CS 684 Project Report

Scarecrow Robot

Deepak Babu Sam (133050095)
Narendra Kumar Singh (133050064)
Rajeev Kumar Verma (133050096)
Praveen Verma (133050086)

11/13/2013

ABSTRACT

Man has always wondered about himself. Robotics is one way of emulating our lives to understand how we work. This thesis is born out of such curiosities in which we investigate the design of a robot that can imitate at least a minute piece of human perception, cognition and actuation. In this project, we are building a fully autonomous intelligent robot that is capable of finding and shooting an intruder which has trespassed into the greenhouse. [3]**Greenhouse** is a structure, primarily of glass in which temperature and humidity can be controlled for the cultivation or protection of plants. It has to work like an actual human capable of seeing through the eyes, interpreting its environment and acting appropriately in order to reach the desired situation.

Table Of Contents

1. INTRODUCTION

1.1 Background and Motivation

Robots are no longer only found on the pages of science fiction books or in the scripts of Hollywood movies. There are now robots being developed to achieve goals varying from vacuuming inside one's home to protecting outside of home and also to crops and plants. Furthermore, the goal of human like behavioural robot research is not only to create robots that will achieve specific tasks, but also "*to develop, validate and disseminate a broadly applicable set of rules which specify, relative to the state of robot cognition, when and where a robot should act and how to interpret the corresponding situations*".

Global warming is defined as *"a gradual increase in the overall temperature of the earth's atmosphere generally attributed to the greenhouse effect caused by increased levels of carbon dioxide, chlorofluorocarbons, and other pollutants."*

Global warming is one of the most pressing environmental issues today's generation of farmers are facing and if it is not controlled or if some alternatives are not found the next generation of farmers and hence all people will severely suffer from this. Due to global warming, it is very challenging for plants/crops around the world to survive due to extreme hot or cold weather. [2]According to one research paper, in US, crop yields at the end of the century are predicted to decrease by 44% for corn, 33-34% for soybeans, and 26-31% for cotton under a slow warming scenario and 79-80%, 71-72%, and 60-78%, respectively, under a quick warming scenario. The long-term goal of Scarecrow robot is to build a few more lookalike robots so that the damage is minimised or completely overcome.

1.2 Objectives

Major objectives of this project are:

[1] To create an intelligent system that can perceive its environment through its own senses, learn how to act so it can reach its objectives and perform its response.

[2] To enable the robot to perform the task of

(a) Identifying objects like the coloured box (in our case but many improvements can be done.

(b) Navigate to reach near the intruder

(c) Shoot using the laser once within a range

This project will serve as a building block upon which continued implementation of human like robot systems can be built. The greater goal is to create a robot that moves in a “human like” manner, learns from its own experiences and acts so as to attain its objectives.

2. INTELLIGENT SYSTEM (IS)

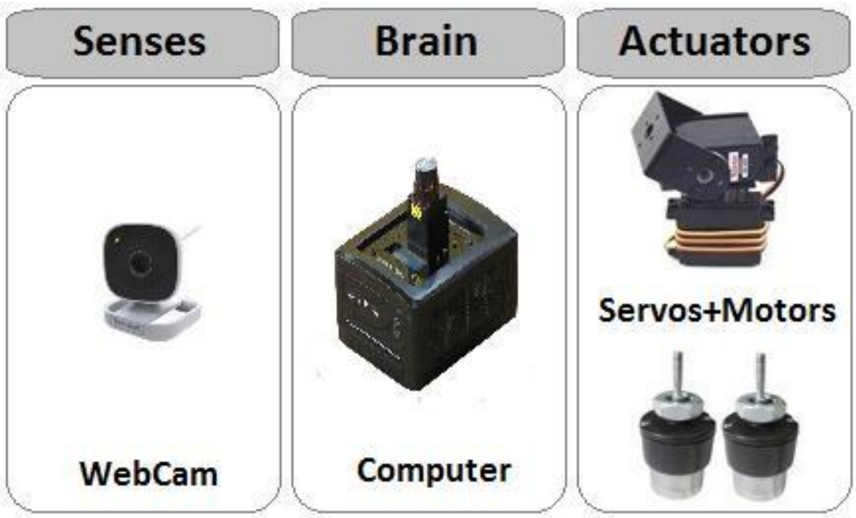

FIGURE 2.1: IS model of Scarecrow Robot

Our Scarecrow IS Robot has the following structural entities:

- **Senses** – 1.3Mega Pixel Webcam
- **Brain** – computer (atom microprocessor)
- **Actuators *Pan-Tilt Servos*** - Camera is mounted on pan and tilt servos (2 degree of freedoms) giving 180 degree yaw and 90 degree pitch
- ***DC Motors*** **-** Two dc geared motors to power the differential drive of the robot

The IS sees the environment through the webcam. The images captured by the camera are fed via USB to the computer. The computer performs image processing techniques to identify the robot's surroundings and converts these information to situations so that the IS can process. It then selects appropriate response rule for the current situation and drives the actuators.

