Elektrotehniški vestnik 68(5): 306–312, 2001
Electrotechnical Review, Ljubljana, Slovenija

Robust and efficient vision system for mobile robots control—application to soccer robots

Gregor Klančar[1], Omar Orqueda[2], Drago Matko[1], Rihard Karba[1]

[1]*University of Ljubljana, Faculty of Electrical Engineering*
Tržaška 25, SI-1000 Ljubljana, Slovenia
[2]*National Council for Scientific and Technical Research (CONICET), Department of Electrical Engineering, National University of the South, Avda. Alem 1253, (8000) Bahía Blanca, Argentina*
E-mail: gregor.klancar@fe.uni-lj.si

Abstract. In the paper a global vision scheme for estimation of positions and orientations of robots is presented. It can be divided into two steps. In the first, the incoming image is scanned and pixels are classified into a finite number of classes. At the same time, a segmentation algorithm is used to find corresponding regions belonging to one of the classes. In the second step, all the regions are examined. Selection of the ones, that are a part of the observed object is made by means of simple logic procedures. The novelty is focused on optimisation of the processing time needed to finish the estimation of possible object positions. Also, an application of this algorithm to robot control is presented. Reference trajectories are generated by the fourth order Beziér curves and a nonholonomic trajectory follower controller is used to assure the robots to follow the prescribed curves.

Key words: computer vision, multiple thresholding, classification, segmentation, nonholonomic control

Zanesljiv in učinkovit računalniški vid za vodenje mobilnih robotov in njegova aplikacija v robotskem nogometu

Povzetek. V članku je predstavljen razvoj globalnega računalniškega vida, uporabljenega za ocenjevanje pozicij in orientacij objektov pri robotskem nogometu. V grobem lahko njegovo delovanje povzamemo v dveh korakih. Z uporabo večkratnega hkratnega upragovljanja izvedemo rojenje slikovnih elementov slike v končno število rojev. Nato iz razvrščenih slikovnih elementov z algoritmom za razčlenjevanje slike na področja določimo enovita področja na sliki. V drugem koraku z uporabo metod preprostega sklepanja izmed vseh področij slike določimo tista, ki so del objektov razpoznavanja.

Predstavljeni pristop odlikuje kratek čas, potreben za obdelavo celotne slike. Prikazan je tudi primer aplikacije robotskega vida pri vodenju robota. Z uporabo Beziérjevih krivulj četrtega reda je določena referenčna krivulja, ki jo skuša robot slediti. Pri tem je uporabljen neholonomski regulator.

Ključne besede: računalniški vid, večkratno hkratno upragovljanje, rojenje, razčlenjevanje slike, neholonomsko vodenje

1 Introduction

In recent years, mobile robots playing soccer have gained much popularity among researchers worldwide. This is mainly due to the fact they serve as an excellent test bed in several areas of research interests, such as path planning, obstacle avoidance, multi-agent cooperation, game strategy, real-time data and image processing, robotic vision, artificial intelligence and control. There are also many challenges in mechanics, e.g. how to make robots smaller, faster, equipped with many sensors or in other words - how to make them better. The area has proven to be an excellent approach in engineering education not only because of the reasons explained above but also because its attractiveness [9]. Students can get the results of their achievements through the game and immediate feedback enables them to find bugs in their algorithms.

The paper presents a design of a global vision system for estimating current object positions and orientations on the playground. The MiroSot category soccer robots we are interested in are without on-board sensors. Thus a precise and fast global vision has to be designed for robots control and navigation in an unknown, dynamically changing area. When designing the vision system, the following requirements have to be accomplished:

- computational efficiency,
- high reliability,
- good precision, and
- robustness to noise, changing of lighting and different colour schemes.

Received 5 February 2001
Accepted 25 September 2001

The last characteristic is essential for the system to function well when using it under different conditions present at competitions [14].

