

ROBOFEI TEAM DESCRIPTION 2007

ANDRÉ DE O. SANTOS, BRUNO G. CARNAES, DIEGO F. DE OLIVEIRA, GUILHERME CORDEIRO, FERNANDO P. TAVARES, JOSÉ ANGELO G. JUNIOR, LEONARDO D. BELLAN, LUIZ H. D. KORNRUMPF, MURILO F. MARTINS, RONALDO K.I SATOMI, VALQUIRIA F. PEREIRA, FLAVIO TONIDANDEL, REINALDO A. C. BIANCHI

*Robotic and Artificial Intelligence Lab – Centro Universitário da FEI
Av. Humberto de Alencar Castelo Branco, 3972 – São Bernardo do Campo - SP*

E-mails: flaviot@fei.edu.br; rbianchi@fei.edu.br

Abstract—This article presents the techniques of computational vision, robot control and game strategy used by RoboFEI Robot-Soccer team. The computational vision is based on the Hough Transformation for circles and the opponent detection is based on colors, the robot control has a Proportional-derivative control with Kalman filter, and the strategy is based on game zones actions.

Keywords— Mobile Robots, Robot Soccer, Hough Transformation, Proportional-Derivative control

Resumo— Este artigo descreve as técnicas de Visão Computacional, controle de robôs e estratégia de jogo usado pela equipe de futebol de robôs RoboFEI. A Visão Computacional é baseada em Transformada de Hough para círculos e a detecção do adversário baseada em cores, o controle dos robôs são através de um controle Proporcional-derivativo com filtro de Kalman e a estratégia é baseada em ações em zonas de jogo.

Palavras-chave— Robôs Móveis, Futebol de Robôs, Transformada de Hough, Controle Proporcional-Derivativo

1 Introdução

Since its beginning, the Robot-Soccer challenge has been a motivating platform to research in Artificial Intelligence and autonomous mobile robotics development. It happens because the development of a Robot-Soccer team involves several research topics, like Computer Vision, robot hardware, robot mechanics, control systems and Artificial Intelligence techniques for the strategy of the team.

This paper presents the RoboFEI team 2007 and describes the vision system, strategy, path planning and control of the mobile robots used.

2. Computer Vision system

In the Mirosoft [1] Robot-Soccer league, the robots are distinguished by the colors on their top face labels, which are determined by the rules.

The geometric shape of the robot color labels are not stipulated by the rules, just the minimum area of the color, 12.25 cm².

Detecting objects by their shape needs an *a priori* knowledge of its shape. Identify the objects by their color may cause malfunction of the vision system when luminosity variation occurs.

The vision system of RoboFEI team detects adversarial robots of any shape by a specific color and the teammates by a circle shape, and the system just consider the color information to identify which are the robots.

Described in details by [2], the vision system was

developed by using techniques like the Canny edge filter [3] and the Hough Transformation [4].

This vision system has seven steps to process the image. They are: image capture, background subtraction, color to gray-scale conversion, the Canny edge filter, Hough space generation, determination of the points that have high probability of being circle centers and the objects recognition (robots and ball).

The image capture is done with a capture board that acquires 30 frames per second at 640x480 maximum resolution.

For the Hough Transformation to detect circles, just edges information is necessary. So the background subtraction ensures that just objects will remain in the image. The Figure 1 illustrates the background subtraction.



Figure 1: Background subtraction

Then, the colors are converted to gray-scale and the Canny edge filter is applied. This processing results in an image only with the objects edges, ready for the Hough Transformation, as can be seen in figure 2.

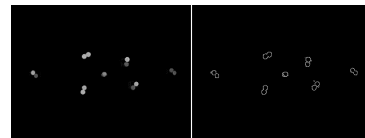


Figure 2: Canny edge filter result

Given the objects edge image, the Hough space is generated. This Hough space is a new gray-scale image which points represents the probability of that point to be a center of a circle in the original image captured.

Considering that the Canny edge filter doesn't result in perfect edges, many of the points in the Hough space can represent fake circle centers. The Hough space is showed in Figure 3.

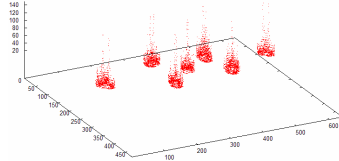


Figure 3: Hough space in 3D

To detect adversarial robots, the system is an extension of the work described in [5]. This detection, that uses color information, ignores the shape that is placed on the robots label, as described in details in [6]. Basically, the idea is to search in the image, jumping pixels in horizontal and vertical axes, any pixel with the specified color and trace internal segments for the pixels that have the same specified color. This pixels that belongs to an object is called *object point*. The segments are traced on the horizontal and a middle point is calculated. Then, a vertical segment is traced and its middle point is calculated. This process is called *cross* (figure 4).

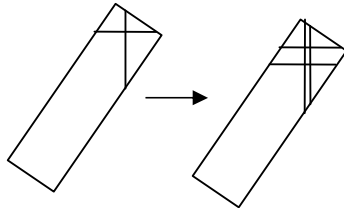


Figure 4 – Example of multiples cross processing

In a circle, only one cross processing is necessary to find its center. However, there are other shapes or images with noise where the cross processing must be performed more than one time for each object point.

