

Robust and efficient vision system for group of cooperating mobile robots with application to soccer robots

Gregor Klančar,* Matej Kristan, Stanislav Kovačič

University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, SI-1000 Ljubljana, Slovenia

Omar Orqueda

*National Council for Scientific and Technical Research (CONICET), Department of Electrical Engineering,
National University of the South, Avda. Alem 1253, (8000) Baha Blanca, Argentina*

(Received 9 December 2002; accepted 28 September 2003)

Abstract

In this paper a global vision scheme for estimation of positions and orientations of mobile robots is presented. It is applied to robot soccer application which is a fast dynamic game and therefore needs an efficient and robust vision system implemented. General applicability of the vision system can be found in other robot applications such as mobile transport robots in production, warehouses, attendant robots, fast vision tracking of targets of interest and entertainment robotics. Basic operation of the vision system is 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 optimization of the processing time needed to finish the estimation of possible object positions. Better results of the vision system are achieved by implementing camera calibration and shading correction algorithm. The former corrects camera lens distortion, while the latter increases robustness to irregular illumination conditions. © 2004 ISA—The Instrumentation, Systems, and Automation Society.

Keywords: Computer vision; Classification; Segmentation; Camera calibration; Non-uniform illumination correction

1. Introduction

In recent years, mobile robots playing soccer have gained much popularity among researchers worldwide. This is mainly due to the fact that 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 [1]. Students can get the results of their achievements through the game and immediate feedback enables them to evaluate their algorithms.

The paper presents a design of a global vision system for estimating current object positions and orientations on the playground. The MiroSot cat-

*Tel.: +386 1 4768701; fax: +386 1 426 4631. *E-mail address:* gregor.klancar@fe.uni.lj.si

egory soccer robots we are interested in are without on-board position sensors. Thus a precise and fast global vision has to be designed for robots' control and navigation in a partially controlled, dynamically changing environment. When designing the vision system, the following requirements have to be accomplished:

- computational efficiency,
- high reliability,
- good precision, and
- robustness to noise, non-uniform illumination, and different color schemes.

The last characteristic is essential for the system to function well when using it under different conditions present at competitions [2].

With color cameras there are many possible ways to carry out the detection of moving objects. Sargent *et al.* [3] developed a fast real-time vision system by the aid of a special hardware accelerated system, which only makes sense when software optimizations or accelerations are not possible. More reliable vision tracking of moving objects could be achieved by robust statistics and probability distributions used. A good example of the latter is given in color-based face tracking implemented by Bradski [4]. Bruce *et al.* [5] suggested a fast vision system for mobile robots by efficient color segmentation and two pass connected region determination algorithm. Another important contribution in robot soccer vision design is introduced by Wyeth *et al.* [2] with special consideration given to robustness of varying playground illumination conditions. Most of the approaches try to classify pixels of an image into one of a predefined number of classes. The most common are: linear color thresholding, K-nearest neighbor classification, neural net-based classifiers, classification trees, and probabilistic methods [6–8].

In this paper, a fast approach with constant thresholding and back-stepping algorithm is presented where special attention is given to the efficiency aspect. The thresholds can be presented as boxes in three-dimensional color spaces. These thresholds are determined by means of off-line learning. If an incoming pixel color falls inside one of the predefined boxes, then it is classified as belonging to the class associated with this box. This first step is followed by the second step where the pixels belonging to one class (a con-

nected region) are distinctively labeled. 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 optimization 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. To improve results of the vision system the camera calibration and non-uniform illumination correction algorithm are implemented. The former corrects distortion caused by the camera lens, thus achieving a more accurate and precise objects positions estimation, while the latter improves robustness to irregular illumination and non-uniform illumination conditions.

The paper is organized as follows. In Section 2 a brief overview of the system is given. The method used for pixel classification is explained in Section 3. Section 4 focuses on the algorithms for image segmentation and region labeling. The algorithm for object estimation is illustrated in Section 5. Section 6 resumes the camera calibration and non-uniform illumination correction implementation together with experimental results shown. The paper ends with conclusions and some ideas for future work.

2. System overview

The soccer robot setup, Fig. 1, consists of ten MiroSot category robots (forming two teams) of size 7.5 cm³, rectangular playground of size 2.2 × 1.8 m², color NTSC camera, frame-grabber Matrox Meteor 2, and personal computer Pentium 4. The vision part of the program processes the incoming images, of a resolution of 640 × 480 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 w , that the robots should have in the next sample time according to the current situation on the playground. These reference speeds are sent to the robots by a wireless connection. To identify the orientations, each robot has to have two color patches. One is the team color and the other is the identification color patch. According to FIRA (Federation of International Robot-soccer Association) rules, the team color must be blue or yellow, the ball must

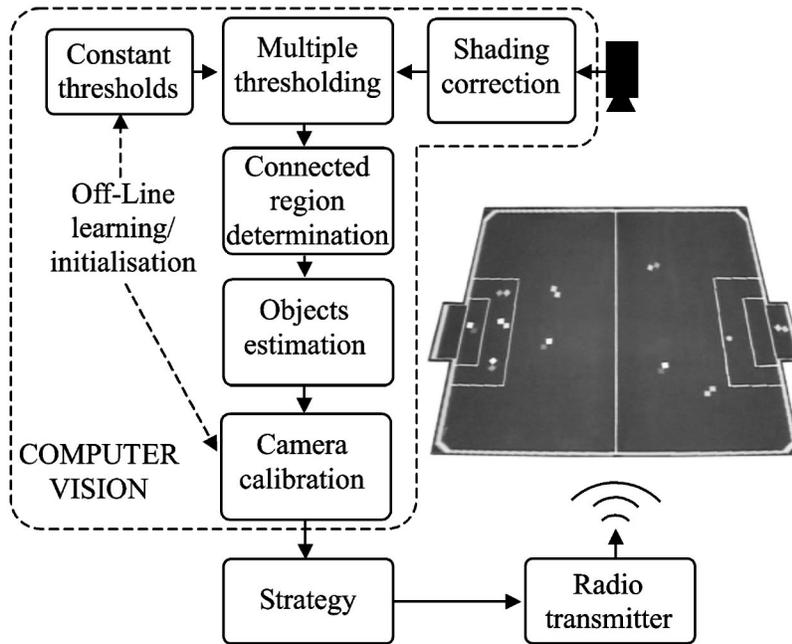


Fig. 1. System overview.

be orange, and identification colors can be any color except the team and ball color. The patch positions and shapes can be chosen freely. In our case, square patches were used. They were placed diagonally, with the team color being closer to the front part of the robot, Fig. 2.