With colour cameras, there are many possible ways to carry out the detection of robots wearing colour dresses. All of them try to classify pixels of an image into one of a predefined number of classes. The most common approaches are: linear colour thresholding, K-nearest neighbour classification, neural net-based classifiers, classification trees and probabilistic methods, [10], [1], [8], [11].

In this paper, a fast approach with constant thresholding and back-stepping algorithm is presented where a special effort is put into the efficiency aspect. The thresholds can be presented as boxes in 3-dimensional colour spaces. These thresholds are determined by means of off-line learning. In the first step, if an incoming pixel colour falls inside one of the predefined boxes, then this pixel is classified as belonging to the class associated with this box. In the second step, the pixels belonging to one class (a connected region) are univocally labelled. With the main purpose of obtaining all fully connected regions, a back- stepping algorithm is applied. Both steps are done with just one scan of the image. Then the logic part and a simple optimisation method are employed to select the proper regions from the previously generated ones. After this logic, the positions and orientations of the objects on the playground are estimated. In the last part of the program, the reference path generation and control of a single agent are used.

The paper is organised as follows. In chapter 2 a brief overview of the system is given. The method used for pixel classification is explained in chapter 3. Chapter 4 focuses on the algorithms for image segmentation and region labelling. The algorithm for object estimation is illustrated in the next chapter. Chapter 6 resumes the control system applied to the soccer robots. Experimental results are shown in chapter 7. The paper ends with conclusions and some ideas for future work.

2 System Overview

The soccer robot set-up, Fig. 1, consists of six MiroSot category robots (generating two teams) of size 7.5cm cubed, rectangular playground of size 1.5×1.3m, JAI MCL-1500 camera, frame-grabber Matrox Meteor 1, and 350 Mhz Pentium II computer. The vision part of the program processes the incoming images, of a resolution of 376×291 pixels, to identify the positions and orientations of the robots and the position of the ball. Finally, the control part of the program calculates the linear and angular speeds, v and ω, that the robots should have in the next sample time according to current situation on the playground. These reference speeds are sent to the robots by a radio connection.

To identify the orientations, each robot has to have two colour patches. One is the team colour and the other is the identification colour patch. According to FIRA (Federation of International Robot-soccer Association) rules, the team colour must be blue or yellow, the ball must be orange and identification colours can be any colour except the team and ball colour. The patche positions and shapes can be chosen freely. In our case, square patches were used. They were placed diagonally, with the team colour being closer to the front part of the robot, Fig. 2.

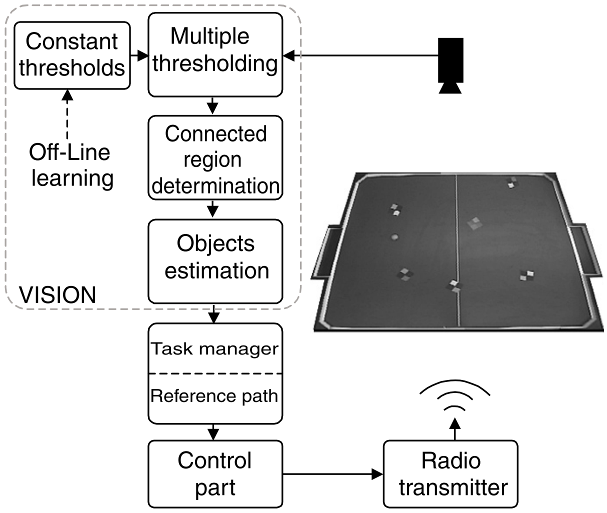

Figure 1. System overview

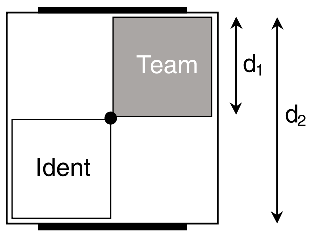

Figure 2. Colour patches on the robot

3 Pixel Classification

To enable detection of different colour patches, each pixel has to be classified into one of the predefined colours.

