Real time ball tracker and Goalkeeper

Capstone Project Proposal

Submitted by:

Vaibhav Sood 101610093

Taranjeet Singh 101610091

Tushar Mahajan 101611057

Utkarsh Chugh 101603364

BE Third Year, COE

CPG No. 27

Under the Mentorship of

--------Dr. Raman Goyal--------

Associate Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

March 2019

TABLE OF CONTENTS

Mentor Consent Form

I hereby agree to be the mentor of the following Capstone Project Team

Project Title:		
Roll No	**Name**	**Signatures**
101610093	Vaibhav Sood	
101610091	Taranjeet Singh	
101611057	Tushar Mahajan	
101610064	Utkarsh Chugh	

NAME of Mentor: Dr. Raman Goyal

SIGNATURE of Mentor: . . . .

Project Overview

Our Group is determined to design an Autonomous goalkeeper capable of intercepting balls that are aimed towards the goal post, the keeper is programmed to protect.

The Project involves a number of state-of-the-Art technologies like ball tracking and motorised mechanised movement. Our design will only work in 2 dimensions. We will employ a number of Machine learning algorithms and image processing algorithms.

Need Analysis

Our Autonomous goalkeeper can be generalized as an Autonomous target interceptor. It has a number of uses in sports and entertainment industry. But most importantly it can be used in defense to develop Ballistic missiles and target interceptor for the Indian forces. Currently, the technology is under research by DRDO. This tech has already been developed by Israeli military to counter any Arabic missiles and Palestinians attacks.

To develop it in 3d is close to impossible thus, our group is trying to make it in 2d as a first step to improve the technology.

The main requirements to make it a success are-

1. The Bot(goalkeeper) must have a motor that can allow the bot to move quickly.
2. A camera that can record the ball movement in at least 40Fps.
3. The camera must employ a raspberry Pi to process the live feed, we can also transfer the live feed to laptop computer via Bluetooth or Wi-Fi.
4. The computer (laptop / raspberry pi) must compute the predicted the landing point for the ball and must instruct the bot to move accordingly.
5. The computer will communicate with the bot through a wireless communication device namely, ZigBee.

Literature Survey

ABSTRACT:

In this project we present the design and implementation of our Small Size Robotic Soccer Goalkeeper. We explain the three main components of our architecture: Vision System, AI System and Robots. Each element is an independent entity and therefore the explanation focus on the improvements made to our object tracking techniques and the way these changes interact with the rest of the elements in the architecture.

KEYWORDS: Small-size, robo-football, autonomous, vision, architecture.

Technologies:

VISION SYSTEM:

The vision system is the only source of feedback in the whole architecture, if the data returned by the vision system is inaccurate or incorrect the overall performance of the goalkeeper will be severely affected. That's why the vision system should be robust enough to compensate for any possible errors.

The main object characteristics used by the vision system are the colors defined in the rules of the SSL [Verret 1995]. The ball is a standard orange golf ball

The main tasks of the vision system are:

1. Capture video from the cameras mounted on top of the field in real time.

2. Recognize the set of colors assigned in the rules to the objects of interest in the field (robot and ball).

3. Identify and compute the orientation and position of the robot and ball.

4. Transmit the information back to the AI system.

5. Adapt to different light conditions (color calibration procedure).

The system has several modules; each module is a functional block with a specific task.

CAPTURE MODULE. We use two off the shelf cameras with an IEEE 1394 link. The frame capture is done with MS DirectShow that allows us to configure the resolution of the image, the space color and the frame rate. By default, we capture RGB images with a 720x480 resolution at 30 fps.

PREPROCESSING MODULE. The preprocessing module is used to improve the quality of the image, such as brightness, contrast, gamma, etc. Although this module is crucial for a proper operation its relevance has been reduced to a minimum thanks to the color calibration procedure.

OBJECT CALIBRATION MODULE. This module is a tool to establish the thresholds of each component according to the space color for every object of interest (robot and ball). The calibration is done in HSV color space where selected thresholds are more robust to changes in field lighting and color changes. The HSV thresholds are then transformed to RGB values to improve segmentation speed and avoid costly color space translation processing. This procedure is the most important improvement to our previous vision system because it allows to accurately discriminate the colors, we are interested in. With this change the errors in the localization of the robots has been reduced significantly.

