Mobile Robot Navigation based on CNN Images Processing – An Experimental Setup

I. GAVRILUT[1], V. TIPONUT[2], A. GACSADI[1]
[1]Electronics Department, University of Oradea, Str. Universității, No. 1, 410087,
[2]Applied Electronics Department, Politehnica University from Timişoara, B-dul Vasile Pârvan, No. 2, 300223,
ROMANIA,
e-mail: gavrilut@uoradea.ro, agacsadi@uoradea.ro, tiponut@etc.utt.ro.

Abstract: – This paper presents results of our work in development of a path-planning algorithm for obstacle avoidance of a mobile robot in a real workspace. The gray-scale images processing of the robot's workspace (global path-planning) is realized by using cellular neural networks (CNNs). Besides that, two IR sensors, mounted in front of the robot were used for fast obstacle avoidance (local path-planning).

Keywords: - mobile robot, cellular neural networks, image processing, path-planning, IR sensors

1 Introduction

Autonomously navigating robots have become increasingly important. Motion planning is one of the important tasks in intelligent control of an autonomous mobile robot. The path-planning problem is generating a collision-free path in an environment with obstacles and optimizing it with respect to some criterion [1],[2].

Mobile robot navigation is divided into three major areas: perception of the environment, path planning and control of the robot. The path-planning component is again divided in two sections: global path planning and local path planning.

Global path planning requires the environment to be completely known and the terrain should be static. In this approach the algorithm generates a complete path from the start point to the destination point before the robot starts its motion. On the other hand, local path planning means that path planning is done while the robot is moving, so the algorithm is capable of producing a new path in response to environmental changes.

The path planning is a complex process starting with the perception of the environment based on maps or images. A central supervisor can do that (global path planning), but in many cases, the corrections of the path, based on sensorial information obtained online through robot's sensors, are required (local path planning). The most frequent sensors for mobile robot are the proximity sensors (laser, IR, sonar) and the visual sensors (video cameras) and

IR sensors are simple, commonly employed, and relatively low-cost sensing modalities to perform the local navigation tasks (especially, fast obstacles avoidance). Sometimes, IR sensors may be preferable to ultrasonic sensors due to their faster response time, narrower beam width, and lower cost. Unfortunately, the intensity of the light detected depends on several parameters including the surface reflectance properties, the distance to the surface, and the relative orientation of the emitter, the detector, and the surface. Due to single intensity readings not providing sufficiently accurate information about an object's position and properties, the recognition capabilities of simple IR sensors have been underused in many applications. Although these devices are inexpensive, practical, and widely available, their use has been mostly limited to detection the presence or absence of objects in the environment (proximity detection) for applications such as obstacle avoidance, counting or wall-following [3].

Though recent research using a camera includes efficient localization methods due to the wealth of information, efficient processing using limited computing power is still not an easy task.

By using cellular neural networks [4],[5] which have very short image processing time a good displacement speed for the mobile robots, can be obtained. The CNN methods have been considered a solution for images processing in autonomous mobile robots guidance [6],[7],[8],[9],[10]. The choice of CNNs is based on the possibility of their hardware implementation in large networks on a single VLSI chip [5],[11],[12].

First, we will present a brief introduction on the two-dimensional cellular neural networks.

1.1 Cellular Neural Networks

A cellular neural network (CNN - Cellular Neural Network) is an analog, nonlinear, dynamic, multi-dimensional circuit having locally recurrent topology. The basic circuit units named cells or artificial neurons are connected only to its neighbor units. The basic cellular neural network [1],[5], has a two-dimensional rectangular structure composed from identical, nonlinear analog circuits (cells) arranged, for example, in M rows and N columns (see Fig. 1).

Due to their locally connections, the field areas occupied on the chip by the connection wire is minimized so that these networks could be implemented in the present VLSI technology, [11], [13]. Cells that are not directly connected together may affect each other indirectly because of the propagation effects of the continuous-time dynamics of cellular neural networks.

A CNN is entirely characterized by a set of nonlinear differential equations associated with the cells in the circuit. The mathematical model for the state equation of the single cell C_{ij} is given by (1).

$$\dot{x} = \frac{dx_{ij}}{dt} = -x_{ij} + \sum_{C_{kl} \in S_r} A_{ij,kl} y_{kl} + \sum_{C_{kl} \in S_r} B_{ij,kl} u_{kl} + z_{ij} \qquad (1)$$

where x_{ij} denotes the state of the cell C_{ij}; y_{kl}, u_{kl} denote the output and input respectively of cells C_{kl} located in the sphere of influence with radius r, Sr, from C_{ij} cell, $C_{kl} \in S_r$; A(ij,kl) and, B(ij,kl) are the feedback and control templates respectively; z_{ij} is the bias term.

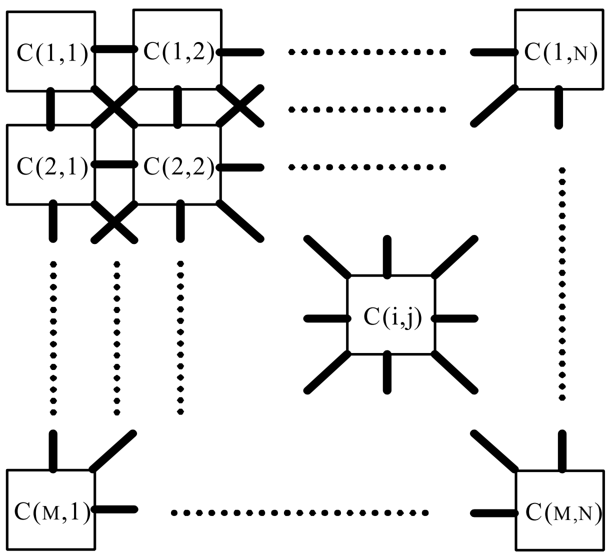

Fig. 1. A two-dimensional cellular neural network with M rows and N columns.

