An Integrated Environment for Mobile Robot Navigation based on CNN Images Processing

I. GAVRILUT[1], V. TIPONUT[2], A. GACSADI[1]
[1]Electronics Department, University of Oradea, Str. Universitatii No. 1, 410087 Oradea,
[2]Applied Electronics Department, POLITEHNICA University Timisoara, B-dul Vasile Parvan No. 2, 300223 Timisoara,
ROMANIA

Abstract: – The paper presents an integrated environment for mobile robot guidance based on images acquired from the robot's workspace. The gray-scale images are processed using cellular neural networks (CNNs), well known for their high speed processing and the capability to be implemented on a VLSI chip.

Keywords: **-** mobile robot, cellular neural networks, path planning, integrated environment**.**

1. Introduction

An important research theme in mobile robotics is the design of the path planning system for a real environment with obstacles. These systems should be able to provide the trajectory for one or more robots, starting from an initial position to the target position, avoiding the obstacles located in the workspace. If we take into account that both obstacles and the target can change dynamically their position and that the obstacles can have any shape, it becomes clear that the above mentioned problem is not a trivial one.

The path planning is a complex process, starting with the perception of the environment based on maps or images. A central supervisor can do this task, but in many cases, the corrections of the path based on sensorial information obtained online through robot's sensors are required. The most frequently used sensors for mobile robot are the visual sensors (video cameras) and proximity sensors (laser, IR, sonar). 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 [1],[11], having a very short image processing time, a good displacement speed for the mobile robots can be obtained. In the last time, CNN methods have been often considered as a solution for images processing in autonomous mobile robots guidance [2],[3],[4],[5],[6],[7],[8],[9],[10].

The choice of CNNs for the visual processing is based on their possibility to be hardware implemented in large networks on a single VLSI chip [11],[12],[13].

Usually, for mobile robot path planning by using CNN, the image of the environment with obstacles must be divided into discrete images (pixels), which can be represent through a standard neural network, having $m \times n$ cells. The gray level, corresponding to each pixel, have values belonging to the interval [-1, 1], known as the standard CNN domain. For binary images, these values can be only +1 for the black pixels and –1 for the white pixels.

2. The structure of the Integrated Environment

In Fig. 1, the structure of the integrated environment used for trajectory planning and movement control of a mobile robot based on the real workspace images is presented.

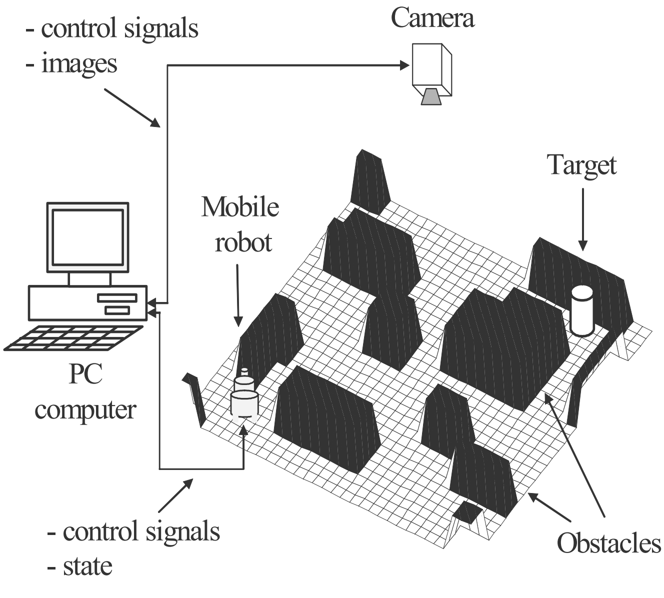


Fig. 1. The structure of the integrated environment.

The robot has to take the shortest way toward the target avoiding the obstacles located between the start and the target position. The personal computer (PC) supervises the whole activity of the robot by processing images of the workspace, acquired by a visual sensor (video camera). It should be noted that the images of the whole environment are captured at discrete moments of time. After each acquired image is processed, the PC will plan a new direction for the mobile robot displacement and a control signal will be sent to the robot, accordingly.

