5

Towards Model-based Vision Systems for Robot Soccer Teams

Murilo Fernandes Martins, Flavio Tonidandel and Reinaldo A. C. Bianchi
Centro Universitário da FEI
Brazil

1. Introduction

Since it's beginning, Robot Soccer has been a platform for research and development of independent mobile robots and multi-agent systems, involving the most diverse areas of engineering and computer science. There are some problems to be solved in this domain, such as mechanical construction, electronics and control of mobile robots. But the main challenge is found in the areas related to Artificial Intelligence, as multi-agent systems, machine learning and computer vision. The problems and challenges mentioned above are not trivial, since Robot Soccer is dynamic, uncertain and probabilistic.

A computer vision system for a Robot Soccer team must be fast and robust, and it is desirable that it can handle noise and luminous intensity variations. A number of techniques can be applied for object recognition in the domain of Robot Soccer, as described by (Grittani et al., 2000).

The research of (Grittani et al., 2000) is based only on color information, as well as the research of (Weiss & Hildebrand, 2004) that uses color information to reduce the amount of information contained in each image frame through a called "relevance point filter".

Other researches uses the shape model of the objects to detect on the image, technique generally used in local vision systems. The research of (Gönner et al., 2005), for instance, detects the ball through it's shape model projected on the image, a circumference, but still uses color-only information to recognize the robots.

No matter which technique is used to solve the Robot Soccer computer vision challenge, it must be able to determine position and angle of the robots and the ball with maximum accuracy and minimal processing time possible, because the success of the strategy and control system depends on the information given by the computer vision system.

This chapter extends the work presented by (Martins et al., 2006a), which considers the use of a well known image segmentation technique – the Hough Transform – to locate the mobile robots and the ball on global vision images, taking advantage of the domain characteristics – the robots and ball shape. To implement the Hough Transform technique, which is in most cases implemented in robotic systems using special hardware, only an off-the-shelf frame grabber and a personal computer are used. A new approach to interpret the Hough space is proposed, as well as the method used to recognize objects, which is based on a constraint satisfaction approach.

The chapter is organized as follows: Section 2 describes the most commonly used color spaces in Robot Soccer computer vision systems – RGB and HSI. The Section 3 introduces the Hough Transform. Section 4 describes the system implement in details and Section 5 presents and discusses the results obtained. Section 6 concludes this chapter and presents suggestions for future work, as well as the work that has already been done to enhance the system.

2. Color Spaces – RGB and HSI

Color is the output of a vision system to the perception of different wave lengths reflected by the objects in an observation. Color spaces are representation models that define the primitives do describe the colors. Many image editor softwares work with different color spaces, including the RGB (Red, Green, Blue) and HSI (Hue, Saturation, Intensity) color spaces (Forsyth & Ponce, 2002). The RGB color space is characterized by a cube with R, G and B axis. The Fig. 1 illustrates the RGB cube. Notice that similar color can be represented by many RGB values, what makes hard to build a threshold color filter based on the RGB parameters.
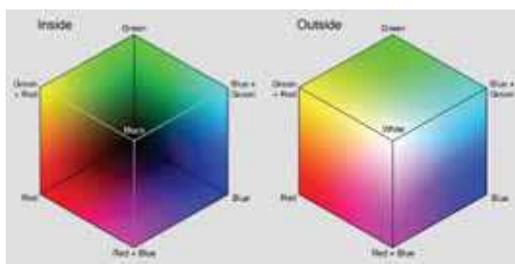

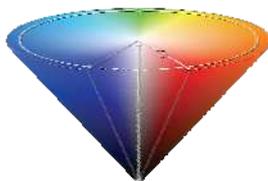Fig. 1. The RGB cube representation

Fig. 2. The HSI conical representation

The parameters R, G and B represents three channels of colors red, green and blue, where, generally, each channel is represented by an 8-bit, implying in 255 different levels of color. The merge of the three channels results in the well known 24-bit true color.

The HSI color space represents a color by Hue, Saturation and Value, where Hue is the color, Saturation identifies how strong the color is and Intensity represents the luminosity intensity of the color. The Fig. 2 shows the conical HSI representation.

The HSI color space is represented by a cone, with a 0 to 360 degrees interval for Hue and a 0 to 100 per cent interval for Saturation and Intensity. Notice that, for any value of Hue, if Saturation is set to 0, then only colors in a gray scale level can be described.

As described by (Penharbel et al., 2004), the HSI color space is better to build a filter to separate colors independent of the luminosity than the RGB color space. The main challenge of using HSI color space is the need of converting the image pixels from RGB, which is tipically the format delivered by ordinary frame grabbers, to HSI. This conversion has a computational cost that can avoid the use of HSI color space and many researches tend to use the RGB color space on their filters, like the filter described in (Bianchi & Reali-Costa, 2000).