3. IMAGE PROCESSING (IP)

3.1 Introduction

Image Processing is the process of extracting useful information from image data. IP is done using the OpenCV library. IP is used to identify and track objects interesting for the IS.

What is OpenCV?

OpenCV is an open source computer vision library available from *http://SourceForge.net/projects/opencvlibrary*. The library is written in C and C++ and runs under Linux, Windows and Mac OS X.

OpenCV was designed for computational efficiency and with a strong focus on real-time applications. One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. The OpenCV library contains over 500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics. OpenCV also contains a full, general-purpose Machine Learning Library (MLL) focused on statistical pattern recognition and clustering.

3.2 Object Recognition

In order to avoid complexity, in this project we decided to limit object recognition of blobs. A blob is closed area in an image of one particular colour or colour combination. Each of the objects like ball, goal post etc is made of one colour and can be considered as a ***blob***. The function of *Blob Tracker* is to find out current position of the blob from the input image.

3.2.1 Mean-Shift Algorithm

The mean-shift algorithm is a robust method of finding local extrema in the density distribution of a data set. Mean-shift ignores data points that are far away from peaks in the data. It does so by processing only those points within a local window of the data and then moving that window.

The mean-shift algorithm runs as follows.

1) Choose a search window:
   - its initial location
   - its type (uniform, polynomial, exponential, or Gaussian)
   - its shape (symmetric or skewed, possibly rotated, rounded or rectangular)
   - its size (extent at which it rolls off or is cut off).
2) Compute the window's (possibly weighted) center of mass.
3) Center the window at the center of mass.

4) Return to step 2 until the window stops moving.

To give a little more formal sense of what the mean-shift algorithm is: it is related to the discipline of kernel density estimation, where by "kernel" we refer to a function that has mostly local focus (e.g., a Gaussian distribution). With enough appropriately weighted and sized kernels located at enough points, one can express a distribution of data entirely in terms of those kernels. Mean-shift diverges from kernel density estimation in that it seeks only to estimate the gradient (direction of change) of the data distribution. When this change is 0, we are at a stable (though perhaps local) peak of the distribution. There might be other peaks nearby or at other scales.

3.2.2 CAM-Shift Algorithm

Cam-shift stands for ***"continuously adaptive mean-shift"*** and is a technique for tracking objects within images. The cam-shift technique enables you to discover features such as size, angle, and position of masses within an image. It performs this function first by attempting to determine the center of mass for the given image and then centering the window on the center of mass. This repeats until the window no longer needs to change positions in order to be centered. OpenCV provides the cvCamShift function which performs the cam-shift technique on a given frame.

It differs from the mean shift in that the search window adjusts itself in size. If you have well-segmented distributions (say face features that stay compact), then this algorithm will automatically adjust itself for the size of face as the person moves closer to and further from the camera.

