   2.2 Digital Signal Processing .................................................................. 7
   2.3 Computer Vision ............................................................................... 9
   2.4 Ping pong robot system ...................................................................... 15
      2.4.1 Vision system ............................................................................... 16
      2.4.2 Trajectory Analysis ..................................................................... 23
      2.4.3 Expert controller ......................................................................... 24

3. System Configuration ........................................................................... 27
   3.1 Introduction ....................................................................................... 27
   3.2 Multimedia system ............................................................................ 28
3.2.1 Workstation ................................................. 28
3.2.2 Camera .................................................... 30
3.3 Robot ....................................................... 30
4. System Implementation ........................................ 33
  4.1 Introduction ............................................... 33
  4.2 System Design ............................................. 33
    4.2.1 Frame acquisition ..................................... 34
    4.2.2 Motion and Ball detection ............................ 35
    4.2.3 Ball location in the image plane ...................... 46
    4.2.4 Camera calibration and 3-D Imaging ................ 50
    4.2.5 Motion Estimation .................................... 59
    4.2.6 Target point computation ............................ 64
    4.2.7 Robot motion .......................................... 67
5. Experimental Procedure and Analysis .......................... 70
  5.1 Foreword ................................................ 70
  5.2 System setup ............................................ 70
    5.2.1 Camera calibration ................................... 70
    5.2.2 Paddle interface and robot positioning ............. 76
    5.2.3 Software control ..................................... 77
  5.3 System performance ..................................... 78
    5.3.1 Frame capture rate and processing time ............ 78
    5.3.2 DFD .................................................. 79
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.3</td>
<td>Motion detection</td>
<td>80</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Ball detection</td>
<td>82</td>
</tr>
<tr>
<td>5.3.5</td>
<td>Ball location in the image plane</td>
<td>83</td>
</tr>
<tr>
<td>5.3.6</td>
<td>3-D Imaging</td>
<td>84</td>
</tr>
<tr>
<td>5.3.7</td>
<td>Displacement</td>
<td>89</td>
</tr>
<tr>
<td>5.3.8</td>
<td>Trajectory estimation and target point computation</td>
<td>92</td>
</tr>
<tr>
<td>6.</td>
<td>Conclusion</td>
<td>94</td>
</tr>
<tr>
<td>6.1</td>
<td>Synopsis</td>
<td>94</td>
</tr>
<tr>
<td>6.2</td>
<td>Performance analysis</td>
<td>95</td>
</tr>
<tr>
<td>6.3</td>
<td>Future work based on current limitations</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Appendix A</td>
<td>106</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1-1 Robot navigation and control set up ......................................................... 4
Figure 2-1 Image formation on camera ........................................................................... 9
Figure 2-2 Structural flow of Computer Vision ................................................................. 10
Figure 2-3. Hardware and software categorization of computer vision blocks ............ 14
Figure 3-1 Ping-pong system .......................................................................................... 27
Figure 3-2 Robot parts and co-ordinates ......................................................................... 31
Figure 3-3 Major axes of the robot .................................................................................. 32
Figure 4-1 System block diagram ................................................................................... 33
Figure 4-2 Acquisition data flow ..................................................................................... 34
Figure 4-3 Video as a sequence of frames ........................................................................ 35
Figure 4-4 DPCM: Redundancy reduction ...................................................................... 37
Figure 4-5 Motion detection region in DFD ..................................................................... 39
Figure 4-6 Motion sensing ............................................................................................... 40
Figure 4-7 DFD for detecting changes in scene ................................................................. 41
Figure 4-9 Ball silhouette ................................................................................................. 46
Figure 4-10 Circle-ellipse similarity ................................................................................ 47
Figure 4-11 Point representation on a circle and an ellipse .............................................. 48
Figure 4-12 Tilt in the XY plane ...................................................................................... 48
Figure 4-13 Focal length of camera ................................................................................ 51
Figure 4-14 Field of view measurement .......................................................................... 53
Figure 4-15 Pixel to millimeter transformation ................................................................. 54
Figure 4-16 3-D Imaging based on scene and object topology ........................................... 56
Figure 4-17 Free body diagram of a projectile ................................................................. 61
Figure 4-18 Displacement in 3-D .................................................................................. 62
Figure 4-19 Coefficient of restitution ............................................................................. 65
Figure 4-20 Motion estimation ....................................................................................... 66
Figure 4-21 Target point computation ............................................................................. 67
Figure 4-22 Robot motion ............................................................................................... 68
Figure 5-1 Experimental top view of focal length computation ...................................... 71
Figure 5-2 Experimental observation for change in ball size ........................................ 72
Figure 5-3 Ball distance vs. size ..................................................................................... 73
Figure 5-4 Resolution in mm .......................................................................................... 74
Figure 5-5 Side view of the scene topology ..................................................................... 76
Figure 5-6 DFD identifying a color ball ........................................................................... 79
Figure 5-7 DFD identifying a white ball ......................................................................... 80
Figure 5-8 Binarization for color ball ............................................................................. 81
Figure 5-9 Binarization for white ball ............................................................................. 82
Figure 5-10 Ball segmentation ......................................................................................... 83
Figure 5-11 Single camera based 3-D tracking of color ball ......................................... 87
Figure 5-12 Single camera based 3-D tracking of white ball ........................................ 88
........................................................................................................................................... 89
Figure 5-14 Velocity plot ............................................................................................... 91
Figure 5-15 Estimated trajectory

Figure 5-16 Updated trajectory with target point
List of Tables

Table 1-1 Computer vision applications ................................................................. 2
Table 2-1 Classification of signals ............................................................................ 8
Table 2-2 Acquisition components for ping pong robot system ............................. 16
Table 2-3 Robot and controller specifications ......................................................... 25
Table 4-1 Trajectory based motion estimation .......................................................... 63
Table 4-2 Trajectory matrix ...................................................................................... 64
Table 5-2 Horizontal resolution ................................................................................ 84
Table 5-3 Vertical resolution ...................................................................................... 84
Table 5-4 Focal length ............................................................................................... 85
Table 5-5 Ball velocity based on displacement in time ............................................. 90
1. Introduction

1.1 Motivation

Primates adapt themselves to their habitat by perceiving changes through their senses and responding appropriately to them. Vision is their most powerful sense as it provides a detailed three dimensional knowledge of the surrounding world. This enables them to notice the dynamics in the visualized environment and interact accordingly. Over the last twenty years, science and engineering have developed rapidly and are concurrently applied to increase machine (artificial) intelligence. For machines also, vision is a potential interface with the system as it facilitates sensing of objects and tracking changes in the scene.

Human-computer interaction is a relatively new discipline, which aims at the scientific study of people’s communication with computers and applications of associated expertise. This sociotechnological field has advanced enormously towards recognizing and improving our relationship with computer based technologies. Human-robot interaction is an applied area that intends to comprehend the behavioral aspects between human and a robot and includes study, design and development of computing systems for a measure of their joint performance. This performance can be evaluated through techniques, which compare human and robot productivity in a team environment [1]. Computer vision is such a scientific tool, which is used to measure the mutual

\[ \text{a timely manner.} \]
A significant number of technical publications on computer vision address various research areas such as three dimensional scene analysis, image analysis and synthesis, range finding, segmentation, stereo vision, shape modeling, object recognition and tracking [2, 3, 4, 5, 6, 7, 8, 9]. Applications that were considered creative imaginations are successfully employed in the real world today. Table 1-1 summarizes the computer vision methods that have been applied to solve various real world problems.

**Table 1-1 Computer vision applications**

<table>
<thead>
<tr>
<th>Industry</th>
<th>Computer vision techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics and multimedia</td>
<td>3-D Pipelining (transformation, lighting, setup, rendering), virtual reality, telepresence, video indexing, inter and intra frame coding, object layer based video coding.</td>
</tr>
<tr>
<td>Medical imaging</td>
<td>Image registration, image fusion, X-Ray based computed tomography (CT), magnetic resonance imaging (MRI), optical diagnosis.</td>
</tr>
<tr>
<td>Sports</td>
<td>3-D virtual simulation, player tracking, object tracking, scene analysis.</td>
</tr>
<tr>
<td>Security and surveillance</td>
<td>Automatic target recognition, motion detection and monitoring, aerial navigation, number plate identification, biometric measurements.</td>
</tr>
<tr>
<td>Power-driven and</td>
<td>Inspections, defect detection and quality control, spray painting, grading, part assembly.</td>
</tr>
</tbody>
</table>


These applications are based on signal-processing techniques and algorithms, which acquire and process spatial and temporal information either from multiple-sensed data sets or from real-time sequences using sensory devices. Another application of computer vision focuses on developing guiding tools for persons with visual impairments, including low vision and blindness. Currently, surgical systems like the da Vinci use 3-D imaging technology that allows a human surgeon to get closer to the surgical site than human vision and hence enables high precision endoscopic operations [10]. Thus, vision is an instructive means of feedback for survey, navigation and control.

This research and development is driven by the desire to understand and design a vision based system that can be navigated and controlled for interaction with humans.

1.2 Objective

In this work, a robotic vision system based on spatio-temporally sampled visual information is developed. This time based vision data is analyzed in three dimensions of the real world and a feedback is generated to communicate with the robot in planning its motion to achieve the task of hitting the ball.

Development of advanced sensing systems is vital in the continued advancement of robotics field [11]. Analogous with the human characteristics, visual information is highly significant to acquire information of the surroundings and navigate the robot for a specific task. For extraction of real time knowledge about the dynamically changing environment, visual sensors are included in the feedback loop [12]. This work based information as an effective means of feedback for
computation of real world dynamics. These motion dynamics are further applied to the robot for striking the ball.

Video sequences sampled regularly every 15 hertz are acquired through a Charged Couple device (CCD) camera interface to obtain scene information. Motion is detected through the temporal change in the brightness values (gray levels) in the image plane [6, 13]. Motion based image segmentation and object detection techniques are used to locate the ball in the image plane. 3–D image reconstruction is performed based on the geometry of the universal coordinates and frame transformations developed for perspective camera vision. Furthermore, a trajectory path symbolizing the velocity of ball in three dimensions is extrapolated. This motion data is updated every fifteenth of a second and is used as feedback to guide the robotic arm for movement, through a serial computer interface. Figure 1-1 illustrates a pictorial representation of the camera based robot system developed.

![Figure 1-1 Robot navigation and control set up](image-url)
Thus, the temporally updated sensory information is used as feedback to communicate with the robotic arm. In response to the feedback, the robotic arm produces corresponding movement for hitting the ball. The main objective of this work is to develop a single camera vision based system for surveying the scene and tracking a ball in three dimensions. This dynamic information is further analyzed and utilized as feedback to navigate a robot to hit the ball at the specified target.

1.3 Nomenclature

Subsequent chapters in this documentation describe

- Literature review of computer vision technology and its application, particularly for the development of the ping pong robot system.
- Controller system configuration.
- Algorithm for tracking the ball in a 3-D space using single camera.
- Trajectory problem and computation.
- System design and experimentation.
- Simulation and performance results of the realized ping-pong robot system.

Chapter 2 contains a literature survey of the computer vision techniques and various ping-pong robot systems developed. This provides the background theory for computational aspects of the spatio-temporal sensory information and the robot's response to the visual feedback.

This section discusses about the vision system and controller configuration.
Chapter 4 describes the system implementation. In this chapter, the physics of the environment is explored and image plane co-ordinates are mapped to the robot world. A conceptual framework stating the processes involved in the system is explained. The essential mathematical and geometrical formulations used for developing the algorithm are also explained. Information with respect to the camera positioning and base mounting are discussed in this chapter.

Chapter 5 lists the experimental procedure for the ping pong playing robot system and presents the simulation results based on this experimental framework described. Performance results validating the outcome of this research and development are also mentioned in this chapter.

Chapter 6 discusses the conclusions derived from this research and development. Various hardware and software parameters that influence the real time system performance are also discussed.

References list the citations and books studied for this thesis.

Appendix A consists of a CD-ROM that contains the source code for

- Simulation and testing of the developed algorithm using Matlab
- Source code for real time system implementation in C
- Standard library functions in C++.
2. System Theory

2.1 Introduction

This chapter presents a literature review on signal processing and the computer vision technology, particularly as an application for the ping pong system design. Further, theoretical knowledge for the development of this system is reviewed. An evaluation based on a comparison of various ping pong systems with the system developed in this thesis is summarized at the end of this chapter.

2.2 Digital Signal Processing

Digital Signal Processing is an area of science and engineering that has developed rapidly over the past thirty years as a result of the significant advances in digital computer technology. A signal is defined as any physical quantity that varies with time, space or any other independent variable or variables. Mathematically, a signal is a function of one or more independent variables, e.g. speech, music, picture, video. Various types of signals are defined depending on the nature of the independent variables and the value of the function defining the signal. For instance, a 1-D signal is a function of one independent variable; 2-D signal is a function of two independent variables and so on. Thus, a signal carries information and the goal of signal processing is to extract useful information carried by the signal [14]. Table 2-1 classifies different types of signals.