3. The Hough Transform

The Hough Transform (HT) (Hough, 1959) is one technique of image segmentation used to detect objects through models adjustment. This technique requires that an object class is determined and such class must be able to describe all possible instances of the referred object. The parameterization of an object class defines the form of this object, therefore, variations of color on the image, or even on the objects, do not affect the performance and efficiency of the HT. To detect objects on an image, the HT tries to match the edges found on the image with the parameterized model of the object.

The HT has rarely been used in robotic systems operating in real time and, when used, it generally needs specific hardware due to its computational complexity. In Robot Soccer, the HT is only used to locate the ball – but not the robots – as described in (Gönner et al., 2005) and (Jonker et al., 2000).

3.1 Circles Detection with Hough Transform

Circles are a very common geometric structure in Robot Soccer since all objects can be represented by one or more circles. The ball for instance is a sphere, but it becomes a circle when projected on the captured image. In FIRA Mirosot and RoboCup Small Size categories, the robots are identified through labels on their upper part. These labels can be of any form and must contain determined a priori colored areas to allow distinction among the robots of different teams. The label used in this work has two circles at 45 degrees with respect to the front of the robot, which complies with FIRA rules (Fig. 3). These circles have the same diameter of the ball used.
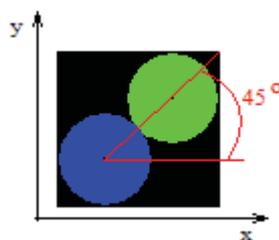

Fig. 3. Label used to identify the robots

Circles are parameterized by the triplet (x_c, y_c, r), which defines a set of equidistant points in r from the central point represented by the Cartesian coordinate (x_c, y_c). A circle can be parameterized using polar coordinates by the equations:

$$x = x_c + r \cos \theta \tag{1a}$$

$$y = y_c + r \sin \theta \tag{1b}$$

In this manner, for known values of the triplet (x_c, y_c, r) and varying the angle θ in all 0 – 360 interval, a complete circumference can be drawn. The space of parameters, also called Hough Space, is three-dimensional. In Robot Soccer, objects move in two dimensions since there is no depth variation of objects on the image, allowing a constant value for radius r to be employed. Thus, the space of parameters becomes bidimensional and it is represented by the point (x_c, y_c).

3.2 The Hough Space
To detect circles of constant radius, on an image that contains only (x, y) edge points, using the HT consists on determining which points belong to the edge of the circle centered in (x_c, y_c) and of radius r. The HT algorithm determines for each edge point of the image a set of possible centers in the Hough Space, set which will be defined iteratively by the variation of θ. Equations (1a) and (1b) become:

$$x_c = x - r \cos \theta \tag{2a}$$

$$y_c = y - r \sin \theta \tag{2b}$$

Fig. 4 demonstrates the algorithm execution for three points on a circle edge of the image on the left, and the respective Hough Space generated is shown on the right. Three circles of radius r drawn on the Hough Space, from 3 points on the edge of the circle with center (x_c, y_c) on the image, intersect themselves in only one point, which is exactly the central point of the circle on the image. Each edge point on the image generates a circle in the Hough Space. Each edge point of this circle in the Hough Space receives a vote. The greater the number of votes a point receives, the greater the probability of this point being a circle center. These points with greater probability are relative maximum of the Hough Space and they define the centers of existing objects on the image.
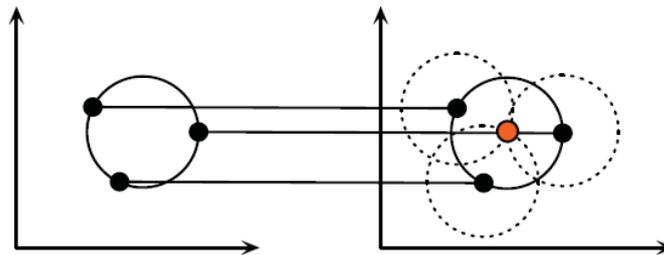

Fig. 4. Example of Hough Space generation

4. Description of the Implemented System

The system described in this section was developed to work in two different situations: first, when the two teams have the same kind of robot label on the top of them and second, when the opponent have a different label. In the first case, the system will recognize the opponent in the same way it recognizes its own players. In the second case, the detection of the opponent robots is done by the color information. Both cases are described below.