3. Pixel classification

To enable detection of different color patches, each pixel has to be classified into one of the pre-defined colors.

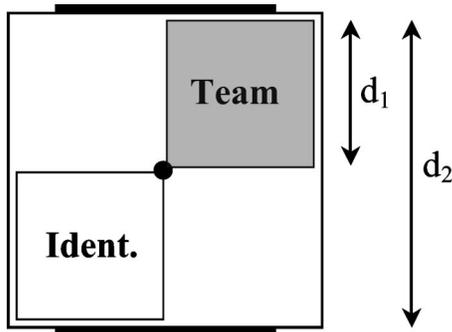


Fig. 2. Color patches on the robot.

3.1. Color spaces

The color image can be presented by the use of different color spaces such as RGB, HSI, YUV, and others. By using the RGB space, the regions with the same color are best presented in a three-dimensional color space with conical volumes. If constant thresholds which form blocks (rectangular prism) in RGB space are used, then the description of color regions becomes rather difficult [5]. When using simple thresholds for pixel classification, HSI and YUV color 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 color spaces, any particular color 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. Also these color spaces are more robust to different illumination conditions. Both RGB and YUV spaces were used in our experiments with constant thresholds, and, as expected, the second color space gave better classification results. However, to obtain the YUV color representation, a transformation from the original RGB space had to be

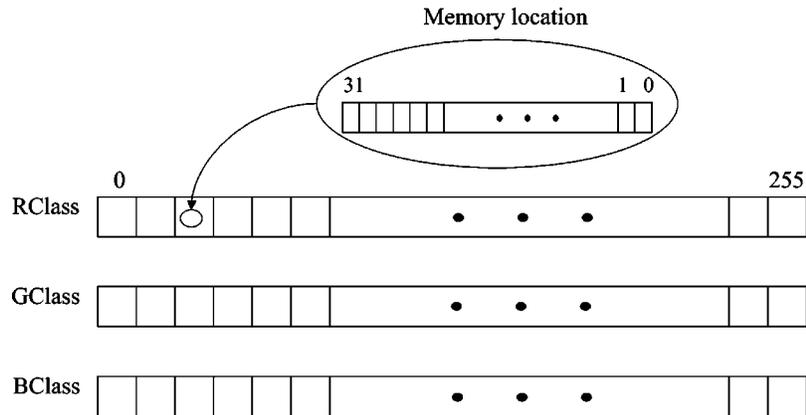


Fig. 3. Look-up tables for pixel classification.

done. This transformation was time consuming although the optimization by using look-up tables was used. Including the optimization it requires some 30 ms, while the rest of the program takes only 7 ms to identify objects from the image. Therefore, YUV or any other color space should be used only when it can be directly obtained from the frame grabber. In the application the analog NTSC camera (RGB color space) was used with image production time of 33 ms (30 Hz). The image processing was done in double buffering mode which involves grabbing into one image while processing (object identification) the previously grabbed image. Double buffering mode is usually used in real-time vision applications for it enables concurrent image grabbing and processing.

