RoboCup-99 Team Descriptions 43
Small Robots League, Team *FutBotIII*, pages 43–47
http://www.ep.liu.se/ea/cis/1999/006/07/

FutBotIII: Towards a Robust Centralized Vision System for RoboCup Small League

FutBotIII

Gilberto Gallinelli, Gianmichele Grittani, José Ramirez

GGa, GGi, JR: Universidad Simón Bolívar

Abstract. *Past RoboCup competitions have shown the existence of many common problems with vision systems, that lead to bad tracking and loss of objects, causing loss of play time. The vision system is a fundamental piece of the soccer system, since it`s the main source of information. A robust and fault tolerant vision system is required, to allow the correct building and testing of better control systems upon it. The use of image processing techniques, together with statistics and a correct knowledge representation are good tools for solving these problems. Our goal is to study and test these techniques in order to contribute with a better vision system.*

1 Introduction

The experience of past RoboCup Small League competitions has shown that the vision system is a delicate part of the whole soccer system. Many teams lost games not due to an inadequate control system but because their vision system failed to recognize and track objects correctly. Our goal is to construct a software based, robust, fault tolerant vision system. In this paper we describe some problems encountered during past competitions, and their possible solutions implemented to be used for our team FutBotIII.

2 General Architecture

The system is made up of three basic parts: the robots, the vision system and the control system. These parts communicate with each other, interchanging data as needed to complete their tasks, as shown in figure 1.

2.1 Robots

We built our own robots on an architecture similar to that of the HandyBoard. Each has a M68HC16 microcontroller, two stepping motors with a maximum velocity of 2.5 m/s that allow a higher precision in movements,

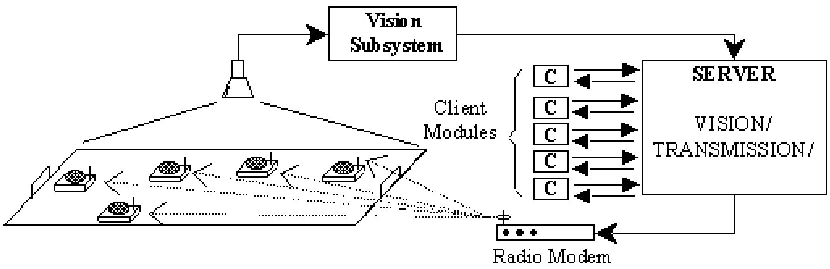

Figure 1: FutBot system schema showing communication between hardware and software parts.

two independent flippers mounted at the front end of the robot [1] that allow a better control of the ball and a radio system that receives commands sent from a transmitter attached to an off-board computer. The commands are low level movement, rotation and flipper angle motion instructions. Each robot is uniquely identified by a binary code, used to know if data broadcast is intended for it [2]. The robots only task is to receive these commands and execute them immediately.

2.2 Vision System

The main data gathering engine for the soccer system is a global vision system that captures a top view of the playing field. A Hitachi 3ccd color camera takes the scene and sends an NTSC video signal through a coaxial cable to a Matrox Meteor-II real time color frame grabber mounted on a PentiumII 300Mhz based computer running Windows NT [3] dedicated exclusively to image capturing and processing tasks. This computer has a serial link to another off-board PC dedicated to robot control. A frame is captured and processed, to extract the information which is sent to the control system computer. This lets us gain parallelism, since the control system can process the information obtained from a frame, while the vision system is processing the next frame.

2.3 Control System

The control system software runs on a Pentium 166Mhz under Linux. It is really made up of many independent processes that represent the brains of the five players. These processes act as clients to a server running on the same machine, that receives and serves the vision information given by the vision computer, to the clients. It also acts as the handler of the radio link with the robots. Then, clients would ask for visual information, receive it, decide what its controlled robot should do, and send the action to the server, for transmission to the robot.

[1]This idea was borrowed from the iXs robot.
[2]These codes work just like Ethernet addresses.
[3]This choice was forced by the libraries that came with the Meteor-II board.

3 Vision System Design

The vision system is a very important piece of software since it`s the only source of information about the state of the world. This means that the accuracy of the information is of extreme importance and since the introduction of noise into the frames is unavoidable, it is important to have a robust, fault tolerant vision system. Image processing speed is also an important issue, since it is necessary to keep a rate of processed frames high enough to capture object movements in time. In our case, objects would be separated by approximately 4 cm. between two frames at maximum speed. If the image processing software misses every other frame, we would have 8cm. between consecutive processed frames, which starts to become significant considering that robots fit in a 18 cm. cylinder. We cope with speed problems using efficient algorithms and processing images only where needed, together with the use of a fast processor.