The equation, which expresses the output value of C_{ij} cell, is given in the equation (2):

$$y_{ij} = f(x_{ij}) = \frac{1}{2}\left[\left|x_{ij} + 1\right| - \left|x_{ij} - 1\right|\right] \qquad (2)$$

where y_{ij} denotes output value of C_{ij}.

In Fig. 2 is presented how the two-dimensional signals are processed with a standard cellular neural network having templates of 3×3 dimensions. Applying the image U on the CNN input and having at state an initial image X, the CNN output image Y is obtained by using operators A, B, z, when that equilibrium point is reached.

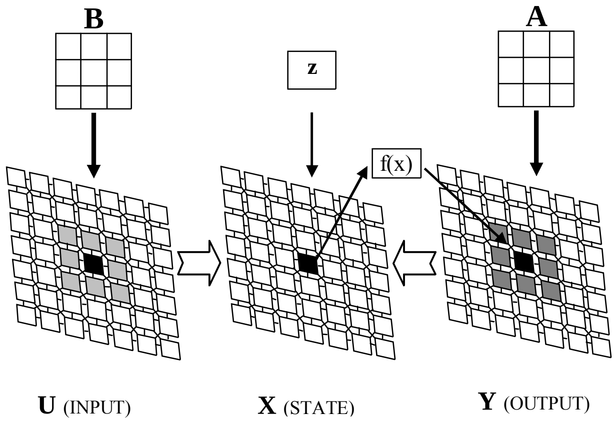

Fig. 2. Signals processing with a standard cellular neural network having templates of 3×3 dimensions.

There is a huge library of templates, available to the whole scientific community, whose content is continuously updated. The difficult task in an implementation of a CNN application is to choose the most appropriate series of templates, corresponding to the processing operations to be performed.

Cellular neural networks are very suited for high-speed parallel signal processing like image or other two-dimensional signals processing. In the same time CNN was used for solving partial differential equations (PDEs). For example, it is presented a numerical solution of a class of PDEs by using emulated digital CNN-UM implemented on FPGAs [14].

Usually, for mobile robot path planning by using CNN, the image of the environment with obstacles must be divided into discrete image and in this way it is possible to represent the workspace through a standard neural network having m×n cells. The processed images have the value of the pixel in the interval [-1, 1], known as the standard CNN domain. For binary images, these values could be only +1 for the black pixels and –1 for the white pixels.

2 The Experiment Presentation

In Fig. 3 the components of the experiment used for trajectory planning and movement control of a mobile robot based on the real workspace images are presented.

The robot has to take the shortest way toward the target avoiding the obstacles located between the initial position and the target position. The PC supervises the whole activity of the robot by images processing of the work place, acquired by a visual sensor (camera). That observes the whole environment and captures images of the workspace at discrete moments. After each image processing operation, the PC will plan, if it is necessary, a new direction for the mobile robot displacement and a control signal will be sent to the robot.

In the same time, if any unexpected obstacle is meeting in the robot path, it is detected by the robot sensors. In this case the robot will take alone the decision for obstacle avoidance.

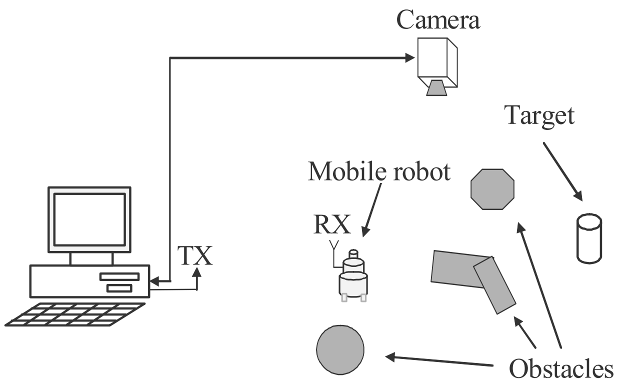

Fig. 3. The components of the experiment.

The flowchart of the whole algorithm used for image-based path planning and control of the mobile robot is presented in Fig. 4.

The color images of the whole workspace are acquired by a camera. In these images will be identified, the mobile robot position and the target position, respectively. Then, these are each represented by one pixel for future processing, even the real dimensions of the robot is bigger (see Fig. 5). In fact, the pixels above mentioned are the symmetrical points in the robot's image and the target's image, respectively.