3.1 Colour Space Transformations

The colour image can be presented by the use of different colour space notations such as *RGB*, *HSI*, *YUV* and others. By using the *RGB* notation the regions with the same colour are best presented in a three-dimensional colour space with conical volumes. On the other hand constant thresholds form blocks (rectangular prism), which make the description of colour regions rather difficult [2].

When using simple thresholds for pixel classification, *HSI* and *YUV* colour spaces are most appropriate. They code the information about chrominance in two dimensions (*H* and *S* or *U* and *V*) and only one dimension includes the information about intensity (*I* or *Y*). With these colour spaces, any particular colour on the playground can be described with wide areas between thresholds in the intensity dimension, while the threshold areas for other chrominance dimensions are narrow. However, these colour spaces are more robust to different lightening conditions.

Both *RGB* and *YUV* spaces were used in our experiments with constant thresholds, and, as expected, the second colour space gave better classification results. However, to obtain the *YUV* colour representation, a transformation from the original *RGB* space had to be done. This transformation was time consuming although the optimisation by using lookup tables was used. Including the optimisation it requires some 30 ms, while the rest of the program takes only 10 ms to identify objects from the image. Therefore, *YUV* or any other colour space should be used only when it can be directly obtained from the frame grabber.

3.2 Thresholding

The basic idea is to classify each pixel according to the remembered colour thresholds of each object. Initially, this part was done with the following code:

```
for i=1 to number of colours on playground
  if (R >= R_lower_boud AND R <=R_upper_bound AND
      G >= G_lower_boud AND G <=G_upper_bound AND
      B >= B_lower_boud AND B <=B_upper_bound  )
    Pixel_colour =i-th colour;
  else Pixel_colour=background;
end
```

This simple part of the code requires 6 relational and 5 AND operations for each pixel classification. This code is repeated for each colour that we want to classify.

To improve this operation, i.e. to check all colours at the same time, an idea of parallelism was considered. Three $N\times$32-bit integer arrays were allocated. Where N corresponds in size to the number of colour levels (usually N=256) and the maximum number of colours to be classified is 32 respectively (Fig. 3). Each bit in a 32- bit memory location is associated with one colour. Although the algorithm is able to classify 32 colours, only 9 different colours are enough for the purpose of the robot soccer game. Because the computer microprocessor has 32-bit arithmetics, the computational burden is the same as for only the 16-bit memory location.

Let us suppose that we want to classify a yellow patch with the colour values in the following range:

```
200 <= R <= 220
230 <= G <= 250
10  <= B <= 30
```

Suppose this colour is associated with the 31st bit of each memory location. So in this case the highest bits in the memory locations between 200 and 220 in *RClass*, 230 and 250 in *GClass* and 10 and 30 in *Bclass* are set to 1, respectively. The same procedure is performed also for bits 0-30 for other colour patches.

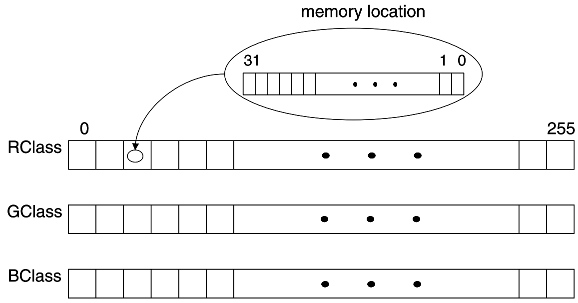

Figure 3. Look-up tables for pixel classification

At the run time, memory locations with the index corresponding to current pixel *R*, *G*, *B* values are taken. The bitwise AND operation between the chosen memory locations gives the information about the classification of pixels. If the result has the 31st bit set to 1, then the pixel is recognised as yellow.

With this methodology, a multiple thresholding (the thresholds for all colour patches checked at the time) is made in only one scan of the image. As the multiple threshold operation takes just two AND operations, it significantly reduces computational burden.