An image can be defined as a two-dimensional function, \( f(x, y) \) where \( x \) and corresponding to row \( x \) and column \( y \). The amplitude of \( f \) at any pair of coordinates \((x, y)\) is called the intensity or gray level of the image at
that point [15]. This intensity is proportional to the radiant energy received in the electromagnetic band.

**Table 2-1 Classification of signals**

<table>
<thead>
<tr>
<th>Signal</th>
<th>Dimensions</th>
<th>Independent Variable(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech</td>
<td>1- D</td>
<td>Time</td>
</tr>
<tr>
<td>Image (Photograph)</td>
<td>2 - D</td>
<td>Spatial variables</td>
</tr>
<tr>
<td>Frame of black-and-white video signal</td>
<td>2 - D</td>
<td>Two discrete spatial variables</td>
</tr>
<tr>
<td>Black-and White Video Signal</td>
<td>3 - D</td>
<td>Two spatial variables and time</td>
</tr>
<tr>
<td>Color Video Signal</td>
<td>3 - D</td>
<td>Three 3-D signals representing primary colors: Red Green and Blue.</td>
</tr>
</tbody>
</table>

In biological vision, the eye is a simple optical instrument as compared to the process of recognition. With internal images projected from the objects in the outside world, it is Plato’s cave with a lens [16]. The visual ability of humans and other animals is the interaction of light, eyes and the brain. Visual perception takes place by the relative excitation of light receptors, which transform radiant energy into electrical impulses that are ultimately decoded by the brain [15]. The role of camera in computer vision is in parity with that of an eye in biological vision. The optical component of the camera is the lens, which is a curved piece of glass or plastic. Light waves (consisting of energy in the
form of electric and magnetic fields) strike a surface and are reflected back at the same angle. These bouncing beams of light are collected by the lens and redirected to the sensor of a camera where electromagnetic energy is transformed into electrical signals to form an image, as illustrated by Figure 2-1.

![Figure 2-1 Image formation on camera](image)

Thus, an image is a spatial representation of an object, a two-dimensional or three-dimensional scene, or another image. It can be real or virtual, as in optics [17].

### 2.3 Computer Vision

In order to interact with the surrounding, it is necessary to gain knowledge about the data in the surrounding. Vision is an efficient source of data acquisition and is used to regulate behavior with the environment in a desired way. Computer vision is an applied science that develops the theoretical and algorithmic basis by which useful information about the world can be automatically extracted and analyzed from an observed image, image set, or image sequences [17]. This helps in making decisions for developing applications and systems for which the image matching and statistical analysis methods alone are insufficient.
Computer vision encompasses the spaces which deal with the recognition of real world objects and an analysis for extracting useful information about the scene [18]. This information can be used either as a feedback for applications, or as data for synthesized display. As shown in Figure 2-2, the structural flow of computer vision is systematized by spaces cached for image, user and computer.

Figure 2-2 Structural flow of Computer Vision

Image space is involved with the formation of a picture on the sensor of a camera and its representation in $2-D$. A video stream can be considered as a sequence of
images. Thus, images formed on a sensor represent a video signal over a period of time. A light source illuminates an everyday scene and an image of the real world is formed on the sensor of the camera. These images are sensed from video sequences over a period of time $t$. Each image formed on the camera sensor is a still representation of the real world objects. An image is read in the units of pixel (picture element). A pixel has properties of position and value but is different than the units of observation. Computer vision deals with the mapping of these pixel values into real world observation values through image processing, analysis, storage and modeling.

As illustrated previously, in Figure 2-2, an image is represented as a 2-D array of numeric values which are read pixel wise. In order to recognize individual regions or objects in an image explicitly, further processing is required. Such a method whose inputs and outputs are images is called image processing and the function of image processing is to represent an image using an enhanced approach. This comprises of various pre processing and post processing steps for noise suppression, blur removal, and edge detection. These steps are utilized for the removal of unwanted information present in the form of noise and extract essential features. The essential attributes are of specific dimensions and through image processing, can be measured in units of pixel positions and levels of brightness, either at that specific location or with reference to the total array size. There exists a geometric correspondence between the points in the scene and the points in an image. This relation determines the real world location, shape and size of an object in the scene through the pixel information in an image. An analysis technique that helps in modeling the objects in the scene through their representation in an image is
required to determine the geometric correlation between the observation values and the pixel values.

Image analysis can be performed upon color data or processed grayscale (black and white) data, read from the array values. These values are required throughout the experimental analysis and need to be stored for further retrieval. Memory of a computer is the physical space where these values are stored on a temporary or permanent basis throughout the program and are utilized by the routine steps performed to evaluate the efficiency of processing and analysis. These procedural steps are performed in the user space and computer space to provide a complete recognition of objects in the real world. User space is consumed by the user-defined algorithms, which encompass the course of routines to be executed for achieving desired purposes. These routines are executed instruction wise and are evaluated by a processing unit of the computer on a timely basis. Computer space is used for the linkage of the procedures and the system devices to provide a computationally efficient analysis of this data and its storage. User and computer spaces are collectively employed to represent the information in a numerical form. User defined instructions evaluate this mathematical data to implement the objective of the system. A perfect synchronization between the user defined instructions and a proportional execution, results in an optimal system design. Thus, the structure of computer vision entails

- Representation of a world scene in image coordinates
- Processing of images for accentuating essential information about the objects
• Analysis of image for determining measurements of the objects relative to the scene
• Storage and model based retrieval of the measured values for feedback applications or synthesized display.

Image, user and computer spaces are articulate spaces associated with the structural formation of computer vision.

A typical system consists of
• Hardware unit comprising of device(s) like computer(s) and other equipment(s) directly involved in the performance of data processing or communications functions.
• Software unit consisting of programs, rules, routines and symbolic languages that control the functioning of the hardware and direct its operation [19].

The purpose of a computer (machine) vision system is to produce a symbolic description of sight being imaged and this description may then be used as a feedback, either to direct the interaction of a system with its environment or interpretation by a human. For interpreting and analyzing images of the scene, the tools of a distinctive computer vision system include
• Hardware for acquiring and storing digital information as representation of real world.
• Software for processing the information and communicating results to other automated systems or users.
The designers and users of computer vision systems depend on a repertoire of software and hardware building blocks [18]. Based on the structural flow of computer vision, Figure 2-3 categorizes the hardware and software building blocks.

**Figure 2-3. Hardware and software categorization of computer vision blocks**

Vision systems utilize the algorithms that form perception. The software components of vision system include image processing, understanding, user interfaces, libraries and databases. Hardware in the form of image acquisition devices and data processing equipments uphold the structural flow of computer vision.

This thesis demonstrates the utilization of computer vision as feedback in development of the ping pong playing robot.
2.4 Ping pong robot system

A robot is defined as a mechanical device that performs automated tasks through direct human supervision or programmed control and Robotics is a science that deals with the study of robots. The word robot comes from the Czech word robota meaning “labor.” It was first used in English in reference to Karel Capek’s play Rossum’s Universal Robots. The ISO 8373:1994 standard Manipulating Industrial Robots-vocabulary defines an industrial robot as an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes. A robot may include a feedback-driven connection between sense and action, not under direct human control [20]. The system developed in this work is such an example, which observes stipulated programming to provide a temporally updated feedback for navigating the robotic arm movement to play ping pong. The sport, now called table tennis was originally termed ping pong after the sound the ball makes when it hits the table. It is the most popular racquet sport in the world and is ranked second overall in terms of participation. This thesis aims at extending the sport to robotics through an appropriate interface with computer vision.

The principal system components of the ping pong playing robot system include

- 3-D vision system that locates the ball in space
- Trajectory analyzer that extrapolates the path of ball motion
- An expert controller in form of a computational machine that computes the target point and a robot that hits the ball in a desired approach.
2.4.1 Vision system

2.4.1.1 Camera vision

A video sequence is a much richer source of visual information than a still image [21]. This is because a single image provides a single shot about the scene while the capture of motion reveals about the dynamics in the image. Thus motion carries a lot of information about the spatiotemporal relationships between image objects. Analogous to all other perceptual systems, vision systems are existent amongst applications that acquire motion dynamics about the external world from sensed images. An acquisition device in form of a sensor or camera is used to capture the video sequences. Table 2-2 reviews acquisition components for ping pong systems developed till date. [22-26].

Table 2-2 Acquisition components for ping pong robot system

<table>
<thead>
<tr>
<th>Developer(s)</th>
<th>Vision element</th>
<th>Spatial resolution</th>
<th>Time resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andersson, 1986 [22]</td>
<td>Four cameras</td>
<td>756x242</td>
<td>60 hertz</td>
</tr>
<tr>
<td>Hashimoto, Ozaki, Asano, Osuka, 1987 [23]</td>
<td>Binocular camera with four sensors</td>
<td>2048x2048</td>
<td>100 hertz</td>
</tr>
<tr>
<td>Fassler, Zurich 1990 [24]</td>
<td>Two CCD cameras</td>
<td>422x579</td>
<td>50 hertz</td>
</tr>
<tr>
<td>Nagdy, Wyatt, Tran, 1993 [25]</td>
<td>Two cameras with frame grabber</td>
<td>1024x1024</td>
<td>≈ 60 hertz</td>
</tr>
<tr>
<td>Miyazaki, Kusano, 2002 [26]</td>
<td>Quick MAG III</td>
<td>640x416</td>
<td>60 hertz</td>
</tr>
</tbody>
</table>
Advances in the hardware industry and efficient image and video processing algorithms enable tracking of objects in video sequences and determine motion. This allows camera based vision system to act as sensor information, as compared to the use of active sensor devices [27]. The ping pong playing robot system developed at the University of La Laguna, Spain employs a single CCD video camera along with a standard image-acquisition card based on the chip BT878 (768x576 pixels @ 40 hertz) to compute the ball location [28]. Computer cameras, where the conversion from analog to digital data is synchronized with the serial register read out, have the following advantages in comparison to the use of video cameras [29].

- Flexibility of image acquisition as a result of
  - sub-array scanning (acquisition time reduction)
  - asynchronous triggering of exposure (arbitrary refresh rate using pulsed illumination)
  - horizontal and vertical pixel binning (fast read out of full image area)
  - time delay and integration (TDI) (reduction in effect of motion blur)
  - high speed TDI (rapidity in representing TDI data via row transfer of charge)
  - multiple camera simultaneous image acquisition (sharing of identical exposure time between two or more cameras)

- Noise and dynamic range considerations due to control in
  - exposure time
  - anti-blooming
  - on-chip pixel binning
• Synchronization between CCD pixel and A/D conversion clocks for elimination of
  • aliasing effect (low frequency sampling due to which a larger digital storage is
    needed relative to the spatial information in the image), that alleviates
    measurement inaccuracies and leads to higher effective resolution
  • pixel jitter (varying presence of locations of the peaks and valleys of signal
    representing the difference between number of pixels per line and number of
    A/D samples per line)

In this system development, a USB communicative computer camera is interfaced to
acquire a frame consisting of (640×480) picture elements every 66 milliseconds.

Information obtained from the vision system is used to control the robot motion in
real time, as opposed to older systems in which the vision systems derived an initial
representation of the world that is then used to plan robot motions [30]. The ability to
track objects through motion perceived from video sequences, allows a robot to rely on
vision-based navigation techniques and avoid using active sensors or sophisticated stereo
imagers for distance measurement [31]. According to Jain [32], there are three main
phases in motion perception
• Peripheral: Identification of areas in the field of view with persistent changes from
  frame to frame.
• Attentive: Attention onto one of image areas in order to investigate it in more detail
• Cognitive: Relate the observation derived from the subimage sequence to the
  knowledge about the section of the real world covered by the field of view.
The purpose of vision in a ping pong robot system is to extract information about
the scene dynamics and locate the ball in space \((x, y, z)\) at particular time \(t\). This
3-D expanse data is concurrently employed with time statistics to track the ball in real
time. In this system design, the vision interface assists the robot to perceive ball motion
by utilizing these three phases simultaneously for ball motion segmentation, ball
detection in the image plane and 3-D ball location in the scene.

2.4.1.2 Ball motion segmentation:

Segmentation refers to the identification of regions that are homogenous in some
sense. Clustering is one method of identifying this homogeneity by measuring the
closeness between the pixels. It is a process of grouping feature vectors into classes
developed into self-organized groups. This technique is based on the assumptions that
image motion across a cluster can be approximated by the motion of the centroid of the
cluster and the clusters belonging to the same object have roughly parallel trajectories
[33]. Another popular tool used in image segmentation is thresholding (binarization),
where the pixel intensities are represented at two levels. [5] presents a survey of various
global (image based) and local (region based) thresholding techniques.

In this work, motion is measured as a function of changes in intensity level as the
ball moves through space. In order to reduce the amount of data to be processed, the three
channel color image is symbolized using a single channel as a gray level image. One of
the simplest approaches for detecting changes between two image frames \(f(x, y, t_i)\) and
\(f(x, y, t_j)\) taken at times \(t_i\) and \(t_j\) respectively, is to compare two images pixel by pixel
and this comparison is performed by computing the difference between the two images. The displaced frame difference ($DFD$), representing the difference in brightness levels of two image sequences is utilized to determine motion in a specific area of interest, based on a global threshold value [6, 13]. Thus, global thresholding is utilized to detect motion and identify the region having the potential of being the ball, through binary pixel values.