It is important to highlight that the cross processing is performed many times on a single object since the searching process with jumps will find many different object points of the same object. This will result in many centers that we call provisory centers (figure 5).

The center of an object is estimated by a calculation of provisory centers that belongs to an object (centers where the Euclidian distance is less than an specific value). This calculation considers the concentration of points on the object and it finds a estimate center that is relatively closed to the real center of the object. One important result is that this process works even in noisy images and only needs a good calibration of opponent color. More details in [6].

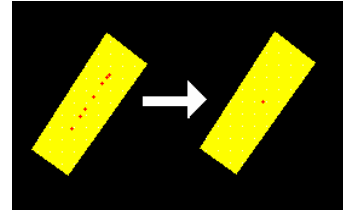


Figure 5 – Provisory centers and the estimate center

The performance of the vision system was analyzed in [6]. For the tests, images with 640x480 pixels were considered, the diameter of the circles was fixed in 16 pixels, and the votes were lower bounded in 16.

The canny parameters were 75 (below) and 150 (high) and the for the cross processing were considered the worst scenario.

Table 1. Execution times

Function	Execution Time
<i>Background extraction</i>	<i>~5ms</i>
<i>Gray Scale + Canny</i>	<i>~8ms</i>
<i>Hough Space</i>	<i>~6ms</i>
<i>Circle Centers Determination</i>	<i>~0ms</i>
<i>Objects Recognition by color</i>	<i>~0ms</i>
<i>Cross Processing</i>	<i>~6ms</i>
Total	<i>~25ms</i>

All functions in Table 1 were operated in sequence, except the cross processing that can be parallelized.

3. Team Strategy

The strategy developed for the team consists in divide the field in seven or eight (middle M can be divided in two) zones, three zones on the attack side, three zones on the defense side and one zone on the middle, as can be seen in Figure 6.

The strategy starts analyzing the zones in which the robots of the team, the adversarial robots and the ball are. This information is passed to a state machine that analyses the possibility for attack and defense.

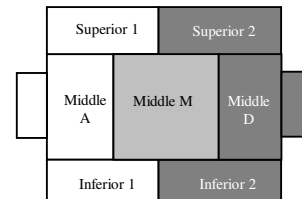


Figure 6: field zones

The states are composed by variables that indicates the zones where the robots are. Depending on the state of these variables, the strategy defines the action for each robot. For example, to intercept the ball, the nearest and relative to x coordinate behind the ball robot is chosen.

Depending on the zone the robot is, an action has a weight greater than the others. In the zone Middle A, for example, a “Kick to Goal” action has priority.

There are few actions that can be performed that, when performed in sequence, can show complex movements. The actions are: kick to goal, ball interception (the same as opponent obstruction), run with ball, pass (kick ball to a specific zone), stop and spinning (to kick the ball in corners or to retrieve ball from opponent).

To move the robots through the field, is applied a technique known as Piecewise Bezier [7] to trace the path of each robot of the team. The first path traced is composed just by line segments, as can be seen in Figure 7(a). Each object that intersects a line segment has a potential field that makes the algorithm to retrace the line segment to avoid that obstacle.

Given the line segments, Bezier is used to smooth the path and the points are passed to the control system. Figure 7(b) shows the smooth path generated by Bezier algorithm.

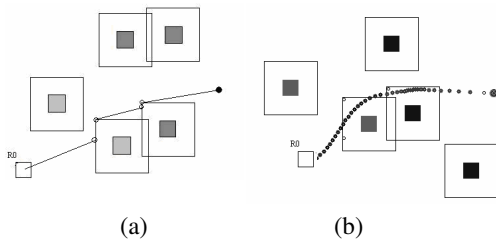


Figure 7: Path planning using Piecewise Bezier

The control of the path is done based on the estimated time for the robot to follow the path. The next point that the robot must follow in the next 30 milliseconds is determined and it’s applied to the control system. This time is fixed in 30 milliseconds because the system captures 30 images per second. So, the vision system, strategy and control must spend a maximum of 30 milliseconds per iteration. If it won’t happen, image frames will be lost.

4. Control System

The robot must follow a path given by the strategy, from a (x,y) initial coordinate another (x,y) final coordinate, like in Figure 8. The control system sets the motor’s velocity of the robots according to the initial point to reach the final point.

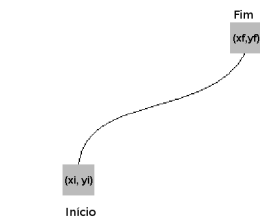


Figure 8: Following a path

The mathematical model used is based on the robot physical characteristics. The robot is a two-wheel differential driver. The differential kinematics was used to describe the mathematical model of the robot. Figure 9 shows the kinematics model of the robot.

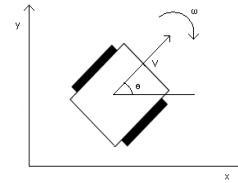


Figure 9: robot kinematics

Where:

- ω is the angular velocity
- v is the linear velocity
- θ is the rotation angle
- (x,y) are the Cartesian coordinates

By the independent variation of the wheels velocity, a linear or angular movement can be determined.