4 Image Segmentation and Component Labelling

To estimate patch positions, first the regions belonging to the ball, team and opponent team patches and all identification patches have to be located. The number of those regions on the playground (K) can be higher than the number of all patches due to noise. Image segmentation in K regions and labelling is done fulfilling the following five conditions [10]:

i. $\bigcup_{i=1}^{K} R_i = R$,

ii. $R_i \bigcap R_j = \emptyset$, $i, j = 1, 2, ..., K$, and $i \neq j$,

iii. R_i is a connected region of pixels,

iv. $P(R_i) = 1, \forall i$,

v. $P(R_i \bigcup R_j) = 0$, $i \neq j$, and R_i, R_j are neighbours,

where $P(x)$ is a logical predicate, which takes the value one if all the pixels of the region accomplish a criterion of homogeneity. In our case, the homogeneity criterion is the equality in colour.

According to the first and second conditions, the regions R_i together must occupy the entire image R and the regions must not have common pixels. Due to the third condition, there must be at least one path of pixels of the same colour connecting any two pixels in the region. Moreover, the regions must be homogeneous with respect to the colour and the neighbour regions must not have the same colour, as stated in conditions four and five.

4.1 Image Segmentation and Labelling Algorithm

Pixel classification and image segmentation are merged by the aid of a corresponding algorithm. Its main property is that the image classification and segmentation is done with only one pass through the image, what considerably contributes to time efficiency. The results of the mentioned algorithm are labelled regions with the following information:

- region number,
- colour,
- number of pixels belonging to this region,
- pointers to each pixel belonging to this region,
- coordinates of the centre of the region, x_{avg} and y_{avg}.

Before the algorithm starts running, one colour space is selected for each component to be identified: ball, robot1, robot2, robot3, team, opponent robots and opponent team.

Algorithm:

- The algorithm starts analysing the 1st pixel of a given region of interest (in our case the whole image).
- If the colour of the pixel under study is a *valid colour** and is different from the colour of the upper and left neighbour pixels, a new region is created (Fig. 4a).
- If the colour of the pixel under study is a valid colour and is equal to the colour of the upper or left pixel, then the pixel under study is added to the region of the upper or left pixel (Fig. 4b).
- If the colour of the pixel under study is a valid colour and is equal to the upper and left pixel colours, then (Fig. 4c):
  - if the upper and left pixels belong to the same region, the pixel under study is added to this region,
  - otherwise, the pixel under study is added to the region with a bigger number of pixels, the pixels of the region with lower quantity of elements are copied to the bigger region, and then the region is deleted.

**"valid colour"* means that the colour belongs to one of the predefined colour labels.

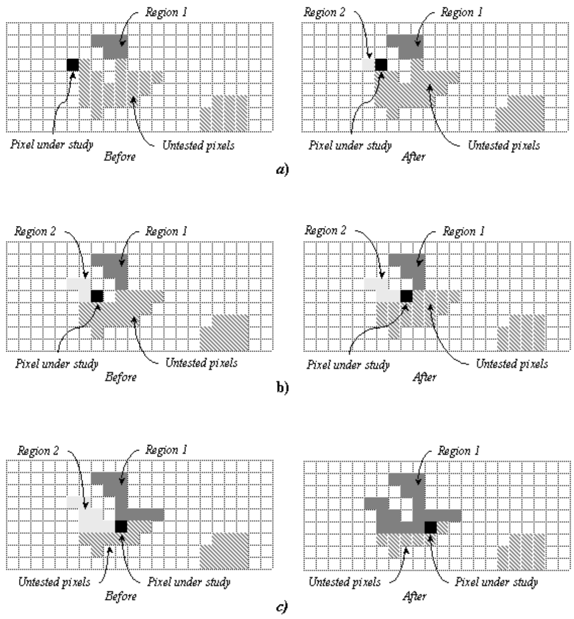

Figure 4. Image segmentation and labelling

5 Object Estimation

From all the possible valid regions identified as described in the previous chapter, a proper number of regions with the biggest area is selected. This step maximises the probability that correct regions are selected and not the ones due to noise.