3 CNN based Image Processing

The obstacles positions from the environment are identified based on the gray-scale image. That was transferred into standard CNN domain, having the value of the pixel in the interval [-1, 1] from white

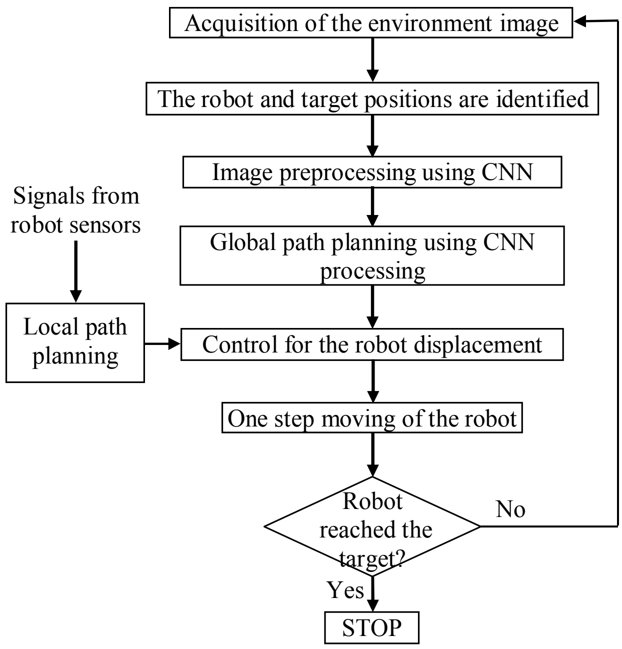

Fig. 4. The flowchart for mobile robot navigation based on CNN Images Processing

to black. In this way the image can be processed with a standard cellular neural network. In our paper the MATCNN toolbox [15], from simulation environment Matlab was used.

If the obstacles from the environment have the luminance more lower than the free space, in the captured image, for their identifying the template TRESHOLD [15], given by (3) can be use:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad z = 0 \qquad (3)$$

On the INPUT and the STATE, respectively, of the cellular neural network the gray-scale image of the environment is applied. After the threshold template applying, the binary image of the environment is obtained on the OUTPUT of the cellular neural network.

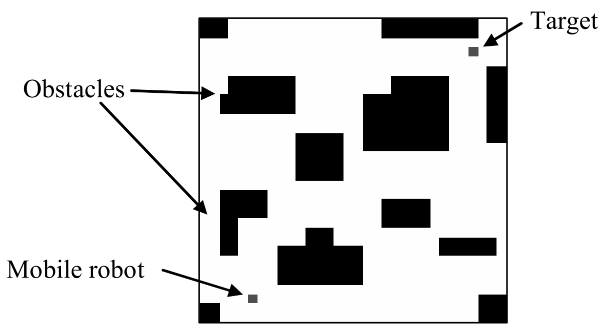

Fig. 5. The binary image of the workspace.

Depending on the illumination conditions, in the acquired image different noises can be found, so that same portions from the free space are interpreted like obstacles. These noises can be removed by applying the template EROSION [15], given by (4).

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad z = -4 \tag{4}$$

After the above processing, the size of obstacles in the image can be affected so that the template DILATION [15] given by the (5) is used.

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad z = 8 \tag{5}$$

The robot and target positions are each identified by a single pixel. In our example, the occupied pixels having values +1 (black) represent the forbidden positions where the robot can't move and the pixels having values –1 (white) represent the free positions accessible for the mobile robot.

4 Path Planning

The robot displacement will be made step by step over free space of the workspace avoiding the obstacles, until the target position is reached. The trajectory must be planned so that the robot no reaches the obstacles and more, that will keeping a fixed distance away from any obstacle.

4.1 Artificial potential field method

For the position estimation between the target and the points from the workspace, an artificial potential field [16],[17] will be created based on the discretized image (m×n resolution). This is composed from an attractive potential field and a repulsive potential field, respectively.

The attractive field is created based on the function $U_{att}(i,j)$ which, in any point from the workspace, has the values given by (6).

$$U_{att}(i,j) = \frac{1}{2} \cdot k \cdot d(i,j) \quad \text{for } \forall \quad i = 1 \ldots m, j = 1 \ldots n \tag{6}$$

where k is a positive scaling factor and d(i,j) represent the Euclidean distance between the target point and the point (i,j) from image.

In this way, for all point from the discretization area which represent the environment is allocated a proportionally value with the distance (number of pixels) between the points and the target point.

If denote the target point with T, having the coordinates (x_t,y_t), the distance between a point (i,j) from an environment up to the target point is given by (7).

$$d(i,j) = \sqrt{(x_t - i)^2 + (y_t - j)^2} \quad \text{for } \forall \quad i = 1 \ldots m, j = 1 \ldots n. \tag{7}$$

The attractive potential has the minimum value where the target point is placed and for other points from the workspace the potential value is proportionally with the distance between the point and the target point.

The repulsive field is created based on the function $U_{rep}(i,j)$ which has the values given by relation (8).

$$U_{rep}(i,j) = \begin{cases} U_{max} & \text{if } i = z \text{ and } j = c \\ n \cdot \left(\dfrac{1}{d((z,c),(i,j))} \right)^2 & \text{if} \\ i \in [z-q, z+q] \text{ or } j \in [c-q, c+q]. \end{cases} \tag{8}$$

where n represent a positive integer number and q is the radius of action of that field around the obstacles positions (z,c).

Finally, based on the total potential field the robot will be "attracted" by the target and in the same time will be "pushing" away from the obstacle (9).

$$U_{att}(i,j) = U_{att}(i,j) + U_{rep}(i,j) \text{ for } i = 1 \ldots m, \; j = 1 \ldots n. \tag{9}$$

Based on that potential field the mobile robot will be coordinates to choose, every time, the optimally direction toward the target which corresponds to the minimum potential around the pixel representing the current position of the robot. The robot movement is on the same direction until the attractive potential is decreased.

