Object Recognition in Robot Football Using a one Dimensional Image

Hatice Köse and H. Levent Akin

Bogaziçi University, Department of Computer Engineering, 80815 Bebek, Istanbul, TURKEY
{kose,akin}@boun.edu.tr

ABSTRACT

Robot football is a challenging task in the world of autonomous robots and related research areas. One of the main problems in this task is to determine the presence and the location of objects from the robot's point of view. This paper introduces methods to recognize the objects in a football field, by the aid of vision. The methods are tested on a simulator [1] of a Khepera [2] autonomous mobile robot with an onboard one dimensional, 256 gray level black-white camera. The objects in the field are the light gray walls, a simulated yellow tennis ball, and a black goal area. The specifications of the robot, and the field, are determined by the FIRA association [3]. Three methods are introduced in this work. In the threshold based method, a high pass filter is used multiple times to detect the rising and falling edges, then a threshold is applied over the detected edges to eliminate the false edges. After the detection of the edges, regions are detected. By using the number of regions, the mean value of each region, and the standard deviation of the regions, several rules, which are needed to classify the regions, are extracted. These rules use the relations between the objects, relative to each other. The second method is based on a neural network, and the last method uses fuzzy expert systems, to recognize the objects.

I. INTRODUCTION

It has been a challenging dream to apply AI and Robotics to real-life activities. Football is a good test bed for these fields for determining the level of their usability and robustness, since it allows a complex, dynamic and multi-agent environment and attracts the public attention with its entertainment side, as well. Detailed information related with Robot Football can be found in [4].

There are several issues related with Robot Football. Navigation, planning, perception and analysis of external data are the most vital issues. Generally vision is used for the detection of presence and location of the objects in the football field, relative to the robot. Also in some works, infrared sensors are used to refine the vision data, with the vision module.

The vision module, which is a specific hardware, gathers the vital data about the football field. Then this data is analyzed and used by the robot to decide the next move. In one approach, the vision module is a global top camera, and a main computer processes the visual data taken by the camera. The results are then send to the robot. In another approach, vision module is onboard. In this work, the second approach is followed.

The aim of this study is to recognize the objects in a simulated football field, by the aid of a one dimensional, black-white camera. The output of the camera is a linear array of 1*64 data points of 256 gray levels. The objects are the light gray walls, simulated yellow tennis ball, and a black goal area, as stated before. Since the image contains only gray level data, it is not very hard to detect the objects by using their colors, and since the image is one dimensional, the shapes of the objects cannot be detected. Just the gray level of the objects relative to each other is known. Also the distance between the robot and the objects affect the tone of the objects recognized by the robot. A ball could have the same color tone with the wall when it is near and, and its color tone would get closer to goal when it is far away from the robot. This is the challenging problem. It is not trivial to recognize the objects from the raw image.

So several different approaches were tried. The first approach is to use thresholds, and rules to distinguish the objects. The second approach is to use a fuzzy expert system and third approach is to use a neural network. The details of these approaches could be found in the following sections.

After the implementation of these methods, the output taken from this methods can be used in robot football. Our main aim is to implement a football game by using a Khepera mini-autonomous mobile robot. The robot would use the visual output coming from the methods, which are described in this paper to detect the presence and the location of the objects in the football field. Then this data would be used to develop strategies to play the game. Simple behaviors would be constructed based on this data., like “search for the ball”, “approach to the ball”, “dribble the ball towards the goal area”, “avoid collisions to the opponent player and the walls”. To control these behaviors a layered architecture could be used. The design and the control of the behaviors are the subjects of a future work.

II. BACKGROUND

There are several works on using vision in robot football [5,6]. Most of these works are related to 3D color cameras. We are interested in the works, which use a similar development environment as our work. In [7], a robot simulator with 2D color camera is used to detect the goal area. A modified version of the Sobel algorithm is used in this work. In [8], control algorithms for Khepera [2] robots are implemented, using clipping, high pass filters and low pass filters with mask of 5 pixels. Also Fourier transformations are implemented on the visual data in this work. In Danish Championship in Robot soccer, which is done in 1997, and 1998, Khepera mini mobile robots were used for robot soccer. A threshold based method is used for vision in [9], which is the one of the participants of the tournament in 1997.

III. DEVELOPMENT ENVIRONMENT

In this section the development environment namely the robot and the simulator used will be described.