3.2. Thresholding

The basic idea is to classify each pixel according to the preset color thresholds of each object. Initially, this part was done using the following code:

```

for i=1 to number of colors on playground
  if (R>=R_lower_bound AND
      R<=R_upper_bound AND
      G>=G_lower_bound AND
      G<=G_upper_bound AND
      B>=B_lower_bound AND
      B<=B_upper_bound)
    Pixel_color=i-th color;
  else Pixel_color=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 pixel and color we want to classify.

To improve this operation, i.e., to check all colors at the same time, the idea of parallelism by means of look-up table is considered. Three $N \times 32$ bit integer arrays are allocated, where N corresponds to the number of color levels (usually $N=256$) and the maximum number of colors to be classified is 32 (Fig. 3). Each bit in a 32 bit memory location is associated with one color. Although the algorithm is able to classify 32 different colors, only 13 are enough for the purpose of the robot soccer game. Because the computer has a 32 bit arithmetic, the computational burden is the same as for 16 bit memory location only.

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

$$\begin{aligned}
 200 &\leq R \leq 220, \\
 230 &\leq G \leq 250, \\
 10 &\leq B \leq 30.
 \end{aligned}$$

Suppose this color 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 then performed for bits 0–30 for other color patches.

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 classified as yellow.

With this methodology, a multiple thresholding (the thresholds for all color patches checked at the same 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 labeling

To estimate patch positions, first all identification patches and the regions belonging to the ball, team, and opponent team 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 camera noise. Image segmentation in K regions and labeling is done fulfilling the following five conditions[6]:

- $\cup_{i=1}^K R_i = R$,
- $R_i \cap R_j = \emptyset$, $i, j = 1, 2, \dots, K$, and $i \neq j$,
- R_i is a connected region of pixels,
- $P(R_i) = 1, \forall i$,
- $P(R_i \cup R_j) = 0$, $i \neq j$, and R_i, R_j are neighbors,

where $P(x)$ is a logical predicate, which takes the value 1 if all the pixels of the region accomplish a criterion of homogeneity. In our case, the homogeneity criterion is the equality in color.

According to the first two 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 color connecting any two pixels in the region. Moreover, the regions must be homogeneous with respect to the color. Additionally the neighboring regions must not be of the same color, as stated in conditions four and five.

The algorithm. Sequential connected component labeling algorithm, originally developed by Rosenfield and Pfaltz [9], is a well-known technique for efficient image segmentation. It requires two passes through the image. In the first all pixel labels are generated with equivalent labels being stored. The second pass replaces each label with its representative label. A number of researchers tried to improve the efficiency of the above algorithm mostly by optimizing the second pass of the algorithm. Our approach is presented in the se-

quel, focusing on one pass variant only. Pixel classification and image segmentation are merged with the aid of a corresponding algorithm. Its main property is that the image classification and segmentation is done with only one pass (row-vice) through the image, which considerably contributes to time efficiency. The results of the mentioned algorithm are labeled regions with the following information:

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

Prior to the start of the algorithm, the color thresholds are selected for each component to be identified: ball, robot1, robot2, robot3, team, opponent robots, and opponent team.

Connected component labeling algorithm.

- The algorithm starts analyzing the first pixel of a given region of interest (in our case the whole image).
 - If the color of the pixel under study is a valid color and at the same time different from the color of the upper and left neighbor pixels, a new region is created [Fig. 4(a)].
 - If the color of the pixel under study is a valid color and is also equal to the color of the upper or left pixel, then the pixel under study is added to the region of the upper or left pixel [Fig. 4(b)].