4.2 CNN methods

For distance evaluation between the target point and the other free position points in the workspace, a wave is generated in the image plane as can be seen in Fig. 6 [6]. The origin of the source, which generates the wave, is actually the position of the target point. The image for distance evaluation can be achieved using the template EXPLORE, defined by (10)

The template EXPLORE is nonlinear because the parameter a is a nonlinear function, and depends on the difference, between the output value of the C_{ij}

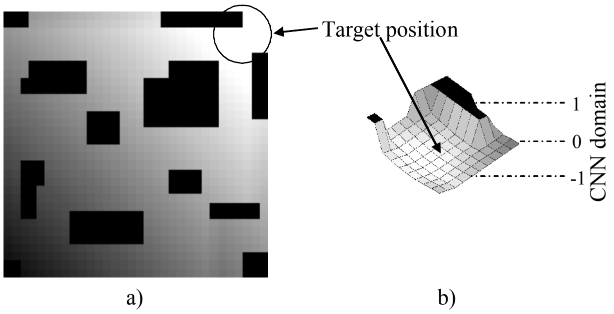

Fig. 6. The principle of determination of the distances through wave propagation: a) the image of the wave moving to the target point b) the values of the pixels located around the target point.

cell and the output value of the C_{kl} cell, situated in her neighborhood (y_{ij}-y_{kl}).

$$A = \begin{pmatrix} 0 & a & 0 \\ a & 1 & a \\ 0 & a & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad z = 0 \qquad (10)$$

The template EXPLORE is nonlinear because the parameter a is a nonlinear function, and depends on the difference, between the output value of the C_{ij} cell and the output value of the C_{kl} cell, situated in her neighborhood (y_{ij}-y_{kl}).

Because of these operations, the value of the pixel corresponding to the target position in the output image remains unchanged at its initial value –1, while the other pixels, which represent the free workspace, will have values proportionally with the distance between their position and the position of the target point.

The mobile robot trajectory is determined by choosing, at each step, the optimal direction, given by the pixel having the minimal value within a 3×3 neighborhood (with r=1). This pixel can be obtained through successive comparisons between all pixels situated in the 3×3 neighborhood of the actual position of the robot [6], or by applying the template PATH [7]. By using EXPLORE there are not two pixels, in the wave image, with the same minimal value in any 3×3 neighborhood.

In case of the successive comparison method, the choice of the optimal direction is realized by extracting the pixel value from a gray-scale image. Using a local method, a neighbor cell from eight possible directions N, S, E, V, SE, NE, NV, SV is chosen in such a way that the output of that cell has the smallest possible value. Practically, the pixel values from the robot neighborhood will be compared in order to choose that pixel which has the minimal value. In this respect, elementary processing AMC [18], are used here, based on the template family SHIFT, corresponding to the eight directions mentioned above (11).

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} se & s & sv \\ e & 0 & v \\ ne & n & nv \end{pmatrix} \qquad (11)$$
$$z = 0$$

where, depending of the considered direction, the only one element from operator B is equal to 1, the other elements having 0 values.

In the case of the second method, by applying the template PATH, a mask image representing the robot neighborhood (r=1) is realized so that the wave obtained with EXPLORE only in the robot neighborhood is selected. After the CNN processing operations on that image, by applying PATH, an image, which indicates the next direction for the robot, is obtained. The template PATH has the form:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & -.1 & 0 \\ -.1 & -8 & -.1 \\ 0 & -.1 & 0 \end{pmatrix} \qquad (12)$$
$$z = 0.3$$

The image processing operations take place as follows. The environment image having the updated value of the wave (Fig. 7a) is applied to the input of the cellular neural network while on the initial state $x(t_0)$ an image having all the pixels at value 0 (Fig. 7b) is applied. An image that has a single black pixel (the others being white) is obtained like output image of the network (Fig. 7c). This pixel indicates the next position (direction) of the robot (Fig. 7d).

After an optimal direction has been determined, the robot movement toward the target can be realized on that direction as long as the current direction allows the robot to move closer to the target or pixel by pixel.

4.2 Determining of the trajectory

The mobile robot trajectory is determined pixel by pixel starting with the pixel which indicates the initial position of the robot (i_R, j_R). The next position will be that pixel which has the minimal value from the neighborhood, with radius r=1, of the current pixel.

If the actual position of the mobile robot is represented by the pixel (i,j), from the processed image, the possible direction of movement are shown in Fig. 8.

The pixels values around the current pixel are representing by a line matrix X and the minimal value is given by the parameter d (13).

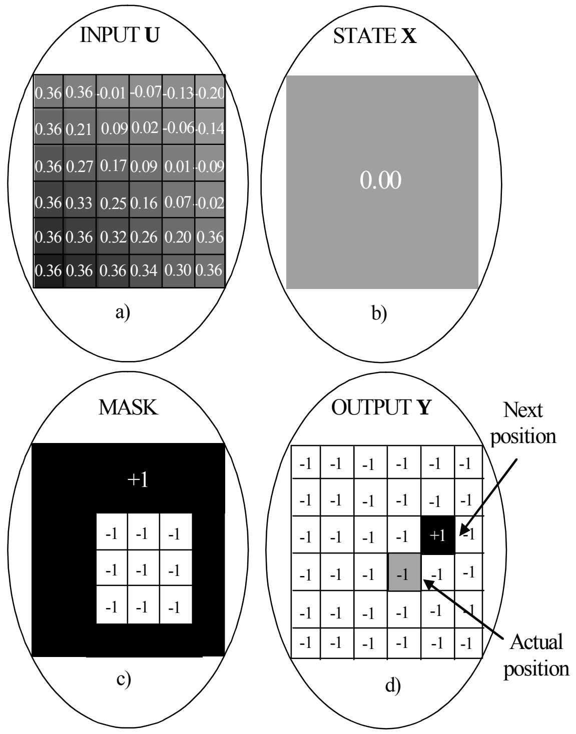

Fig. 7. Determination of the next position for mobile robot; a) the wave image after processing b) the image having all pixels at value 0, c) the mask image, d) the image that indicates the future position of the robot.