III.1 Khepera

In the real world tests, a mini robot of Khepera type shown in Figure 1 will be used. Khepera is a mini mobile robot (5 cm diameter, 70 g.) developed at EPFL - LAMI [2]. It is commercially available from K-Team S.A. It allows real world testing of algorithms developed in the simulation for trajectory planning, obstacle avoidance, pre-processing of sensory information, and hypotheses on behavior processing, among others. It has two motor driven wheels and is equipped with 8 infra-red sensors allowing it to detect obstacles all around it in a range of about 5 cm. These sensors can also provide information about light measurement.

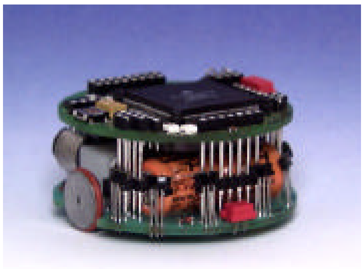


Figure 1. Simple Khepera Robot with no external turrets.

III.2 K213 Vision Turret

The K213 turret is the vision module of Khepera. The K213 Vision Turret is mainly composed two sensors, specific hardware and a processor. There are light intensity and image sensors. The image sensor is a linear array of elements that charge themselves under light exposure. This module does not have auto-iris. The light sensor is needed to give a scanning frequency, which fits the scanning needs of the image sensor. This action corresponds to auto-iris feature.

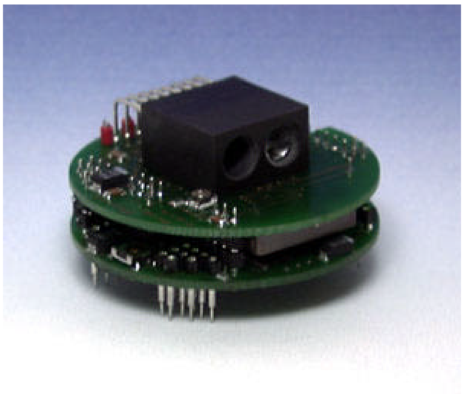


Figure 2. K213 Vision Turret

III.3 Webots Simulator

At the moment the algorithms are tested in a simulated version of the real robot and the vision module. But algorithms are designed robust and efficient enough to work on the real robot and real world environment. Noise is added to the simulator to make it closer to the real world environment.

We use Webots, which is a realistic mobile robots simulator. It includes models for the Khepera and Alice the robots as well as extension turrets for obstacle detection, vision, gripping objects. The user can program virtual robots using a C / C++ library which is compatible with the real the robot. A 3D environment editor allows customizing the robotics scenarios [1].

IV. APPLICATION

In Figure 3, a screenshot from the game is seen. Robot is standing in the middle of the field, and looking towards the blue goal area. It sees the yellow ball, the goal area and the walls. In Figure 4, the analyzed image of the screenshot from the point of view of the robot is seen. The walls are denoted by ‘o’, the ball is denoted by ‘*’, and ‘+’s show the goal area.

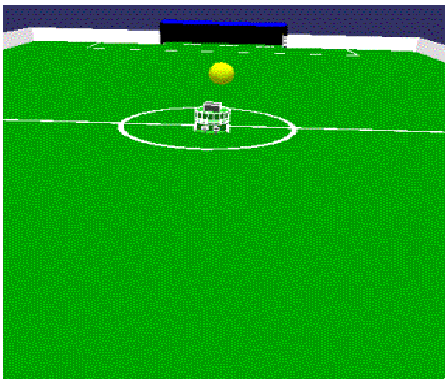


Figure 3. A screenshot from the game

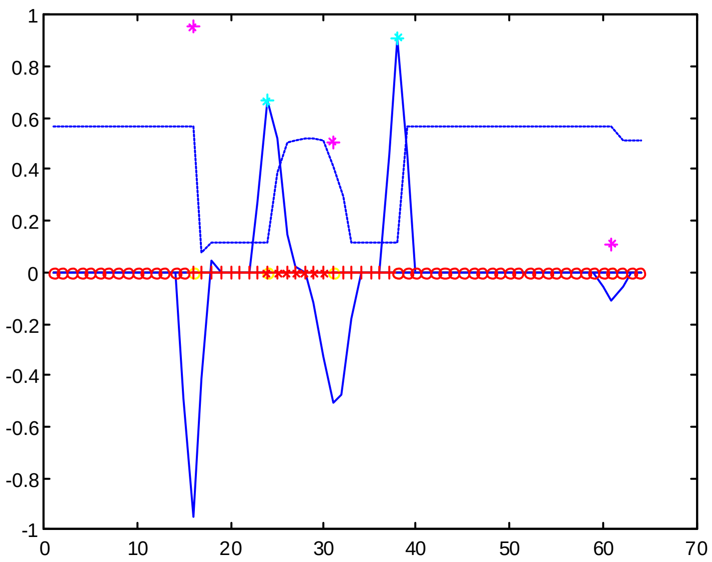


Figure 4. The analysis of the screenshot in Figure 3, from the robots point of view