The flowchart of the whole algorithm used for image-based path planning and control of the mobile robot is presented in Fig. 2.

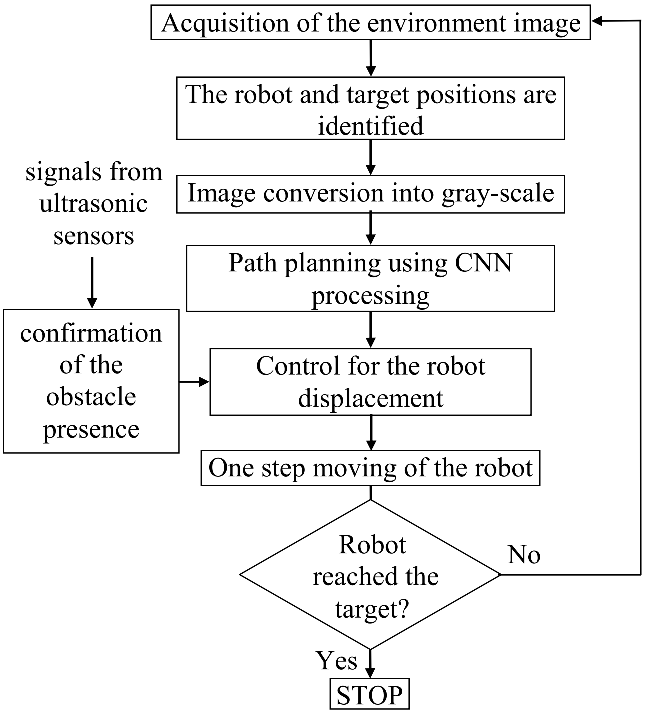


Fig. 2. The flowchart for mobile robot path planning and control process.

The color image of the workspace, acquired by the video camera, is transferred into Matlab environment. It results three matrixes which have the element values proportionally with primary color components (red, green, blue) of each pixel. In these images will be then identified the mobile robot position and the target position, respectively. For the shake of simplicity, both the robot and the target positions will be represented in the next phase of the image processing by only one pixel each, called the central point (of the robot and the target, respectively). Actually, the central point of an object within a binary image represents the pixel located on horizontal and vertical axes, equally spaced from the extreme points of that object on both axes.

In order to simplify the finding task of the robot and target positions, a monochrome light source (LED) can be attached to the up side of the robot and the target, respectively. If the color of the light source (LED) corresponds to one of the fundamental color (red, green or blue), than it is easy to identify the positions of the robot and the target based on the color images given by the camera.

Practically, in this step, the line and the column corresponding to each of these positions are determined. For this purpose, the discrete step (resolution) of images of the workspace has to be adequately chosen.

3. CNN based image processing

The obstacle positions from the environment are identified based on a gray-scale image. This image is transferred into standard CNN domain, having the value of each pixel in the interval [-1, 1], from white to black. In this way the image can be processed with a standard cellular neural network. In our paper the MATCNN toolbox [14], from simulation environment Matlab was used for this purpose.

If the pixels corresponding to the obstacles placed in the image of the workspace have the luminance lower than the pixels representing the free space, then obstacles in the captured image can be identified through a CNN image processing using the template TRESHOLD [14]. This template is given by the relation (1):

$$A=\begin{pmatrix}0&0&0\\0&2&0\\0&0&0\end{pmatrix}\quad B=\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}\quad z=0 \tag{1}$$

The gray-scale image of the environment is applied to the input and state layers of the cellular neural network. By the end of the image processing, using threshold template, the binary image of the environment is obtained at the output layer of the cellular neural network.

Depending on the illumination conditions, the acquired images can include different noises. As a result, some portions from the free space can be interpreted like obstacles. These noises can be removed by applying the template EROSION [14], having the form (2):

$$A=\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}\quad B=\begin{pmatrix}0&1&0\\1&1&1\\0&1&0\end{pmatrix}\quad z=-4 \tag{2}$$

The erosion procedure can affect, in the same time, the obstacles dimensions in the processed images. In order to prevent that, a new image processing is needed, using the template DILATION [14]. This template is given by (3):

$$A=\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}\quad B=\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}\quad z=8 \tag{3}$$

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

4. Path planning

The robot displacement will be made step by step over the free space of the workspace, avoiding obstacles, until the target position is reached. Moreover, the planned trajectory have to keep a fixed distance away from any obstacle.