$$X = [N \quad S \quad E \quad W \quad NE \quad NW \quad SE \quad SW], \\ d = \min(X). \tag{13}$$

The next pixel of the trajectory will be obtained through comparison of *d* with the matrix elements. That pixel becomes actual pixel and so on, until the pixel representing the target point will be reached.

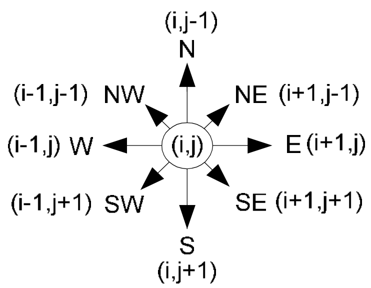

Fig. 8. The possible directions of the robot movement.

5 The Robot Displacement

The robot displacement toward the target along the planned trajectory can be done after the three main steps are completed. These steps are: generating and transmitting of the command toward the locomotion system, robot orientation on the specified direction and the robot movement.

In our experiment, the miniature mobile robot Robby RP5 (see Fig. 9) was used [19].

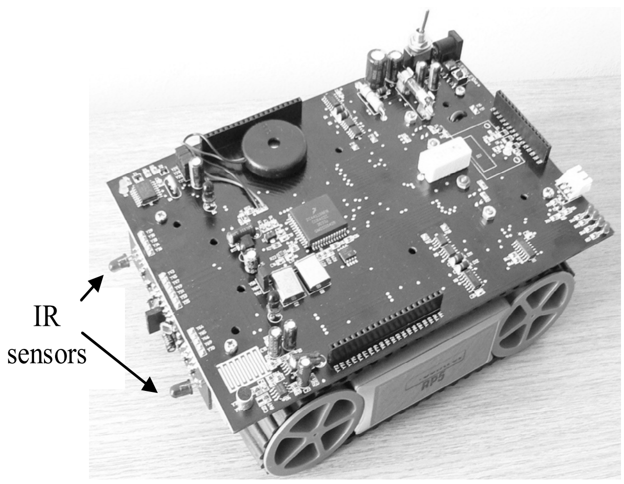

Fig. 9. The mobile robot Robby RP5.

5.1 The mobile robot Robby RP5

The locomotion system of the robot is composed from two symmetrical trays. Both of the DC motors and the spur gear transmissions are integrated therein. The wheel axles and drive shafts are supported in sintered bearings. Two independently controllable electric motors ensure highest mobility of the chassis.

The robot uses the D/A converters, in this case better referred to as PWM outputs, to switch the drive motor voltage, so that, the speed and direction of each track is freely controllable.

The command system of the robot is presented in Fig. 10.

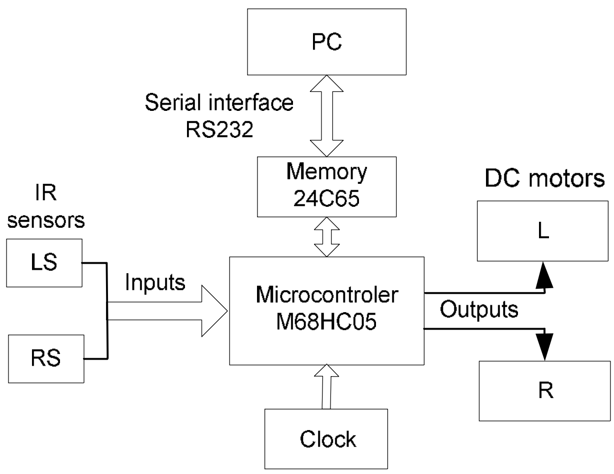

Fig. 10. The command system of Robby RP5.

The microcontroller on the robot is a computer of the C-Control series. This compact unit features universal capabilities for measuring, controlling and steering as well as serial data communication and data storage. The microprocessor allows programming in the well-known BASIC programming language. Through a few lines of BASIC (simplification variant CC-BASIC) source code the computer is able to handle a task like the "brain" of a small autonomous mobile robot.

To communicate with its environment, it has eight analogue inputs, two analogue outputs and sixteen digital port lines randomly usable as in/or outputs.

The signals given by IR sensors (LS and RS) are directed to microcontroller inputs and the signals commands are used for speed control of DC motor drive. In the same time these signals can be used to turn the mobile robot with different turning radius.

There is an Integrated Design Environment (IDE) for the development of application programs for the robot. The IDE is equipped with a standard mouse-controllable graphical user interface with drop-down menus and allows the development of source code (Editor), translating into machine language (Compiler) and uploading the C-Control program to the robot (Loader).