SEGMENTATION MODULE. Assigns each pixel of the images into object classes. The module consists of two segmenters, each one using the thresholds values assigned to the camera for every object of interest. The RGB thresholds are mapped to an array where a logical AND is computed to segment 32 colors at the same time.

BLOB BUILDER MODULE. Connects the segmented pixels into blobs. Before reaching this module, the image is composed of separate pixels; when a blob is constructed useful information is computed such as the area, centroid, bounding box, etc. An RLE and four-neighbor search are computed. A joint list of blobs for the two cameras is generated for each color.

ACTIVATION/DEACTIVATION MODULE. Enables or disables the use of the robot.

RECOGNITION MODULE. Selects the regions that adjust better to the objects searched. It has selection criteria for every kind of object. For the ball we select the orange blob that is nearest to an area of 85 pixels (with an image resolution of 720x480).

GEOMETRIC CALIBRATION MODULE. This module computes the internal and external parameters of the cameras using the Tsai method [Tsai 1987] in OpenCV [OpenCV 2005]. These parameters are used to correct the distortion produced by the lenses of the camera.

LOCALIZATION MODULE. Computes the position of all objects in the field. It uses the camera parameters obtained in the Geometric Calibration module to undistort the image. Also computes the orientation of robot. This computation is done using a set of user-defined points. Each point represents a well-known landmark of the field (corners, midline point, etc), when a point appears in both camera images their coordinates are used to match them and discard duplicate pixels data.

GRAPHIC DISPLAY MODULE. It's responsible for displaying video images in the screen and for generating basic drawing functions such as lines, circles, etc. in the video image.

TRANSMISSION MODULE. A UDP network link is setup for the communication between the vision system and the AI system. The module builds a structure appropriate for data transmission. In practice the vision system can perform communication with two or more hosts allowing for distributed AI system processing if necessary.

AI SYSTEM The AI System or High-Level Control Application is formed by seven modules named: Artificial Intelligence, Collision Detection, Transceiver Communication, Omnidirectional Drive Control, User Interface, Vision's System Communication and Game Control. The system has a main thread that loops all the time and calls function in each of the different modules. This system is designed in a way that the user can test each module separately. It doesn't need all modules connected and working to make simple tests. The system also has a dynamics simulator in order to test system functions initially in the computer, including collision detection, AI and robot control. Figure 3 shows the interaction between the AI modules.

MAIN THREAD. The Control System has a function that loops all the time that we call the main thread. In the iteration we call each of the functions of the different

modules to finally get the commands sent to each one of the robots. The main thread first checks for the vision system's information to know where the robots are. Then we have to check the game state we are in (game state is controlled by the referee). With the robot's coordinates and the game state we call the AI function that returns the desired position to move for each robot and the actions to take. Each robot has to avoid collisions so the future positions are sent to a potential field function that returns the moving vector that avoids the obstacles. With the moving vector we have to calculate the speed for each one of the four wheels of the robot. This task is done by the drive control function. Finally, we build the packets for all the robots with the speed and the actions and we send them with the help of the transceiver.

AI MODULE:

This module receives the robot's positions, the ball position, the robot's angles, the game state, the robot's roles, the strategy and the direction we're shooting. With all this information the system calculates the future position and the actions for robot. We can program and integrate a new strategy in a very easy way thanks to the simple design of this module.

We have the role named: Goalkeeper. The task of the goalie is avoiding at all cost receiving a goal. His behavior is simple and effective; at all times we compute the maximum blocking area we can accomplish with the robot inside its own defense area.

COLLISION DETECTION MODULE. This module receives the movement vector, robot's positions, the ball position, and the robot's angles and returns the new movement vector that assures that there will be no collisions on the movement. The vector is the outcome of a potential field that is the combination of multiple potential fields, each field is locally assigned according to its position on the field, the object and the robot behavior.

TRANSCEIVER COMMUNICATION MODULE. This module receives the speed for each one of the robot's motors and the actions to take. This module builds the packets that we will send using our transceiver. It also makes sure the communication is active at all times.

OMNI-DIRECTIONAL DRIVE CONTROL MODULE. This module receives the movement vector and returns the speed for each one of the four robot's motors. Since we have four omni-directional wheels we need this module to know the speed on each motor so that we move on the desired direction.