2.4.1.3 Image plane analysis

The goal of image analysis is the construction of scene and object descriptions on the basis of information extracted from image or image sequences. The methods by which an object can be described are varied, but in primitive terms the description may consist of a set of surfaces, edges or vertices. Any one of these three sets can be used in a description process. Edges are the intersections of object surfaces, while vertices are the intersections of the surface borders. Both edges and vertices represent spatial discontinuities of a three-dimensional object and characterize the geometrical structure of the object in three dimensions [7]. Edges are curves in the image where rapid changes occur in the brightness or in the spatial derivatives of brightness and arise from discontinuities in surface orientation and surface reflectance properties [34]. Thus edge detection can be considered complementary to image segmentation, since edges can be used to break up images into regions that correspond to different surfaces [13]. John Canny described a computational approach for the design of edge detectors for arbitrary edge profiles [35]. A ping pong ball has very clearly defined boundaries for a human, but for a machine vision system, detecting smooth, round translucent edges is a difficult task, as the moving ball introduces noise and blurred images [36]. Edges in images often result
from occluding contours of objects. Illusory contours can be induced along directions approximately collinear to edges or approximately perpendicular to ends of lines. The exact role of edges and line-ends in illusory contour formation is elucidated by Lesher and Mingolla in [37]. Anderson and Barth [38] demonstrated that the velocity of contour terminations and the direction of motion of a partially occluded figure regulate the perceived shape and apparent movement of illusory contours formed from moving image sequences. Also, binary images introduce a significant amount of aliasing, limiting the vision system's accuracy. In short, each pixel around the ball's boundary may randomly be included or excluded from the ball, depending on the pixel's exact intensity. Since there are relatively few pixels in the object, each global thresholding decision induces substantial noise in the ball's apparent position [22]. These practical reasons coerce the use of a region based local threshold that extracts the exact ball region from noise and other scene changes. Edges are high frequency components and exhibit high energy. A set of points which have been determined to lie on an edge can be represented by active contours (snakes). Tracking of the complete object can be achieved by employing the active contours by minimizing the energy function

\[
E(\Gamma) = \int_{0}^{1} E_{\text{internal}}(\nu) + E_{\text{image}}(\nu) ds
\]  

(2-1)

where \(s\) is the arc length of contour \(\Gamma\), \(E_{\text{image}}\) signifies the energy based on image observations, and \(E_{\text{internal}}\) prevents gaps and rapid bending [9].

In this thesis, motion is detected via background subtraction and a correspondence from frame to frame is established. Further, contours are formed for
motion detected region that has the feasibility of being the ball. This technique is implemented based on a fast algorithm for active contours and curvature estimation developed by Williams and Shah [39]. The shape of the observed contour is modeled as a single shape approximation to the simple compact 2-D region to fit an ellipse. The center of this ellipse determines the center of ball location in the image plane.

2.4.1.4 3-D Imaging

A perspective transformation (also called imaging transformation) projects 3-D points onto a plane, thus providing an approximation to the manner in which an image is formed by viewing a three-dimensional world. Displacement of a point in the image plane corresponds to its displacement in the real world. This relation can be determined by geometric scene analysis through techniques that map a 3-D scene onto an image plane using a many-to-one transformation. However, a single image point does not uniquely determine the location of a corresponding world point. This missing depth information can be obtained by performing stereoscopic imaging that involves obtaining two separate views of an object of interest [40]. Also known as binocular vision, stereopsis is the system where two cameras are spaced at a distance apart and the apparent change in distance between two points on two successive images is computed using triangulation [36]. This change is known as disparity and yields three-dimensional motion from two-dimensional information. A method for finding a straight line and plane correspondences in stereopair images based on image analysis [3], three-dimensional scene analysis [2] and matching object features using relational tables from
stereo image [7] is explained in [41]. A computer algorithm for reconstructing a scene from a correlated pair of perspective projections is formulated in [42].

Ping pong playing robot systems described in [22, 23, 24, 25, 26] employ stereo imaging using two or more cameras. The system developed in [28] estimates the 3-D ball location using triangulation between ball coordinates and its shadow, through color tracking. In this system development, calibration of a single camera based on the geometry of the image plane and analysis of the image plane co-ordinates, is used to determine the location of ball center in the 3-D world, relative to its size and position in the image plane. Further, processor architecture is utilized to compute the time for change in ball location and additionally maintain synchronization between the frame capture rate and system processing time. Thus, ball center \((x, y, z, t)\) is computed at the end of each frame illustrating the peripheral, attentive and cognitive phases of motion perception.

2.4.2 Trajectory Analysis

In physics, motion means a change in the position of a body with respect to time, as measured by a particular observer in a particular frame of reference. Thus motion is the act or process of changing place or position and is defined in the proportion of space to time. In other words, the properties of space and time determine the nature of motion and the properties of motion, in turn, determine the nature of force. Therefore, relative space and relative time result in the relative motion. This thesis utilizes ball position relative to each frame to compute its displacement in respective directions. The rate of change of this displacement is measured as the ball's velocity and is further utilized to
analyze the ball motion in time. The time and position for first and second bounce are estimated in two individual free flight trajectories. A ball in flight is acted upon by forces of gravity, air drag, and spin [43]. In this work, the trajectory problem is modeled using projectile physics in which the horizontal and vertical motion are independent of each other. Due to absence of acceleration in the horizontal direction, the velocity in horizontal direction remains unchanged [44]. The velocity in vertical direction changes due to free fall acceleration. An experimental value of coefficient of restitution determines the change in velocity of the ball after the first bounce. This new velocity is used to estimate the second trajectory. Since the robot has to hit the ball before it bounces a second time, a target point relative to the robot’s workspace in \((x, y, z)\) is computed based on this estimated trajectory. The robot illustrates motion towards the ball, if the target point within its workspace.

2.4.3 Expert controller

According to Andersson [22], an expert controller is the juxtaposition of an expert system and a robot controller and is a technique to increase the intelligence of the robot controller, while maintaining real time response. The essential features of an expert controller are exploitation of task redundancy, continual integration of new sensor data, regular improvement of task planning, fast execution time, accuracy based on physics models, flexible internal architecture, and robustness to failure. Table 2-3 outlines the different robots along with their controller architecture, used in various ping pong playing systems [22, 23, 24, 25, 27].
<table>
<thead>
<tr>
<th>Place</th>
<th>Degrees of freedom</th>
<th>speed (m/s)</th>
<th>Controller architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT&amp; T Bell Labs, USA, 1986 [22]</td>
<td>Total = 6&lt;br&gt;Used = 5</td>
<td>1.0</td>
<td>Four 32 bit MC68020 and 68881 co-proc.</td>
</tr>
<tr>
<td>Toshiba Research Center, Japan, 1987 [23]</td>
<td>Total = 7&lt;br&gt;Used = 6</td>
<td>10.0</td>
<td>Three 32 bit MC68020 and four 16 bit MC 68000.</td>
</tr>
<tr>
<td>Univ. of Wollongong, Australia. 1993 [25]</td>
<td>Total = 2&lt;br&gt;Used = 2</td>
<td>not given</td>
<td>16 bit Transputer architecture based TRAM.</td>
</tr>
<tr>
<td>University of La Laguna, Spain, 2003 [27]</td>
<td>Total = 5&lt;br&gt;Used = 5</td>
<td>2.2</td>
<td>32 bit Intel Pentium II, MMX set.</td>
</tr>
</tbody>
</table>

In this system, a 64 bit Intel Pentium IV processor, which has a clock frequency of 64 giga hertz, performs numerical processing to calculate the distance by which to lead the ball, such that robot can arrive at that position at the same time as the ball. This defines the robot’s object retrieval rate. Ping pong requires five degrees of freedom in form of the ball positions along the trajectory, angle at which the ball should be hit and
orientation of the paddle [22]. The robot used in this system is a commercial Mitsubishi industrial robotic arm (MIR), RV 2AJ with five degrees of freedom and an operating speed of 640 mm/sec. In this work, robot motion is an integration of the five joint movements, which ensure that the ball lands on table.

In a human ping pong game, a player tracks the ball continuously and hits it at a suitable position and angle after several trials and attempts. The goal of this thesis is to derive a mathematical paradigm of the human game and apply the strategic skills in a robot, through effective use of computer vision. Analogous to the ping pong playing robot systems developed in [22, 23, 24, 25, 27], in this system development, the ball is tracked in 3-D through DFD employing a single camera for color independence. Additionally, the system is implemented utilizing simple controller architecture. The system configuration is featured in the next chapter.
3. System Configuration

3.1 Introduction

The ping-pong robot system developed in this work consists of two main hardware units.

- Multimedia system.
- Five degree of freedom robotic arm.

Figure 3-1 shows the ping-pong system.

![Ping-pong system diagram]

**Figure 3-1 Ping-pong system**

Forthcoming sections explain individual configuration in detail.
3.2 Multimedia system

The multimedia system consists of two components

- Workstation
- USB [Universal Serial Bus] operated computer camera

3.2.1 Workstation

The workstation used is Dell Precision 350 that runs on an Intel Pentium 4 processor. Technical specifications about the workstation are cited in [46]. In computer technology, there are several types of interfaces (methods of interaction) [47]

- User interface: the keyboard, mouse, menus of a computer system. The user interface allows the user to communicate with the operating system. A program interface that takes advantage of the computer's graphics capabilities to make the program easier to use is a graphical user interface (GUI).
- Software interface: the languages and codes that the applications use to communicate with each other and with the hardware.
- Hardware interface: the wires, plugs and sockets that hardware devices use to communicate with each other.

Operating systems perform basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices. Operating systems can be classified as follows:

- Multi-user: Allows two or more users to run programs at the same time. Some operating systems permit hundreds or even thousands of concurrent users.
- Multiprocessing: Supports running a program on more than one processor.
- Multitasking: Allows more than one program to run concurrently.
- Multithreading: Allows different parts of a single program to run concurrently.

POSIX (portable operating system interface for UNIX), is a set of IEEE and ISO standards that define an interface between programs and operating systems and is currently maintained by PASC (portable applications standard committee), an arm of the IEEE. The operating system used in this work, is LINUX, a freely-distributable open source operating system that runs on a number of hardware platforms the system is implemented using a POSIX 1003.1c thread based GNU compiler (version3.3.3) on a Fedora Core 2 kernel (2.6.7 1.494.2.2) for pipelined processing.

A port is an interface on a computer to which a device can be connected. Almost all personal computers come with a serial (one bit at a time transfer) RS - 232C port or RS 422 port for connecting a modem or mouse and a parallel (concurrent transfer of multiple bits) port (25-pin connector interface) for connecting a printer. The workstation communicates with the robot arm through a RS - 232C port. USB is an external bus standard that supports data transfer rates of 12Mbps. A single USB port can be used to connect up to 127 peripheral devices, such as mice, modems, and keyboards. The camera used in this system is USB operative.

The camera captures frames at a regularly sampled rate and this data is transferred to the work station through USB to generate a visual analyzed feedback. This feedback is provided to Picasso through the programmed data via the serial port.
3.2.2 Camera

Charged couple device (CCD) cameras are an important source of geometric vision [48]. Logitech quick cam pro 4000 with CCD based sensor technology, is used to acquire the visual data from the environment. This camera has an in built compatibility for interface with Windows platform and the technical specifications are mentioned in [49]. For compatibility with Linux, an individual driver based application interface with video for Linux (V4L) is developed. The current interface enables the camera to capture 15 frames per second at VGA 640×480 resolution. Analog to digital conversion is synchronized with the CCD pixel clock. These motion samples are further processed for tracking the ball and the processed information is provided as effective visual data to the robot. The next section explains the configuration of the robot.

3.3 Robot

The robotic arm used, Picasso, is a MIR arm RV 2AJ. Degree of freedom (DOF) is a measure of variability, which expresses the number of options available within a variable or space. In a system with n states, the degree of freedom is n. Thus in engineering terms, degree of freedom describes flexibility of motion. A mechanism that has complete freedom of motion (even though limited area or envelope) has six degrees of freedom. Three modes are translation; ability to move in each of the three dimensions and other three are rotation; the ability to change angle around the three perpendicular axes. Picasso is a vertical five-axis multiple-jointed type robot. RV 2AJ has the following degrees of freedom.
- **Translation (3 DOF):** The robot arm exhibits translation in each of the three dimensions $X, Y, Z$ and the corresponding axes are $J_1, J_2, J_3$.

- **Rotation (2 DOF):** The arm demonstrates angular movement about axes $J_5$ and $J_6$. $J_4$ axis does not exist for RV 2AJ model. Thus orientation can be specified only in $J_5$ and $J_6$ axes as shown in Figure 3-2.

In total, it has five degrees of freedom and can rotate almost 180 degrees at its base. A pneumatic air gripper is attached to the robot arm and the ping-pong paddle is assembled to the gripper with proper mechanical interface. Figure 3-2 shows the various parts in the robotic arm.