The developed BASIC program, determining the actions and reactions of the robot, will be translated into a sequence of command bytes by the compiler. The commands and the related parameters may then be transferred via serial interface to the microcontroller, and stored into the EEPROM memory (24C65). The interface connection between PC and robot is only necessary while uploading the program. When the robot is programmed (the program was transferred or uploaded into robot memory), it may be disconnected before starting the robot.

5.2 The actions of the mobile robot

The control signals of the DC motors are:

- MR – activation of the right motor;
- ML – activation of the left motor;
- SR – sense for the right motor;
- SL – sense for the right motor.

Based on these signals, the possible actions of the mobile robot are presented in Tab. 1.

As can be seen in Fig. 10 the mobile robot is equipped with two IR sensors, for obstacle detection, each of them composed from an emitter and a receiver. If during driving the robot, obstacles which were not detected by camera or moving obstacles, appear in front of it, they can be detected by infrared sensors. Depending on position of these obstacles, the control system of the robot will prepare orders for the locomotion system in order to avoid them.

MR	**ML**	**SR**	**SL**	**Robot action**
0	1	-	1	Take right with advance
0	1	-	0	Take left with withdrawal
1	0	1	-	Take left with advance
1	0	1	-	Take right with withdrawal
1	1	0	0	Moving back
1	1	0	1	Rotate right
1	1	1	0	Rotate left
1	1	1	1	Moving forward

Tab. 1. The control signals of the robot motors.

The distance of area covered by the IR sensors can be set up on three levels (L1, L2, L3 and R1, R2, R3 respectively) (see Fig. 11). In case of our robot these distances are set up at: 30, 60 and 100 cm, respectively.

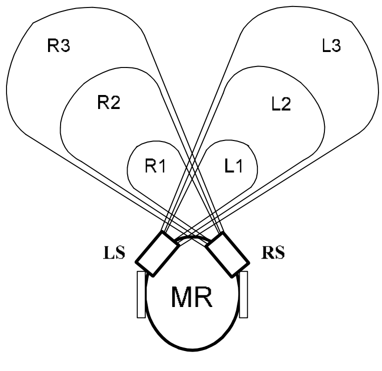

Fig. 11. The positions of IR sensors.

On time of the robot moving, that distances will be alternatively set up, in increasing order. In order to plan the mobile robot actions, the flowchart presented in Fig. 12, is used. If neither sensor detects an obstacle then the robot moving will be at maximum speed. When one (or both) sensor detects one (or more) obstacle, at level L3 and/or R3, then the robot moving will be at medium speed. When one (or both) sensor detects one (or more) obstacle, at level L2 and/or R2, then the robot moving will be

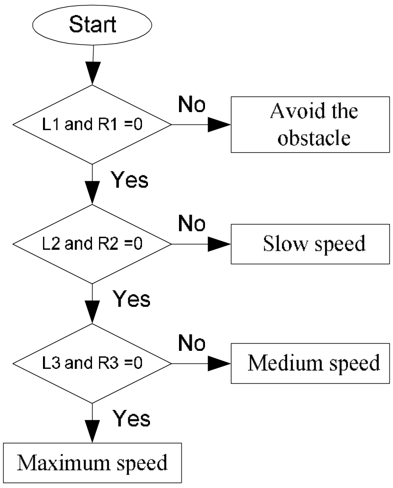

Fig. 12. The actions of the mobile robot.

at slow speed. If one (or both) sensor detects one (or more) obstacle, at level L1 and/or R1, then the robot will turn left or right, depending on the obstacles position and their mission.

In case of global navigation, for control signal transmitting between the PC computer and the mobile robot, the parallel port of the PC can be used.

In this paper, for controlling of the displacement robot, the D connector pins (Data0, Data1, Data2 and Data3) were used. These can be configured from the Matlab environment like the output pins. The communication between the parallel port and the mobile robot can be made by wire, through IR communication or using radio communication.

For local navigation, the mobile robot uses their microcontroller (M68HC05) and the EEPROM memory (24C65) where a program for obstacles avoidance is stored. Of course, the obstacles are detected by the two IR sensors, mounted in front of the robot.

6 Experimental Results

Mobile robot navigation based on images processing was experimentally tested in an indoor environment with static obstacles.

In the first version, a web camera *USB PC Camera 305* mounted on the USB port of a PC computer was used for environment image acquisition. The camera control for establish the image acquisition moments is realized based on a program named VFM (***V****ision* ***F****or* ***M****atlab*) [8],[20],[21]. Practically, that program transfers into simulation environment Matlab the acquired images in form of three matrixes, each of these representing one of the primary color weight (red, green and blue) for every pixel from current image. The images resolution can be modified in five levels from 160×120 pixels up to 640×480 pixels. After their transferring into CNN domain the image processing for obstacles detection was made using the MATCNN toolbox [15].

An image acquired by the video camera, having the resolution 160×120 pixels is presented in Fig. 13a. That represents the gray-scale image of the real environment and was used for system testing. The binary image obtained through CNN processing is shown in Fig. 13b. In that image the obstacles are representing by black pixels and the free space are representing by white pixels. After the EROSION template was applied the image from Fig. 13c is obtained. Finally, by applying the DILATION template, the image used for path planning is shown in Fig. 13d.

Connection between the robot and the PC computer was achieved, in the first phase, by five wire connection. Then, a remote control based on unidirectional radio communication (40 MHz), was made. This type of communication offers a great flexibility so that the robot will not be disturbed by obstacles during its journey compared with the communication by wire connection.