4.1 A Model-based Approach to Recognize the Objects

The implemented computer vision system has seven stages, as follows: image acquisition, background subtraction, application of edge filter, Hough Space generation, determination of high probability points to be centers of circles on the image and objects recognition (robots and ball).

4.1.1 Image Acquisition

The image acquisition system consists of an analogical camera with a composite video output and an off-the-shelf video frame grabber based on Bt-878 chipset. This equipment can acquire up to 30 frames per second. In this work, two image resolutions were used: 320x240 and 640x480 pixels, both with color depth of 24 bits. Thirty pictures were captured for each resolution. Fig. 5-left presents one of the images used.

4.1.2 Background Subtraction

As previously mentioned, only the edges of the image are relevant to the HT. Each point of the edge is an iteration of the HT algorithm. To optimize the performance of this algorithm, a simple method of background subtraction was used. It computes the difference between the image captured and a background image, without the moving objects. The background image is updated each frame time, using a method known as Running Average (Piccardi, 2004), according the equation below:

$$B_{i+1} = \alpha F_i + (1-\alpha)B_i \tag{3}$$

where the background image is represented by B, the captured image is represented by F and α is a learning rate used to determine how fast static objects become part of the background image. Although the method is simple, it is efficient for this application because the background does not suffer major modifications. The final image contains only the mobile objects, resulting in relevant edges only. The background image is presented on Fig. 5-center and the result of the background subtraction can be seen on Fig. 5-right.

Fig. 5. (left) Captured image containing a ball and two complete teams of robots (center) background and (right) result of background subtraction

4.1.3 Canny Edge Filter

There are many different techniques capable of extracting edge information on an image, as described by (Forsyth & Ponce, 2002). The present work uses a well known technique for edge detection, the Canny filter (Canny, 1986), which produces binary images. Fig. 6-left shows the result of edges detection with the Canny filter on an image that the background was subtracted (Fig. 5-right) and was converted into gray scale (to lower the processing time).

4.1.4 Hough Space Generation

The Hough Space can be generated from the resultant binary image produced with the Canny filter. The generation of the Hough Space with the algorithm described in Section 2 is correct, but not efficient. Although this algorithm generates the Hough Space correctly, it does several redundant iterations to produce the points of possible circle centers. This redundancy happens because when varying the angle θ, it generates 360 centers points for each edge point with decimal precision. However, the digital images are composed of pixels located on a grid, where each pixel position is an integer number. Therefore, the use of decimal precision is irrelevant and redundant.

To eliminate redundancy, an algorithm for circle drawing proposed by (Bresenham, 1965) was used. This algorithm determines points of a circle using only integers, through the addition of a value determined a priori. Moreover, the algorithm takes advantage of points symmetry in a circle: points position is computed in only one octant and, by symmetry, the points of the other 7 octants are determined, without repetition of previously drawn points. In this way, the processing time for the generation of the Hough Space is minimized, allowing the use of applications in real time. The Hough Space generated for the edge points of the image in Fig. 6-left can be observed in Fig. 6-right and Fig. 7 shows the same Hough Space in a 3D view, where the Z-axis represents the probability of a point being a center of an existing circle on the image.

4.1.5 Circles Determination from the Hough Space

After the Hough Space is generated, the following step is to find out the points that received more votes in the space in order to detect the possible circle centers (x_c, y_c), with r kept constant. It is possible to consider r constant because the distance between the camera and the field is greater than the field dimensions, and the radius variation is less than 5%. This fact also implies in no significant distortion in the images.

This stage is the one that presents greater problems in terms of circle detection. As any edge point generates a set of points (in a circle) that receives votes in the Hough Space, there might be misrepresenting votes producing false relative maximums.

The implemented algorithm verifies whether a voted point reached a minimum number of votes – determined a priori – as the Hough Space is being generated. If a point exceeds this threshold, it is stored only once in a vector of possible centers. At the end of the Hough Space generation, this vector stores the number of votes for each point that exceeded the minimum number of votes.

To guarantee that all points representing real circle centers on the image are in the vector, a low minimum number of votes is defined. But, because of this low threshold, there might be false relative maximums in this vector. Another problem is that, due to the nature of the HT

and the Canny filter, a circle in the image may generate several possible centers, lying close to each other.
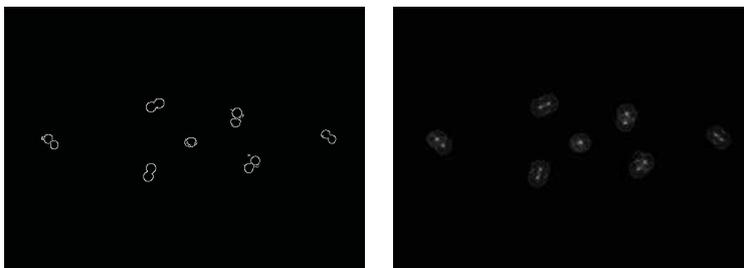

