A Robotic Architecture for RoboCup

M. Bracho[1], M. Castro[2] and J. A. Moreno[2]

[1]Laboratorio de Inteligencia Artificial
Decanato de Ciencias,
Universidad Centroccidental "Lisandro Alvarado",
Barquisimeto, Venezuela
`mbracho@delfos.ucla.edu.ve`

[2]Laboratorio de Computación Emergente,
Facultades de Ciencias e Ingeniería,
Universidad Central de Venezuela. Caracas, Venezuela.
`jose@neurona.ciens.ucv.ve`

Abstract: A Robotic architecture tailored for the Small-Size Robot (F180) League of the RoboCup competition is presented. It is described in terms of its mayor hardware and software components.The main hardware components are the robots with wireless communication to the offboard control PC, the global vision system with a digital camara in a host PC interconnected to two other client PCs all communicating in TCP/IP mode. The software environment is of modular nature and is organized on a distributed computing environment. Both the vision and robot control systems are based on Cellular Automata used as computational tools. The image of the workspace is represented in a convenient cellular space for filtering, color classification, object tracking and path planning. Experimental results show that the proposed architecture renders reasonable real time response so as to face the challenges posed in the RoboCup Small-Size Robot league.

Keywords: Path Planning, Heuristic, Cellular Automata, Exact Cell Decomposition

1 Introduction.

The ultimate goal of robotics is to build artificial agents capable of displaying rational and complex behaviors in the accomplishment of a specific task. The tasks under consideration are characterized by the need of meaningful interactions of the agent with a real dynamic world through a physical body. The achievement of this goal is a chal-

lenging interdisciplinary enterprise involving an important research effort on some innovative technologies such as artificial intelligence, artificial vision, image processing and emergent computing, [1].
In this paper a Robotic architecture tailored for the Small-Size Robot (F180) League of the RoboCup competition [2,3,4] is presented. It is described in terms of its mayor hardware and software components. In respect to hardware, the radio controlled robots, the global vision system together with three interconnected PCs, communicating in TCP/IP mode, are the main components. The software is of modular nature and is organized on a distributed computing environment. Both the vision and robot control systems are based on Cellular Automata used as computational tools. The image of the workspace is represented in a convenient cellular space for filtering, color classification, object tracking and path planning. Experimental results show that the proposed architecture renders reasonable real time response so as to face the challenges posed in the RoboCup Small-Size Robot league.

The organization of the paper is as follows: In the second section the hardware architecture is presented, the robot, the global vision system and the PC architecture are briefly described. In the third section the software of the system, including the image processing algorithms, robot tracking and path planning algorithms are presented. In the fourth section the Flood Algorithm and its complexity is discussed, and in the lasts section experimental results are discussed.

2 Hardware of the UCV Robot Team

2.1 Robot

The UCV-robot has been conceived as an experimental tool that allows the execution of tests and benchmarking, of diverse robotic algorithms, in a real world environment. Its construction is based on the use of cheap hardware parts. UCV-robot is a simple small radio controlled mobile robot. Its dimensions are width: 12 cm., length: 12 cm. and height: 10 cm. It is built in two layers. The lower platform holds two FMA Direct S3501AM, Futaba type, servo motors, one for each of the two wheels of 6.4cm diameter, in wheelchair position. The wheels are powered independently for both steering and drive. An incremental encoder is placed into the servo motor gear outputting 400 pulse per wheel revolution. The servos are superior to stepper motors due to their low current requirements, high torque, low electromagnetic interference, high resolution of rotation, light weight, and built-in current amplification. They are easily positioned by a simple control pulse. The control pulse is defined by a *width* that represents "*stop*" and a *width change* or *delta_w* that yields "*full travel*". A typical value for the stop width on the Futaba R/C type servo is 1.50 ms., and the range of maximum and minimum pulse widths are typically between 1.0 ms and 2.0 ms. The servo requires the reception of the control pulse approximately every 20 ms or 50 times a second. Thus, the motor controller keeps the motors "*on*" at the current setting until either a new pulse arrives with the same or different width or approximately 20 ms has elapsed without receiving a new pulse. The second platform of the robot holds the battery pack, a voltage regulator, a communication and a CPU board. Each board is equipped with an interconnected microchip pic16c84 processor. The robot communicates with the main computer by a radio trans/ceiver device. The micro controller on the communication board transmits, receives and does data verification. The micro controller on the CPU board command the servos. Figures 1 and 2 show different views of the UCV Robot. Que tipo de radio