 - If the color of the pixel under study is a valid color and at the same time equal to the upper and left pixel colors, then [Fig. 4(c)]:
 - 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.

5. Object position estimation

From all of the possible valid regions identified in a way described in the previous section, a proper number of regions with the biggest area is selected. This step maximizes the probability of correct regions being selected and not those due to

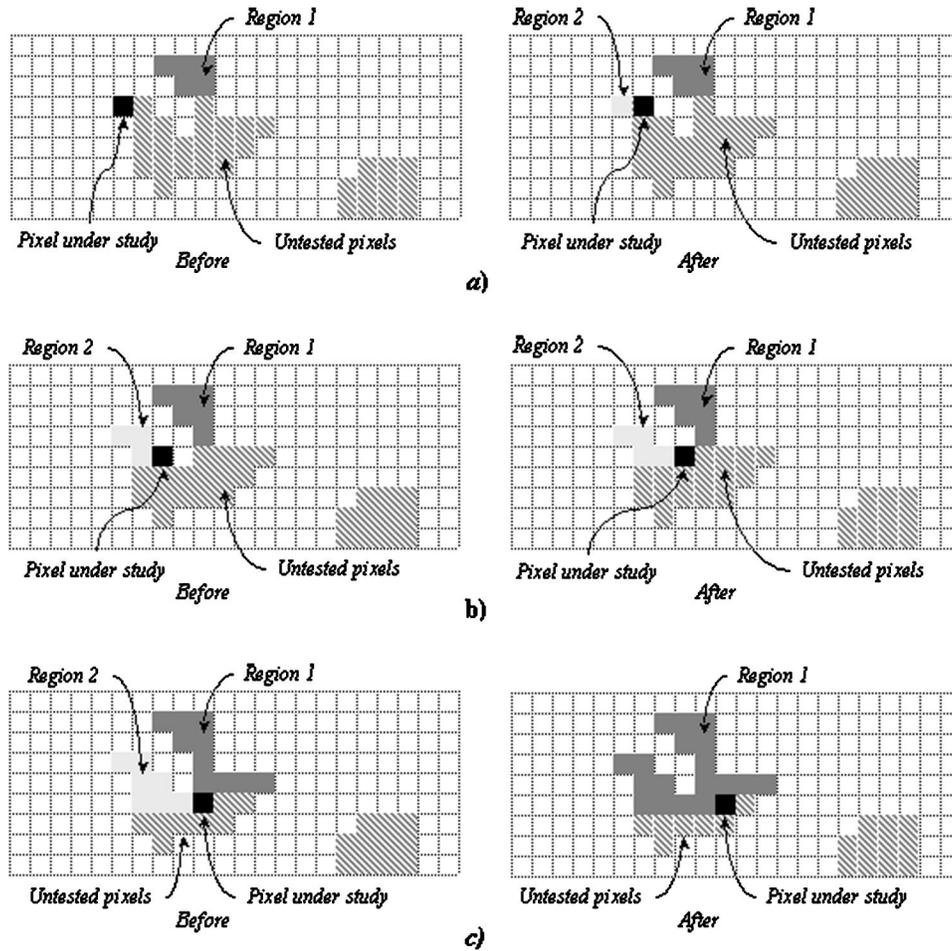


Fig. 4. Image segmentation and labeling.

noise which usually have small areas (one or at most two pixels). The algorithm only selects the first few biggest areas and therefore small areas caused by noise are automatically excluded.

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

$$\text{dist}(\text{region}, \text{team}_i) > K_1, \quad (1)$$

where dist is Euclidean distance, region is the current testing region, team_i are already chosen team regions, and K_1 is positive constant. In the case of the presented example the best results are obtained by $K_1 = \frac{3}{4}d_1$, where d_1 is the size of the color patch (see Fig. 2).

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

$$K_1 < \text{dist}(\text{region}, \text{ident}_i) < K_2, \quad (2)$$

where region is the current testing region, ident_i are already chosen other identification regions (other robots) and positive constant K_2 in the presented example chosen as $K_2 = d_2$ where d_2 is the robot size (see Fig. 2).

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

Table 1
Tables of possible pairs from Fig. 5.

	(a)			(b)			(c)		
	A	B	C	A	B	C	A	B	C
1			1	1			1		1
2	1				1	1	2	1	1
3		1		3	1	1	3	1	

found following a simple procedure, where the element indices represent the correct pair:

- if the row has just one element equal to 1, or
- if the column has just one element equal to 1.

For the robots that cannot 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.

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

In the first table [Table 1(a)] three pairs can easily be found (1C, 2A, and 3B). The regions 4 and D (see Fig. 5) are not considered because their area is small, and their existence is probably due to noise. In the second table [Table 1(b)], taking the first row and the first two columns, all pairs are found (1C, 2B, and 3A). In the last table [Table 1(c)], taking the last row and the last column, two pairs are found (1C and 3A). The second robot could not be found, thus leaving B as the only possible team color not chosen yet in the second row, the pair being 2B.

When the right regions representing color patches are found, the final estimated patch position can be improved by taking the weighted average of all region positions with the same classified color 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:

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

with x_{T_i} , y_{T_i} denoting i th center position of the team patch and x_{I_i} , y_{I_i} denoting i th center position of the identification patch. Calculated orientation φ_i points to the direction of robot front side (the side next to team color, see Fig. 2).

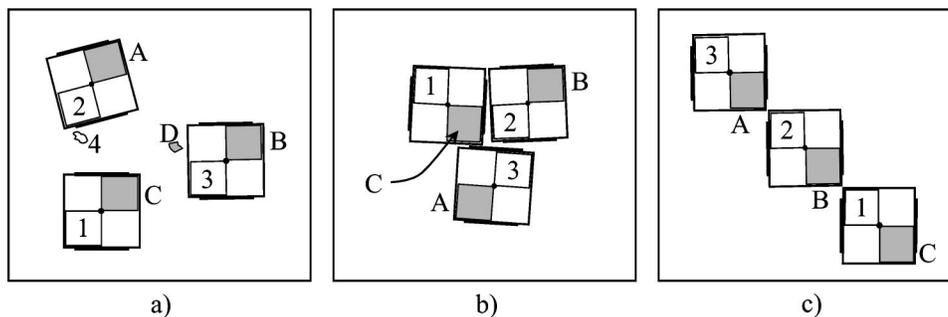


Fig. 5. Different robots placement.

6. Camera calibration and non-uniform illumination correction

To improve the results of the presented vision system the camera calibration and non-uniform illumination correction algorithm are implemented. The first corrects distortion caused by camera lens, thus enabling more accurate and precise estimated objects positions achievement while the second improves robustness to irregular illumination and unequal non-uniform illumination conditions.

6.1. Camera calibration

Due to the need for a good position estimation based on acquired images the relationship between the pixel coordinates of the image and the playground have to be known. This relationship also includes distortions caused by camera lenses. Several types of lens distortions exist; radial distortion is the most problematic [10], especially when inexpensive wide-angle lenses are used. The problem of camera calibration has therefore received wide attention in computer vision applications. The most widely used method is the polynomial model for camera distortion [11,12]. Perš *et al.* [10] suggested a mathematical model of radial distortion based on camera and lens projection geometry. Their idea is followed in the approach below. The camera calibration routine consists of two steps originating from the demand for an easy-to-use and fully automated system, and from the fact that there are two main reasons for camera distortion. The first is radial distortion caused by the zoom (lenses) and the second is perspective distortion caused by the tilt of the camera.