USER INTERFACE MODULE. This module is constantly displaying all the information we need for each robot. We have the coordinates, the angle, the motor's speeds, the desired positions, the id, the actions, the game state, the referee's commands and the game positions for each robot. The coordinates, angles, desired positions and actions are displayed graphically in a GUI system programmed using OpenGL [OpenGL 2005].

VISION SYSTEM COMMUNICATION MODULE. In this module we receive the packets sent by the vision system containing what we call the scenario that has the robot's coordinates, the ball coordinate and the robot's angles.

GAME CONTROL MODULE. This module receives all the referee commands through the serial interface and returns the game state of the game. This module is based on the SSL rules.

Robot:

The robot receives commands from the AI system and executes them. To accomplish that each robot has the next functional elements:

circuit manually setup with a dipswitch making it easy to modify its id if necessary, any time during the match.

DSP. The robot micro controller is the Texas Instruments TMS320LF2812 fixed-point single chip DSP (Digital Signal Processor). This device offers low power and high-performance processing capabilities, optimized for digital motor and motion control. The DSP consists of six major blocks of logic: (1) External program and data memory, (2) Analog Interface, (3) I/O Interface, (4) Expansion interface,

(5) JTAG Interface, and (6) Parallel Port JTAG Controller Interface, while we currently use only three of these: • External program and data memory. The RAM module is used in debugging the software with the Parallel Port JTAG Controller Interface. • I/O Interface. The interface contains different kinds of pins:

Capture units: used for capturing the rising pulses generated by the motor encoders which can be used to measure speed and direction of the moving motor.

PWM outputs: these pins have an associated compare unit. A periodic value is established to determine the size of the PWM, and the compare value is used to change the duty cycle.

Standard I/O: used to read and write values for transceiver communication, motor, kicker and dribbler control.

Assumptions and Constraints.

S. No.	Constraints
1	The ball shot by the player must not gain any height, that is, the design will only work in 2 dimensions and not in 3 dimensions.
2	There must be a minimum lag between the live feed of the ball's movement and the final machine learning prediction of the ball's landing position.

S. No.	Assumptions
1	The ball will move in 2 dimensions only, i.e. There will not be any height involved
2	Since the acceleration by the motor will give rise to unpredictable error, we will assume that motor will have no acceleration.
3	The speed of the ball will not exceed a given maximum speed due to technical limitations of computer.
4	The ball will be moving on a plain, uniform friction surface and surface will be of uniform color to ease the image processing commutations.
5	The ball will be of spherical shape and not of any other shape or size.
6	The live feed will have minimum delay.

Standards

- ISO 20954 – Image Stabilization standard for camera
- ISO 20490 – Auto Focus Standard for Camera
- ISO 19093 – Low Light Performance Standard

- IEEE 802.11 – Wireless Communication
- ISO 10218-1:2011 specifies requirements and guidelines for the inherent safe design, protective measures and information for use of industrial robots.

Objectives

- The aim of the project is to design an autonomous goalkeeper.
- The goalkeeper will be able to intercept the football before it hits the goal post.
- The camera-computer combination will be able to track the ball's position accurately and predict its route.
- The bot will have high speed motor to ensure quick movement.

Methodology

- Use State of the art object tracking technology to track the movement of the ball shot by the player.
- Employ a camera that will transfer the live feed of the ball's movement to the computer.
- The computer will predict the most probable final landing position of the ball by employing various machine learning, neural network and image processing technologies.
- The computer will ascertain the bot's current location and accordingly compute the direction and time in which the bot must move to intercept the target or the ball.
- The bot will communicate with the computer through a Zigbee.
- The bot will employ an Arduino that will provide all the control signals.