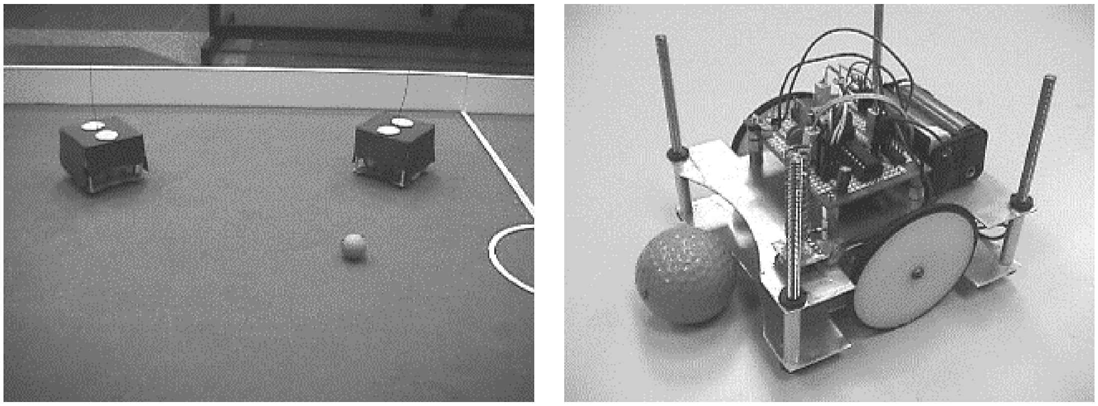


Figure 1. Detail of the UCV-Robot (left). Two robots chasing the ball (right) Note the two color patches on top of the robots for positioning purposes

2.2 Sensing

The architecture uses a global vision system as a primary sensorial source. It operates as a global positioning system for the robots and the ball in the playing field. At the present time the system consists on a Connectix Quick Cam digital camera, placed 2.5 m. directly above the playing field and connected to a PC by its parallel port. The camera is operated in 24bit mode with a resolution of 320x240 pixels, with a frame rate of approximately 10 frames per second.

2.3 Communication

The communication protocol uses a master-slave technique, in which only one device (the host) can initiate transactions (queries). The other devices (the robots) respond by taking the action requested in the query or by supplying the requested data to the host. The host can address individual robots. The robots return a message (response) to queries that are addressed to them individually. The protocol establishes the format for the host's query by placing it in the pretransmition bytes, it consists in the device address, a function code defining the requested action, any data to be sent, and an error-checking field. The robot's response message contains fields confirming the action taken, any data to be returned, and an error-checking field.

The Query: The function code in the query tells the addressed robot what kind of action to perform. The data bytes contain some additional information that the robot will need to perform the function. For example, function code 01 will query the robot to move forward and respond with the incremental encoder contents. The data field must contain the information telling the robot how many counts to advance. The error check field provides a method for the robot to validate the integrity of the of the message.

The Response: If the robot makes a normal response, the function code in the response is an echo of the pretransmition bytes, and the device address in the query. The data bytes contain the data collected by the robot, such as incremental encoders counts or status. If an error occurs, the robot ignores the message. The error check field allows the host to confirm that the message contents are valid. If an error occurred or if the robot does not respond the host will retransmit the message.

3 Software of the UCV Robot Team

3.1 Overall Software Architecture

The software architecture of the system addresses a combination of high-level and low-level processes. It is conceived as the combination of the vision system in a host computer server and several client computers where the control processes of the individual robots are executed. Data is shared between the computers using a TCP/IP communication protocol.