Fig. 6. (left) Result of the application of the Canny filter on Fig. 5-right and (right) Hough Space generated for the edge points, where brighter points have more votes
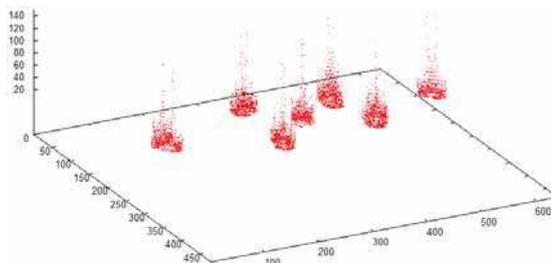

Fig. 7. A 3D view of the Hough Space shown on Fig. 6 (right)

The first step to separate the false center from points where real circle centers are located is to order the points in the vector by the number of votes received using the Quicksort algorithm. After this ordination, the point in the first position in the vector represents the global maximum and is considered a circle center and is inserted in a new vector, the vector of centers.

As the real center of a circle can be defined as the point that was voted the most, and overlapping between two circles do not occurs, all the points that the Euclidean distance to the first center is less $2r$ can be removed from the vector of possible centers. After this, the second position will represent the second maximum and can be considered as a center, and so on.

The algorithm continues until iterations reach a maximum number of circles determined, or until the end of the vector of possible center is reached. This algorithm results in a vector with points distant enough from each other to be considered different circles.

4.1.6 Object Recognition

This stage is the only one that considers color information and image illumination. Therefore, to successfully recognize objects it is necessary to distinguish them, what can only be done when the main colors, defined by the competition rules and different for each team, as well as the secondary colors, used in order to differentiate the robots of the same team, are known.

To define the colors in the image first the mean color of each circle is computed, using a 5 x 5 mask centered in the points found during circle detection. Then, these colors are converted from the RGB to the HSI color space. As mentioned in Section 2, in this color space, pixels are described by the triple (Hue, Saturation, Intensity). The Hue component describes the color of the pixel. It ranges from zero to 360 degrees, where zero degree is red, 60 yellow, 120 green, 240 blue and 300 degrees magenta. The Saturation component signals how white color is present, and the Intensity component represents illumination. In this color space, illumination variations do not modify the hue value of a pixel.

Using this color space, it is easy to define the colors in the image: the mean colors of the circles are ordered by the Hue component using the Quicksort algorithm. As the colors in the HSI color space are always in the same order and at least 30 degrees apart, and the number of circles of each color is known a priori, it is very easy to define the colors of the objects. For the circles in Fig. 5-right, the following colors were found: one orange circle (Hue = 21), three yellow (H = 45, 48 and 50), two green (H = 154 and 160), two cyan (h = 200 and 207), three blue (all at 223) and two pink (H = 348 and 354). The histogram of the Hue component of the same image is presented in Fig. 8.
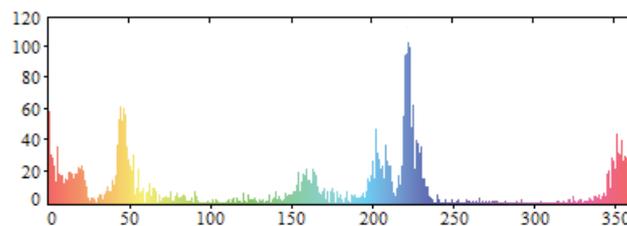

Fig. 8. Histogram for the Hue component of the pixels present in Fig. 5-right

Now that the color of each circle is known, deciding which circle is the ball and which ones are parts of the same robot can be done by solving a problem of constraint satisfaction. According to (Russell & Norvig, 2002) a constraint satisfaction problem is "a special kind of search problem that satisfies additional structural proprieties beyond the basic requirements form problems in general". In this kind problem, the states are defined by the value of a set of variables and the goals specify constraints that these values must obey. In the robot recognition problem, the constraints are that the two circles that are in the robot identification label must be at a fixed distance, $2r$. Another constraint is that each circle of a primary color must be matched with one circle of a secondary color.

To identify which circles belongs to each robot, the algorithm searches in the vector of centers which circles are of a primary color (blue or yellow): this circles are defined as roots of three trees. After defining the roots, the algorithm searches in the vector for circles that are at $2r$ from the primary color circles and adds them as child nodes of the corresponding tree.