First, the team and opponent team regions are investigated. The region is a probable team patch if it is classified as team colour and if the positions of other team patches satisfy the condition:

$$dist(region, team_i) > \frac{3}{4}d_1, \tag{1}$$

where $dist$ is Euclidean distance, $region$ is the current testing region, $team_i$ are already chosen team regions and d_1 is the size of the colour patch (Fig. 2).

To find the right identification region among all which are classified as a particular identification colour, the following condition must be fulfilled:

$$\frac{3}{4}d_1 < dist(region, indent_i) < d_2, \tag{2}$$

where $region$ is the current testing region, $ident_i$ are already chosen other identification regions (other robots) and d_2 is the robot size (Fig. 2).

A table of possible pairs is generated from selected team and identification regions with rows presenting selected identification regions and columns presenting selected team regions. The entries in the table are set to 1 if the condition (2) is true. The correct pairs (team, identification) are then found with a simple procedure:

- if the row has just one element equal to 1, then the element indices represent the correct pair,
- if the column has just one element equal to 1, then the element indices represent the correct pair,
- for the robots that can not be identified by the above two conditions the row associated with the unidentified robot is investigated and between possible team regions the one that has not been chosen yet is selected.

The same procedure is repeated for opponent players. The procedure is explained in Fig. 5.

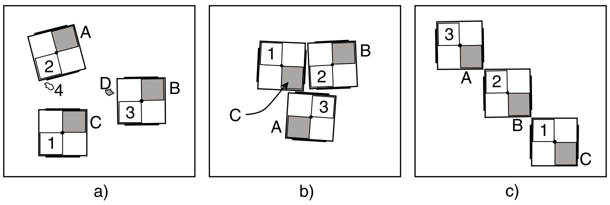

Figure 5. Different robots placement

In Fig. 5, team patches are shaded and marked with different letters, while the identification patches are marked with numbers.

a)	A	B	C
1			1
2	1		
3		1	

b)	A	B	C
1			1
2		1	1
3	1		1

c)	A	B	C
1		1	1
2	1	1	
3	1		

Table 1. Tables of possible pairs from Fig. 5

From the first table three pairs can easily be found (1C, 2A, 3B). The regions 4 and D are not considered because their area is small, and are probably due to noise. From the second table, taking the first row and the first two columns, all pairs are found (1C, 2B, 3A). In the last table, taking the last row and the last column, the pairs 1C and 3A are found. The second robot is not found, so the only possible team colour not chosen yet in the second row is B, the pair is 2B.

When the right regions representing colour patches are found, the final estimated patch position can be improved by taking the weighted average of all region positions with the same classified colour and less than distance d_1 away.

From known positions of the regions belonging to the objects, the object positions and orientations are calculated. The position of the ball is equal to its region position, while the i-th robot data (position x_i, y_i and orientation φ_i) are calculated as follows:

$$\begin{bmatrix} x_i \\ y_i \\ \varphi_i \end{bmatrix} = \begin{bmatrix} \frac{x_{T_i}+x_{I_i}}{2} & \frac{y_{T_i}+y_{I_i}}{2} & \tan^{-1}\left(\frac{y_{T_i}+y_{I_i}}{x_{T_i}-x_{I_i}}\right) - \frac{\pi}{4} \end{bmatrix}^{\mathrm{T}}, \tag{3}$$

with x_{T_i}, y_{T_i} denoting i-th position of the team patch and x_{I_i}, y_{I_i} denoting i-th position of the identification patch.

6 Trajectory Follower Controller

Once the positions of the ball and of the players are estimated, the manager of the soccer robots provides the trajectory that each robot has to follow according to the strategy.