The tasks of the vision system are to detect, recognize and track objects in frames, and where possible, give useful estimates of object positions under conditions of missing objects in frames.

3.1 Detection

Objects are found in a frame, by searching their color markers, according to the RoboCup Small League rules. Since differences in lighting and camera noise tend to change the way colors are perceived, the vision system is trained at startup with examples of the colors as seen. The program calculates the color to be searched for, plus a suitable threshold to cope with differences. To make the program faster, the entire frame is only searched the first time. Later, it is sufficient to search localities of the last place where the object was seen, unless we lose track of an object, in which case, we must search subsequent frames entirely until all objects are found again.

3.2 Recognition

The problem of uniquely identifying robots with different hues has been studied by Veloso et. al. [1], with the conclusion that it is unfeasible. We are studying ways to identify robots with monochrome signs that will code not only the robot's identity, but also its heading. The problem with this approach is the computational time required to process the sign. Many algorithms can be used to match the patterns, or some simple mathematical function (like moments) can be applied on the subimage to recognize the robot. We are studying the different possibilities to see if it is feasible to add this calculation to the image processing. If it`s not possible, then the best approach we have found is the one taken by Veloso et. al. [1].

3.3 Tracking

Once recognized the new position of the object, we can estimate its velocity vector, to predict future positions and make calculations based on these estimates. Since the ball has the additional problem that it can be hidden from view by the robots, when away from the center, we associate the ball

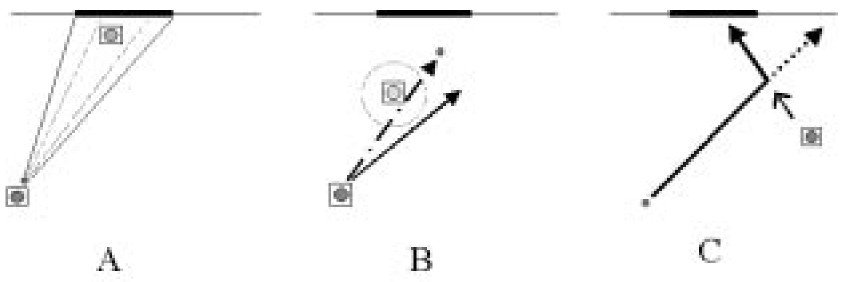

Figure 2: (a) Goalie behavior, minimizing free space at both sides of the goal. (b) Obstacle avoidance technique. (c) Interception of ball´s path.

to the robot that has the most chance of being the one hiding it. In other words, we consider the ball as being possessed by that robot. When the ball is visible, we use the same tracking technique described by Veloso et. al. [1] using a Kalman filter, since it gives a suitable estimate considering noise.

4 Control System Design

Since our main concern is creating a robust and fast software based vision system, our control techniques are simple enough to test such system and let us concentrate on tuning it.

4.1 The Goalie

We consider that after a strong vision system, the key to a good soccer team is a strong goalie. Its strategy is very simple: it seeks to minimize the contiguous free space on both sides of the goalie, as seen from the ball´s point of view, as long as the ball is within a certain threshold distance from the goal (figure 2(a)). Otherwise, the goalie keeps its position at the center of the goal.

4.2 Obstacle Avoidance

Since agents plan their movements according to their goals, they have to replan movements when obstacles are on their direct path. Simply the new direction is calculated aiming the vector towards the edge of the confidence area around the obstacle (figure 2(b)). Since there are two sides we can choose from, to avoid the obstacle, we choose the one which the obstacle is moving away from. This way we avoid a possible crash if both obstacle and robot are moving to the same side.

4.3 Ball Interception

Having the ball´s movement prediction, the robot just moves along the vector that intercepts the ball and bounces it towards the wanted direction (figure 2(c)). This is a simple calculation.

5 Discussion

We retain that the key for a good soccer team is a fast robust vision system. After that, a strong goalie. It is important to build a robust fault tolerant vision system, on which we can later develop a better control system, since it is useless to have an excellent control if vision doesn´t work correctly. The search for fast algorithms, processing only when and where it is needed, and achieving an adequate frame processing rate to gain precision on world measurements are ways to achieve this goal. We are building over the experience of other teams with their vision system, to attempt improvements using the good work of others as a starting base.

References

[1] Manuela Veloso, Peter Stone, Kwun Han, Sorin Achim Prediction, Behaviors and Collaboration in a Team of Robotic Soccer Agents. ICMAS proceedings (1998)

[2] Edward Dougherty, Charles Giardina Image Processing - Continuous to Discrete, Volume1. Prentice Hall. (1987)