If all robots are located distant one from another, this procedure will result in three trees with only one root and one child, defining a robot. Having a center for each labeled circle and the position of the circles known, the algorithm determines the global position (x, y) of the robot on the image and its direction angle in relation to the axis x. As the ball is the only object that can be orange because this color cannot be used for any another object, any orange circle is considered a ball and there is no need to construct a tree to recognize balls.

However, there might be a situation where robots are close to each other, or even touching each other, as in Fig. 9-left. In this case, instead of a tree for each robot, the algorithm will build one tree with three child nodes. It will also build two trees with one child node, as expected (Fig. 9-right). In this case, the algorithm needs to remove child nodes from the tree with three child nodes. To do this, first nodes that are not of a secondary color are removed (it might be another robot's primary color or the ball). And second, the algorithm removes from the wrong tree the circles that are of a secondary color which are already represented in a correct tree. The algorithm will stop when all the trees are correct, representing one robot. The final output of the implemented system can be observed in Fig. 10-right.
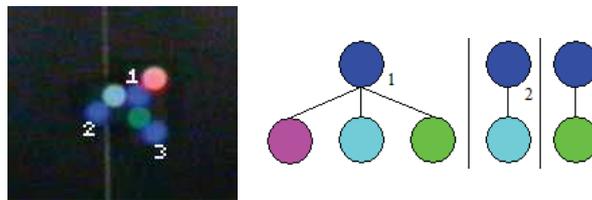
Fig. 9. (left) Image with three robots touching each other and (right) the trees built by the algorithm
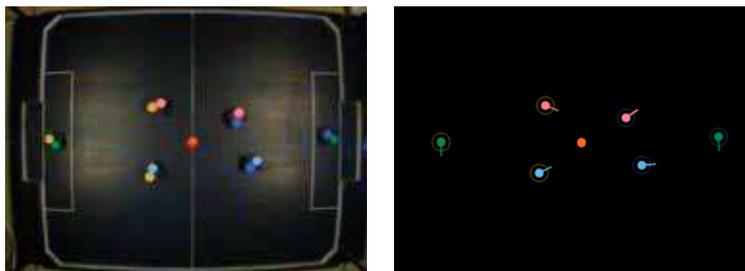
Fig. 10. (left) Captured image containing a ball and two complete teams of robots and (right) result of the execution of the system, showing the computed position of each object

4.2 A Color-based Approach to Recognize the Objects

To detect the opponent robots, the color information is given a priori, but there's no information about the shape of the labels used by opponent. For this case, an approach to recognize the robots by their color, instead of recognizing by the shape model, is used.

This implementation is described in details in (Martins et al., 2006b) and follows the proposal of (Bianchi & Reali-Costa, 2000), where line segments are traced when a pixel of the given color is reached. The system first defines, from the histogram of the image resulting of background subtraction (Fig. 8), the range of the opponent color and then a sparse pixel sampling is done. Notice that the sparser the pixel sampling is, the faster the algorithm will be, but it may imply in an inefficient recognition because this sampling may not reach all opponent's color blobs.

When a pixel of desired color is reached, horizontal and vertical line segments are traced from this pixel position until a non-desired color pixel is reached. The first line segment to be traced is the horizontal, followed by the vertical segment, which is traced from the mid-

point of the horizontal segment. For the process of tracing horizontal and vertical segments, was given the name "cross-processing" (Martins et al., 2006b).

When the cross-processing is applied to a circle, only one iteration is necessary to determine it's center, but when the cross-processing is applied to other shape models or irregular geometric shapes, more iterations are required to determine the center of this object. The cross-processing runs, recursively, n times for each pixel of desired color, where the iteration m starts from the mid-point of the vertical segment achieved at the last $m-1$ iteration.

The cross-processing is executed many times for the same object, because the sparse pixel sampling may result in many pixels of desired color from the same object blob. Each cross-processing execution for each pixel of desired color generates a possible center point.

Some geometric shapes, as the one illustrated on Fig. 11, can have many possible centers after the complete sparse pixel sampling. To determine the center of the object, a mean between the possible center points is computed. This mean is a spatial mean where each dislocates the center to it's nearby. The more a region of the blob has possible center points, the nearer from this region will be the final center computed. The possible center points considered to compute the mean are those which have an Euclidean distance of less than a threshold defined a priori. The Fig. 11 illustrates how the mean computation approximates the possible center points and converges to the center of the object.

Computing the mean allows that, even on noisy images, the objects can have their centers determined. The Fig. 12-right illustrates a noisy image containing objects with different shapes and their centers determined (red point). Notice that there are objects which the centers computed are not the real centers, but these centers remain into the object blob.