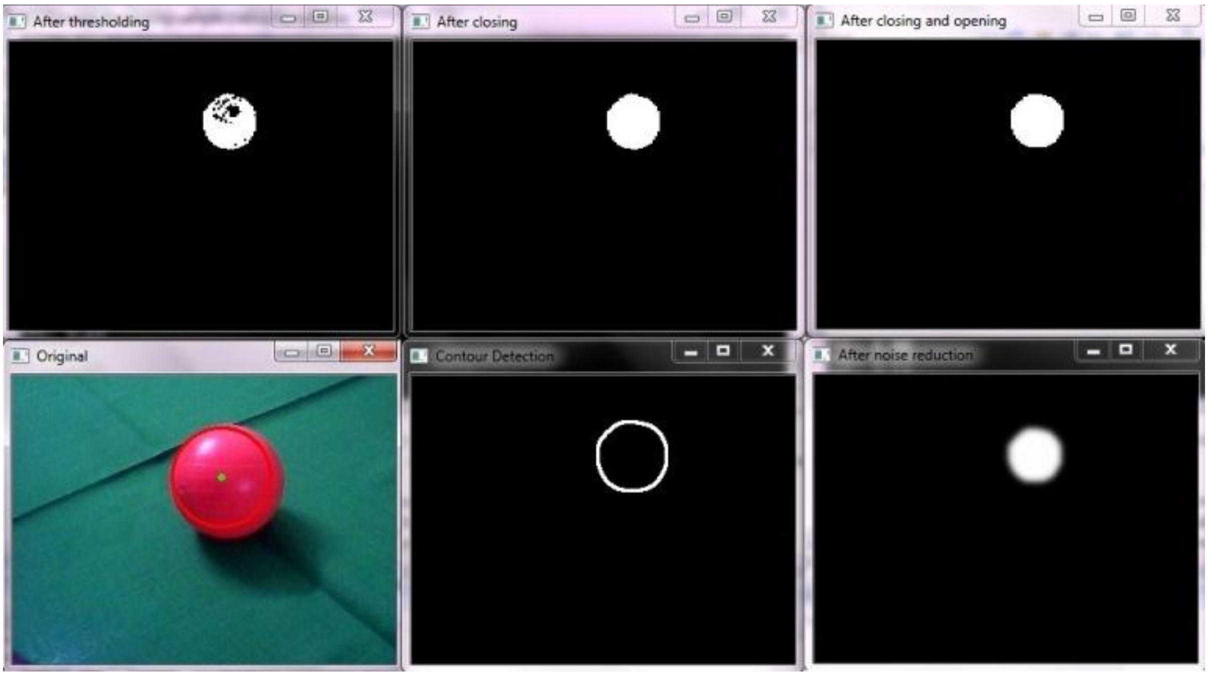

FIGURE 3.2.2.1: Various stages of detection of a ball. In clockwise direction from the top the figures show – colour thresholded, closed, opened, noise reduced, contour detected and the original image

3.2.3 Blob Detection

Blob detection is a process through which objects in a frame are detected by analyzing groups of closely related pixels. These pixel groups are termed blobs. Blobs are typically detected through looking for differences in color or brightness in the areas surrounding the blob. Performing blob detection on a fame results in a series of blobs each which have a corresponding position and size.

3.3 Blob Tracking

Histogram colour model of the blob to be identified is previously recorded. Back projection technique is used to filter out the image for finding blobs of interest. Back projection is a way of recording how well the pixels or patches of pixels fit the distribution of pixels in a histogram model. The filtered image is then fed to a continuously adaptive mean-shift (CAMSHIFT) algorithm to track the blob. The mean-shift algorithm is a robust method of finding local extreme in the density distribution of a data set (here colour histogram). Cam-shift tracker is a related algorithm. It differs from the mean-shift in that the search window adjusts itself in size. In this project, major blobs of interest are the ball and the goal post region.