6.1.1. Radial distortion rectification

To correct radial distortion, a radial distortion model [10] based on camera and lens projection geometry is used

$$R = f(r) = \frac{H}{2} \frac{(1 - e^{-2r/H})}{e^{-r/H}} = H \cdot \sinh\left(\frac{r}{H}\right), \quad (4)$$

where R is the rectified radius, r is the radius from the distorted image [Eq. (5)], and H is the focal length. Let x and y be the coordinates of a pixel in the distorted image, and let X and Y be the coordinates of the same pixel in the rectified image. The origin of the transformation (4) is placed in

the center of the image. The relations between (x, y) and (X, Y) are as follows:

$$r = \sqrt{x^2 + y^2}, \quad \varphi = \arctan 2\left(\frac{y}{x}\right), \quad (5)$$

$$X = R \cos(\varphi), \quad Y = R \sin(\varphi).$$

The only unknown parameter is the focal length H , which can be set manually or by means of automation. Since there are strict rules regarding the playground shape (straight parallel and perpendicular lines) and dimensions, this knowledge can be used to estimate parameter H in model (4). First the Hough transform is used to fit a line to the horizontal playground boundary (the longest one) on the distorted image obtained from the camera. Three points are then selected on the line, two in the corners and one in the middle. By moving in a normal direction from the line the pixels on the thresholded image belonging to the boundary that are closest to the previously selected points are chosen. These three pixels are transformed by means of transformation (4) in an optimization procedure to change parameter H until the most satisfactory collinearity of the transformed pixels is obtained.

6.1.2. Rectifying the effect of titled camera

In a robot soccer game there are two cameras—one for each team. One of them is placed slightly left or right from the playground center. In order for the playground to occupy as much of the acquired image as possible, the camera has to be titled, hence the perspective distortion. With an assumption of perfect projection, e.g., with a pin-hole camera, a set of parallel lines in the scene is projected onto a set of lines in the image that meet at a common point. This point of intersection, perhaps at infinity, is called the vanishing point. The three orthogonal vanishing points form a triangle, and the intersection of the triangle's heights is called the principal point. Many good and robust algorithms are available for the detection of perspective [13,14]. An efficient and robust method of vanishing point detection and transformation that relies only on the information gained from the playground is selected. Its basic idea was introduced by Fangi *et al.* [15], and reads

$$\begin{aligned} (pX - 1)x + qXy &= -X, \\ pYx + (qY - 1)y &= -Y, \end{aligned} \quad (6)$$

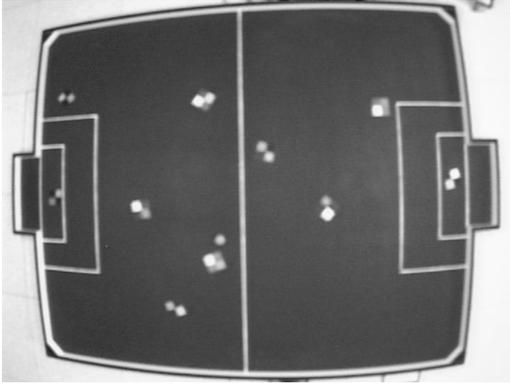


Fig. 6. Distorted image (image from the camera with radial and perspective distortion).

where x and y are the coordinates of rectified image, whereas X and Y are the coordinates of an image with rectified radial distortion only. The parameters p and q are estimated from the location of vanishing points, which are defined by the cross sections of the lines fitted (by means of Hough transformation) to the playground boundaries on image with rectified radial distortion. The principal point is placed in the center of the image. The two lines running from the principal point to the calculated vanishing points are almost perpendicular. From the position of the vanishing points and the principal points, playground rotation is determined. If the detected rotation is not negligible, the rotation should also be rectified prior to perspective transformation. Parameters p and q from Eq. (6) are obtained by

$$\begin{aligned} p &= \frac{1}{d_{v_1}}, \\ q &= \frac{1}{d_{v_2}}, \end{aligned} \quad (7)$$

where d_{v_1} is the distance from the principal point to the horizontal vanishing point and d_{v_2} is the distance to the vertical vanishing point.