Figure 3-2 Robot parts and co-ordinates
Picasso can be controlled manually by the control panel, or through the computer programs. From the work station, it can be controlled using C/C++, Linux shell scripting, or any other language that can communicate through the serial port. The interface of the robot is designed to accept only ASCII characters from the serial port. In this thesis, the robot motion is programmed through a serial interface developed using C programming. Robots are moved relative to different coordinate frames; in each type of coordinate frame, the motions are different [45]. A universal frame is defined by the major axes as shown in Figure 3-3.

![Figure 3-3 Major axes of the robot](image)

Real time ball tracking in three dimensions of the real world coordinates is performed with reference to the universal frame and the robot’s motion is programmed, relative to this reference frame. The robot action is synchronized in time with feedback provided by the vision system.

Chapter 4 explains the system development along with the experimental set up for the ping pong playing robot system. Equivalent frame transformations for 3-D imaging using single camera are also formulated in this chapter.
4. System Implementation

4.1 Introduction

Building a system requires a balance between algorithmic requirements and available resources. This chapter of the thesis documents a description of techniques and procedures used by the system configuration to maintain the structural flow of computer vision and use it as a means of communication with the robot. Time and position data transfer enables it to hit the ball at a specified target. Synoptically, this chapter describes the system implementation that entails the algorithm design and development. This consists of engineering and signal processing formulae for simulating the system as well as its real time functionality.

4.2 System Design

In order to avoid a mismatch between user expectations and the ability to realize a computable system, it is important to specify distinctive steps [18]. Subsequent sections explain specific computer vision techniques used to gather the scene information and their application in realizing the system, as shown in Figure 4-1.

![System block diagram](image-url)

Figure 4-1 System block diagram
4.2.1 Frame acquisition

Frames sampled at a regular interval of $1/15^{th}$ of a second are captured by the camera and read as images through the acquisition architecture described in Figure 4-2 below.

![Acquisition data flow](image)

**Figure 4-2 Acquisition data flow**

The set of colors (color space) for the video sequences captured by the camera used in this system design, are represented mathematically using the $YUV$ color space format, where $Y$ symbolizes luma (luminance) information and $(U, V)$ symbolize chroma (chrominance) information. This format enables transfer of efficient quality real world images. However, V4L (the original video capture/overlay application programming interface of the Linux kernel) uses the $RGB$ color space format. Also, $RGB$ color space is the most prevalent choice for graphic buffers because color cathode ray tubes (CRT) use red, green and blue (primary colors) phosphors to create the desired color. For digital $RGB$ values with a range of 0 to 255, $Y$ has a range of 0 to 255, $U$ has a range of 0 to ±112 and $V$ has a range of 0 to ±157. To simplify implementation, another color space $YCbCr$ is used. This is a scaled and offset version of the $YUV$ color space. $Y$ is defined to have a nominal value range of 16 to 235; $C_b$ and $C_r$ are defined to have a range of 16 to 240, with 128 equal to zero. The sampling format used is
4:2:0 orthogonal sampling with a 2:1 reduction of \( C_b \) and \( C_r \) in both the horizontal and vertical directions of scan lines. Thus, an appropriate conversion to the \( RGB \) color space is performed using the following equations [50].

\[
R = 1.164(Y \cdot 16) - 1.596(C_r \cdot 128) \quad (4-1)
\]

\[
G = 1.164(Y \cdot 16) - 0.813(C_r \cdot 128) - 0.392(C_b \cdot 128) \quad (4-2)
\]

\[
B = 1.164(Y \cdot 16) - 2.017(C_b \cdot 128) \quad (4-3)
\]

### 4.2.2 Motion and Ball detection

#### 4.2.2.1 Video and motion

From the classification of signals, as shown in Table 2-1, video refers to the pictorial (visual) information, including still images and time-varying images. A still image has spatial intensity distribution, which is constant with respect to time, and a time-varying image has a varying spatio-temporal intensity pattern. A video signal is represented by a time sequence of still-frame images (pictures) as shown in Figure 4-3.

![Figure 4-3 Video as a sequence of frames](image)

Motion estimation may refer to image-plane motion (2-D motion) or object-motion (3-D motion) estimation. 2-D motion refers to the orthographic or perspective
projection of the 3-D motion onto the image plane because time-varying images are 2-D projections of 3-D scenes. Object motion includes the 3-D motion of the objects in the scene as well as 3-D motion of the camera such as zooming and panning [51].

Occlusion refers to the covering uncovering of a surface due to the 3-D motion of the object. In order to track the 3-D ball motion until the target point, it is necessary to detect the ball in each frame. Motion is a powerful cue used by humans to extract objects of interest from a background of irrelevant detail. In imaging applications, motion arises from a relative displacement between the sensing system and the scene being viewed such as in robotic applications, autonomous navigation and dynamic scene analysis [15]. In this algorithm, the motion information is used to segment the ball from the background.

4.2.2.2 Spatial and temporal redundancy

Digital video is video information that is stored and transmitted in digital form. Digital video compression techniques have played an important role in the world of telecommunication and multimedia systems where bandwidth is still a valuable commodity. Hence, video coding techniques are of prime importance for reducing the amount of information needed for a picture sequence without losing much of quality, judged by human viewers. The statistical analysis of video signals indicates that there is a strong correlation both between successive picture frames and within the picture elements themselves [52]. A still image, or a single frame within a video sequence, contains a significant amount of spatial redundancy. Hence, for coding still images, only the spatial
correlation is exploited. Such a coding technique is called *Intraframe* coding and is the basis for JPEG coding. A moving video sequence contains temporal redundancy (i.e., successive frames of video are usually very similar [53]. If temporal correlation is exploited, it is called *Interframe* coding; this predictive coding is the main coding principle in all standard video codecs such as H.261, H.263, MPEG-1, 2, and 4.

A simple method for redundancy reduction is to predict the value of pixels based on the values previously coded, and code the prediction error. This method is called Differential Pulse Code Modulation (*DPCM*), where the differences between the incoming pixels from the predictions in the predictor are quantized and coded for transmission as illustrated by the figure below.

**Figure 4-4 DPCM: Redundancy reduction**

Various other mapping techniques like transform coding are utilized to remove spatial redundancies for low bit rate video coding. For static parts of the image sequence,
temporal differences will be close to zero, and hence are not coded. Those parts which change between the frames, either due to illumination variation or motion of objects, result in significant image error which needs to be coded. By using the differences between successive images, temporal redundancy is reduced. The interframe error can be substantially reduced by motion compensation. For further bit rate reduction, the transform coefficients and the co-ordinates of the motion vectors are variable length coded (VLC).

Thus, the three fundamental redundancy reduction principles employed in video communication are:

- Spatial redundancy reduction: to reduce spatial redundancy among the pixels within a picture (similarity of pixels, within the frames)
- Temporal redundancy reduction: to remove similarities between the successive pictures, by coding their differences.
- Entropy coding: to reduce the redundancy between the compressed data symbols, using variable length coding techniques.

In this thesis, correlation in the direction of ball motion in each frame is utilized to compute the visual feedback information for the robot to play ping-pong. Based on the principles for interframe and intraframe video coding, this correlation is determined by exploiting the redundancy in the spatial and temporal domain. Motion is sensed as a function of reduction in temporal redundancy and ball is detected in this motion segmented region by utilizing the correspondence in the pixel values.
4.2.2.3 Motion sensing

Reducing the amount of data to be processed without eliminating the essential information is the key to real time image processing [11]. Hence, motion is sensed in only partial area of the complete scene difference. Since the object of interest is the ball and the ball motion is over the table, this area corresponds to the region enveloped by the table. This is illustrated in Figure 4-5.

![Figure 4-5 Motion detection region in DFD](image)

The camera is positioned to acquire the frame in a perspective fashion. Hence, in order to maintain the perspective linearity of the scene captured with the corresponding data in the frame buffer, it is important to compute the slope determining the scan resolution for sensing motion from the image plane resolution. DFD scan is performed
row wise. For each row beginning with \(\text{row}_{\text{start}}\), a multiple of the slope \(\frac{y}{x}\) determines the equivalent column numbers to be processed. An integer type cast normalizes this multiple to obtain an integer value. Following equations are used to compute the column values for each row being scanned.

\[
c_{\text{start}} = \text{col}_{\text{start}} - \text{int} \left[ \left( \frac{y}{x} \right) \times (\text{row} - \text{row}_{\text{start}}) \right]
\]

\[
c_{\text{end}} = \text{col}_{\text{end}} + \text{int} \left[ \left( \frac{y}{x} \right) \times (\text{row} - \text{row}_{\text{start}}) \right]
\]

Figure 4-6 illustrates the sensing of motion eventuating from DFD.
Absolute difference between two successive frames is used to sense motion in the scene through the change in intensity values. This difference is computed using equation (4-6).

\[
dfd(row, col) = \text{abs}[\text{frame}_1(row, col) - \text{frame}_2(row, col)]
\]  

(4-6)

Once motion is sensed, frame1 is indexed as the base frame for comparison with successive frames captured and these frames are indexed as the reference frames. If the computational difference is zero, frame2 replaces the frame1 and a new frame is captured and compared, for sensing the changes in the scene. If changes in the intensity values are observed, motion is sensed and the image is further analyzed for ball detection.

Figure 4-7 is an illustrative example of changes detected in scene through DFD.

![Base frame](image1)
![Reference frame](image2)

DFD

**Figure 4-7** DFD for detecting changes in scene
As shown in Figure 4-7, motion is sensed as the changes in two frames captured every $1/15^{th}$ of a second. These changes represent the ball motion and the player motion in serving the ball. The next step is to segment the ball motion from player motion and recognize the ball location in image plane.

4.2.2.4 Ball detection

The intensity values in an image can be represented by a characteristic function $b(x, y)$ that is zero for all image points corresponding to the background and one for points on the objects. This function forms a binary image and an object's region can be further identified as the region corresponding to the brighter area in the image [13]. Using the DFD, changes in a specific area of the scene are detected through change in the brightness values of the pixels. Further, a global threshold representing the sum of RGB values of each pixel, segments motion identified region from the background, and represents the image in binary form (1 bit/sample). This global thresholding operation defines the characteristic function to be one where the brightness value of each pixel is greater than the threshold value and zero vice versa. Pixels having intensity values greater than the threshold are motion identified and exhibit the capacity of being the ball region. The ball region is now known and an exact area has to be recognized.

In order to identify an object region precisely, a map-like representation called the primal sketch is obtained from the images [54]. This consists of image features labeled with their property values. These maps are further symbolized by abstract relational structures in which, e.g., nodes represent regions, labeled with various property values
(color, texture, shape, etc.), and arcs represent relationship amongst regions [3]. In computer vision, image features can be classified as

- Global: representing average grey level, the area in pixel

- Local: a part of image with some special properties, for instance a circle, line or a textured region in an intensity image [55].

In this work, feature extraction based on contour tracing of the boundary edges is used to recognize the exact ball region amongst the motion identified region.

Edges are pixels at or around which the image values undergo a sharp variation. A simple model for an edge in an image is a straight line separating the two regions of constant brightness [13]. Lesher and Mingolla in [37], experimentally measure the dependability of edge type and line-end inducing elements in illusory contour formation on the human retina. This concept is applied to camera vision, where contours are used to represent a set of points which are determined to lie on an object’s edge, since geometrically; edges are the projection of object boundaries. Thus, in order to segment the ball region precisely, contour tracing is performed on the binary image. The fact that a region has shape and position properties as well as statistical properties of the gray levels of the pixels in the region [17] is employed to fit a curve of arbitrary shape to the set of image points on the identified region. This curve is called an active contour, or snake or deformable contour and is based on the association of an energy function to each possible contour shape, in such a way that the image contour to be detected corresponds to a minimum of the energy function. Contour continuity and smoothness is established using the greedy algorithm described in [39, 55], where a connected
component labeling operation is performed to compress horizontal, vertical and diagonal components and the end points are extracted using a two step approach.

- Energy minimization:

  The energy function for a contour \( c = c(s) \), parameterized by its arc length \( s \) is minimized by minimizing the following function

  \[
  E = \int \left( \alpha(s)E_{\text{cont}} + \beta(s)E_{\text{curv}} + \gamma(s)E_{\text{image}} \right) ds
  \]  

  (4-7)

  where, \( E_{\text{cont}} \) (continuity of the contour) and \( E_{\text{curv}} \) (smoothness of the contour) constitute the internal energy and \( E_{\text{image}} \) (edge attraction) constitutes the external energy. \( \alpha, \beta, \gamma \) control the relative influence of the corresponding energy term and can vary along \( c \). For a chain of \( N \) image points, \( p_1, \ldots, p_N \),

  \[
  E_{\text{cont}} = \left( \bar{d} - \| p_i - p_{i-1} \| \right)^2
  \]  

  (4-8)

  \[
  E_{\text{curv}} = \| p_i - 2p_i + p_{i+1} \|^2
  \]  

  (4-9)

  \[
  E_{\text{image}} = -\| \nabla I \|
  \]  

  (4-10)

  where, \( \bar{d} \) is the average distance between the pairs \( (p_i, p_{i-1}) \) and \( \nabla I \) is the spatial gradient of the intensity image \( I(x, y) \), computed as

  \[
  \nabla I = [F_x, F_y] = \frac{\partial F}{\partial x} \hat{\imath} + \frac{\partial F}{\partial y}
  \]  

  (4-11)
Thus, $E_{\text{image}}$ becomes very small (negative) wherever the norm of the spatial gradient is large i.e. near the edges, making $E$ small and attracting the snake towards the image contours.