IV.1 Threshold Based Method

The raw data is not adequate to give useful information to identify the regions in the image. So high pass filtering is used to magnify the region boundaries in the image, so it becomes easier to detect the regions. A mask of [-1 -1 1 1] is applied to detect rising edges and another mask of [1 1 -1 -1] is applied to detect falling edges. The output of the application of the masks to all of the data points is zero or a small number when the image is flat and becomes a relatively high number in case of edges in the image. These edges are the boundaries of the regions in the image and are represented as peaks in the filtered image. So a rising edge, which is represented by a positive peak shows the starting point of an object, and the falling edge which is represented by a negative peak symbolizes the end point of the object. As the color difference between the objects become higher, the height and sharpness of the peaks between them also becomes higher. After the detection of the edges, a threshold is applied over the detected edges to eliminate the false edges, caused by the noise. In Figure 5, the peaks show the rising and falling edges. The peaks with '*' signs are the rising edges. (The dotted lines show raw data and the solid lines are the filters.). Notice that there are false edges caused by noise, and the irregularity of the ball object. By applying a threshold, most of these false edges are eliminated. But the big ones, like the one just before the last peak, remains. A high threshold would eliminate it, but would cause a serious loss in the data.

After the detection of the edges, regions are detected. The number of edges determines the number of regions. Notice that the number of regions is not exactly half of the total number of edges. This is because some regions are enclosed by just one edge, not two. So the greater of the number of rising and falling edges, give the number of regions. If the number of regions is one or two, then the number of regions is equal to the number of objects, exactly. But if the number of regions is more then two, it is not trivial to tell the number of objects. There could be three objects, or two objects, but one object would be partitioned. In this case, the sensor data is used.

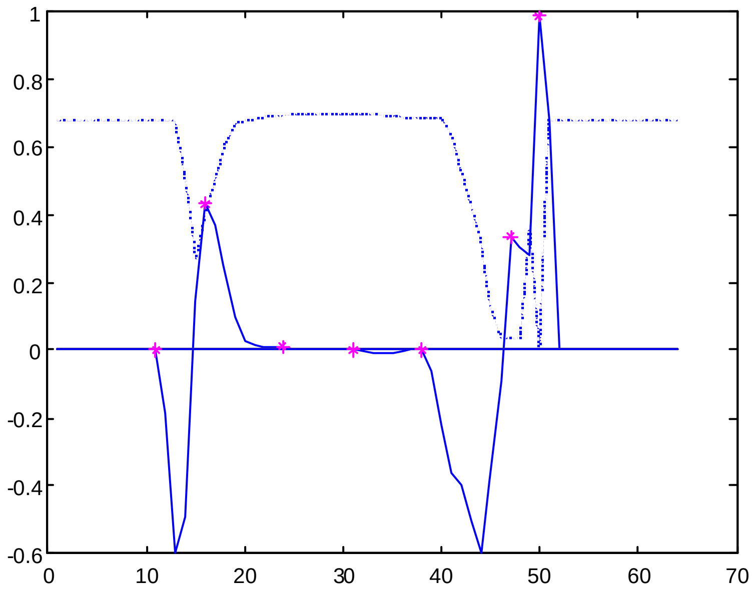


Figure 5. Rising and falling edges, before thresholding

Objects have different gray levels with respect to each other. By using these relationships it is possible to detect rules, and label the regions as ‘wall’, ‘ball’, and ‘goal area’. Unfortunately, the results are not 100 percent correct, due to noise, and the color differences between objects change due to the change in the distance between them. Besides the color of the walls and the ball are not very distinct, even for the human expert.

A training set of 110 states was prepared to train the algorithm (each state has 64 data points). 10 states of wall-ball, 10 state of wall-goal area and 90 states giving the interaction between ball, goal area and robot. Robot, and ball are positioned in different distances (near, middle, far), and positions (left, center, right). Then the data points read by the onboard camera, from these positions are taken as training set. The threshold values were gathered manually from the detailed analysis of the training set of data.