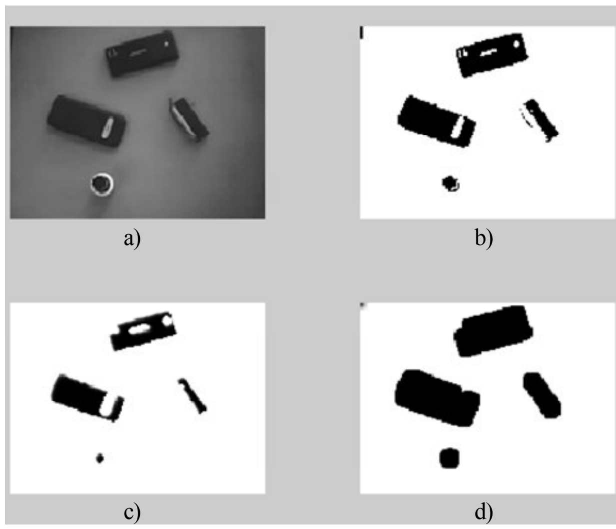

Fig. 13. CNN processing of the real environment image; a) the gray-scale image, b) the binary image obtained by applying the template TRESHOLD, c) applying the template EROSION, d) the finally image after the template DILATION was used.

In the example, above presented, the target (T) was identified situated on the column 130 and line 20, respectively, so that the attractive potential field is presented in Fig. 14.

Based on the images above presented, the entire planned trajectory of the robot is shown in Fig. 15. The initial position of the robot (R) has the

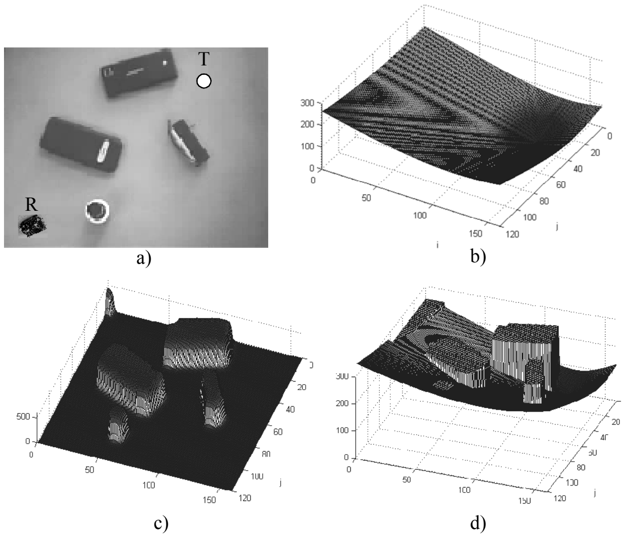

Fig. 14. Example of an artificial attractive potential field; a) image of the environment with obstacles, b) the attractive potential, c) the repulsive potential, d) shape of the total potential.

coordinates (20, 130) and the target point (130, 20), respectively.

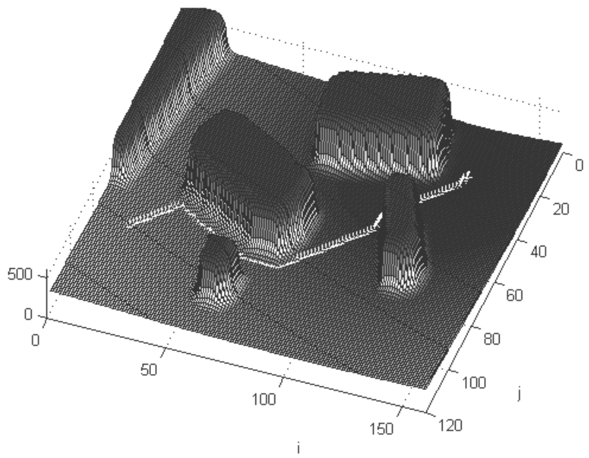

Fig. 15. The planned trajectory of the mobile robot.

In this case, a single image of the environment was captured and based on that the robot can be controlled to reach the target position.

7 Conclusions

The paper presents an experiment for mobile robot navigation based on visual information given by a camera. For images processing the functions from the MATCNN toolbox [15], were used but was necessary, in the same time, some instructions from Matlab.

The total processing time can be reduced if all the signals processing even the signals control are entire realized using only the cellular neural networks (CNN chips). The robot can be recognized after their shape or based on their movement (if that is the single moving object from the environment) by using CNN. On the other hand, the target (if that is fixed) can be identified based on the gray-scale images of the workspace. Starting from these assumptions the camera can be set to acquire, directly, the gray-scale image of the environment.

The path planning based on the artificial potential field method has the disadvantage that the mobile robot can be blocked in local minima if the concave obstacle exists in the environment (having the concavities oriented toward the robot) and these obstacles are placed on the planned trajectory of the mobile robot. This problem can be resolved through CNN processing of the binary environment image which represents the concave obstacles. In this way the concavities can be eliminate and the potential field method can be applied without problems.

Another problem is the light sources position in the workspace. These must be distributed in order to provide a uniform illumination. If the environment illumination is not optimally, some areas from the free space (dark-picture portions) can be identified like obstacles. In the same time, the obstacle shadow can be interpreted like area occupied by the obstacles.

The environment surface is important to complete safety navigation because the slippage of the robot's wheels can be appearing. A good positioning of the robot can be obtained if the odometer sensors are used.