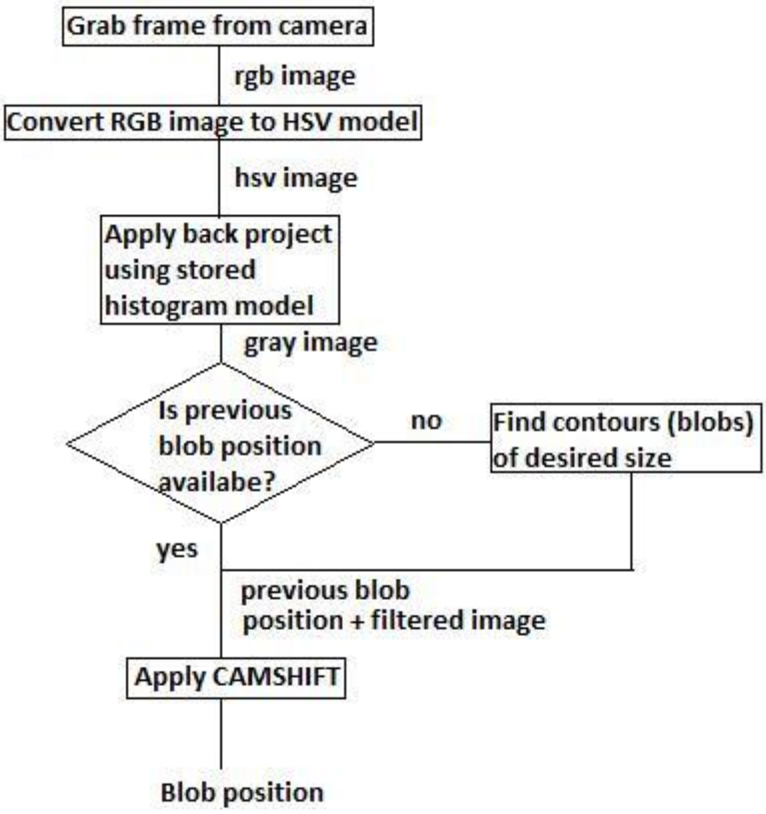

FIGURE 3.3.1: Blob tracking algorithm flowchart

4. MECHANICAL SYSTEMS

4.1 Overview

Mechanical system design is the first step of designing robot. Human body mechanism basically consists of bones, joints, muscles, and tendons. It is impossible to replace all of the muscular-skeletal system by current mechanical and electrical components. Therefore, the primary goal of the mechanical system design is to develop a robot that can imitate equivalent human motions. Inorder to avoid complexity, a wheeled robot is envisaged capable of roaming the arena swiftly. A camera is mounted on a 2 Degree of Freedom (DOF) head. It can rotate around (pan) 180 degrees and bend or tilt 90 degrees replicating human head motion.

4.2 Construction

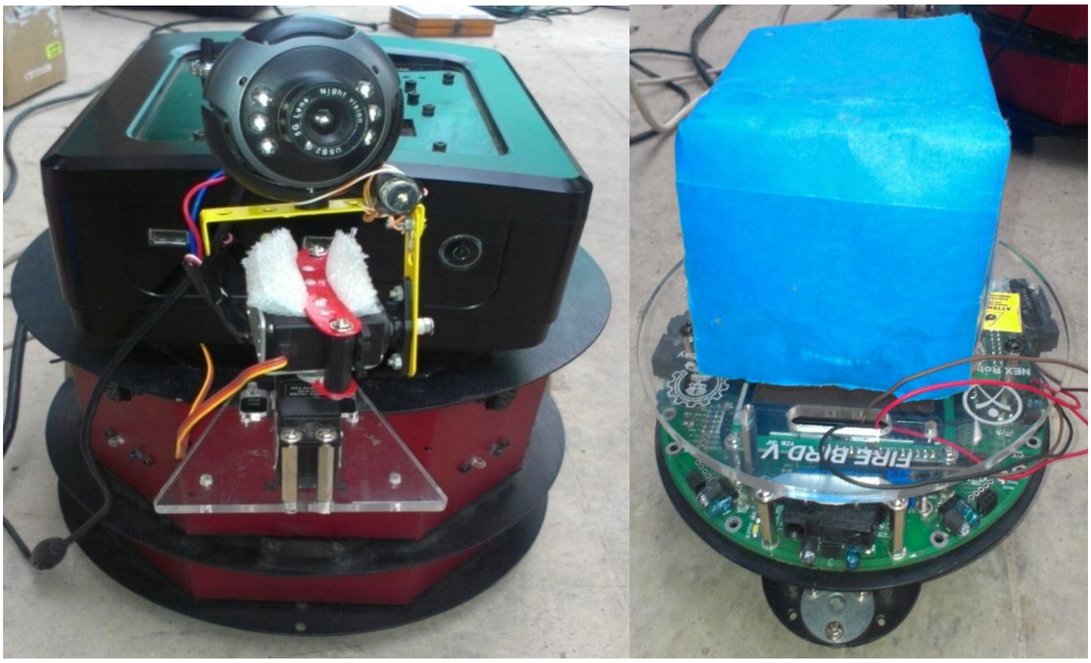

Figure 4.2.1: Complete setup of scarecrow robot

The camera is mounted on Firebird VI which follows the mechanism of pan-tilt motion using two servo motors. The command is given through the atom processor. Here firebird V is used as an intruder having a coloured box being mounted on top of it. The intruder is programmed to move randomly in the greenhouse, of course avoiding the walls or any obstructing thing comes in its way.