This approach doesn't ensure that the real center of the objects will be achieved, but an estimated center close to the real center, no matter the shape of the object, allowing the recognition of the opponent robots for obstacle avoidance and strategy issues, for instance.
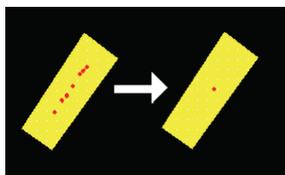

Fig. 11. The cross-processing resulting in the possible center points and the mean computation to determine the center of the object
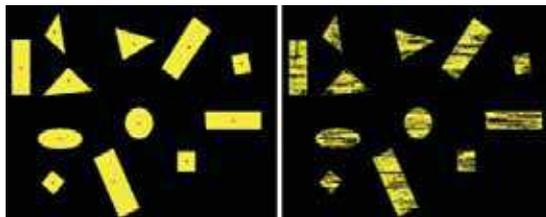

Fig. 12. (left) center detection of objects with different shapes in a standard image (right) center detection on the same image with noise

5. Experiments and Results

The HT and cross-processing implementations were made in C++ and the computer vision library OpenCV (Intel, 2007) was used. The results were obtained in a computer with an Intel Pentium 4 Hyper Threading processor running at 3.2 GHz. The program was executed under Windows XP, configured for priority in real time.

5.1 Execution Times

As the goal was to use the system for an application in real time, the performance evaluation considers as being an acceptable maximum time for the algorithm execution the time interval of an image acquisition. The acquisition systems commonly used in Robot Soccer, as the described in this work, are able to acquire 30 pictures per second. Therefore, the maximum time available for processing is 33 ms.

Table 1 presents the performance results for the HT implemented system, showing the amount of time needed for each processing stage. The values presented are the average of the execution of the system with 30 different images, in two different resolutions (320x240 pixels and 640x480 pixels, both 24 bits NTSC color images). The images used in this test contained six robots and two balls. In each image, the objects are in a different position, spread randomly over the entire field, including the corners. The difference between the sum of the stages times and the total time is small and can be considered rounding error. This results show that the implemented system is capable of recognizing objects not only in real time, but allowing 13 ms for other processes, as strategy and robots control.

As previously mentioned, all adjustable parameters of the system were kept constant for all experiments described in this work. For images with a resolution of 640x480, the radius r was set to 8 pixels, while the minimum number of votes was set to 16. For images with a resolution of 320x240, the radius r was set to 4 pixels and the minimum number to 10. The thresholds of the Canny filter were defined off-line: the low threshold used was 75 and the high threshold was 150.

Task	Image size	
	320x240	640x480
Background subtraction	0,8412 ± 0,001	5,1667 ± 0,003
Color conversion + Canny filter	1,6163 ± 0,002	7,8435 ± 0,005
Hough Space generation	1,4621 ± 0,006	6,3683 ± 0,02
Circle centers determination	0,0334 ± 0,001	0,0267 ± 0,001
Objects recognition	0,0031 ± 0,0001	0,0023 ± 0,0001
Total time	4,0674 ± 0,009	19,4083 ± 0,03

Table 1. Execution Times (in milliseconds)

For the cross-processing the parameter n, the number of iterations, was set to 5, with a pixel sampling interval of 10 pixels for lines and columns, clustering the possible centers with a Euclidean distance of less than the threshold set to 70 pixels. The time measured for images with resolution of 320x240 pixels was 3,037 ms with a standard deviation of 0,0354 and 5,892 ms with a standard deviation of 0,0672 for the 640x480 resolution images.

5.2 Light Intensity Variation Robustness

To verify the robustness of the system in respect to light intensity variation, a second experience was performed: again, 6 robots and 2 balls were placed in a random position of the field and then the light intensity was slowly changed from 300 Lux to 1300 Lux. This experiment was repeated 10 times, each time placing the objects in a different position, randomly chosen. To be able to compare the system described in this paper, the same experiment was performed with a color based system that uses threshold and blob coloring techniques to find the robots, calibrated at 1000 Lux (Penharbel et al., 2004).