The complete system is fully autonomous consisting of a well defined processing cycle. The object recognition and tracking algorithm perceives the dynamical scene in the playing field and processes the images, resulting in the positions and orientations of each robot and the ball. This information, in a cellular automata representation, is distributed to the different robot controlling algorithms in the client computers. Each agent evaluates the world state and uses its strategic knowledge to decide its next action. Actions are motion commands that are sent by the host, through a radio link, to the robots.

3.2 Object Recognition and Tracking System.

The RoboCup rules specify well defined colors for different objects in the field hence the identification of the pixel colors lead to a simple and straightforward manner for the classification and recognition of the objects in the scene. Several methods for this task have been presented in the literature [2,3,4,5,6]. In our case we have found that a simple and fast minimum and maximum threshold method for each color channel is most suitable. In a practical setting the robustness of this method allows the recognition of a 4 color class domain. Considering scenes similar to those occurring in the RoboCup robot soccer cup tournament the four detected colors can be used to identify the ball and the dual color identification of the robots. The field background and white lines do not classify in any of the color classes and are hence identified by default. Figure 1 shows an actual input sample image. The scene shows two types of robots differentiated in colors, a ball, goals A and B, the field and the field lines. The image is of 320 x 240 pixel resolution, RGB coded, recorded by the digital camera above the playing field.

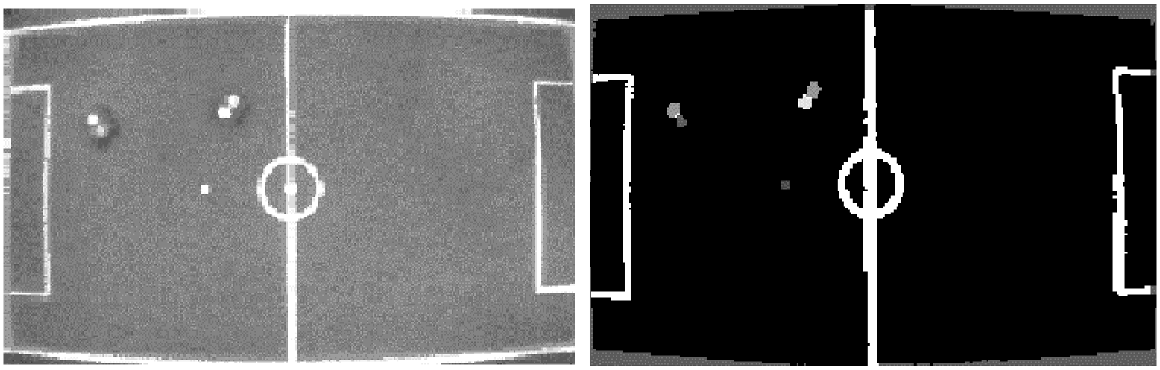


Figure 2. Actual input image (left) and false image after color detection (right).

As the pixels are being classified into one of the 4 possible classes, static and mobile objects are identified along with their relative positions in a specific coordinate system. This information is supplied to the client computers in the form of cell state values in a cellular automata representation providing in this way a discrete version of the workspace. This CA representation of the workspace is used by the path planner to obtain free collision routes for the robot motion. Its initial states are (0) field, (1) lines, (2) robots type A, (3) robots type B, (4) ball, (8) empty spaces.

3.3 Path Planner

Several path planning methods have been presented in the literature [7,8,9]. In the present work a cellular automata approach is applied [10]. For this purpose a free flying robot without dynamic and kinematic constrains is considered. It consists therefore of a single point without any consideration about its orientation. The workspace is decomposed in a finite collection of non intercepting square cells. As previously noted this cell array is the cellular space of a CA and constitutes the configuration space, where the objects in the scene occupy a certain number of cells. Assuming a grid of given size, in the present case a 288x178 grid where each cell corresponds to approximately 1cm^2 of real space, the discrete configuration space can be defined by the coordinates of each cell:

$$C=\{(x,y)\mid x\in\{0,\cdots,x_{max}\}, y\in\{0,\cdots,y_{max}\}\} \tag{1}$$

In this representation each object is a particular set of cells and the free space is the set of cells not belonging to any object. A path in free space is a finite discrete sequence of cells. The input to the algorithm is the CA representation of the workspace provided by the object recognition system, it contains information about the location of each colored object in the workspace. The position coordinates of each object is calculated as the centroid of the corresponding colored pixels. In phase one of the algorithm, the image information is transformed to the discrete cellular space with four possible initial states: (0) free, (1) strange object, (2) initial position (position of the robot considered), (3) goal position (position of the ball or strategic position). In phase two, a predefined template of cells in state (1) is placed over the position of the strange objects in order to account for the physical size of the robot. The size of the template depends on the size of the robot and of the cell, for the typical resolutions used the template is about 24x24 cells. The schematic diagram in Figure 2 shows these processes.

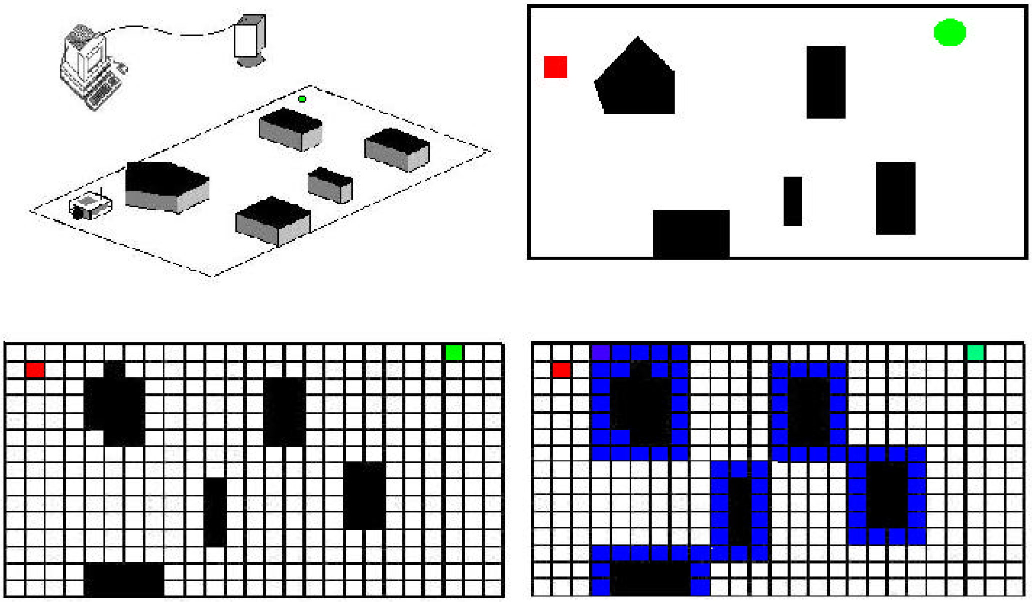


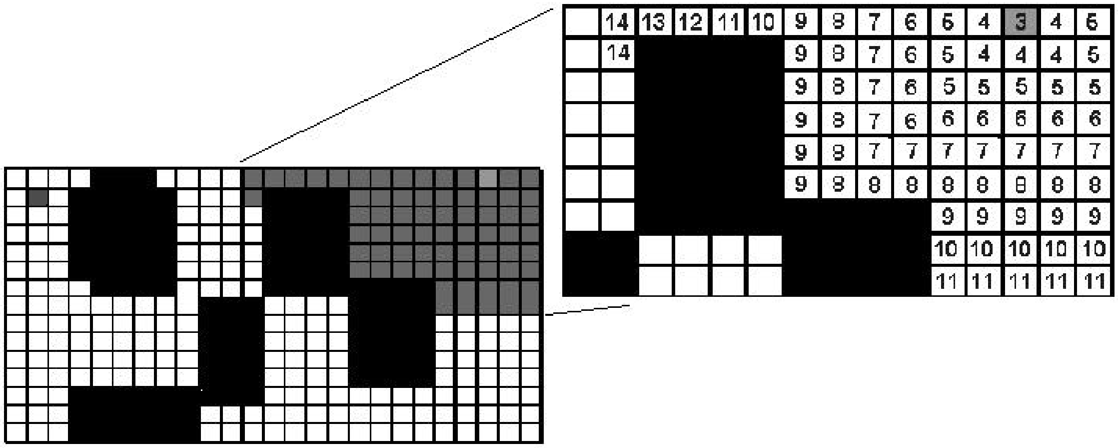
Figure 2.Schematics of the image capturing process and workspace CA representation.