4.3 Pan-Tilt servo head

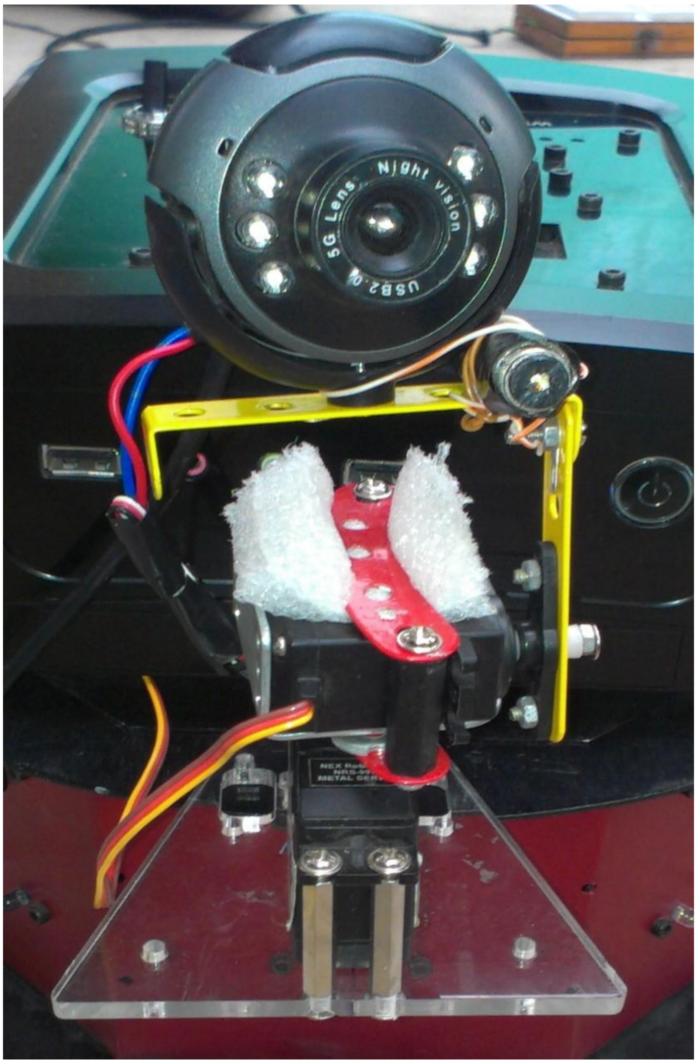

Figure 4.3.1: Pan-Tilt setup

Pan and Tilt capabilities are added to the camera which acts as the eye of the robot. This system consists of the necessary brackets and hardware to allow you to pan and tilt your camera. Two servos are required for its operation. The panning servo is fixed vertically to the base while the tilt servo is mounted on to rotor of the pan servo as shown in the figure. Camera is then mounted on to the top of Firebird VI.

Servo Specifications: Standard Size micro servo Motor with Dual Ball Bearing. Provides 5.5kg/cm at 4.8V and 7kg/cm at 6V. Exact replacement for Futaba S3003 and Hitec HS-311**.**

WebCam Specifications: 1.3 Mega Pixel, Intex WebCam

5. ELECTRONIC SYSTEMS

5.1 Overview

The electronic system primarily acts as a slave to the computer. It receives commands from the computer and controls the differential drive of the robot and the servos accordingly.

FIGURE 5.1.1: The overall electronic system of the robot

5.2 Description

The heart of the circuit is Firebird VI microcontroller. It has in-built modules for serial communication, pulse-width modulation and timed signal generation.

PWM module of the microcontroller is used to generate PWM signals to be given to L293D motor driver. By varying the pulse width of the signals, speed of the motors can be controlled.

The robot main board and all actuators are powered from three cell 12 volt lithium ion battery pack. It is of capacity 1.4ahr. Sufficient capacitors are provided to supply high current during switching of motors and servos.

6. Cognition

6.1 Introduction

The IS first convert its input from the environment to situations so that the brain can process. After the current situation is made, the IS selects appropriate response rule and perform the desired action.