4.1. Artificial potential field method

In order to estimate the distance between the target and different points from the workspace, the artificial potential field method [15],[16] will be used. We will suppose, in the following, that an attractive potential field and a repulsive potential field will be created over a image having a $m \times n$ resolution.

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 the relation (4).

$$U_{att}(i,j) = \frac{1}{2} \cdot k \cdot d(i,j) \qquad (4)$$
$$for\ \forall \quad i = 1 \ldots m,\ j = 1 \ldots n$$

In the above, k is a positive scaling factor and $d(i, j)$ represents the Euclidean distance between the target point and the point (i, j) from the image.

In this way, for each point in the image representing the environment is allocated a value proportionally with the distance between that point and the target.

If we consider that the target point T has the coordinates (x_t, y_t), then the distance between a point (i, j) up to the target is given by the relation (5).

$$d(i,j) = \sqrt{(x_t - i)^2 + (y_t - j)^2} \qquad (5)$$
$$for\ \forall \quad i = 1 \ldots m,\ j = 1 \ldots n.$$

The attractive potential has the minimum value in the target point while in other points from the workspace the potential value is proportionally with the distance between that points and the target point.

The repulsive field is created based on the function $U_{rep}(i, j)$ which has the values given by the relation (6):

$$U_{rep}(i,j) = \begin{cases} U_{max} \quad if\ i = z\ and\ j = c \\ n \cdot \left(\dfrac{1}{d((z,c),(i,j))} \right)^2 if\ i \in [z-q, z+q]\, or \\ j \in [c-q, c+q]. \end{cases} \qquad (6)$$

Here, 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 "pushed" away from the obstacle (relation 7).

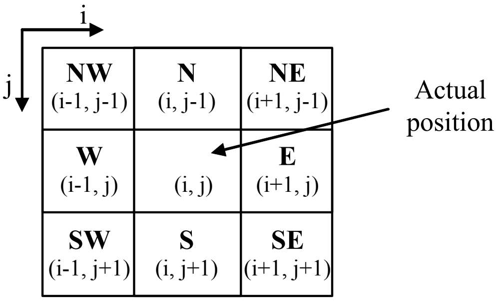
$$U_{att}(i,j) = U_{att}(i,j) + U_{rep}(i,j),\ i = 1..m,\ j = 1..n. \qquad (7)$$

Using the potential field method, the mobile robot will be controlled 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 will keep the same direction until the value of the attractive potential decreases.

4.2. Determination 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 of the robot corresponds to the pixel which has the minimal value of the potential field within a neighborhood with radius $r = 1$ around of the current pixel.

If the actual position of the mobile robot in the processed image is represented by the pixel (i, j), the possible directions of movement are as shown in Fig. 3.



NW (i-1, j-1)	**N** (i, j-1)	**NE** (i+1, j-1)
W (i-1, j)	(i, j)	**E** (i+1, j)
SW (i-1, j+1)	**S** (i, j+1)	**SE** (i+1, j+1)

Fig. 3. The possible movement directions of the robot.

The pixels values around the current pixel are representing by a line matrix X and the minimal value is given by the parameter d (relations 8):

$$X = [N \quad S \quad E \quad V \quad NE \quad NV \quad SE \quad SV],$$
$$d = min(X). \qquad (8)$$

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.

5. The robot displacement

The robot displacement toward the target along the planned trajectory can be done after 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, finally, the robot movement.

5.1. Generating the control signals

The locomotion system (Fig. 4) includes two DC engines MD and MS, which are controlled by *Ud* and *Us* signals (TTL levels). In the front of the chassis a omni directional wheel is mounted. The maximum speed of the robot is 0.92 m/s.