6.1 Mobile reference generation

Because two-wheel driven robots have nonholonomic constraints, [5], [7], [4], [12], [13], feasible trajectories have to be determined. This trajectory is based on a 4^{th} order Beziér curve [7], [4], [6]. The Beziér curve that connects two points P_s and P_f, with velocities V_s and V_f is given by:

$$\begin{bmatrix} x_r(\lambda) \\ y_r(\lambda) \end{bmatrix} = P_s B_5(\lambda) + P_1 B_4(\lambda) + P_2 B_3(\lambda) + P_3 B_2(\lambda) + P_f B_1(\lambda), \tag{4}$$

with $\lambda = t/T_{\max}$, $T_{\max}$ is chosen to satisfy the upper velocity bounds of the robot: $V_{\max}$, and $\Omega_{\max}$; $P_1 = P_s + \alpha V_s$, $P_2 = (P_1 + P_3)/2$, $P_3 = P_f - \alpha V_f$, α is a positive constants; and $B_1(\lambda) = \lambda^4$, $B_2(\lambda) = 4\lambda^3(1-\lambda)$, $B_3(\lambda) = 6\lambda^2(1-\lambda)^2$, $B_4(\lambda) = 4\lambda(1-\lambda)^3$, $B_5(\lambda) = (1-\lambda)^4$.

This Beziér curve has to be mapped to the velocity space of the reference robot (v_r, ω_r). To perform this task, the following set of equations is used:

$$
\begin{aligned}
\varphi_r &= \tan^{-1}\left(\frac{y_r'(\lambda)}{x_r'(\lambda)}\right) \\
v_r(t) &= \frac{1}{T_{\max}}\sqrt{x_r'^2(\lambda)+y_r'^2(\lambda)}\Big|_{\lambda=t/T_{\max}} \\
\omega_r(t) &= \frac{x_r'(\lambda)y_r''(\lambda)-x_r''(\lambda)y_r'(\lambda)}{[x_r'^2(\lambda)+y_r'^2(\lambda)]^{\frac{3}{2}}}\Bigg|_{\lambda=t/T_{\max}} v_r(t),
\end{aligned}
\tag{5}
$$

with $|v_r(t) \leq V_{\max}|$, and $|\omega_r(t) \leq \Omega_{\max}|$, where $V_{\max} > 0$ and $\Omega_{\max} > 0$ are the maximum feasible velocity values of the robot, and $(\cdot)'$ denotes the derivative with respect to the variable λ.

6.2 Control Law

Let (x_e, y_e, φ_e) be the coordinates of the reference robot in the frame of the real robot $\langle \mathbf{0}; X, Y, \Phi\rangle$. Here $q_e(t)$ is the error expressed in the real robot coordinate system (Fig. 6) by:

$$
q_e(t) = \begin{bmatrix} x_e(t) \\ y_e(t) \\ \varphi_e(t) \end{bmatrix} =
\begin{bmatrix} \cos(\varphi(t))(x_r(t)-x(t))+\sin(\varphi(t))(y_r(t)-y(t)) \\ -\sin(\varphi(t))(x_r(t)-x(t))+\cos(\varphi(t))(y_r(t)-y(t)) \\ \varphi_r(t)-\varphi(t) \end{bmatrix}.
\tag{6}
$$

Let (v_r, ω_r) be the inputs of the reference trajectory, and let $q_r(t) = [x_r(t), y_r(t), \varphi_r(t)]^{\mathrm{T}}$ be the parametric curve of the reference trajectory. The control law has the following form [12]:

$$
\begin{cases} v = v_r\cos(\varphi_e) + k_1 x_e \\ \omega = \omega_r + k_2\varphi_e + k_3\frac{\sin(\varphi_e)}{\varphi_e} y_e, \end{cases}
\tag{7}
$$

with k_1, k_2, and $k_3 > 0$.