The equations to control the both wheel velocities are described in details in [8], and they are:

$$vr(t) = (kd_p \cdot Se + kd_d \cdot ve) \cdot d + ka_p \cdot \theta$$

$$vl(t) = (kd_p \cdot Se + kd_d \cdot ve) \cdot d - ka_p \cdot \theta$$

The constants of these equations were determined empirically.

The use of a classical PD controller still gives some error in set the velocities because the inertia of the robots and the processing time spent by the computer until the robot receive the data causes a delay and the position and rotation angle of the robot considered are not the real position and rotation angle. To predict these position and rotation error, the Kalman Filter [9] was used. The constants of the filter was set empirically.

The tests were done in the SimuroSot [10] simulator. The result can be seen in Figure 10.

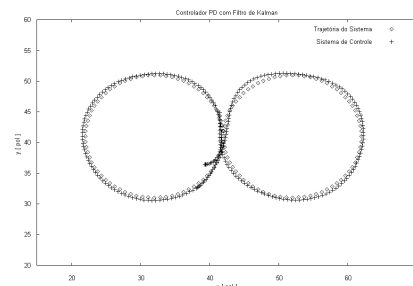


Figure 10: PD controller combined with Kalman Filter

The path given to the control system was a set of points forming the shape of an “eight”.

The same tests were performed in real robots and the results are closely to the simulation results. However, the constants must be set differently (empira-

cally).

5. Future Developments

The control system is completely based in the vision system. In fact, the control feedback is done by the positions of the robots extracted from the images. This method introduces imprecision to the controller that can be avoided if each robot has an internal PD to control each wheel. Odometers are considered in each wheel and, during the gap between to images (30ms) the internal PD can keep the robot on track.

Some preliminary results show that the robot can only perform a straight line if the internal PD with odometers is presented. Without this internal PD, a little difference between wheel motors or little modification of the angle of the robot in vision system is sufficient to make the robot to oscillate instead of follow a straight line.

Therefore, the control system will be improved with an internal PD control in a near future.

6. Conclusions

The new RoboFEI team, if compared to the old ones, has a vision system that handles better with noise, luminosity variation and perform a faster time.

The strategy has more actions to choose for the robots, given the position and rotation angle by the vision system more accurately than the old system. Now the system has an obstacle avoidance and path planning algorithm for the better move of the robots.

The control system is capable to control the robots accurately, estimating the future position and rotation angle and calculating the velocities for the motors.

The next steps are testing exhaustively the new system and find the lacks to be improved.

References

- [1] FIRA (2006). FIRA Small League MiroSot Game Rules. Available at: <http://www.fira.net/soccer/mirosot/rules_slm.html>. Last access: 09/30/06.
- [2] Martins, M. F.; Tonidandel, F.; Bianchi, R. A. C. (2006). A Fast Model Based Vision System for a Robot Soccer Team. Lecture Notes in Computer Science, 2006.
- [3] Canny, J. (1986) A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.8, n. 6, p.679-698.
- [4] McLaughlin, R. A.; Alder, M. D. (1998). The Hough Transform versus the UpWrite. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 4, p.396-400, 1998.
- [5] Bianchi, R. A. C.; Reali-Costa, A. H. (2000). O Sistema de Visão Computacional do Time FutePOLI de Futebol de Robôs. In: CBA 2000 – Congresso Brasileiro de Automática, Florianópolis, 2000. Anais. Florianópolis, Sociedade Brasileira de Automática, 2000. p.2156-2161(in Portuguese).
- [6] Martins, M. F.; Tonidandel, F.; Bianchi, R. A. C. (2006). Reconhecimento de Objetos em Tempo Real para Futebol de Robôs. (2006). In: Jornada de Robótica Inteligente – Encontro Nacional de Robótica Inteligente, 2006, Campo Grande. Anais do XXVI Congresso da Sociedade Brasileira de Computação. Campo Grande: Sociedade Brasileira de Computação. p. 173-182 (In Portuguese).
- [7] Hwang, J.H, Arkin, R. C. and Kwon, D.S. (2003) "Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.
- [8] Sen Gupta, G., Messom, C, Demidenko, S. (2004). Vision Assisted Measurement for Optimization of Robot Motion and Position Control Functions, *IEEE Instrumentation and Measurement Technology Conference*, Como, Italy, pp 297- 302, ISBN 0-7803-8248-1.M. Young, *The Technical Writers Handbook*. Mill Valley, CA: University Science, 1989.
- [9] Sen Gupta, G., Messom, C, & Demidenko, S. (2003) "Global Vision Based Optimization of Control Functions and Kalman Filtering for Robotic Applications", *Proceedings of IEEE International Conference on Computational Intelligence, Robotics and Autonomous Systems*, (CIRAS 2003), ISSN 0219-6131.
- [10] FIRA. FIRA Middle League SimuroSot. Available at: <<http://www.fira.net/soccer/simurosot/overview.html>>. Last access: 09/30/06.