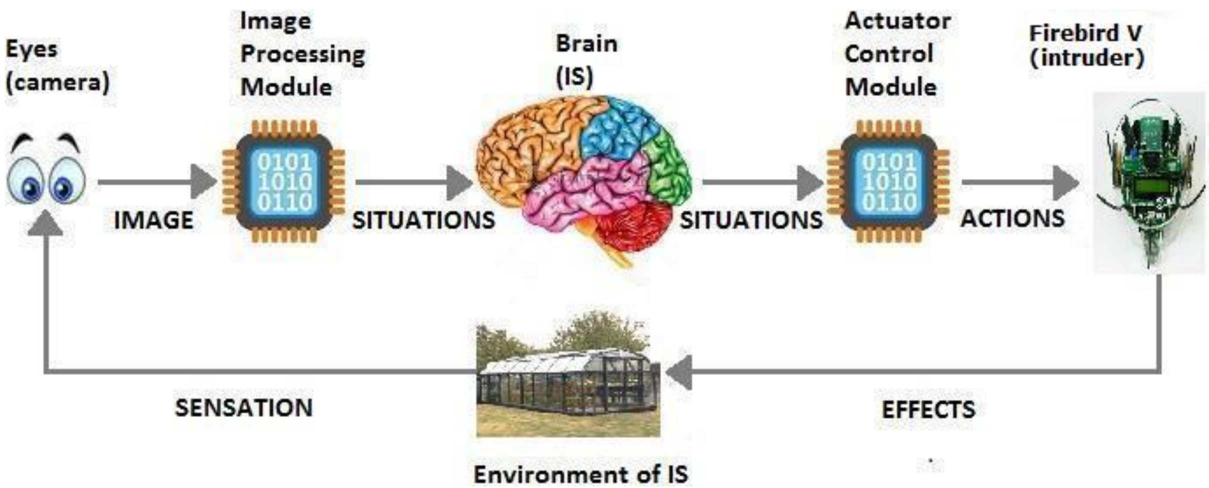

FIGURE 6.1.1: Overview of cognitive functioning

The figure shows the cognitive functioning of the scarecrow robot IS. The eye (camera) of the IS receives stimuli from the environment. The images from the camera are processed by the image processing module and extracts information from the image feeds. It represents the current state of the environment (greenhouse) as a series of situations. The brain can process only with situations. An appropriate response rule that best describe the current situation is searched in the IS's memory. Once the rule is found, the IS takes its response part. If no response rule for current situation is found, IS can't proceed further and produces error message. The response could be either again a series of situations or an elementary action. In both cases, the output of the brain is again in terms of situations (like search_intruder etc). The module dealing with actuator control then takes up these response situations and sends commands to the actuators. Actions by the actuators may affect the situations of the environment and again may produce sensation to senses which acts as a feedback. This feedback process continues till the objectives of the IS are achieved.

6.2 State chart diagram

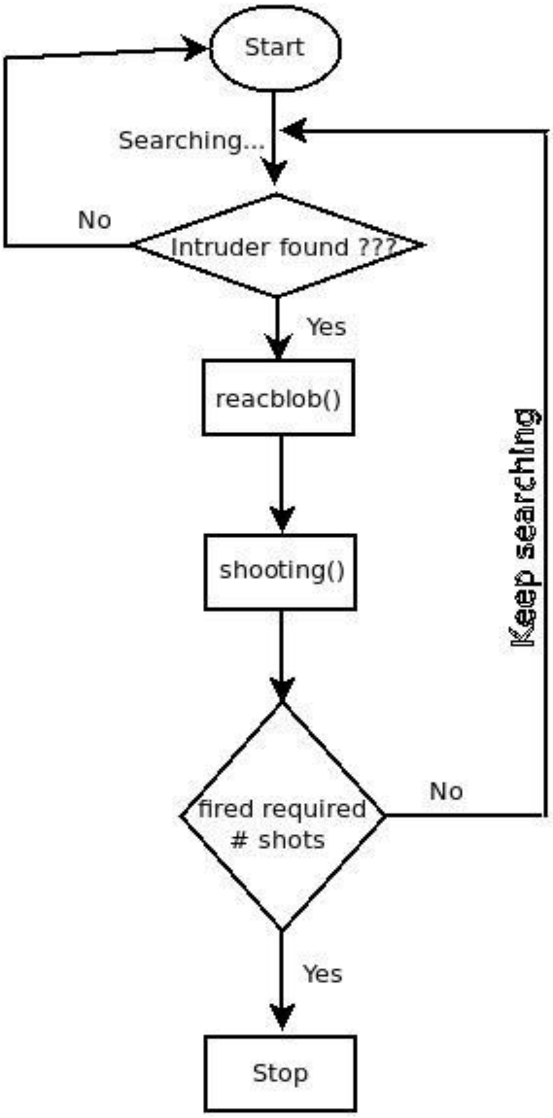

Figure 6.2.1: State chart diagram

On start-up the current situation (CS) is {"*roaming in greenhouse*"}. A response rule relates between some situations and corresponding action. Action is invoked by calling elementary functions. These C functions act and perform the desired actions. Actions may change the CS of the IS.

Eg: {"*roaming in greenhouse*"} → find_intruder()

If the CS is {"*roaming in greenhouse*"} then find_intruder() will be called and intruder is searched. If the intruder is found, the CS changes to {"*roaming in greenhouse*", "*intruder_found*"} and then corresponding rule is selected.