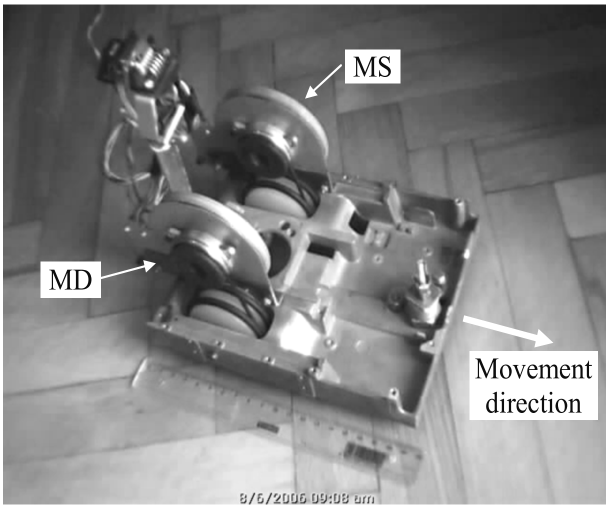


Fig. 4. The locomotion system of the mobile robot.

The movement realized by the robot and the corresponding command signals are as follows:

- moving forward ($Ud = 1$, $Us = 1$);
- moving to the right ($Ud = 0$, $Us = 1$);
- moving to the left ($Ud = 1$, $Us = 0$).

The turn angle depends on the active time of the control signals. Experimentally, the following parameters of the motion have been determined:

- turn at 45 degrees - $t = 0.2$ sec.;
- turn at 90 degrees - $t = 0.4$ sec.;
- turn at 135 degrees - $t = 0.6$ sec.;
- turn at 180 degrees - $t = 0.8$ sec..

It should be mentioned that the locomotion system is open loop controlled, no position sensors are used for this purpose.

5.2. Transmitting the control signals

The control signal is transmitted to the mobile robot through the parallel port of the PC. In the same time, this port can receive data regarding the position of obstacles, acquired by sensors that equip the robot.

In the present experiment, Data0 and Data1 lines of the parallel port of the PC are used in order to control the robot movement. These lines can be configured from the Matlab environment as output pins. Data0 pin outputs the signal *Ud* and Data1 pin gives the signal *Us*, respectively.

The communication between the parallel port and the mobile robot can be done in different way: wired transfer or wireless, using IR or radio communication.

5.3. The robot orientation

In this experiment the mobile robot receives a control signal which includes the parameters *Ud*, *Us*, *t*. If the actual orientation of the robot is different from the orientation provided by the received command, the robot will be turn on the specified direction.

For example, if the robot initial orientation is SE, the possible orientations are E, NE, N (turn left) and S, SW, W, NW (turn right) like in Fig. 5.

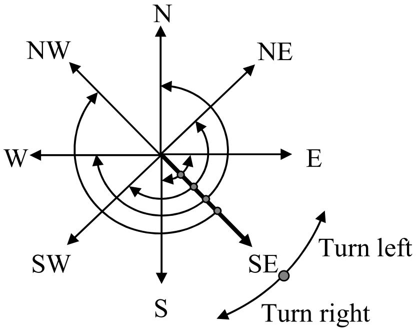


Fig. 5. The possible orientations when actual orientation of the robot is SE.

After the robot orientation is completed the robot will move on the planned trajectory ($Ud = 1$, $Us = 1$). The time interval for this command is proportionally with the number of pixels on the same unchanged direction toward the target.