In the third phase, the final configuration resulting in the prior phase is used as initial configuration for a CA dynamics that computes the Manhattan distance between the initial and goal positions. In this cellular automata the possible states for the cells are the following: (0) free space, (1) obstacles (strange objects), (2) initial position, (3) goal position, (4) Manhattan distance 1 to the goal,......, $(3+l)$ Manhattan distance l to the goal. The transition rule applied to evolve this cellular automata is

$$s_i^{t+1} = \left\{ \begin{array}{c} s_x^t + 1 \ \ if \ s_i^t = 0 \wedge \exists \, x \in h_i^t \mid s_x^t \geq 3 \\ s_i^t \end{array} \right\} \tag{2}$$

In Figure 3 this process that resembles a flood from the goal to the initial position is shown.

Figura 3. Inserte aquí la leyenda de su figura.



	14	13	12	11	10	9	8	7	6	5	4	3	4	5
	14	■	■	■	■	9	8	7	6	5	4	4	4	5
		■	■	■	■	9	8	7	6	5	5	5	5	5
		■	■	■	■	9	8	7	6	6	6	6	6	6
		■	■	■	■	9	8	7	7	7	7	7	7	7
		■	■	■	■	9	8	8	8	8	8	8	8	8
		■	■	■	■	■	■	■	■	9	9	9	9	9
■	■					■	■	■	■	10	10	10	10	10
■	■					■	■	■	■	11	11	11	11	11

The flood dynamics is stopped either when the cell with the initial position is reached or when all cells in the cellular space are different from 0 in which case no path is possible. In the former case a path is calculated by going backwards from the goal to the start positions in a descending manner. Due to the existence of saddle points in the navigation function, the shortest path between an initial and goal position is not well defined. Often a cell has more than one neighbor with the same Manhattan distance to the goal. To follow always the steepest descend of the function may not work because there could be cases where the gradient may have the same value in several directions. In order to select a reasonable good path two heuristics are applied among the neighbors of a cell: (I) The cells that fulfill the condition of being in direction of the steepest descend and allows the conservation of the direction of movement for the robot are selected in the path with the highest priority or (II) those cells which are in direction North, South, West and East and have a small angle to the moving direction of the robot are also selected with high priority. In general it is found experimentally that these two heuristics efficiently produces reasonable good paths with few minimal changes of the direction of movement (commands for the robot). Figure 4 shows this process:

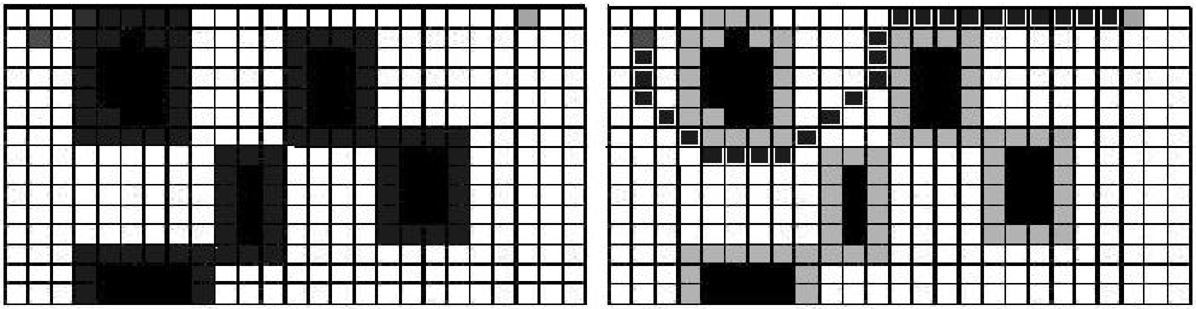


Figura 4. Original CA representation of the workspace (left). Calculated path (right)

The knowledge of the sequence of cells that define a path together with the initial direction of movement of the robot allows, in a straightforward manner, the production of a list of commands which guides the robot along the path to its goal.