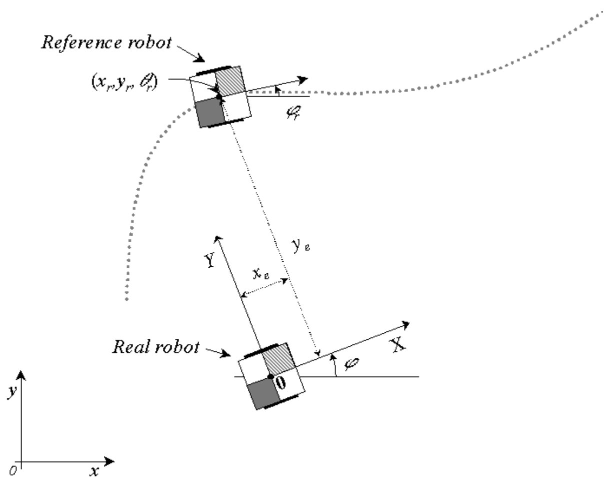

Figure 6. Relative positions of the robot and reference

7 Experimental Results

In this experiment, the robot starts running from several different starting points. The trajectory generation layer projects a straight line from the goal to the ball position plus 10 centimetres (point P_f). Then the robot has two possibilities. If it is located between the goal and point P_f, an intermediate point is located to attack the ball. If the robot is located behind point P_f, then the robot attacks the ball directly, as suggested in [6].

In Fig. 7, the robot attacks the ball by following the trajectory, which is provided by the robot soccer manager (chapter 6). The reference trajectory is calculated from equation (4) with respecting robot initial conditions (position and velocity) and final desired robot position and velocity, i.e. ball position and robot velocity at the kick. The task of the robot is to follow the path determined by the trajectory generation layer. The emphasis of the experiment is to follow the curve as well as possible without considering the path optimality thus the path itself is not the optimal one.

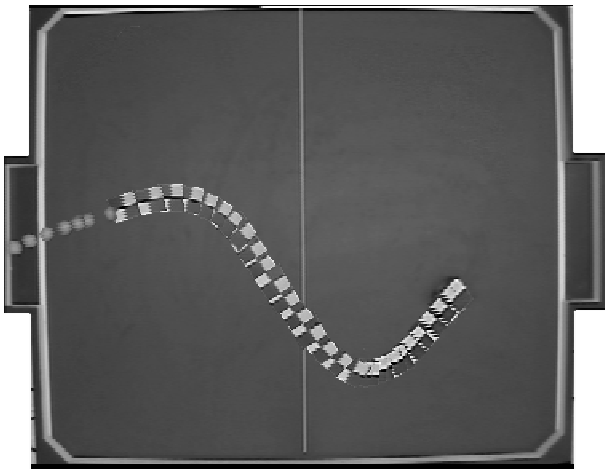

Figure 7. Robot attacking the ball

8 Conclusions

Our work is one of the first approaches in Slovenia towards establishing a vision system and the control for the purpose of robot soccer game. The developed software is divided into two separate applications each running in different threads and communicating through a shared memory. Thus multi task processing is easily achieved and programming is more transparent.

From the vision part perspective, our intention was to efficiently merge classification and connected region determination and labelling. The applied approach is confirmed by the short time needed for an incoming image processing. The time required to finish the entire position estimation is no more than 10 ms on 350 Mhz Pentium II computer. Thus the theoretical limit is 100 Hz image processing irrespective of if the capabilities of the frame grabber and camera.

Our main problem was the irregular lightening, which was some half of the one recommended (1000 luxes).

Future work will be focused on the improvement of lighting conditions, complete realistic simulator development and also acquisition of a new frame grabber capable of outputting other colour spaces.

Acknowledgement

This work was partly supported by the research grant ES/A99-HX1/01 of the International Cooperation between the Slovenian and Argentinean governments.

9 References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[2] J. Bruce, T. Balch, and M. Veloso, Fast and Cheap Color Image Segmentation for Interactive Robots, *Proceedings of IROS-2000*, Takamatsu, Japan, October 2000, also available on: http://citeseer.nj.nec.com/303543.html.