- Corner elimination:

  In the second step, the greedy algorithm searches for corners as a curvature maxima along the contour by controlling $E_{\text{curv}}$. For instance, if a curvature maximum is found at point $p_j$, $\beta_j$ is set to zero [55].

  A contour scan performing these two steps, is run throughout the binary image and based on a local threshold characterizing the size of the ball, the bounding contour of the ball is identified using this traversing process. Figure 4-8 demonstrates the segmentation of exact ball region from the changes detected in the scene.

![Figure 4-8 Ball detection via contour tracing](a) Changes in the scene  
(b) Ball region extraction

Additionally, points along the ball contour are utilized to compute the silhouette of the ball region, as shown in Figure 4-9.
Object recognition involves recovering information about its reflectance and shape [13]. In this thesis, reflectance information is utilized to gain knowledge about the presence of a ball in the scene and a region dependent threshold based on the retrieved contour points is further applied to determine the ball silhouette in the image plane. If there is not a single contour with equivalent points within the specified range, a new frame is captured for detecting ball motion. Consequently, the two-step approach is utilized to form contours throughout the image and detect the exact ball region along with other noise and motion contours. The next step is to determine the location of the ball in 2–D of the image plane.

4.2.3 Ball location in the image plane

In binary images, a region’s position can be defined by the center of its area [13]. Duda and Hart considered the use of the Hough transform, to detect complex
patterns of points in binary images. This technique, named after Paul Hough who patented the technique in 1962, votes for all combinations of parameters which may have produced if it were part of the target curve [56]. The Hough transform is performed between Cartesian space and some parameter space in which the straight line (or other boundary formulation) can be defined. Since the edge image is converted to a bi-level form for thresholding points belonging to a boundary, this technique is fine for images with strong contrast, but reduces its applicability for shaded images. The fact that each point in the edge map votes for more than one line in the parameter space creates a false peak in the histogram, when there are several lines in the image [57]. A rapid increase of the search time results with the number of parameters in the curve's representation. Also, line detection can be disturbed by low-curvature circles in form of spurious peaks [55]. In order to locate the center of the ball contour, a best curve interpolating the points on the contour needs to be modeled. Figure 4-10 illustrates that the image of a 3-D circle, as viewed from the top, is an ellipse.

(a) Circle in 2-D plane  (b) Circle in 3-D plane

Figure 4-10 Circle-ellipse similarity
Figure 4-11 represents an arbitrary point on a circle and an ellipse from the direction perpendicular to the planes.

As illustrated above, the plane of the ellipse intersects the plane of the circle along the Y axis; hence the Y coordinate \( n \) equals \( j \) and \( b = r \). The constant \( a \) of the ellipse is given by

\[
a = \left( \frac{r}{\cos(\varphi)} \right)
\]  

(4-12)

where, \( \varphi \) is the tilt between two planes as shown in the figure below,

An ellipse that is not rotated or translated and has its center at the origin, satisfies

\[
\left( \frac{x^2}{a^2} \right) + \left( \frac{y^2}{b^2} \right) = 1
\]  

(4-13)
Substituting \((m, n)\) for \((x, y)\) and \(b = r\),

\[
\left(\frac{m^2}{a^2}\right) + \left(\frac{n^2}{r^2}\right) = 1
\]  

(4-14)

From figure 4-12, \(m = i/\cos(\varphi)\); substituting this along with equation (4-13), in equation (4-15),

\[
\left(\frac{i^2}{r^2}\right) + \left(\frac{n^2}{r^2}\right) = 1
\]  

(4-15)

Substituting \(n = j\) and multiplying by \(r^2\)

\[i^2 + j^2 = 1\]  

(4-16)

This is equivalent to the equation of a circle in canonical form. Thus, an arbitrary ellipse is understood in terms of the operations transforming a unit radius circle centered at origin to an ellipse having a specified major axis length, minor axis length, orientation and position [17]. Since, the ball is circular in shape, the points on the ball contour are used as sequences to fit an ellipse as a single shape approximation to the compact 2-D region formed by the ball contour. An ellipse \((x_0, y_0)\) matching the ball silhouette satisfies the equation

\[
\begin{bmatrix}
  a & b/2 \\
  b/2 & c
\end{bmatrix}
\begin{bmatrix}
x_0 \\
y_0
\end{bmatrix}
+ \begin{bmatrix}
d/2 \\
e/2
\end{bmatrix} = \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]  

(4-17)

\(a, b, c, d, e\) are the Eigen vector components determining the length of the major axis, minor axis and orientation of ellipse [17]. Thus, the topological properties of objects in a scene are used to locate the ball center in image plane. In 3-D computer vision, the process of image feature extraction not only obtains the 2-D maps, but also utilizes them
for further application. 2D location of the ball in the image plane is utilized to track the ball in 3D through camera calibration.

4.2.4 Camera calibration and 3-D Imaging

Stereo vision refers to the ability to infer information on the 3D structure and distance of a scene from two or more images taken from different view points [55]. Computer vision establishes the geometrical correspondence between 3D location of points in a scene and their position in the image plane with the aid of stereopsis, discussed in Chapter 2. From a computational standpoint, a stereo system must solve two problems:

- Correspondence: Identification of part of the left and right images as projections of the same scene element.
- Reconstruction: 3D location and structure of the objects in the scene through the geometry of the stereo system.

Camera calibration is a technique that utilizes the intrinsic and extrinsic parameters of the camera to locate an object in 3D space [55]. The system developed in this work, makes use of camera calibrated sensory information to reconstruct 3D information of the ball, based on the image plane data. The correspondence problem is eliminated through the use of a single camera. Motion accuracy is dependent on camera calibration due to the use of a single camera. Hence, camera calibration is employed as a function of position, orientation and optical characteristics of the camera.
4.2.4.1 Intrinsic camera calibration:

The intrinsic parameters of a camera are the parameters necessary to link the pixel co-ordinates of an image point with the corresponding coordinates of the camera. These parameters constitute the optical, geometric and digital characteristics of the camera.

- **Focal length:**

For a computer camera, the focal length is indefinite. However, an object’s size increases as it approaches towards the camera. Shape approximation using ellipse provides the ball location and size in the image plane. The fact that size of an object is inversely proportional to its distance from image plane is utilized to compute the distance of the ball (focal length) from the focal point as a function of ball size in pixels. Figure 4-13 illustrates the computation of the distance of the ball from the camera focus, based on a series of experiments.

![Figure 4-13 Focal length of camera](image-url)
An exponential curve is a best fit to determine the ball distance $Z_b$ (mm) along line of focus of the camera. This curve is characterized by the following equation:

$$y = 22094 \cdot x^{-0.8966}$$  \hspace{1cm} (4-18)

- **Field of view:**

Each frame is acquired at a VGA resolution, thus providing an aspect ratio of 4:3. Thus, the horizontal resolution and the vertical resolution of the camera are different. Thus, the camera observes the centers in the horizontal plane at an angle ($\alpha$) and vertical plane at an angle ($\beta$). An experiment that determines these angles as a function of the distance of the camera from the center of the image plane is performed. Further, modeling based using an exponential curve provides a generic formula to compute the angles for a ball detected in each frame during real time implementation. Figure 4-14 demonstrates this computation.
Angular field of view for horizontal resolution

\[ y = 22.746e^{-0.0001x} \]

(a) Horizontal field of view resolution

Angular field of view for vertical resolution

\[ y = 17.514e^{-0.05x} \]

(b) Vertical field of view resolution

Figure 4-14 Field of view measurement
Figure 4-15 illustrates transformation from pixel units to real world units (mm) in the image plane as a function of $Z_b$, whose value is computed using equation (4-18).

**Figure 4-15 Pixel to millimeter transformation**

Experiment based values of $\alpha$ and $\beta$ are used to map pixel values $(X_i, Y_i)$ to physical resolution values $(F_b(x), F_b(y))$ using following equations

\[
X_i = (Z_b \cdot \tan(\alpha)), Y_i = (Z_b \cdot \tan(\beta)) \tag{4-19}
\]

\[
F_b(x) = \left(\frac{F_p(x) \cdot X_i}{\text{col}/2}\right), F_b(y) = \left(\frac{F_p(y) \cdot Y_i}{\text{row}/2}\right) \tag{4-20}
\]

Thus, intrinsic camera calibration based on focal length and field of view is utilized to locate the ball in real world units on the image plane.
4.2.4.2 Extrinsic camera calibration:

The extrinsic camera parameters are the parameters that define the location and orientation of the camera reference frame with respect to a world reference frame (universal frame). The transformation between camera and the world reference frame is based on:

- A pure translation: describing relative positions of the origin and representing movement without a change in its orientation.
- A pure rotation: describing the alignment of frame axes and representing movement about an axis.
- A combination of translations and/or rotations.

Since the directional vectors do not change in a pure translation, the transformation $T$ is

$$ T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 4} \tag{4-21} $$

where $d_x, d_y, d_z$ are the three components of a pure translation vector $\vec{d}$, relative to the $x, y, z$ axes of the reference frame.

Rotations about $x, y, z$ axes of the frame are given by following equations

$$ Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 4} \tag{4-22} $$
\[
Rot(y, \theta) = \begin{bmatrix}
\cos(\theta) & 0 & \sin(\theta) & 0 \\
0 & 1 & 0 & 0 \\
-\sin(\theta) & 0 & \cos(\theta) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_{4 \times 4}
\]

\[
Rot(z, \theta) = \begin{bmatrix}
\cos(\theta) & -\sin(\theta) & 0 & 0 \\
\sin(\theta) & \cos(\theta) & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_{4 \times 4}
\]

where \( \theta \) is the angle about which the rotation occurs [45].

A geometric model of the scene based on reference coordinates of universe frame \( U(x, y, z) \) based on the robot linear axes, ball frame \( B(x, y, z) \) camera frame \( C(x, y, z) \), real image frame \( F_p(x, y) \) and pixel image frame \( I(row, col) \) is developed as shown in Figure 4-16.
A perspective image of the world scene is acquired by the camera as its optical axis is perpendicular to the image plane. Alignment of width centers of this image frame $I(0, \text{col}/2)$ and $I(\text{row}, \text{col}/2)$ with universal axis $U_x$ ensures a zero degree pan in the image plane. Real image frame coordinates $F_p(x, y)$ and ball dimensions are computed from pixel information in that frame. Geometric analysis of the scene is used to devise the transformations for 3-D camera vision. Based on the right hand rule, subsequent formulae are derived linear and rotational matrices for single camera vision, determining position and orientation of ball in space, from universal point $U$ as shown in as illustrated in Figure 4-16. Also, rotations are performed about the angle $\theta$ which is a measure of the tilt of the camera and is computed as

$$\theta = \left( \frac{x_{center} + x_{camera}}{z_{camera}} \right)$$

(4-25)

$$T^U_C = \begin{bmatrix} 1 & 0 & 0 & -x_{camera} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z_{camera} \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 4}$$

(4-26)

$$T^{C}_{fo} = \begin{bmatrix} \cos(-\theta) & 0 & \sin(-\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\theta) & 0 & \cos(-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 4} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 4}$$

(4-27)

$$T^{f_0}_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -Zb \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 4}$$

(4-28)
\[
T_I^F = \begin{bmatrix}
\cos(-90) & -\sin(-90) & 0 & 0 \\
\sin(-90) & \cos(-90) & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}_{4\times4} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}_{4\times4}
\]

(4-29)

\(T_C^U\) represents a transformation from world reference point to camera through a combination of translations along \(x\_\text{camera}\) and \(z\_\text{camera}\). \(T_{f\theta}^C\) represents the camera tilt for a perspective view of the scene through rotation about \(\theta\). \(T_{f\theta}^o\) symbolizes linear translation along the focal axis of the camera. This transformation maps the camera axis to image plane \(I\) in physical measurements. \(T_I^F\) is the transformation required to align the image plane axes with the frame axes. Thus, the transformations from the universal point \(U\) to a frame represented in pixels is computed as

\[
T_I^U = \text{Trans}(-x\_\text{camera},0,z\_\text{camera})\ast \text{Rot}(y,-\theta)\ast \text{Trans}(0,0,-Z_b)\ast \text{Rot}(z,-90)
\]

(4-30)

\[
T_I^U = T_C^U \ast T_{f\theta}^C \ast T_{f\theta}^o \ast T_I^F = \begin{bmatrix}
0 & \cos(\theta) & -\sin(\theta) & (Z_b \ast \sin(\theta) - x\_\text{camera}) \\
-1 & 0 & 0 & 0 \\
0 & \sin(\theta) & \cos(\theta) & -(Z_b \ast \cos(\theta) + z\_\text{camera}) \\
0 & 0 & 0 & 1
\end{bmatrix}_{4\times4}
\]

(4-31)

The ball coordinates in real world measurements are computed using equation (4-15) and are represented as a transformation from frame coordinates using the following equation

\[
T_B^F = \begin{bmatrix}
F_b(x) \\
F_b(y) \\
0 \\
1
\end{bmatrix}_{4\times1}
\]

(4-32)
Thus, the transformation from universal point to 3-D ball position in the real world is calculated as

\[
T_B^U \begin{bmatrix} 4x1 \\ \end{bmatrix} = T_F^U \begin{bmatrix} 4x4 \\ \end{bmatrix} * T_B^F \begin{bmatrix} 4x1 \end{bmatrix}
\] (4-33)

Multiplying equations (4-31) and (4-32), we get,

\[
T_B^U = \begin{bmatrix} B_x \\ B_y \\ B_z \\ 1 \end{bmatrix}_{4x1} = \begin{bmatrix} (F_b(y)\cos(\theta))+(Z_b\sin(\theta)) - x_{camera} \\ -F_b(x) \\ (F_b(y)\sin(\theta))-(Z_b\cos(\theta))+z_{camera} \\ 1 \end{bmatrix}_{4x1}
\] (4-34)

\(B_x, B_y, B_z\) are ball centers in respective coordinates, tracked from the universal point \(U_x, U_y, U_z\). Thus, parametric calibration in pan, tilt, focus, and position, proportional to the topological parameters of the scene and ball, enable locating the ball in 3-D from image plane.