The experiments indicate that the CA based path planning algorithm allow real time motion planning. In effect, using as input an image of 288*178 pixels resolution, an average response of 10 ms per path was measured.

4 Experimentation

Experiments carried out on a table tennis size football playing field using a pair of UCV-robots chasing a ball on a table tennis sized football field. It is observed that the actual execution time of the algorithm depends on several factors, the most important of which are the frame rate of the camera and the data communication mode by the parallel port that limits the transfer velocity. In fact, the executing program waits for the camera, in its process of acquiring the image and transferring the data, in order to compute the path.

5 Conclusions

A cellular automata approach for solving the robot path planning problem that yields very efficient experimental performance on real life path planning situations is proposed. The cellular automata algorithms were tested with different workspace configurations and cellular space sizes over real time images from a digital camera. The computer effort depends on the size of the cellular space and the length of the resulting path.

This reduced time complexity together with the simplicity of the cellular automata simulation allows the algorithm to perform quite well on serial machines. Some of the key practical advantages of the method are: it does not require parameter tuning, real time performance for any kind of workspace scene, consistent behavior over repeated experiments, it can handle path planning in dynamic environment. In conclusion the practical performance is very satisfactory however further study involving many more benchmarking examples are necessary.

References

1.- D. Kortenkamp, R.P. Bonasso y R. Murphy, "Artificial Intelligence and Mobile Robots",AAAI Press/The MIT Press, Melo Park 1998

2.- Kitano H. (Eds). (1998). RoboCup –97: Robot Soccer World Cup I. Lectures Notes in Computer Science; Vol 1935: Lectures Notes in Artificial Intelligence. Berlin: Springer-Verlag.

3.- Veloso M., Pagello E. and Kitano H. (Eds), (2000) RoboCup-99 : Robo Soccer World Cup III, Lecture Notes in Computer Science, Vol 1856. Lectures Notes in Artificial Intelligence. Berlin: Springer-Verlag.

4.- Asada M. and Kitano H. (Eds). (1999). RoboCup-98 : Robot Soccer World Cup II Lecture Notes in Computer Science, Vol 1604. Lectures Notes in Artificial Intelligence. Berlin: Springer-Verlag.

5.- Bruce, J., Balch, T. and Veloso, M. (1999). "Fast Color Image Segmentation Using Commodity Hardware". School of Computer Science. Carniege Mellon University.

6.- Sanchez, J.G., Bracho, M., and Moreno, J.A., "Object Recognition an Tracking in a Digital Image", Proceedings of the International Joint Conference IBERAMIA-SBIA'2000, pp.264-273, 2000

7.- Latombe, J. C. (1991). "Robot Motion Planning". Kluwer Academic Publishers, Boston, MA.

8.- Lee S., and Kardaras G., "Elastic String Based Global Path Planning Using Neural Networks", Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '97) June 10-11, 1997 Monterey, CA

9.- Bracho M., and Moreno J.A., (2000). "Heuristic Algorithm for Robot Path Planning Based on Real Space Renormalization", Lecture Notes in Computer Science, Vol 1952. Lectures Notes in Artificial Intelligence. Berlin: Springer-Verlag.

10.- Behring C., Bracho M., Castro M., and Moreno J.A., (2000) "An algorithm for robot path planning with cellular automata", Proceedings of the Fourth International Conference on Cellular Autómata for Research and Industry ACRI'2000 in Theoretical and Practical Issues on Cellular Automata, pp.11-19, Springer-Verlag