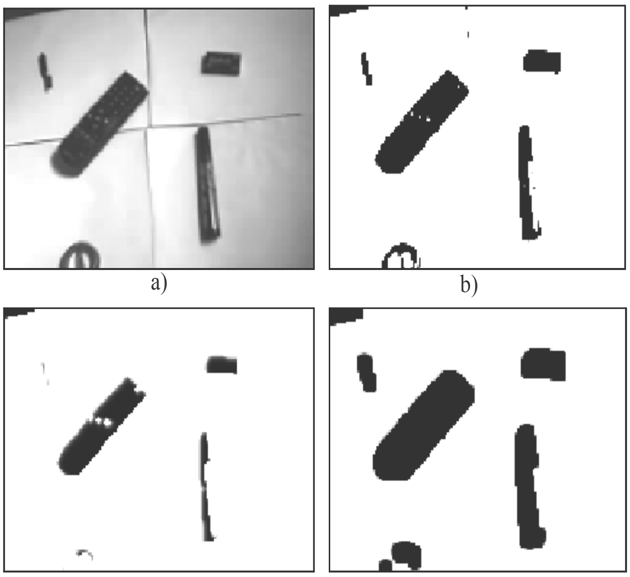
5. Experimental results

The components of the integrated environment have been experimentally tested. In the first version, a web camera *USB PC Camera 305* connected to the USB port of a Pentium IV computer was used for image of the environment acquisition. The behavior of the video camera and the moments of image acquisition are controlled using VFM software application (***V**ision **F**or **M**atlab*) [17],[18].

Practically, this program transfers into Matlab simulation environment the acquired images in form of three matrixes, each of these representing one of the primary color weight (red, green and blue) for each pixel from the current image. The resolution of the acquired images can be modified in five levels, from 160×120 pixels up to 640×480 pixels. These images are then converted into the CNN domain in order to be processed using the MATCNN toolbox [14].

In the following it will be presented, briefly, the CNN based image processing for obstacles detection.

An image acquired by the video camera and having the resolution 160×120 pixels is presented in Fig. 6a. The picture 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. 6b. In this figure, black pixels represent obstacles and the free space is represented by white pixels. After a CNN processing, using EROSION template, results the image given in Fig. 6c. Finally, we obtain the image used for path planning (Fig. 6d), by applying the DILATION template.



a) b) c) d)

Fig. 6. 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.

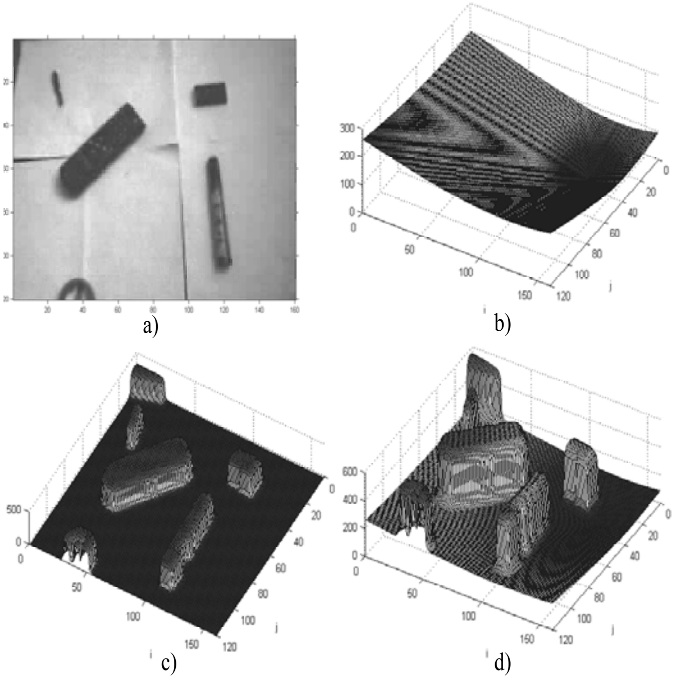
In the example presented above, the target was identified situated on the column 140 and line 40, respectively, so that the attractive potential field is presented in Fig. 7.

As a result of the images presented in Fig. 7, the entire planned trajectory of the robot can be determined as is shown in Fig. 8. The initial position of the robot has the coordinates (15, 100) and the target point (130, 40), respectively.

In this case, having only static obstacles, a single image of the environment was captured and processed in order to control the robot to reach the desired target position.

6. Conclusions

The paper presents an integrated environment for mobile robot navigation based on visual information given by a video camera. The acquired images have been processed using functions from the MATCNN toolbox [14], and some instructions from Matlab.



a) b) c) d)

Fig. 7. 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.