The result of these experiments is presented in Fig. 13. It can be seen that, while the system proposed in this work is robust to light intensity variation, detecting all objects in all the trials, the color based system only performed well when light intensity was near 1000 Lux. This experiment also indicates that color noise do not affect the system. As it is model-based, different illumination in the same image may change the color of an object, but, nevertheless, will not affect the system capability to compute the position of any object. Finally, the system was tested while controlling the robots during a real game, with the robots moving in all positions of the field, presenting the same performance as in the two experiences described above.
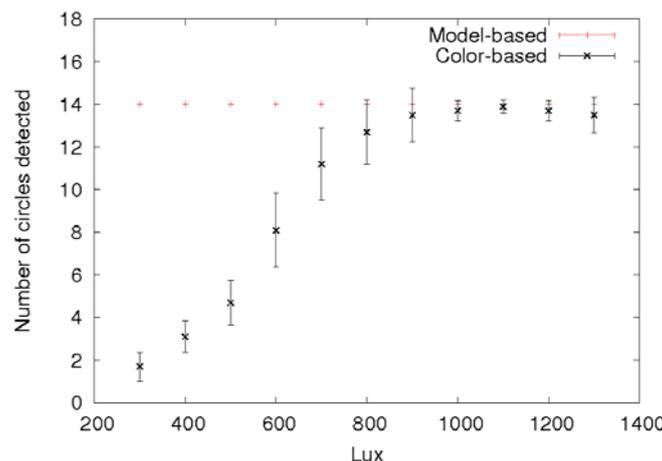

Fig. 13. Number of detected objects versus light intensity variation for the model-based system proposed in this chapter and the color-based system implemented by (Penharbel et al., 2004)

5.3 A Specific Case in which the System May Fail

The system is able to recognize all the robots in any case but one. After the tests, a very specific case was observed, which is very difficult to occur in a Robot Soccer match, where the system may fail. The robots arrangement in the image is one that even the human vision is unable to distinguish the robots with certain. The Fig. 14 illustrates two image sequences of the system processing steps where it may fail and not recognize the robots. Each line shows an example with the captured image, followed by the Canny filter, the Hough Space generated, the circles detected and the objects recognized.
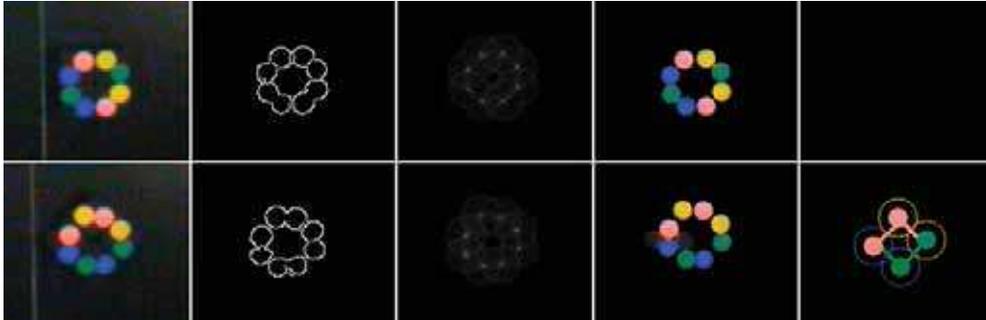
Fig. 14. Specific case where the system may fail, not detecting the robots

On the first image sequence (upper line) of Fig. 14 the robots can't be recognized because every circle of primary color has two child nodes and the constraint satisfaction approach fails. On the Second image sequence (lower line) of Fig. 14 one of the robots is slightly dislocated compared to the first image sequence, resulting in a circle of primary color with only one child node, which results in the recognition of all the robots.

6. Conclusion and Future Work

This chapter described the use of artificial intelligence and computer vision techniques to create a fast and robust real time vision system for a robot soccer team. To increase the system performance, this work proposes a new approach to interpret the space created by the Hough Transform, as well as a fast object recognition method based on constraint satisfaction techniques. The system was implemented entirely in software using an off-the-shelf frame grabber.

Experiments using real time image acquisition allow concluding that the implemented system is robust and tolerant to noises and color variation since it considers just the objects form, and automatically determines the color information, needed only to distinguish the robots among themselves. Robots are well detected in every position of the field, even in the corners or inside the goal area, where light intensity is lower than in the center of the field. The measured execution performance and the tests of object recognition demonstrate that it is possible to use the described system in real time, since it fulfills the demands on performance, precision and robustness existing in a domain as the Robot Soccer.

Future works include the implementation of the control of the camera parameters, such as aperture, zoom, focus, gain and others, in real time. To be able to construct this new part of the system, a camera that allows the control of these parameters through a serial port was bought and is being tested. Finally, distortion lens was not mentioned in this work and, although being small, will be addressed in a future implementation. Another work that is in initial phase of development is the automatic color calibration, in which is intended to use the Hue histogram of colors and clustering algorithms, like K-Means (Forsyth & Ponce, 2002) and is based on recent researches, like (Sridharan & Stone, 2005).

7. References

Bianchi, R. A. C. & Reali-Costa, A. H. (2000). O Sistema de Visão Computacional do Time FUTEPOLI de Futebol de Robôs. *Anais do CBA 2000 – Congresso Brasileiro de Automática*, pp. 2156-2161, Florianópolis, 2000, Sociedade Brasileira de Automática, Brasil.