A modification based on standard deviation is made to this algorithm to improve the success rate (Figure 7). The algorithms, which are based solely on the knowledge about the means of the regions, are not so robust because as the location of the objects relative to the robot change, their brightness also changes, so there are no strict rules that can be extracted to distinguish the objects. The walls would be brighter than the ball, if the ball were far away from the robot, whereas the ball becomes brighter than the walls if it is near to the robot. Besides since a linear one-dimensional data is gathered, the shape information of the objects is not available. Fortunately, a feature of the ball makes it distinguishable from the other objects. Since the ball is a rough object unlike the rectangular wall and goal, the points belonging to ball have a distinguishably high standard deviation from their mean, relative to the points representing the wall and goal objects. But the presence of noise, taking the data from different angles and distances, and some bugs in the simulator prevents a 100 percent success even when the standard deviation is used. The modified version of the algorithm is given in Figure 7.

```
mask=[-1 -1 1 1] // for rising edges
mask1=[-1 -1 1 1] // for falling edges

//finds the rising edges
for i=1 to all data in a state
    filter(i)=filter(i)+training data(i+j)*mask(j) // j=-1..2 //mask is applied to a
window of size=4
for i=1 to all data in a state
 find the peak points in the filter // these peaks give the location of edges

//finds the falling edges
for i=1 to all data in a state
    filter(i)=filter(i)+training data(i+j)*mask1(j)
for i=1 to all data in a state
 find the peak points in the filter

for i=1 to all data in a state
 detect the number of regions between the edges

if number_of_regions==1
    then
        number_of_objects=1
    else
        if number_of_regions==2
            then
                number_of_objects=2
            else
                number_of_objects=3
for i=1 to all data in a state
 detect the mean of all the points in each region
threshold1=0.4 // calculated by analyzing the 90 states
//The below part is modified in the second version of the algorithm
if the mean in a region is < threshold1
    then
        region is goal area
    else
        if # of objects=2
            then
                if mean of a region is>threshold2 // threshold2=0.7
                    then
                        region is wall
                    else
                        region is ball
if # of objects=3
    then
        region with highest mean is the wall and
        the other is the ball
// In some cases the walls are detected as ball and ball is detected as walls.
// Since there can not be two pieces of walls, this ambiguity is solved as below:
for i=1 to all data in a state
 detect the # regions which are labelled as ball
if # of regions labelled as ball>1
    then
        label the ball regions as wall,
        and wall regions as ball
```

Figure 6. Threshold Algorithm

```
if the mean in a region is < threshold1
    then
        region is goal area
    else
        if # of objects=2
            then
                if standard_deviation of a region is>threshold2 % threshold2=0.3
                    then
                        region is ball
                    else
                        region is wall
```

Figure 7. Modified threshold Algorithm

IV.2 Neural Network Based Method

In order to compare the results a neural network was used to detect the presence and the position of the ball [10]. The detailed information about the neural network used in this study is given in Table 2. The system has three position variables as, *near*, *medium* ,and *far* for the relative position of the goal area and the ball to the with respect to the robot unlike the other two methods.

Table 1. Parameters of the Neural Network

Parameter	Value
Number Of Input Units	65
Number Of Hidden Units	4
Number Of Output Units	2
Number Of Epochs	100000
Learning Rate	0.05
Momentum	0.90

IV.3 Fuzzy Logic Based Method

The Fuzzy Logic [11,12] based method is constructed by redefining the thresholds in the threshold based method in terms of linguistic fuzzy terms, and membership functions. In this work three different fuzzy expert systems (FES) were developed. They differ by the rules and membership functions used. The output variable *object* is calculated by Takegi-Sugeno type fuzzy control mechanism, in three of the expert systems. The expert systems are called *version 1*, *version 2*, and *version 3*. In the rules part, variable MEAN stands for variable *mean_of_the_region*, and STDEV stands for *standard-deviation_of_the_region* (Figure 8).