Other response rules:

{"*roamin_ in_greenhouse*", "*intruder_not_found*"}	→	find_intruder()
{"*roaming_in_ greenhouse*", "*intruder_found*"}	→	reach_intruder()
{"*roaming_in_greenhouse*", "*intruder_near*", "*intruder_inrange*"}	→	shoot_intruder()

Response rules are written and stored in a text file. The rule file is loaded and read by the scarecrow robot program. All the situations and elementary function names are predefined and are loaded to the rule memory of the IS. After determining the current situation each time, the IS looks for any available rule corresponding to the present situations and if found, its elementary function is executed. Figure 6.2.1 illustrates this working.

6.3 Rule Functions

6.3.1 findBlob() function

findBlob() function searches for a given object (blob) in the robot's environment. Its operation is general and is used for finding either ball or goal post. If the current situation (CS) is "*intruder_not_found*", the function looks for the intruder. Once the intruder is found, the CS will be "*intruder_found*" or "*intruder_near*".

Searching is done by manipulating the pan-tilt servos so that the head camera sweeps entire environment ahead of the robot. First when the servos are in normal position, the camera can see only long distance objects (as there is no tilt or bend) and the pan servo is rotated from one side to another by incrementing pan angle by some value (5°) covering entire 180°. If the blob is not found, the head is bent by commanding the tilt servo. Now the robot can see more nearby objects and is rotated through 180°. If again the required blob is not detected, this cycle is repeated by tilting the camera more and more by some angle (20°), so that the camera scans nearby areas. This process continues until the blob is found or the bend angle reaches maximum value (145°) at which the camera points directly downwards (in such case the IS stops functioning as it can't find the blob).

When the blob is detected, now the robot has to be turned to position it in the direction of the blob, so that it can navigate to the blob. Hence the camera is quickly rotated to normal pan angle (90°) without changing the camera tilt so that it can detect blob when the robot is turned. Now the robot rotates to left or right based on the previous position of the blob. The robot turns till the blob comes almost at the center of the camera frame. The camera tilt is also slightly adjusted to keep the blob at the center. PID control actuates the differential turning and servo tilting and helps quickly bringing the blob at frame center with less overshoot. After that, the emf returns happily suitably changing the CS to "*intruderl_found*".

6.5.3 reachBlob() function

reachBlob() helps the robot in navigating and reaching near a blob (intruder). If the CS is "*intruder_near*", it runs to goal post region else to the ball. The blob position is extracted from the camera frame by the blob tracker. If the blob is on the left of the middle of the frame, then right motor is turned on at full speed. If the blob is on the right of the middle of the screen, then left motor is turned on at full speed. Hence the robot traces its path towards the blob. As it move towards the blob, the robots camera has to bent more and more down (pitch of the camera) to keep the blob at the centre of the screen. Finally when the blob is near the robot, the camera can no more bend and the CS is changed to "*blob_near*".

6.5.3 shoot_intruder() function

shoot_intruder() simply shoots the intruder till it retreat.