ACKNOWLEDGMENT

This work was supported by a grant from the Romanian National University Research Council, PN II Program ID-668/2009, No.645/19.01.2009.

References:

[1] M. Seda, *Relaxation of Roadmap Methods of Robot Motion Planning*, WSEAS Trans. on Systems, ISSN 1109-2777, Issue 8, Vol. 5, pp. 1808-1813, 2006.

[2] N. Rahman, A. R. Jafri, M. U. Keerio, *Fuzzy Behaviour Based Navigation of a Mobile Robot for Tracking Multiple Targets in an Unstructured Environment*, WSEAS Trans. on Systems, ISSN 1109-2777, Issue 6, Vol. 5, pp. 1425-1490, 2006

[3] R. Clark, A. El-Osery, K. Wedeward and S. Bruder, *A Navigation and Obstacle Avoidance Algorithm for Mobile Robots Operating in*

Unknown, Maze-Type Environments, Proc. Intern. Test and Evaluation Association Workshop on Modeling and Simulation, Las Cruces, NM, December 2004.

[4] L. O. Chua, L. Yang, *Cellular Neural Networks: Theory and Application*, IEEE Trans. On Circuits and Systems, (CAS), Vol. 35, pp. 1257-1290, 1988.

[5] T. Roska, L. O. Chua, *The CNN universal machine: an analogical array computer*, IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 40, pp. 163-173, 1993.

[6] A. Gacsádi, T. Maghiar, V. Tiponuţ, *Path planning for a mobile robot in an environment with obstacles using cellular neural network*, Intern. Workshop on CNN and their Applications, (CNNA 2002), Frankfurt/Main, pp. 188-194, 2002.

[7] I. Gavriluţ, A. Gacsádi, C. Grava, V. Tiponuţ, *Vision based algorithm for path planning of a mobile robot by using cellular neural networks*, Intern. Conf. on Automation, Quality &Testing, Robotics, AQTR 2006, pp. 306-311, Cluj-Napoca, 2006.

[8] I. Gavriluţ, V. Tiponuţ, A. Gacsádi, *An Integrated Environment for Mobile Robot Navigation based on CNN Images Processing*, Systems Theory and Applications, Vol. 2 of the Proc. of the 11-th WSEAS Intern. Multiconf. on CSCC, Agios Nikolaos, Crete Island, Greece, July 23-28, 2007.

[9] V. Tiponuţ, I. Gavriluţ, C. Căleanu, A. Gacsádi, *Development of a Neural Network Guided Mobile Robot Collectivity*, WSEAS Trans. on Circuits and Systems, Issue 6, Vol. 5, pp. 805-812, June 2006.

[10] P. Arena, L. Fortuna, M. Frasca, G. Vagliasindi, M. E. Yalcin, A. Basile, J.A.K. Suykens, *CNN Wave Based Computation for Robot Navigation on ACE16k*, IEEE Intern. Symp. on Circuits and Systems, ISCAS 2005, pp. 5818-5821, 23-26 May, 2005.

[11] G. L. Cembrano, A. Rodríguez-Vázquez, S. Espejo-Meana, R. Domínguez-Castro, *ACE16k: A 128x128 Focal Plane Analog Processor with Digital I/O*. Intern. Journal Neural Systems, 13(6), pp. 427-434, 2003.

[12] P. Arena, A. Basile, L. Fortuna, A. Virzì, *Visual Feedback by Using a CNN Chip Prototype System*, Proc. of the IEEE Intern. Workshop on CNN and their Applications, (CNNA 2002), pp. 291-297, Frankfurt, 2002.

[13] G. Linan, S. Espejo, R. Dominguez Castro, E. Roca, A. Rodriguez-Vazquez, *CNNUM3: A mixed-signal 64x64 CNN Universal Chip*, Proc. Of MicroNeuro99, pp. 61-68, Granada, 1999.

[14] Z. Nagy, P. Szolgay, *Numerical Solution of a Class of PDEs by Using Emulated Digital CNN-UM on FPGAs*, Proc. of the European Conf. on Circuit Theory and Design, Cracow, Poland, September, 2003.

[15] C. Rekeczky, *MATCNN - Analogic Simulation Toolbox for Matlab*, Version 1.0, DNS-11-1997, Technical Report, Analogical and Neural Computing Laboratory, Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest, September, 1997.

[16] J. Borenstein, Y. Koren, *The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robot*s, IEEE Transaction of Robotics and Automation Vol. 7, No 3, pp. 278-288, June, 1991.

[17] Z. Liu, M. H. Ang Jr., W. K. G. Seah, *A Potential field based Approaches for Multi-Robot Tracking of Multiple Moving Targets*, The First Humanoid, Nanotechnology, Information Technology, Communication and Control Environment and Management (HNICEM) Intern. Conf., March 27-30, Manila, Philippines, 2003.

[18] ***, *Cadetwin-99, Extended Analogic Macro Code (AMC) and Interpreter*, Reference Manual, Version 3.0, Analogical and Neural Computing Laboratory. Computer and Automation Institute, Hungarian Academy of Science, Budapest, 1999.

[19] www.c-robotics.com.

[20] http://www.isr.ist.utl.pt/~alex/Resources/vfm.

[21] A. Soinio, *A Lego-robot with camera controlled by Matlab*, LEGO Group Company, http://www.abo.fi/fak/tkf/ rt/robot, 2003.