### 4.2.5 Motion Estimation

The motion of the ball is analyzed as a proportion of the displacement of the ball to the time interval between each frame. Section 4.2.4.1 explains the concepts of displacement and velocity and Section 4.2.4.2 extends these concepts to measure the motion information of the ball, relative to each frame.

#### 4.2.5.1 Displacement and Velocity

A change from one position \(x_1\) to another position \(x_2\) is called displacement \(\Delta x\), and is computed as

\[
\Delta x = x_2 - x_1
\] (4-35)
Thus displacement is a vector quantity that has both magnitude and direction. The rate of change of displacement is termed velocity. The ratio of displacement $\Delta x$ that occurs during a particular time interval $\Delta t$ to that interval is termed the average velocity $v_{\text{avg}}$ and is given by the following equation.

$$v_{\text{avg}} = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{t_2 - t_1} \quad (4-36)$$

The notation means that the position is $x_1$ at time $t_1$ and $x_2$ at time $t_2$.

The change in an object's velocity with respect to the change in time is called acceleration and is determined as a second derivative of the object's position $x(t)$ with respect to time. Thus in equation form,

$$a = \frac{dv}{dy} = \frac{d}{dt}\left(\frac{dx}{dt}\right) = \frac{d^2x}{dt^2} \quad (4-37)$$

If an object moves in a vertical plane with some initial velocity $\vec{v}_0$, but its acceleration is always the free fall acceleration $\vec{g}$, which is downward, such an object is called a projectile (meaning it is projected or launched) and its motion is called projectile motion [44]. A projectile is any object which once projected continues in motion by its own inertia and is influenced only by the downward force of gravity. Figure 4-17 shows the free body diagram which illustrates that a projectile is any object upon which the only force acting is the force due to gravitational acceleration $\vec{g}$, whose value is constant equal to $9860 \text{mm/s}^2$. In projectile motion, the horizontal motion and the vertical motion are independent of each other, that is, neither motion affects the other [58]. Thus the
horizontal motion has zero acceleration and vertical motion has constant downward acceleration equal to $g$.

![Figure 4-17 Free body diagram of a projectile](image)

The real world coordinates of the ball are reconstructed using the 2-D data as described in Section 4.2.3. This 2-D spatial information of the ball is used to compute the change in position of the ball and velocity of the ball in each direction is obtained based on projectile physics.

### 4.2.5.2 Motion analysis

Motion is evaluated as the spatio-temporal displacement of $B_x, B_y, B_z$ in frame $n$. Velocity is a vector measurement of the rate and direction of motion. The vertical motion is influenced by downward acceleration constant $-g = -9807\, mm/sec^2$ as the ball exhibits projectile motion [44]. Thus the motion of the ball is analyzed using the velocity information and is computed using equations (4-38) – (4-40). $\Delta_t$ is the time interval measured in milliseconds between two successive ball locations and is maintained at
maintained at 0.066 seconds through synchronization between the processor ticks and the CCD clocking rate.

\[ \Delta t = \frac{\text{processor \_ clocks}}{\text{clocks \_ per \_ sec}} (ms) \]  

(4-38)

\[ V_x^n = \frac{\Delta x}{\Delta t} = \frac{B_x^n - B_x^{n-1}}{\Delta t} \]  

(4-39)

\[ V_y^n = \frac{\Delta y}{\Delta t} = \frac{B_y^n - B_y^{n-1}}{\Delta t} \]  

(4-40)

\[ V_z^n = \left[ \frac{B_z^n - B_z^{n-1}}{\Delta t} - \left( \frac{1}{2} g \ast \Delta t \right) \right] = \left[ \frac{\Delta z}{\Delta t} - \left( \frac{1}{2} g \ast \Delta t \right) \right] \]  

(4-41)

Thus, in the working environment of the system, displacements and velocities are computed relative to the three coordinates. Figure 4-18 demonstrates the displacements and velocities in the three directions x, y, z.

![Figure 4-18 Displacement in 3-D](image-url)
4.2.5.3 Trajectory estimation:

Projectile motion equations for trajectory estimation in three dimensions are derived as a function of the velocities in respective co-ordinates with reference to the ball co-ordinates. Following table summarizes the trajectory estimation formulae for one ball position.

**Table 4-1 Trajectory based motion estimation**

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Velocity</th>
<th>Initial position</th>
<th>Estimated position</th>
<th>Motion in the direction of the dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>$V_x$</td>
<td>$X^0$</td>
<td>$X^1$</td>
<td>$X^1 = X^0 + V_{x_0} \cdot t - \frac{1}{2} \cdot a_x \cdot t^2$</td>
</tr>
<tr>
<td>Y</td>
<td>$V_y$</td>
<td>$Y^0$</td>
<td>$Y^1$</td>
<td>$Y^1 = Y^0 + V_{y_0} \cdot t - \frac{1}{2} \cdot a_y \cdot t^2$</td>
</tr>
<tr>
<td>Z</td>
<td>$V_z$</td>
<td>$Z^0$</td>
<td>$Z^1$</td>
<td>$Z^1 = Z^0 + V_{z_0} \cdot t - \frac{1}{2} \cdot a_z \cdot t^2$</td>
</tr>
</tbody>
</table>

In this case, acceleration due to gravity is considered only in the Z direction due to which, the ball tends to bounce on the table. Hence $a_x = a_y = 0$. For the current frame $n$ and initial positions $X^0 Y^0 Z^0$, the revised trajectory estimation equations are

\[
X^{n+1} = X^0 + V_x^n \cdot t \tag{4-42}
\]

\[
Y^{n+1} = Y^0 + V_y^n \cdot t \tag{4-43}
\]

\[
Z^{n+1} = Z^0 + V_z^n \cdot t - \frac{1}{2} \cdot a_z \cdot t^2 \tag{4-44}
\]
Based on equations (4-42) – (4-44), Table 4-2 is the formation of the trajectory matrix that entails the trajectory path computed from the above equations over frames $l$.

**Table 4-2 Trajectory matrix**

<table>
<thead>
<tr>
<th>Frame</th>
<th>$n$</th>
<th>$n-1$</th>
<th>$n+2$</th>
<th>$n-1*t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 1/\Delta t$</td>
<td>0</td>
<td>1/15</td>
<td>2/15</td>
<td>$n-1*t$</td>
</tr>
<tr>
<td>$X$</td>
<td>$X^0$</td>
<td>$X^{n+1}$</td>
<td>$X^{n+2}$</td>
<td>$X^{n+l}$</td>
</tr>
<tr>
<td>$Y$</td>
<td>$y^0$</td>
<td>$y^{n+1}$</td>
<td>$y^{n+2}$</td>
<td>$y^{n+l}$</td>
</tr>
<tr>
<td>$Z$</td>
<td>$z^0$</td>
<td>$z^{n+1}$</td>
<td>$z^{n+2}$</td>
<td>$z^{n+l}$</td>
</tr>
</tbody>
</table>

Subsequent equations for trajectory estimation are derived as a function of the velocities with reference to the co-ordinates of ball centers in each frame.

\[
B_x^{n+1} = B_x^n + V_x^n \Delta t \quad (4-45)
\]

\[
B_y^{n+1} = B_y^n + V_y^n \Delta t \quad (4-46)
\]

\[
B_z^{n+1} = B_z^n + V_z^n \Delta t + \frac{1}{2} g \Delta t^2 \quad (4-47)
\]

**4.2.6 Target point computation**

Since the projectile path of any object is parabolic [44], the trajectory of the ball is modeled as a parabolic function. Individual ball trajectory before a bounce occurs is calculated based on equations (4-45)-(4-47). Comparing equation (4-47) to a form $ax^2 + bx + c$, it is seen that the equation is quadratic in variable $t$. When a bounce occurs, the ball is on the table; hence it has zero elevation from the ground. In this
work, since the ball is tracked at the center, an elevation equal to the radius of the ball (mm) accounts for a bounce. Thus equation (4-47) can be revised as

$$\frac{1}{2} g \Delta t^2 + V_z^n \Delta t + B_z^n = 19.8$$

(4-48)

The time at which the first bounce occurs is computed using the quadratic solution

$$t_{bounce} = \frac{-V_z^n + \sqrt{(V_z^n)^2 - 2g(B_z^n - 19.8)}}{g}$$

(4-49)

This time is used to predict the center location of the ball in \( n_x, n_y \) at the time of bounce, using equations (4-45) and (4-46) respectively.

$$x_{bounce} = B_x^n + V_x^n \cdot t_{bounce}$$

(4-50)

$$y_{bounce} = B_y^n + V_y^n \cdot t_{bounce}$$

(4-51)

After the first impact, change in velocity in the vertical direction is restituted by a dimensionless collision coefficient. This coefficient of restitution is computed based on the change in height of bounce and is illustrated in the figure below.

Figure 4-19 Coefficient of restitution
Based on a series of trials, a value of 0.78 is computed using the formula

\[
e = \frac{\sqrt{2gh}}{\sqrt{2gh}} = \frac{h'}{h'}
\]  

(4-52)

Thus, after the first bounce \( V_z \) is multiplied by 0.78 to balance the ball velocity after bounce. Figure 4-20 illustrates an approximation of trajectory path of the ball.

**Figure 4-20 Motion estimation**

The robot has to hit the ball before the second bounce occurs; therefore time and corresponding \( x, y \) locations for second bounce are also computed using equations (4-49) – (4-51). The ball is tracked for its center location in \( x, y, z \), immediately after the first bounce occurs. The displacement \( x \) is the target for the robot in \( x \). Figure 4-21 illustrates a linear interpolation model developed for the computation of the target point.
Figure 4-21 Target point computation

The hit point for the robot is \( f(x, y, z) \) using

\[
T \arg \{x, y, z\} = ((l \cdot xbounce2), (ybounce2), (h))
\] (4-53)

The time for the ball to reach the impact point is gauged using equation (4-49).

4.2.7 Robot motion

The target point computed using equation (4-53) is specified to the robot through an RS 232C interface. A ping pong paddle is attached to the end gripper of the robot arm. The robot along with the interfaced ping pong paddle is positioned with its tip at a specific elevation from the universal reference position \( U \), from which the ball is tracked. This is the home position for the robot given by

\[
\text{Robot}_{\text{home}} = (x_{\text{home}}, y_{\text{home}}, z_{\text{home}}) = U(X, Y, (Z + \text{robotbase\_to\_endgripper}))
\] (4-54)

where \( X, Y, Z \) correspond to the respective \( J_1, J_2, J_3 \) axes of the MIR arm as shown in Figure 4-22. \( J_1J_2J_3 \) exhibit linear movements along the base, shoulder and the upper arm. \( \text{robotbase\_to\_endgripper} \) is the elevation provided along \( J_3 \) to constrain the contact of the paddle tip with the table. Ping pong requires five degrees of freedom [22].
The remaining two degrees of freedom correspond to the angular motion about the forearm and mechanical interface at the end of the gripper and is demonstrated by $J_4, J_5$ respectively.

![Figure 4-22 Robot motion](image)

A target point in $x, y, z$ is computed using equation (4-53). The robot has to demonstrate motion and hit the ball such that

- center of paddle is in contact with the center of the ball (specified as target)
- ball lands on the table.