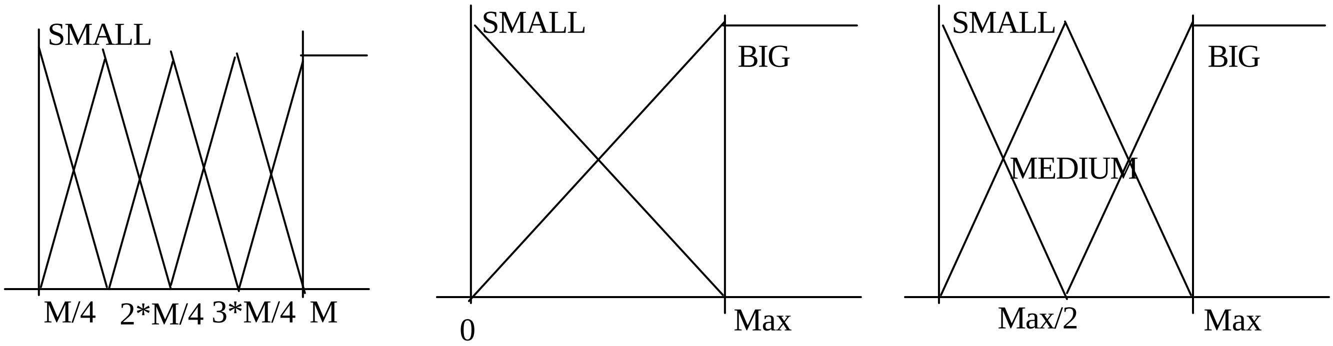
In version 1, just the membership function SMALL is used. The use of others did not make an improvement in the success so excluded. M is 1 and Max is 0.1. For the Mean of version 2, and version 3, Max is 1. For STDEV of version 2, and version 3, Max is 0.07 (Figure 9). These numerical values are gathered from the careful analysis of the statistics taken from the behavior of the train data, by the human expert.

```
For i=1 to number of
regions
 if MEAN is SMALL AND
STDEV is SMALL
    then object is GOAL
  if MEAN is NOT SMALL AND
STDEV is SMALL
    then object is WALL
 if MEAN is NOT SMALL AND
STDEV is BIG
    then object is BALL
            (a)
```

```
for i=1 to number of
regions
 if MEAN is SMALL
    then object is GOAL
  if MEAN is BIG AND STDEV
is MEDIUM
    then object is WALL
 if MEAN is MEDIUM AND
STDEV is BIG
    then object is BALL
            (b)
```

```
for i=1 to number of
regions
 if MEAN is SMALL
    then object is GOAL
  if MEAN is BIG AND STDEV
is MEDIUM
    then object is WALL
 if MEAN is MEDIUM AND
STDEV is BIG OR MEDIUM
    then object is BALL
            (c)
```

Figure 8. Rules of the Fuzzy Expert System. (a)Version 1 (b) Version 2 and (c) Version 3



(a) (b) (c)

Figure 9. Membership functions denoting the variables MEAN and STDEV for all versions (a) Membership Functions for MEAN for version 1 (b) Membership Functions for STDEV for version 1 (c) Membership Functions for version 2 and 3, for MEAN & STDEV

V. TEST RESULTS AND DISCUSSION

The algorithms were tested using a data set collected at 100 positions taken from the simulator, while the robot and the ball objects are moving, stationary, and at different angles and distances from each other, including the walls and the goal area. In the following, *rate of successful classifications* describes the ratio of the successfully detected locations to the successfully detected presence of the objects. This ratio shows how successful the methods are, in classifying the objects after detecting the presence of them.

It should be noted that the success of the algorithms depends strongly on the thresholds that are used in the algorithms.

Table 2. Success rate of the algorithms

	Threshold method Without Standard Deviation (Percent)	Threshold method With Standard Deviation (Percent)	Neural Network Success Rate (Percent)	Fuzzy Expert System Success Rate (Percent) (Version 1)	Fuzzy Expert System Success Rate (Percent) (Version 2)	Fuzzy Expert System Success Rate (Percent) (Version 3)
Detecting the presence/absence of the ball	78	90	75	78	83	94
Detecting the location of the ball	48	77	42	74	79	77
Rate of successful classifications	62	85	58	95	95	82

Rate of successful classifications is the highest in the Fuzzy Logic based methods. Notice that, as the parameters of the fuzzy membership functions change, the success improves. Dependent on the distance between objects, and the robot, the features of the objects change. So it is hard to restrict the specifications of the objects with rules. Especially, when both the walls and the ball are far away from the robot and close to each other, their Mean and Stdev get closer. In this case, even for the human expert, it is not trivial to differ them. In the FES version 3, a modification is done to version 2 to improve the success. This modification allows the FES, accepting the objects with Medium Stdev as ball if their Mean is Medium. So the lower limit of the Stdev of the Ball object is decreased. This allows the robot to see the ball when it is too far from the robot so, becomes too small (this case is ignored in similar works generally). Unfortunately, in this case, the walls could also be recognized as ball, by mistake. So, this fact decreases the rate of classification although success in detecting the presence of the ball increases. But a simple filtering could solve this problem. There cannot be more than one Ball object so the object with highest Stdev would be taken as Ball and the other wrong Ball objects would be classified as Walls. As well, in version 2, in cases where the ball is too near and behind the walls there are walls, just the two sides of the ball are detected as ball, and the middle of the sides is detected as wall. A simple filtering could be used to combine these two sides, to solve the problem.