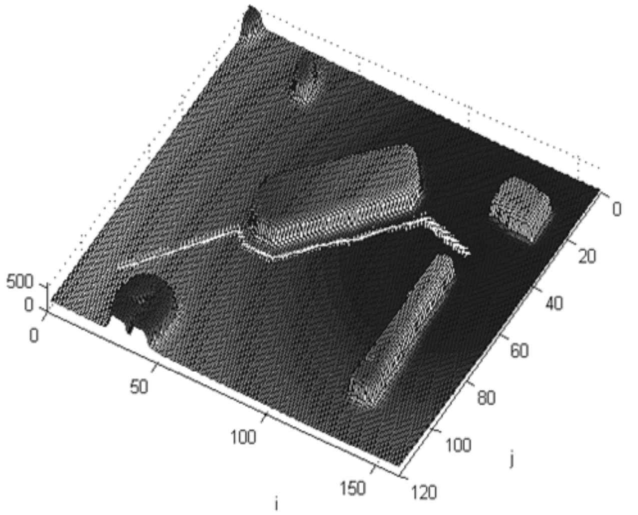


Fig. 8. The planned trajectory of the mobile robot.

The total processing time can be reduced if all images and even the control signals are entirely processed using cellular neural networks (CNN chips). The robot can be recognized after his shape or based on his movement, using CNN procedures (we suppose that the robot is the only moving object in the whole workspace). 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 concave obstacles are present in the environment (having the concavities oriented toward the robot) and these obstacles are placed on the planned trajectory of the robot. The solution to this problem is to eliminate the concavities from the acquired images, using CNN based procedures. After that, the potential field method can be applied without any difficult.

Another problem we should pay attention is the light sources position in the workspace. For the best results, the light sources must be distributed over the whole workspace 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 obstacles.

The environment surface is important in order to achieve a safety navigation because of the slippage of the robot's wheels. A good positioning of the robot can be obtained if robot is provided with odometers.

References:

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

[2] 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.

[3] I. Gavriluţ, A. Gacsádi, L. Ţepelea, V. Tiponuţ, *Motion planning for two mobile robots in an environment with obstacles by using cellular neural networks*, The 7-th Intern. Symposium on Signals, Circuits and Systems, (ISSCS 2005), pp. 801-804, Iaşi, 2005.

[4] 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.

[5] I. Gavriluţ, V. Tiponut, A. Gacsádi, *Path Planning of Mobile Robots by Using Cellular Neural Networks*, Proceedings of the 10-th IEEE International Workshop on CNN, CNNA 2006, pp. 108-113, Istanbul, Turkey, 28-30 August, 2006.

[6] 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, Volume 5, pp. 805-812, Cavtat, Croaţia, 2006.

[7] B. Siemiatkowska, A. Dubrawski, *Cellular Neural Networks for Navigation of a Mobile Robot*, L. Polkowski and A. Skowron (Eds.): RSCTC'98, LNAI 1424, pp. 147-154, Springer-Verlag Berlin Heidelberg, 1998.

[8] D. L. Vilarino, C. Rekeczky, *Shortest path problem with pixels level snakes: Application to robot path planning*, Proc. of the IEEE Intern. Workshop on CNN and their Applications, (CNNA 2004), pp. 135-140, Budapest, 2004.

[9] A. Adamatzky, P. Arena, A. Basile, R. Carmona-Galan, B. De Lacy Costello, L. Fortuna, M. Frasca and A. Rodriguez-Vazquez, *Reaction-Diffusion Navigation Robot Control: From Chemical to VLSI Analogic Processors*, IEEE Trans. on Circuits and Systems – I: Regular Papers, Vol. 51, No. 5, May, 2004.

[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. Symposium on Circuits and Systems, ISCAS 2005, pp. 5818-5821, 23-26 May, 2005.

[11] 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.

[12] 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.

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

[14] 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.

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

[16] Z. Liu, M. H. Ang Jr., W. K. G. Seah, *A Potential field based Approaches for Multi-Robot Tracking of Multiple Moving Targets*, HNICEM Intern. Conf., March 27-30, Manila, Philippines, 2003.

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

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