7. RESULT

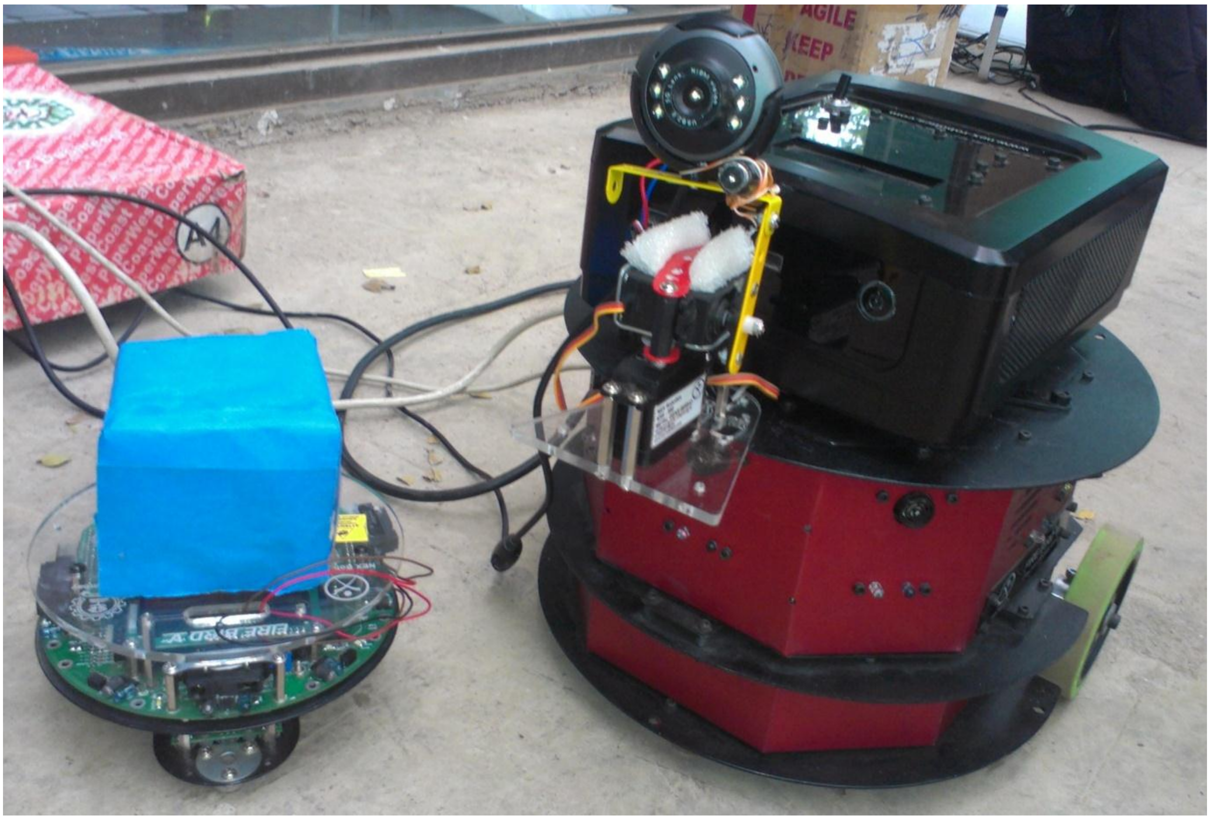

Figure 7.1: Final Project snapshot

Finally we used firebird VI as scarecrow robot and for the purpose of testing and running we used firebird V as an intruder. The firebird VI robot was left in the greenhouse to roam and whenever it sees the intruder it goes closer to it and fires laser on it. Once a specified number of laser shoots is fired on the intruder, the intruder stops by detecting those shoots and again the firebird VI robot starts the task of detecting and removing the intruders. For the purpose of detection we are mounting a coloured box on top of the intruder.

8. CONCLUSIONS

8.1 Achievements

Over the last few months the project has put together a complete design for the development of a robot capable of protecting the greenhouse from intruders. To that end the design for the vision processing and interface software has been developed. A significant amount of learning has taken place during the project especially in the areas of image processing and cognition. The aim of this thesis was to design an intelligent system (IS) capable of vigilance of the greenhouse on small scale. In particular the head design with camera to act as the "eye", design of interface software for the control of robot and the human like cognitive functioning forms the centre of attention for this project. Almost all the objectives initially planned are achieved by the end of this project work.

8.2 Future Works

- An important next step would be to replace the wheeled structure of the robot(firebird VI) by a humanoid platform replicating an actual scarecrow.
- Another improvement we hope is in the image processing technique used for object recognition. Current simple colour based blob detection could be supplemented with much more sophisticated shape recognition or machine learning algorithms so as to eliminate false object identifications.
- Cognition is a large area for future works. The robot brain could be enhanced with more rule and current elementary functions could be decomposed in to smaller functions increasing flexibility. We hope, in the future, the robot could learn the rules by itself from its own experiences just like a human player does.

8.3 Concluding Remarks

We expect that our work will be used as groundwork for further human like robot building research. We believe that the vision and control system will be very useful in almost all aspects of future research. The vision system currently has a limited scope of what it detects, but is set up in such a way which makes it easily expandable. The detection strategies with improvements could be copied and modified in order to detect other objects, and although it was not meant to be a completely robust robot system, for the purposes of our project it worked very well. There is a significant space for studies in the cognition area that has yet to reach any significant leaps, but will be eventually so that we can learn more about, as stated in the beginning, *how we work*.

9. References

[1] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, September, 2008

[2] Estimating the impact of climate change on crop yields: the importance of non-linear temperature effects by Wolfram Schlenker and Michael J.Roberts September-2006

[3] Definition of Greenhouse http://www.thefreedictionary.com/greenhouse

[4] Firebird VI Manual by Nex-Robotics