$J_1, J_2$ coincide with the $x, y$ locations of the universal point and are the starting point for the end gripper along with the paddle. A linear transformation along $J_3$, computed with reference to the robot base ensures that the paddle center meets the ball center in elevation. Thus the robot observes linear motion in three directions using

$$Robotmotion = Robot_{home} \cdot \{Target(x, y, (z - robotbase\_to\_paddlecenter))\} \quad (4-55)$$

where $robotbase\_to\_paddlecenter$ is the linear distance which confirms a contact with the paddle center along $z$. $J_4, J_5$ specify the angle of roll and pitch joints and are set at a value to ensure the angular momentum of the paddle in such a way that the ball lands on
the table. The orientation of the robot motion is always positive along $J_1$ since the robot moves forward. The orientation of robot motion along $J_2, J_3$ depends on the direction of ball motion in the $Y, Z$ coordinates of the system. The orientation of robot motion about $J_4, J_5$ depends on the direction of the estimated $ybounce2$ to produce an inward movement such that the robot hits the ball to land on the table. A work space is defined based on the range in which the robot can illustrate motion using all five degrees. If the target of the ball is within the workspace of the robot and if a second bounce has not occurred, the robot exhibits motion to hit the ball with the paddle using its five degrees of movement, according to the following equation

$$Robotmotion = Robot_{home} \cdot (Target(x, y, (z - robotbase_to_paddlecenter))a,b)$$

(4-56)

where $a, b$ signify specified angular movements about the roll and pitch joints.

A summary of the experimental procedure for the course of action explained in this chapter is enumerated in the next chapter. Further, analysis results, demonstrate the validity of the ping pong playing robot system.
5. Experimental Procedure and Analysis

5.1 Foreword

This chapter lists the experimental procedure for the current system set up which facilitates the robot to play the sport of ping pong. Furthermore, results indicative of the system’s performance, based on these procedural steps are also presented in this chapter.

5.2 System set up

The system set up is configured based on the individual modus operandi of

- Camera calibration
- Paddle interface and robot positioning
- Software control

5.2.1 Camera calibration

Camera calibrating is performed utilizing the intrinsic parameters; that define the optical characteristics of the camera and extrinsic parameters; that identify the geometry of the scene, viewed by the camera.

5.2.1.1 Intrinsic camera calibration

The parameters computed using intrinsic camera calibration is the focal length that establishes the correspondence between the ball size in pixels and its distance from the image plane on the CCD sensor chip (which is identified physically as the center of the camera) and the angular field of view that determines the resolution conversion.

Subsequent iterations are used to compute the focal length experimentally.
- The ball is positioned and glued to the paper such that the ball center approximately aligns with the center of the paper, which is glued to the table. This ensures the ball to be stationary while performing the experiment and is illustrated as shown in the Figure 5-1 below.

![Figure 5-1 Experimental top view of focal length computation](image)

- Note that the paper is arranged in a landscape fashion. This ascertains the alignment of the image plane axis which provides a graphical resolution of $640 \times 480$, featuring a landscape display.

- The camera is fixed to a stand and faces vertically downwards such that the image plane is perpendicular to the scene. Straightness of the image plane is confirmed by aligning the image frame axis parallel to the table and the paper. The distance of the camera center to the ball center is maximum, as it is easy to slide the camera down for observing changes in the ball size.
The ball diameter is measured using a Vernier calipers and the center is computed as its radius, on a millimeter scale. The distance in mm, between the camera center and the ball center is measured using a level scale. A snapshot of the scene is taken using *camstream*; a multimedia video enhancement tool on Linux system. The ball size in pixels is computed using the GNU image manipulation program (*GIMP*).

The camera position is lowered and a new reading measuring the distance of the ball center from the camera center in mm along with the corresponding ball size in pixels is taken.

This experiment is repeated for a set of approximately 50 readings with a uniform down slide of 30 mm in the camera's position, maintaining the straightness in the image plane. The change in the ball's size is noticed after a certain interval. Figure 5-2 illustrates two instances where changes in the ball size are observed, while performing this experiment.

![Figure 5-2 Experimental observation for change in ball size](image_url)

(a) Distance = 344 mm  
Ball size = 101 pixels

(b) Distance = 180 mm  
Ball size = 222 pixels
A graph of the distance of the ball center (mm) versus the ball size (pixels) is utilized to fit an exponential curve, which determines the relation employed in computing the focal length of the camera and is shown in Figure 5-3.

![Focal length graph](image)

*Figure 5-3 Ball distance vs. size*

The camera frame has an aspect ratio of 4:3; as a result there is difference in resolution along the horizontal and vertical axes of the image plane. An experiment based on the following steps is utilized to compute the geometric correspondence determining the scale which represents the image plane in physical units of measurement.

- A level marked with physical units of measurement (for e.g. yardstick) is placed on the table such that it aligns along the center of the horizontal axis of the image plane. The corresponding area covered by 640 pixels is noted along with the distance from the camera center.
The above step is repeated with the level measure aligned with the center along the vertical axis of the image plane.

This procedure is repeated for a set of readings and the angular correspondence determining the horizontal and vertical resolution in physical units is computed as shown in the figure below:

(a) Resolution in the horizontal plane (column measure)

(b) Resolution in the vertical plane (row measure)

Figure 5-4 Resolution in mm
$Z_d$ is the distance between the camera center and yardstick. A plot between $\alpha$ and $\beta$ with respective values of $Z_d$ is used to fit an exponential curve that characterizes relation between image plane units and physical world units to identify the ball in physical units using equation (4-19) and (4-20). This plot is as shown in Figure 4-14.

5.2.1.2 Extrinsic camera calibration

The goal of extrinsic camera calibration is to locate the center of the image plane in the scene and compute the tilt of view, based on the topological positioning of the camera. The following steps are performed in identifying the center of the image plane.

- The camera is centered on a base plate, which is mounted on a tripod stand, fixed to the table. A tilt of the camera, commensurate with the field of view occupying the scene, enables perspective frame capture. A snapshot is taken and center of the image plane in the scene is recognized based on the center (row, column) in the snapshot.

- The next step is to compute the tilt in the camera based on its distance (height and length) from the center of the image plane in the scene. This is illustrated in Figure 5-5. $x_{\text{center}}$, $x_{\text{camera}}$ and $z_{\text{camera}}$ are physically measured in mm. Accordingly, $x_{\text{imageplane}}$ is computed using the following equation

$$x_{\text{imageplane}} = x_{\text{center}} - x_{\text{camera}}$$

(5-1)

The tilt is further computed as

$$\theta = \arctan\left(\frac{x_{\text{imageplane}}}{z_{\text{camera}}}\right)$$

(5-2)
Further, $Z_d$ is computed based on $\theta$ and corresponding values of $\alpha$ and $\beta$ are calculated.

Thus, the optical characteristics and computation of parameters characterizing the scene topology are the experimental references employed to locate the ball in space.

### 5.2.2 Paddle interface and robot positioning

A pneumatic gripper is connected to the robotic arm. Two aluminum plates $(25 \times 25 \times 5)$ mm are interfaced to the end gripper. Four screws having a thread size of 0.163 inches $\approx 4.1$ mm are used to attach the ping pong paddle to the aluminum plates, which are held by the end gripper. The ping pong paddle measures 160 mm in length and is circular with a radius of 75 mm. The paddle is attached in such a way that the center of
the circular region of the paddle is utilized by the robot arm to hit the ball center. The robot is positioned with the center of the ping pong paddle in line with the reference point and at a certain height above the table to ensure smooth motion in the robot’s work space.

5.2.3 Software control

The camera is interfaced to the work station through a USB and the robot is serial communicative. Camera driver modules are developed to enable 15 frames per second capture rate at VGA resolution. Binary modules of these drivers are encoded and for every reboot of the system, these modules need to be recompiled. In order to synchronize the binary modules of the camera drivers and the developed interface, the command `modprobe pwcx` has to be executed at the command prompt, for every restart of the system. PWGX contains the list of proprietary binary (hex-encoded) modules that are utilized to develop the appropriate interface of the camera on Linux operating system. The robot is controlled through the serial port and is connected to the serial port identified as S0 of the current system. The communication is RS-232C based and a baud rate is to be specified that enables serial transfer of the data. Once the system is restarted, the command `stty -F/dev/ttyS0 9600 parenb` needs to be executed at the command prompt. This command identifies the serial port and the device (robot in our case) connected as RS-232C communicative. It also specifies the baud rate, which regulates the data flow to and from the robot.

Array based algorithmic modules for image math, connected component labeling and estimation are tested using MATLAB® version 7.0.1.24704 (R14) SP1. The system is implemented using a POSIX 1003.1c thread based GNU compiler (version 3.3.3) on a
Fedora Core 2 kernel (version 2.6.7-1.494.2.2) for pipelined processing. Intel® Image Processing Library functions are used as a benchmark for performing unsigned 8-bit image arithmetic. Dynamic memory storage and vector math for graphical user interface are optimized using Intel® Open Source Computer Vision Library for real time performance.

5.3 System performance

The system's functionality is leveraged by implementation of specific steps mentioned in the previous section and reasonable assumptions based on practical values. System latency is the time from acquisition of data until it is applied to the control output [22]. A global timestamp measures this latency as a function of processor ticks. Additionally, various local parameters ensure a smooth flow in executing individual blocks of the system as well as their synthesized performance.

5.3.1 Frame capture rate and processing time

The developed driver interface for the camera facilitates a capture of one frame every 66.66 milliseconds. Since the processing time for each step varies and is less than the capture rate, a delay has to be introduced. This delay acts as a software interrupt for the processor clock and a proper synchronization between the CCD clocking rate and the processor ticks is achieved. The global timestamp is computed based on this synchronization rate. The delay factor is updated with the inclusion of individual algorithmic steps and the timestamp is accordingly revised.
5.3.2 DFD

Motion is determined for change in background is which is measured as the displaced frame difference between the base frame and reference frame. Figure 5-6 illustrates a difference between two frames and an orange color ball identified through the difference.

![Base frame](image1)
![Reference frame](image2)

**Figure 5-6 DFD identifying a color ball**

Figure 5-7 illustrates a frame difference to identify a white ball.
5.3.3 Motion detection

A global threshold is applied in tabular region in the image plane, to segment ball motion from noise and other background motion. Each frame is represented by $640 \times 480 = 307200$ pixels. However, motion is detected in only the area of interest, i.e. the table, which represents $250 \times 330 = 82500$ pixels. This saves the computation time for more than $1/3^\text{rd}$ area of the image. Comparing Figure 5-6 and 5-7, it is observed that the colored ball is more distinct in display as compared to the white ball in the DFD.
image. However, with elimination of the color factor due to binarization based on a global threshold, it is possible to identify the prospective ball region from the background difference. Figure 5-8 illustrates the detection of prospective ball motion region for the differences observed in Figure 5-6.

![Figure 5-8 Binarization for color ball](image)

Figure 5-9 illustrates the detection of prospective ball motion region for the differences observed in Figure 5-7.
5.3.4 Ball detection

Motion is also detected in the form of noise, along with the ball region. A local threshold based on the contour scanning and number of retrieved points ensures the segmentation of the exact ball region in each frame. Ball segmentation based on contour points representing the ball size, from noisy illusory contours is shown in Figure 5-10(a) and (b) respectively.
The number of points retrieved on the contour for the ball in Figure 5-10(a) is 40 and in Figure 5-10(b) is 29, which represent the ball size in consistency with their respective distances from the camera.

5.3.5 Ball location in the image plane

2 \( - \) \( D \) motion refers to the projection of the \( 3 \) \( - \) \( D \) motion onto the image plane as time-varying images are \( 2 \) \( - \) \( D \) projections of \( 3 \) \( - \) \( D \) scenes [51]. Since the ball is circular in shape and has uniform intensity, the exact motion is unobservable. Hence,
there is a ringing effect on the ball’s silhouette. In order to identify projection of the
2-D circular ball, an ellipse is used as a single shape approximation and the center of
this ellipse determines the center of the circle. The center of the ball shown in Figure 5-9
(a) is \( F_b(x, y) = (289, 168) \) and that of the ball in Figure 5-9 (b) is \( F_b(x, y) = (296, 186) \), on
a pixel scale. These values are confirmed with the values obtained by using the function
\textit{bwlabel} in Matlab, which employs connected component labeling.

### 5.3.6 3-D Imaging

In order to represent the ball on a physical measurement scale, equations 4-19 and
4-20 are utilized. Relations determining the values of \( \alpha \) and \( \beta \), as shown in Figure 5-4
are based on experimental values tabulated as follows

<table>
<thead>
<tr>
<th>Distance (mm)</th>
<th>Angle (alpha) in degrees corresponding to X-resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>520.7</td>
<td>21.32</td>
</tr>
<tr>
<td>673.1</td>
<td>21.15</td>
</tr>
<tr>
<td>806.45</td>
<td>20.7</td>
</tr>
<tr>
<td>1060.45</td>
<td>20.07</td>
</tr>
</tbody>
</table>
Table 5-3 Vertical resolution

<table>
<thead>
<tr>
<th>Distance(mm)</th>
<th>Angle (beta) in degrees corresponding to Y-resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>520.7</td>
<td>16.64</td>
</tr>
<tr>
<td>673.1</td>
<td>16.55</td>
</tr>
<tr>
<td>806.45</td>
<td>16.45</td>
</tr>
<tr>
<td>1060.45</td>
<td>15.88</td>
</tr>
</tbody>
</table>

The ball’s size increases as it gets closer to the image plane. The curve determining the relation between the ball size and its distance from the image plane as formulated in equation 4-18 is based on the experimental values tabularized in Table 5-4.