After the test phase, it is observed that, the most successful method is the Fuzzy Logic method. But this is not all. The success of the methods is strongly related to the parameters that are used, so after the refinement of the parameters, the success of the other methods would be improved. The thresholds of the threshold based method, and the parameters of the neural network approach could be refined. Also the success of the fuzzy logic based method could be improved by defining new rules and new parameters for the current rules.

VI. CONCLUSION

The aim of this study is to recognize the objects in a simulated football field, by the aid of a one-dimensional, 256 gray level camera. Several methods are implemented to solve this problem and the output of these methods is the absence/presence and the location of the objects in the field. First method is purely based on the thresholds defined by a human expert, and in the other two methods, instead of thresholds a neural network and a fuzzy expert system are used. For test and train phases data taken from the simulator is used. The data is taken in a dynamic environment including noise, so it is successful in simulating the real world case. Besides the robot is moving during the collection of data, so the data becomes quite hard to learn, and problematic cases were added to data to gather more robust algorithms. No assumptions, such as ignoring objects below a given size, are allowed. Whereas, these assumptions are generally used in similar works. Likewise, distance, orientation and location of the objects are not limited. Moreover, additional control mechanisms like extra filters and checks, other then the FES tested in this work, are not used to improve the success. These affect the success of the methods. Higher success would be gained in case of data gathered from simpler states, or assumptions were used.

In the real world case, the usage of neural networks and FES would be more robust, then the threshold algorithm. Also a hybrid approach such as a neuro-fuzzy system could be used for improving the success.

The nature of the peaks could be used the refine the results. The past data would be also useful in problematic cases. Usage of extra filters would improve the success in problematic cases, too.

As an object gets nearer to the robot, it becomes brighter. So a near ball is brighter than a far wall, whereas it is not as bright as the wall, when they are equidistant to the robot. So the algorithm could detect the existence of a ball, but could misclassify as the wall object, and could be wrong in detecting the position of the ball. When the ball is far, it is not a very important problem, but if the ball is critically near, it is vital to detect the position of it. Also sensor knowledge would be helpful in this case, to detect the distance of the ball to the robot.

Although the data is noisy, and task is hard due to the inefficiency of the vision module, the methods show a good performance, besides they are simple and rapid. So they could be successfully implemented in a real robot, and it is expected to get a good performance, in the real world tests.

After the implementation of these methods, their output can be used in robot football. The design and the control of the robots behaviors are the subjects of a future work.

ACKNOWLEDGEMENTS

This project is supported by the Bogazici University Foundation and by Bogazici University Research Fund project 01A101. The authors would like to thank Dr. Ethem Alpaydin and Dr. Lale Akarun for valuable discussions. Also thanks to M.Toygar Karadeniz for his kind help in the implementation of the neural network method.

REFERENCES

1. Cyberbotics home page, http://www.cyberbotics.com/, 2001.
2. K-team home page, http://www.k-team.com/, 2001.
3. FIRA Home Page, http://fira.net/, 2001.
4. Robocup Home Page, http://www.robocup.org/, 2001.
5. Garcia, R., et al.," A Fast Vision System for Tracking in Robotic Soccer", FIRA Robot World Cup France'98 Proceedings, 1998.
6. Shen, W., et al, "Building Integrated Mobile Robots for Soccer Competition", In Proceedings of 1998 International Conference on Robotics and Automation, 1998.
7. Ornella, E.," Robot Soccer", http://www.cyberbotics.com/resources/soccer/index.html, 2001.
8. Pantzer, T., "Kcontrol Algoritm",http://www.informatik.uni-leipzig.de/~pantec/khepera/, 1999
9. Ilsoe, M., J., Jensen, and H. Madsen, "Anthony 97", http://www.dcs.qmw.ac.uk/RFC/ARFT98/anthony/anthony.ps, 1997
10. Freeman, James A., "Neural Networks:Algorithms, Applications, and Programming Techniques", Addison-Wesley, 1991.
11. Pedrycz, W., "Fuzzy Control and Fuzzy Systems", Research Studies Press, 1993
12. Takagi T., and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control", IEEE Trans. Syst. Man. Cybem., Vol.15. No.1. January/February, pp. 116-132. 1985