Project Outcomes & Individual Roles

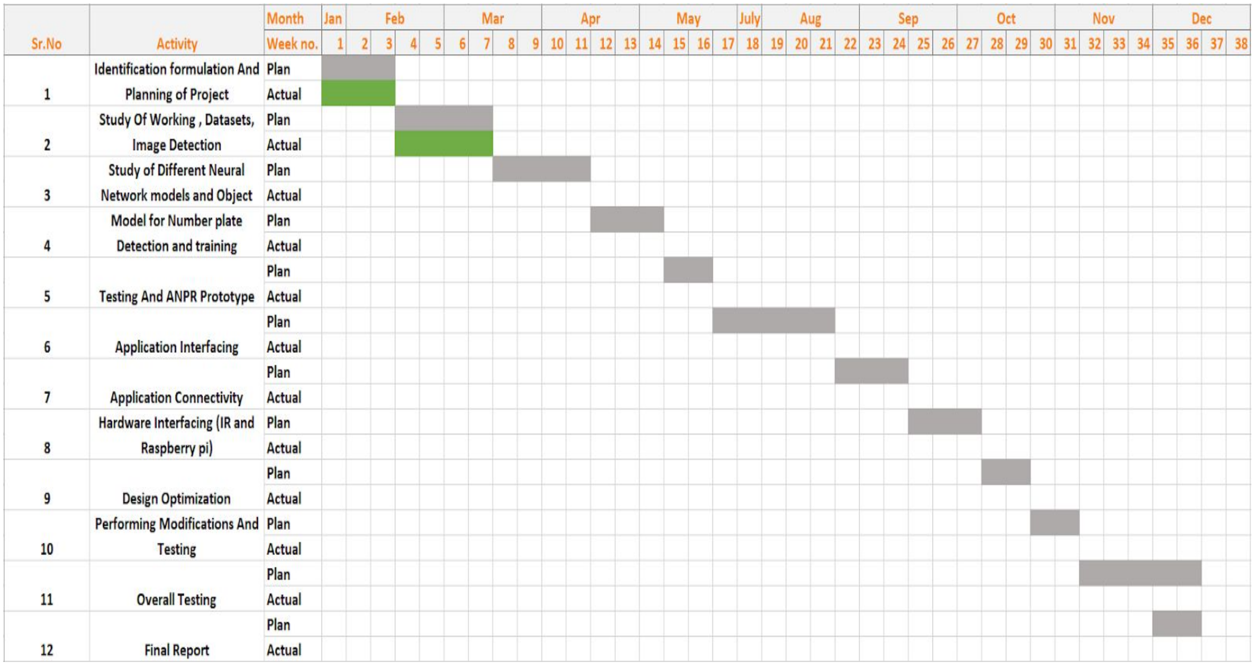
Project Outcomes

- The player or the user will kick or strike the ball.
- The ball will be tracked by the camera-computer arrangement,
- An ML model which will be used to compute the final landing position of the ball.
- A computer will communicate with the goalkeeper bot and order it move accordingly so as to intercept the ball before it hits the target.
- The computer program will assume that the bot has successfully intercepted the target and will loop to the beginning again.

Individual Roles

- Vaibhav Sood – Build the Communication system.
- Taranjeet Singh- Develop the ML algorithm to track the ball and compute the landing position at the goal post.
- Tushar Mahajan –To Find the appropriate hardware needed for various applications throughout the project.
- Utkarsh Chugh – Responsible for the compatibility and dependencies of various parts of the project connected through the network.

Work Plan



Course Subjects

Professional Skills

- Research Skills
- Team Work and Discipline
- Presentation Skills

Technical Skills

- Software Configuration Management
- Building Neural Networks
- Image Processing
- Neural Networks.
- Raspberry Pi
- IOT

REFERENCES

[1] Cormen, T. H.; Leiserson C. E.; & Rivest R. L. (1990) Introduction to Algorithms. MIT Press

[2] Khatib, O.: Real-time Obstacle Avoidance for Manipulators and Mobile Robots.The International Journal of Robotics Research 5 (1986) 90–98

[3] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In Proceedings of the IJCAI-95 Workshop on Entertainment and AI/ALife, 1995.

[4] MicroMo Product Catalog. http://www.faulhaber-group.com/n41799/n.html

[5]OpenComputer Vision Library. http://www.intel.com/research/mrl/research/opencv/

[6] OpenGL Open Graphics Library. http://www.opengl.org/

[7] Tsai, R.Y. A versatile camera calibration technique for high accuracy 3D machine vision using off-the-shell TV cameras and lenses. IEEE Journal of robotics and Automation, 1987

[8] Verret, Ball, Kiat Ng. Laws of the F180 League Release 3.00a. http://www.itee.uq.edu.au/~wyeth/F180%20Rules/in dex.htm.