Bresenham, J. E. (1965). Algorithm for Computer Controlo f a Digital Plotter. *IBM Systems Journal*, Vol. 4, No. 1, 1965, 25-30.

Canny, J. A computational approach to edge detection. (1986). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, Nov. 1986, 679-698.

Forsyth, D. A. & Ponce, J. (2002). *Computer Vision: A Modern Approach*. Prentice Hall, New Jersey.

Gönner, C.; Rous, M. & Kraiss, K. (2005). Real-time adaptative colour segmentation for the robocup middle size league. *Lecture Notes in Computer Science*, Vol. 3276, 2005, 402-409.

Grittani, G.; Gallinelli, G. & Ramírez, J. (2000). FutBot: A Vision System for Robotic Soccer. *Lecture Notes in Artificial Intelligence*, Vol. 1952, Nov. 2000, 350-358.

Hough, P. V. C. (1959). Machine analysis of bubble chamber pictures. *International Conference on High Energy Accelerators and Instrumentation*, 1959.

Intel (2007). Intel Computer Vision Library. Online Available at: <http://www.intel.com/technology/computing/opencv/>. Accessed on: 07/30/2007.

Jonker, P.; Caarls, J. & Bokhove, W. (2000). Fast and accurate robot vision for vision based motion. *Lecture Notes in Computer Science*, Vol. 2019, 2000, 149-158.

Martins, M. F.; Tonidandel, F. & Bianchi, R. A. C. (2006a). A Fast Model-Based Vision System for a Robot Soccer Team. *Lecture Notes in Artificial Intelligence*, Vol. 4293, 2006, 704-714.

Martins, M. F.; Tonidandel, F. & Bianchi, R. A. C. (2006b). Reconhecimento de Objetos em Tempo Real para Futebol de Robôs. *Anais do XXVI Congresso da Sociedade Brasileira de Computação*, pp. 173-182, Campo Grande, 2006, Sociedade Brasileira de Computação, Brasil.

Penharbel, E. A.; Destro, R.; Tonidandel, F. & Bianchi, R. A. C. (2004). Filtro de Imagem Baseado em Matriz RGB de Cores-Padrão para Futebol de Robôs. *Anais do XXIV Congresso da Sociedade Brasileira de Computação*, Salvador, 2004, Sociedade Brasileira de Computação, Brasil.

Piccardi, M. (2004). Background Subtraction Techniques: A Review. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics SMC 2004*, The Hague, 2004, Netherlands.

Russell, S. & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey.

Sridharan, M. & Stone, P. (2005). Towards Eliminating Manual Color Calibration at RoboCup. In: *RoboCup 2005*, Itsuki Noda, Adam Jacoff, Ansgar Brednfeld & Yasutake Takahashi, Springer Verlag, Germany.

Weiss, N. & Hildebrand, L. (2004). An exemplary robots soccer vision system. *Proceedings of the CLAWAR/EURON Workshop on Robots in Entertainment, Leisure and Hobby*, Vienna, 2004, Austria.

Robotic Soccer

Edited by Pedro Lima

ISBN 978-3-902613-21-9

Hard cover, 598 pages

Publisher I-Tech Education and Publishing

Published online 01, December, 2007

Published in print edition December, 2007

Many papers in the book concern advanced research on (multi-)robot subsystems, naturally motivated by the challenges posed by robot soccer, but certainly applicable to other domains: reasoning, multi-criteria decision-making, behavior and team coordination, cooperative perception, localization, mobility systems (namely omni-directional wheeled motion, as well as quadruped and biped locomotion, all strongly developed within RoboCup), and even a couple of papers on a topic apparently solved before Soccer Robotics - color segmentation - but for which several new algorithms were introduced since the mid-nineties by researchers on the field, to solve dynamic illumination and fast color segmentation problems, among others. This book is certainly a small sample of the research activity on Soccer Robotics going on around the globe as you read it, but it surely covers a good deal of what has been done in the field recently, and as such it works as a valuable source for researchers interested in the involved subjects, whether they are currently "soccer roboticists" or not.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Murilo Fernandes Martins, Flavio Tonidandel and Reinaldo A. C. Bianchi (2007). Towards Model-based Vision Systems for Robot Soccer Teams, Robotic Soccer, Pedro Lima (Ed.), ISBN: 978-3-902613-21-9, InTech, Available from: http://www.intechopen.com/books/robotic_soccer/towards_model-based_vision_systems_for_robot_soccer_teams

INTECH

open science | open minds