The example of the complete camera calibrating procedure is shown in figures below. Distorted image obtained from the camera is shown in Fig. 6 and the rectified image is shown in Fig. 7.

Although both the radial distortion and perspective correction are very simple, running the algorithm on every incoming image from the camera is time-consuming. Pixel classification, image seg-

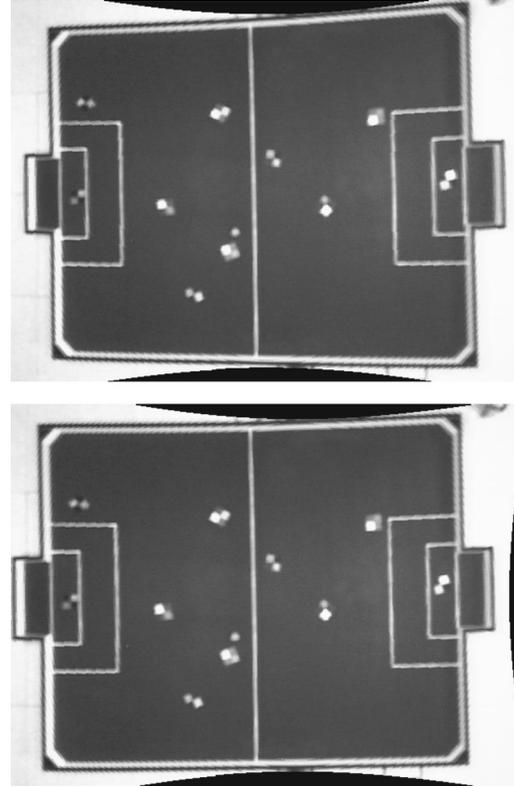


Fig. 7. Rectified image: (a) corrected radial distortion only; (b) corrected radial and perspective distortion.

mentation, and image labeling are therefore performed on a distorted image and only the estimated objects coordinates are transformed (radial distortion and perspective rectification).

6.2. Compensating for irregular illumination conditions

Non-uniform illumination caused by irregular illumination is the main reason for loss (or incorrect estimation) of the players or the ball during the game. It is often the case that the center of the playground is more illuminated than its corners. If a vision system is taught a specific color when the carrier is in the center of the playground, there is a strong possibility that this color will not be recognized when its carrier is positioned in one of the corners. Non-uniform illumination and illumination conditions are more or less constant during the game, thus a static shade model could be built in the initialization phase. The relation between the true (ideal uniform illumination) image

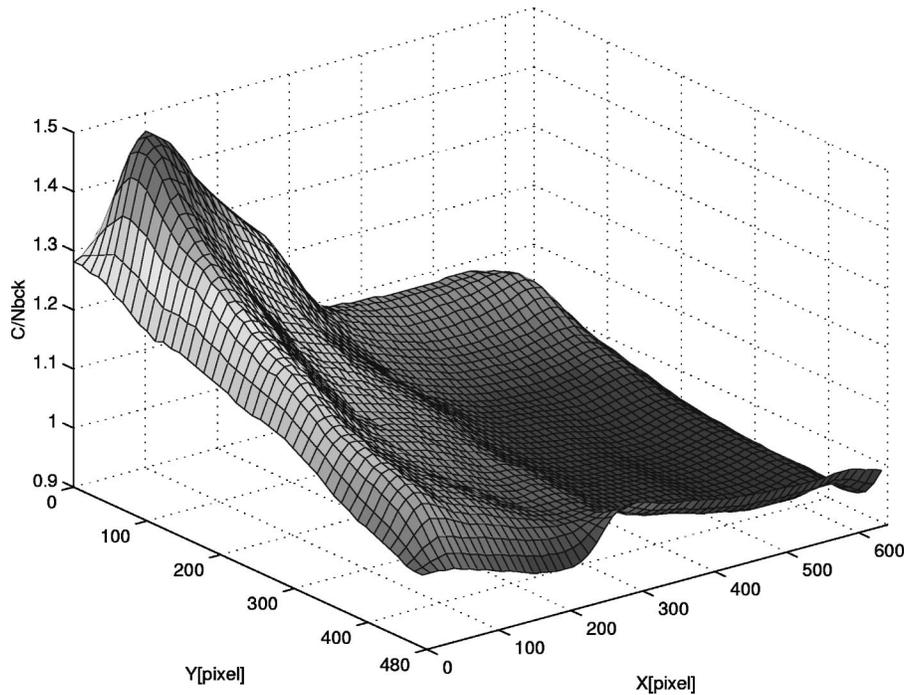


Fig. 8. Calculated multiplicative component ($C/NBCK$) for the red channel.

$U(x,y)$ and acquired image $N(x,y)$ is usually described by means of a linear model of image formation

$$N(x,y) = U(x,y)S_M(x,y) + S_A(x,y), \quad (8)$$

where $S_M(x,y)$ is the multiplicative and $S_A(x,y)$ the additive component. There are two types of non-uniform illumination correction methods, namely, retrospective and acquisition-based. Acquisition-based methods involve taking one or two reference images and interpreting them as multiplicative and additive components for the acquired image, while retrospective methods rely solely on the information content of the acquired image. The most intuitive retrospective methods for correcting multiplicative and smooth intensity variations are homomorphic filtering, image blurring, smoothing, averaging, Fourier-domain filtering, and homomorphic unsharp masking [16]. In Likar *et al.* [17] additive and multiplicative non-uniform illumination components were approximated by means of second-order polynomials. Our test showed that this type of model did not improve detection, and sometimes even made it worse. The reason for such results lies in the shape of the illumination plane. A typical illumination

plane consists of several peaks and valleys which cannot be modeled sufficiently by means of second-order polynomials but by those of higher order. It can be quite difficult to estimate the order of a polynomial in a model, hence a simpler method was chosen. The proposed method is acquisition-based, it employs a background image, and the additive component is discarded. Non-uniform illumination is finally corrected by

$$\hat{U}(x,y) = \frac{N(x,y)}{N_{BCK}(x,y)} C, \quad (9)$$

where $\hat{U}(x,y)$ is the corrected image, $N_{BCK}(x,y)$ is the background image, and C is the normalization constant needed to restore the desired gray level range. To improve computational time efficiency, multiplication by multiplicative component $C/N_{BCK}(x,y)$ in Eq. (9) should be realized by means of a look-up table.

Acquiring the multiplicative component. The court and the background in camera's field of view was covered with gray paper sheets, and the acquired reference image $N_{BCK}(x,y)$ then represented the non-uniform illumination plane of a multiplicative component $C/N_{BCK}(x,y)$ in relation

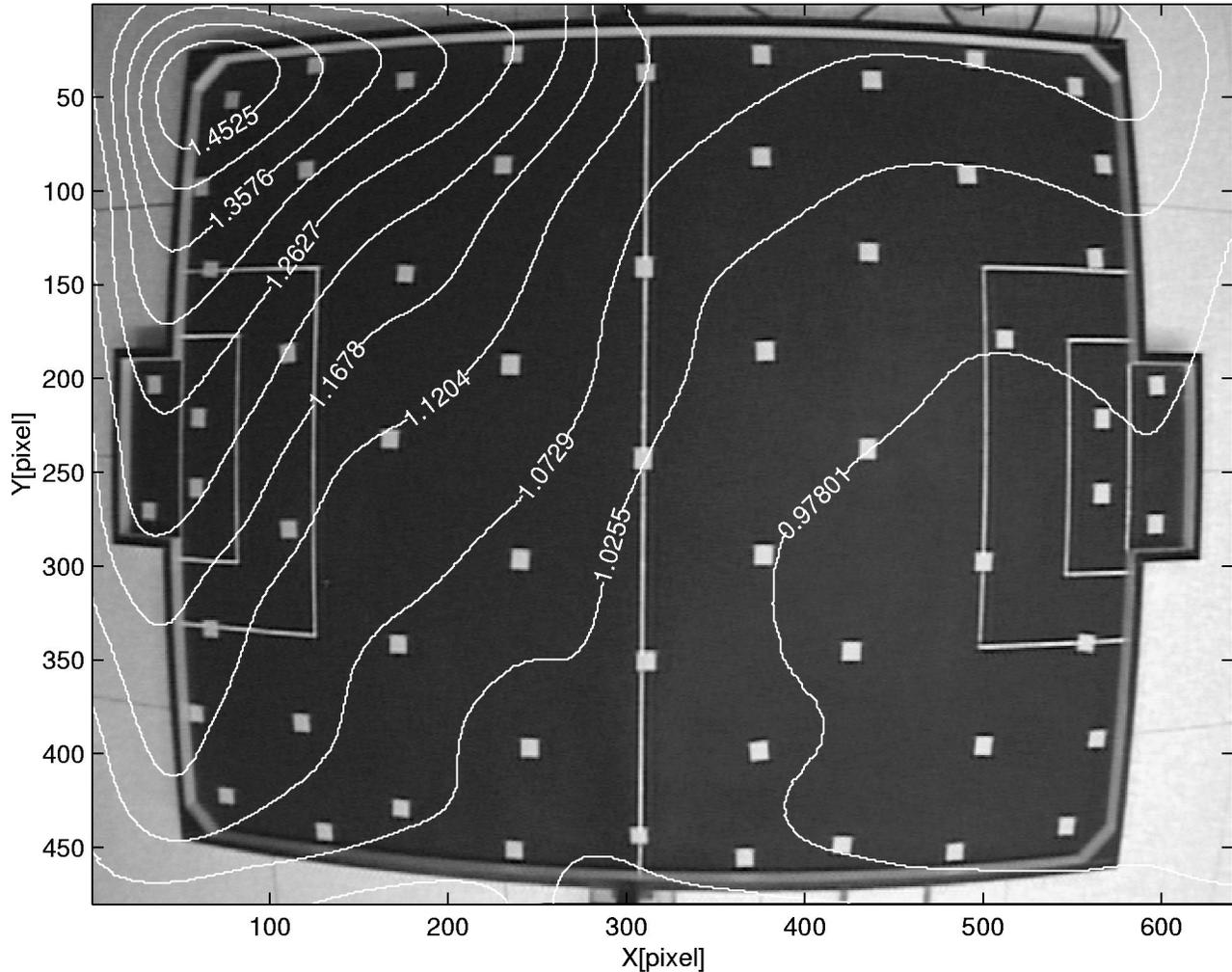


Fig. 9. Camera image with color patches arranged over the playground and intensity illumination contours displayed.

(9). The reference image was retouched with a slight Gaussian blurring to increase the smoothness of the non-uniform illumination plane. The illumination planes were calculated for all three channels (red, green, and blue). Fig. 8 shows only the plane for the red channel, while the other two (green and blue) have a similar shape. The values on axis z in Fig. 8 correspond to illumination strength; more illuminated regions have smaller values of axis z and vice versa.