[3] K. Han and M. Veloso, Reactive Visual Control of Multiple Nonholonomic Robotic Agents, *Proceedings of the International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, also available on: http://www.cs.cmu.edu/˜mmv/produce-bib00.html.

[4] M. Khatib, H. Jaouni, R. Chatila, and J. P. Laumond, Dynamic path modification for car-like non-holonomic mobile robots, *IEEE International Conference on Robotics and Automation (ICRA'97)*, Albuquerque, USA, 20-25 April 1997, pp. 2920-2925.

[5] I. Kolmanovsky and N. H. McClamroch, Developments in Nonholonomic Control Problems, *IEEE Control Systems*, vol. 15, no. 6, pp. 20-36, December 1995.

[6] Y. D. Kwon, D. M. J. M. Won, and J. S. Lee, Multiagents cooperation strategy for soccer robots, *Proceedings of FIRA Robot World Cup 98*, Paris, France, June 1998, also available on: http://tau.postech.ac.kr/international_publications.html.

[7] J.-P. Laumond, Robot Motion Planning and Control, *Lecture Notes in Control and Information Sciences*, no. 229, Springer Verlag, January 1998.

[8] C. G. Looney, *Pattern Recognition using Neural Networks — Theory and Algorithms for Engineers and Scientists*, Oxford University Press, 1997.

[9] A. Minoru, R. D'Andrea, A. Birk, H. Kitano, and M. Veloso, Robotics in Edutainment, *Proceedings of the 2000 IEEE International Conference on Robotics & Automation (ICRA'2000)*, San Francisco, California, April, 2000, pp. 795-800.

[10] N. Pavešić, *Razpoznavanje vzorcev: Uvod v analizo in razumevanje vidnih in slušnih signalov (Pattern Recognition: An Introduction into the Analysis and Understanding of Visual and Audio Signals)*, Faculty of Electrical Engineering, Ljubljana, Slovenia, 2000.

[11] J. Perš and S. Kovačič, A System for Tracking Players in Sport Games by Computer Vision, *Electrotechnical Review — Journal for Electrical Engineering and Computer Science*, vol. 67, no. 5, pp. 281-288, 2000.

[12] C. Samson and K. Ait-Abderrahim, Feedback Control of a Nonholonomic Wheeled Cat in Cartesian Space, *IEEE International Conference on Robots and Automation*, Sacramento, California, 1991, pp. 1136-1141.

[13] N. Sarkar, X. Yun, and V. Kumar, Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robot, *International Journal of Robotic Research*, vol. 13, no. 1, pp. 55-69, 1994.

[14] G. Wyeth and B. Brown, Robust Adaptive Vision for Robot Soccer, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'2000)*, San Francisco, California, April 2000, also available on: http://www.elec.uq.edu.au/˜wyeth/pub.html.

Gregor Klančar received his B.Sc. degree in 1999 from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia, where he is currently employed as a member of the national young researcher scheme. His research interests are in the area of fault diagnosis methods and multiple vehicle coordinated control.

Omar A.A. Orqueda received his B.Sc. degree in 1996 from the Department of Electrical Engineering - National University of the South, Bahía Blanca, Argentina. He is a post-graduate student at the same University. His research interests include path planning and control of autonomous vehicles as well as perception planning.

Drago Matko received his B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering in 1971, 1973 and 1977, respectively, from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia for his work in the field of Adaptive Control Systems. Currently he is a professor of the same faculty of discrete control systems, computer controlled systems, identification and computer aided design of control systems. His research is involved with adaptive systems, computer aided design and modern control techniques.

Rihard Karba received his B.Sc. (1972), M. Sc. (1975) and Ph.D. (1981) degrees from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia where he is currently professor. His research interests include modelling and simulation of dynamic systems, multivariable control systems analysis and design as well as modelling and simulation in biopharmacy and in biomedicine.