Table 5-4 Focal length

<table>
<thead>
<tr>
<th>Distance from the camera lens (mm)</th>
<th>Size of ball (pixels)</th>
<th>Distance from the camera lens (mm)</th>
<th>Size of ball (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1008</td>
<td>32</td>
<td>862</td>
<td>37</td>
</tr>
<tr>
<td>998</td>
<td>32</td>
<td>846</td>
<td>37</td>
</tr>
<tr>
<td>968</td>
<td>32</td>
<td>838</td>
<td>39</td>
</tr>
<tr>
<td>952</td>
<td>32</td>
<td>828</td>
<td>39</td>
</tr>
<tr>
<td>928</td>
<td>34</td>
<td>814</td>
<td>42</td>
</tr>
<tr>
<td>905</td>
<td>36</td>
<td>798</td>
<td>42</td>
</tr>
<tr>
<td>890</td>
<td>37</td>
<td>776</td>
<td>43</td>
</tr>
<tr>
<td>Distance from the camera lens (mm)</td>
<td>Size of ball (pixels)</td>
<td>Distance from the camera lens (mm)</td>
<td>Size of ball (pixels)</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------</td>
<td>-----------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>877</td>
<td>37</td>
<td>479</td>
<td>71</td>
</tr>
<tr>
<td>758</td>
<td>44</td>
<td>466</td>
<td>75</td>
</tr>
<tr>
<td>738</td>
<td>44</td>
<td>447</td>
<td>76</td>
</tr>
<tr>
<td>724</td>
<td>45</td>
<td>428</td>
<td>78</td>
</tr>
<tr>
<td>696</td>
<td>47</td>
<td>416</td>
<td>80</td>
</tr>
<tr>
<td>682</td>
<td>49</td>
<td>402</td>
<td>86</td>
</tr>
<tr>
<td>658</td>
<td>49</td>
<td>385</td>
<td>89</td>
</tr>
<tr>
<td>646</td>
<td>52</td>
<td>368</td>
<td>94</td>
</tr>
<tr>
<td>630</td>
<td>53</td>
<td>357</td>
<td>99</td>
</tr>
<tr>
<td>611</td>
<td>56</td>
<td>344</td>
<td>101</td>
</tr>
<tr>
<td>600</td>
<td>56</td>
<td>316</td>
<td>113</td>
</tr>
<tr>
<td>584</td>
<td>58</td>
<td>303</td>
<td>121</td>
</tr>
<tr>
<td>564</td>
<td>60</td>
<td>277</td>
<td>132</td>
</tr>
<tr>
<td>548</td>
<td>61</td>
<td>265</td>
<td>137</td>
</tr>
<tr>
<td>524</td>
<td>64</td>
<td>247</td>
<td>150</td>
</tr>
<tr>
<td>519</td>
<td>66</td>
<td>225</td>
<td>171</td>
</tr>
<tr>
<td>507</td>
<td>67</td>
<td>201</td>
<td>197</td>
</tr>
<tr>
<td>490</td>
<td>69</td>
<td>180</td>
<td>222</td>
</tr>
</tbody>
</table>

Camera calibration based on these parametric values is utilized to track the ball in
3-D. Figure 5-10 illustrates the tracking of a color ball after binarization and a comparison of values determining the distance of ball in line of camera axis.

(a) Ball detection in real time sequences acquired by the camera

(b) Focal length for a stationary ball

(c) Focal length for ball detected in real time sequences above

(d) 3-D coordinates of the ball centers in above sequences.

Figure 5-11 Single camera based 3-D tracking of color ball
Figure 5-12 illustrates 3-D tracking of a white ball.

(a) Ball detection in real time sequences acquired by the camera

(b) Focal length for a stationary ball

(c) Focal length for ball detected in real time sequences above

(d) 3-D coordinates of the ball centers in above sequences.

Figure 5-12 Single camera based 3-D tracking of white ball
The amount of noise introduced for the difference between two image frames varies with respect to the frames captured at different time intervals. A performance parameter for the frame capture mechanism is the replacement of the base frame with the new frame captured when no motion is detected. A local functioning factor for the 3 - D imaging module is scanning for single contour sequence, matching the number of points on the ball silhouette, only in the motion identified region. The system takes 10 ms to perform these computations. In order to maintain synchronization with the frame capture rate, an 45 ms delay with 20 % I/O tolerance, based on experimental trials is provided.

5.3.7 Displacement

Velocity data of the ball in three directions based on equations 4-39 – 4-41, is used to determine the orientation of ball in flight with reference to the orientation of robot motion. Table 5-5 lists the ball position tracked and their corresponding velocities for the displacement illustrated in Figure 5-13.

![Figure 5-13 3-D displacement](image-url)
### Table 5-5 Ball velocity based on displacement in time

<table>
<thead>
<tr>
<th>Δt (ms)</th>
<th>X (mm)</th>
<th>$V_x$ (mm/ms)</th>
<th>Y (mm)</th>
<th>$V_y$ (mm/ms)</th>
<th>Z (mm)</th>
<th>$V_z$ (mm/ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1151.94</td>
<td>0.0</td>
<td>142.09</td>
<td>0.0</td>
<td>232.84</td>
<td>0.0</td>
</tr>
<tr>
<td>0.020</td>
<td>1067.32</td>
<td>-4231.15</td>
<td>141.38</td>
<td>-35.34</td>
<td>66.98</td>
<td>-8194.30</td>
</tr>
<tr>
<td>0.030</td>
<td>801.11</td>
<td>-8873.53</td>
<td>123.83</td>
<td>-585.06</td>
<td>174.70</td>
<td>3738.67</td>
</tr>
<tr>
<td>0.030</td>
<td>758.42</td>
<td>-1423.05</td>
<td>129.26</td>
<td>180.88</td>
<td>260.53</td>
<td>3008.72</td>
</tr>
<tr>
<td>0.030</td>
<td>608.10</td>
<td>-5010.71</td>
<td>121.81</td>
<td>-248.43</td>
<td>359.52</td>
<td>3447.49</td>
</tr>
<tr>
<td>0.030</td>
<td>496.60</td>
<td>-3714.46</td>
<td>119.25</td>
<td>-85.19</td>
<td>398.37</td>
<td>1443.13</td>
</tr>
<tr>
<td>0.030</td>
<td>483.02</td>
<td>-314.03</td>
<td>127.64</td>
<td>209.87</td>
<td>348.40</td>
<td>-1052.25</td>
</tr>
</tbody>
</table>

Initial time and velocities are zero. Δt, which is measured as function of the processor ticks, is initialized as soon as the ball is located in 3-D space and is computed until a displacement is observed. Additionally, in order to maintain the capture rate at 66 msec, appropriate delay is provided. As soon as a displacement is observed, the
velocities in respective directions are also calculated. The velocity plot is illustrated in Figure 5-14 below.

![Velocity plots](image-url)

**Figure 5-14 Velocity plot**
As shown in Figure 5-14, $V_x$ is always negative; this indicates the ball approaching towards the robot. A slight change in $V_x$ is observed due to the drag and spin of the ball and $V_y$ is almost constant. Change in orientation of $V_z$ occurs due to the effect of gravitational acceleration.

5.3.8 Trajectory estimation and target point computation

Change in the location of the ball in time is used to determine the bounce information using equations 4-49 4-51. A trajectory based on this estimation is extrapolated and is as shown in Figure 5-15 below.
Based on the extrapolated trajectory, a second bounce is estimated and a target point is computed using equation 4-53. The robot motion is exhibited towards the ball at the target specified by equation 4-56. A revised trajectory plot with the second bounce and the target point is illustrated in Figure 5-16 below.

![Updated trajectory with target point](image)

**Figure 5-16 Updated trajectory with target point**

Thus, vision based trajectory information is utilized as a feedback information to control the robot's action in hitting the ball.
6. Conclusion

6.1 Synopsis

This thesis described and demonstrated the notions and implementations in practice, of a ping pong playing robot. The task was a challenging one due to

- Parametric rigidity of the hardware configuration and
- Idealistic inclination towards the design and development of a simple, economical and robust mechanism.

For viable design of the system, the complete task was segregated into different modules based on particular functionality. Overall system performance was based on the functioning of the vision system and the controller architecture and their synchronization. The intelligence of an entire ping pong playing robot system encompasses a sense of time in the continuous domain, since we live in a continuous world [22]. In this system design, a timestamp is associated with all the individual modules so that the robot is aware of the ball location in space at time. With the current design, the system can take advantage of a redundancy that is associated with every individual module. Changes in the scene as a function of the correlation in the image are used to characterize the ball motion. The robot motion is perceptively based on the work space in which the robot moves freely. The utilization of these redundancies in developing and implementing the algorithm proved useful to guide the robot to hit the ball at the specified location in time.

Thus computer vision exploiting calibrated camera information was utilized as a scientific tool to visually track the ball \((x, y, z, t)\) position at the end of each frame and navigate the robot. Processor architecture was efficiently utilized in synchronization with
the CCD clocking rate to capture one frame approximately every 1/66th of a second and use of frame grabbers and video cameras was successfully avoided. The strategic skills of hitting the ball were taught to the robot through the coordination between a camera (which represents the human eye) and an expert controller (represents the human brain). The basic initiative of deriving a mathematical paradigm of the ping pong game played by the humans and investing the analogous planning stratagems in a robot was brought into practice through the system developed in this thesis. Additionally, individual tasks of ball tracking using single camera based perspective vision, at a low sampling rate of 15 Hz. and algorithmic self sufficiency of the system in demonstrating consistency to work for any color ball were also accomplished through this work.

6.2 Performance analysis

Various experimentations were performed to determine the feasibility of the system working conditions. There were no special illumination requirements for the ball to be detected in the image plane; in this work, a diffused fluorescent tube lighting system illuminates the scene. The center of the moving ball is tracked in 3-D based on its size in the image plane and distance from the focal center of a stationary camera. The utilization of a single camera is advantageous in comparison to stereo vision, due to non existence of the correspondence problem in identification of part of the left and right images as projections of the same scene element. Additionally, the amount of noise introduced is less, in significance to the use of more than one camera. The current positioning of the camera above the robot demonstrates maximum accuracy in locating the change in ball size as the ball approaches the robot. The tilt in the camera for
perspective frame capture prevents occlusion of the moving ball. A revised trajectory is extrapolated after a bounce occurs on the side of the robot. This approach enables the computation of an exact target point, irrespective of the angle at which the ball is thrown. Based on the reach of the robot arm, a 3-\textit{D} coordinate working space of the robot is determined. If the target point lies within this space, the robot knows that the ball is hittable and demonstrates corresponding motion. This step also prevents the robot from crashing into the table. The robot arm is programmed to hit the ball at its maximum speed of 640 mm/sec. The decision to move the robot is made after a bounce occurs and the robot hits the ball, which has a speed between 1140 mm/sec and 1400 mm/sec. Due to low acquisition rate and low speed of the robot motion, the robot misses for any change in the specified range of the ball velocity. For the current system design, the air drag and spin dynamics of the ball are not considered in estimating the target point.

6.3 Future work based on current limitations

Due to low sampling rate, the observed trajectory is not used to predict the future trajectory of the ball. An increase in the current acquisition rate of 15 frames per second would be effective for predicting the return path of the ball. In this work, USB version 1.1 is used to interface the camera to the computer. However, VGA resolution in graphics mode is limited to 15 frames per second, as there is not enough bandwidth available on the USB bus to squeeze through more, even with compression. Use of a faster peripheral bus standard IEEE 1394 (a or b) can provide isochronous frame transfer at a higher sampling rate. Additionally, the robot is communicated through a serial interface at a baud rate of 9600 bits/sec. A faster interface should enable high speed data transfer to the
robot and ultimately more time for scene analysis. Prospective work involves utilization of all five degrees of freedom of the robot to exhibit the task of hitting the ball at an angle perpendicular to trajectory path of the ball center unlike the linear interpolating movement of robot joints used in the present system design. Future expectations to hit the ball back at the table and train the system for a rally with the human rely on the mentioned practical analysis.

To summarize, system performance in real time, with an aerodynamic approach is expected to boost with a high acquisition rate, an increase in robot speed and its interface. Thus, a boost in the hardware configuration is applicable for the futuristic design of a sophisticated ping pong playing robot system that can utilize all five degrees of freedom associated with the game of ping pong, to volley with a human.
References


Appendix A

The *CD-ROM* contents include

- Matlab code for design and testing of the algorithmic modules

- C source code for
  - Individual modules of frame capture, motion detection, ball detection, 3-D imaging, trajectory computation, and robot motion
  - An interface of the above modules for the complete system design along with a list of the global parameters

- C++ code of the library functions utilized as the benchmarks.