The illumination plane from Fig. 8 is (according to Fig. 1) applied to incoming camera images. To test the efficiency of the non-uniform illumination correction algorithm, a number of color patches of the same color are arranged over the playground (see Fig. 9). From the color patch in the center of Fig. 9 the representative color thresholds are de-

termined. The pixel classification and segmentation algorithm is presented on the incoming camera image (Fig. 10), and then on the same image with compensated non-uniform illumination (Fig. 11).

In Figs. 10 and 11, the regions with black color patches represent recognized pixels which belong to color patches from Fig. 9. A pixel is classified as belonging to the color patches if it fits the representative color thresholds. Due to non-uniform illumination (see the intensity illumination contours in Figs. 9–11 or the illumination plane in Fig. 8), some color patches in Fig. 10 are purely detected or even undetected.

From Fig. 11 it can be seen that non-uniform illumination compensation improves the efficiency of the classification and segmentation algorithm.

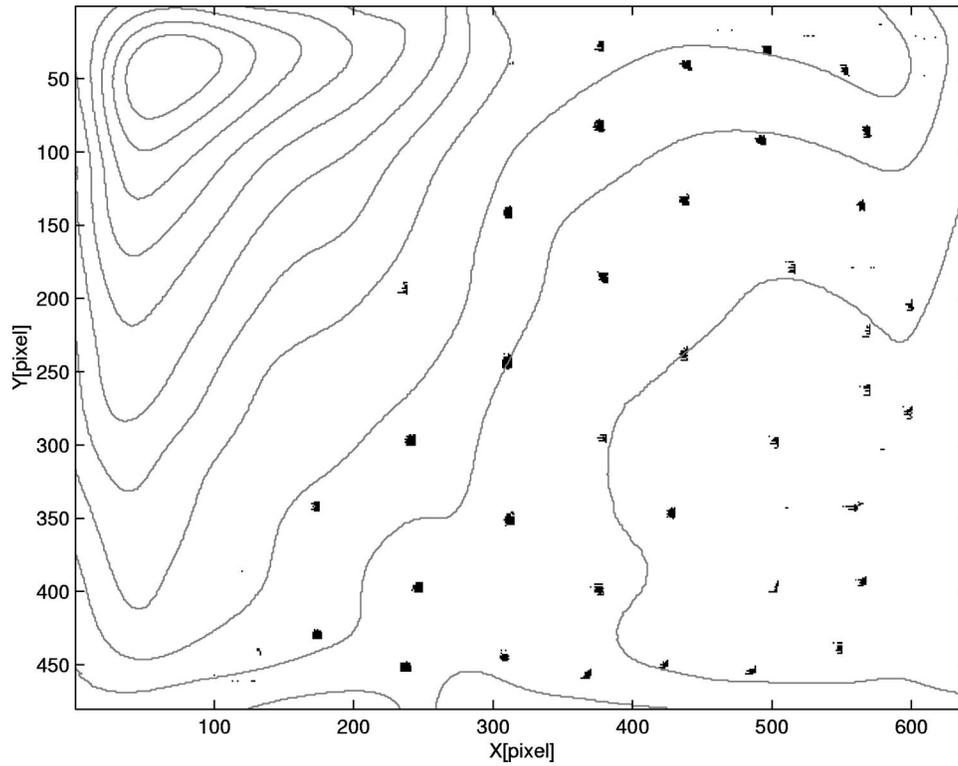


Fig. 10. Classified pixels from the camera image in Fig. 9.

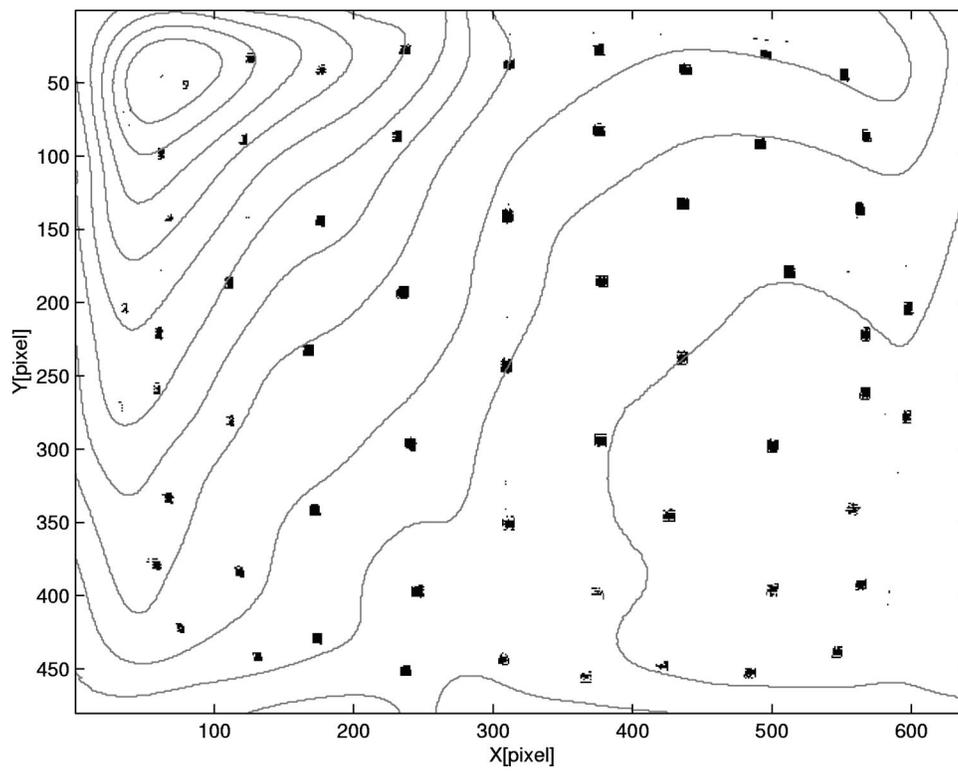


Fig. 11. Classified pixels from the camera image in Fig. 9 with compensated non-uniform illumination.

All of the patches from Fig. 9 can be reliably identified from the compensated image (Fig. 11), while this could not be achieved by using the non-compensated image (Fig. 10).

7. Conclusions

The issues that the paper address are vexing ones for the robot soccer community; issues that are often passed as trivial. However, there is little literature on effective solutions to the problems, which is often a source of frustration to teams who wish to pursue other issues in the soccer domain (such as AI and control).

An approach toward establishing a fast and robust vision system for the purpose of a robot soccer game is presented. Special consideration is given to optimization of computational work and robustness issues. The latter are assured by inclusion of methods for image quality improvement such as correction of non-uniform illumination and camera lens distortions. Robustness is further achieved by time-efficient algorithms which enable global image processing. By contrast, some vision systems used by other robot soccer teams employ local image processing to obtain the desired frame rate of vision system. The major disadvantage of these latter algorithms is loss of one or more objects (robots or ball) for some unpredictable reasons (lightening conditions, collisions, bugs). The local search areas have to be increased until objects are found, which results in larger and irregular sample time. This could not happen with global image processing. However, a disadvantage of the presented approach can appear if a large number (more than 15) of different color patches have to be followed. Some of them could then become quite similar on camera image which could result in wrong objects' estimation. The problem will be dealt with in future work by inclusion of object tracking algorithms.

The developed software is divided into two separate applications (vision and control) 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, connected region determination, and labeling. 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 7 ms on 1.4 GHz Pentium 4 computer. Thus the theoretical limit is 100 Hz image processing, irrespective of the capabilities of the frame grabber and the camera.

Any incorrect or inaccurately estimated position of the robots or the ball results in a poor game apart from perfection of the strategy control algorithm. The robustness of the presented vision system is therefore improved by means of camera calibration algorithms. The suggested procedure for shading correction proved useful when the illumination conditions remained more or less unchanged during the game. The optimized algorithms presented enable the vision system to be used in real-time applications where robustness to irregular illumination and to camera distortions are important.

References

- [1] Minoru, A., D'Andrea, R., Birk, A., Kitano, H., and Veloso, M., Robotics in edutainment, in Proceedings of the 2000 IEEE International Conference on Robotics & Automation (ICRA'2000), San Francisco, California, 2000, pp. 795–800.
- [2] Wyeth, G. F. and Brown, B., Robust adaptive vision for robot soccer, in *Mechatronics and Machine Vision in Practice*, edited by J. Billingsley. Research Studies Press, 2000, pp. 41–48.
- [3] Sargent, R., Bailey, B., Witty, C., and Wright, A., The importance of fast vision in winning the First Micro-Robot World Cup Soccer Tournament. *Rob. Auton. Syst.* **21**, 139–147 (1997).
- [4] Bradski, G. R., Computer vision face tracking for use in a perceptual user interface. *Intel Technol. J.* **2nd quarter**, 1–15 (1998).
- [5] Bruce, J., Balch, T., and Veloso, M., Fast and inexpensive color image segmentation for interactive robots, in Proceedings of IROS-2000, Takamatsu, Japan, 2000, pp. 2061–2066.
- [6] Pavešič, N., 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.
- [7] Bishop, C. M., *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, UK, 1995.
- [8] Looney, C. G., *Pattern Recognition using Neural Networks. Theory and Algorithms for Engineers and Scientists*. Oxford University Press, Oxford, UK, 1997.
- [9] Rosenfeld, A. and Pfaltz, J. L., Sequential operations in digital picture processing. *J. ACM* **3**, 471–494 (1966).
- [10] Perš, J. and Kovačič, S., Nonparametric, model-based radial lens distortion correction using titled camera assumption, in Proceedings of the Computer Vision

Winter Workshop 2002, edited by H. Wildenauer and W. Kropatsch. Bad Aussee, Austria, 2002, pp. 286–295.

- [11] Tsai, R. Y., A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Rob. Autom.* **3**, 323–344 (1987).
- [12] Pajdla, T., Werner, T., and Hlava, V., Correcting radial lens distortion without knowledge of 3-D structure. Technical Report No. TR97-138. Faculty of Electrical Engineering, Czech Technical University, Praha, Czech Republic, 1997.
- [13] Rother, C., A new approach for vanishing point detection in architectural environments, in Proceedings of the 11th British Machine Vision Conference (BMVC'00), Bristol, UK, 2000, pp. 382–391.
- [14] Tuytelaars, T., Proesmans, M., and Gool, L. V., The cascaded Hough transform, in Proceedings of International Conference on Image Processing (ICIP '97), 1997, Vol. 2, pp. 736–739.
- [15] Fangi, G., Gagliardini, G., and Malinverni, E. S., Photo-interpretation and small scale stereoplotting with digitally rectified photographs with geometrical constraints. CIPA Congress 2001, available on: <http://cipa.icomos.org/papers/2001-.htm>, September 2001.
- [16] Likar, B., Viergever, M. A., and Pernuš, F., Retrospective correction of MR intensity inhomogeneity by information minimization, in Lecture Notes in Computer Science Vol. 1935, edited by S. L. Delp, A. M. Di-Gioia, and B. Jaramaz. Medical Image Computing and Computer-Assisted Intervention, Springer, Berlin, 2000, pp. 375–384.
- [17] Likar, B., Maintz, B. A., Viergever, M. A., and Pernuš, F., Retrospective shading correction based on entropy minimization. *J. Microsc.* **197**, 285–295 (2000).



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, multiple vehicle coordinated control, and robot soccer. Since 2001 he has been a member of Slovenian robot soccer team,

which won 5th place in World Championship in Korea, 2nd and 1st place in European Championships in Austria and Slovenia in 2002, and 2003, respectively.



Matej Kristan received his B.Sc. degree in Electrical Engineering from the University of Ljubljana, Faculty of Electrical Engineering, Slovenia, in 2003. His research interests are in the area of computer vision, image processing, and machine vision applications.



Stanislav Kovačič received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Ljubljana, Ljubljana, Slovenia, in 1976, 1979, and 1990, respectively. Since 1976 he has been with the Faculty of Electrical Engineering, University of Ljubljana. Currently he is a professor at the Faculty of electrical engineering, University of Ljubljana. His research interests include computer vision, image processing and analysis, biomedical and machine vision applications.



Omar A.A. Orqueda has received his B.Sc. degree in 1996 from the Department of Electrical Engineering-National University of the South, Bahma Blanca, Argentina. He is PhD student at this University and leader of the Group on Robotics and Simulation of the National Technical University, Faculty of Bahma Blanca. His research interests include motion planning and control, and physical simulation of autonomous robots with multiple degrees of freedom.