CNN Processing Techniques for Image-Based Path Planning of a Mobile Robot

I. GAVRILUȚ, L. ȚEPELEA, A. GACSÁDI
Electronics and Telecommunications Department
University of Oradea
Universitatii Str., 1, Oradea, 410087
ROMANIA
{agacsadi, ltepelea, gavrilut}@uoradea.ro

Abstract: - The paper presents methods for images processing in order to path planning of a mobile robot in an environment with obstacles. The CNNs (Cellular Neural Networks) methods have been considered a good solution for image processing in autonomous mobile robots guidance. The choice of CNNs for the visual processing part is based on the possibility of their hardware implementation in large networks on a single VLSI chip and their capability to make parallel processing.

Key-Words: - cellular neural networks, image processing, path-planning, mobile robot

1 Introduction

An important research topic in mobile robotics is path planning. There are various forms of robot planning, like: path and motion planning, perception planning, navigation planning, and manipulation planning [1]. The path planning problem is to find a feasible geometric path in some environment for moving a mobile robot from a starting position to a goal position. A geometric model of the environment with the obstacles and the free space is supposed to be given. A path is feasible if it meets the kinematics constraints of the mobile robot and if it avoids collision with obstacles.

The path planning is a complex process starting with the perception of the environment based on maps or sensory information. The most frequent sensors for mobile robot are the visual sensors such as cameras using CCD arrays, and range sensors such as laser, IR, or sonar.

By using camera, a lot of information is obtained, which must be processed in the shortest time possible. Using cellular neural networks [2],[3], which have very short image processing time, a good displacement speed for mobile robots, can be obtained. The CNN methods have been considered a solution for images processing in mobile robotics [4],[5],[6].

This paper proposes an image based path planning method by using cellular neural networks. The outline of the paper is as follows: first, Section 2 briefly reviews about the standard cellular neural network. Then, Section 3 describes the path planning algorithm based on images. Section 4 presents simulation results and lastly, Section 5 presents some conclusions.

2 Cellular Neural Networks

A 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 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). The links between the cells indicate that there are interactions between the neighboring cells.

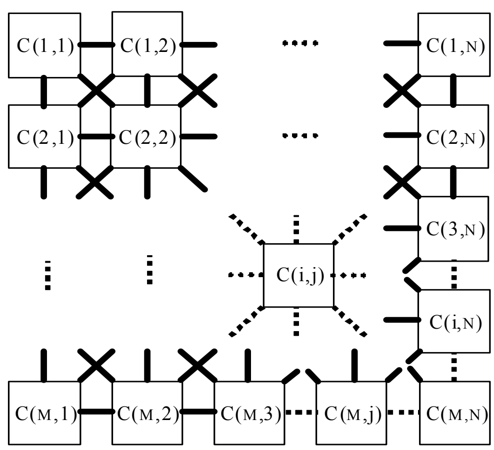

Fig. 1. A basic two-dimensional cellular neural network.

Due to their locally connections, the occupied areas on the chip by the connection wire is minimized so that these networks could be implemented in the present VLSI technology [8]. 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(i,j) is given by the following set of relations:

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

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, S_r, 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.

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

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

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. 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 equilibrium point is reached.

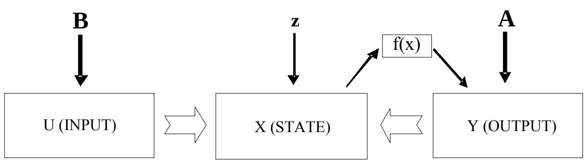

Fig. 2. Signals processing with a standard cellular neural network.

A variety of approaches have been proposed to use CNNs for mobile robots path planning based on images in unstructured environments [7],[9]. Usually, the environment with obstacles must be divided into discrete images and in this way it is possible to represent the workspace in the form of an M×N array, through a standard neural network having M×N cells. The processed images are gray-scale, having 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.

3 Path planning by using CNN

In the current applications, the mobile robot is considered to be placed in a plane workspace, where there are only static obstacles. The robot has to take the shortest way toward the target avoiding the obstacles located between the Start and the Target positions.

The gray-scale images of the environment with obstacles are acquired, using a video camera, then transferred to the cellular neural network or CNN chip, respectively. After CNN elementary preprocessing, the binary image of the workspace is obtained and will be used in the CNN algorithm (see Fig. 3).

After a spatial discretization of the image, which corresponds to the CNN resolution, we suppose that each obstacle is represented by at least one pixel having a fixed value. The robot and target positions are each identified by a single pixel. In our example, the occupied pixels having values +1 represent the forbidden positions where the robot can’t move and the pixels having values –1 represent the free positions accessible for the mobile robot (see Fig. 7).

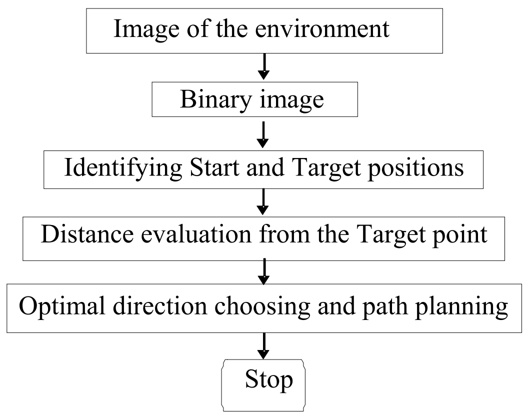

Fig. 3. Flowchart for mobile robot path planning by using cellular neural networks.

3.1 CNN based image pre-processing

The obstacles positions from the environment are identified based on the gray-scale image. If the obstacles from the environment have the luminance more lower than the free space, in the captured image, for their identifying the TRESHOLD template [10], 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.

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 EROSION template [10], given by (4).

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

The obstacles dimensions, in the image, can be affected so that the DILATION template [10] given by (5) is used.

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

3.2 Distance evaluation

For optimal trajectory obtaining between Start and Target positions, the distances between the free points from the workspace, where the mobile robot can be along the algorithm, and the target point must be determined.

In this respect, a wave is generated in the image plane having the source origin situated in the target point. The image for distance evaluation between two different positions in the workspace, one of them being the target position, could be achieved using the EXPLORE template [10], defined by (6).

$$A = \begin{pmatrix} 0 & a & 0 \\ a & 1 & a \\ 0 & a & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad z = 0 \qquad (6)$$

The template mentioned above is nonlinear because parameter "a" is nonlinear function (Fig. 4), and depends on the difference y_{ij}-y_{kl}.

Through its propagation the wave searches all the possible directions in the environment, starting from the target point. As a result 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 pixels having the value +1 are going to be the forbidden positions through which the robot cannot pass. All the other pixels will have values that proportionally increase with the distance between them and the target position (see Fig. 5).

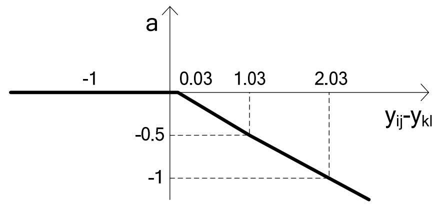

Fig. 4. Transfer characteristic for a.

Thus, starting from the center of the wave source, the value of pixels is increasing approximately with a distance measure unit, when the wave radius is increasing by 1. If the environment is without obstacles the wave has a circular propagation.

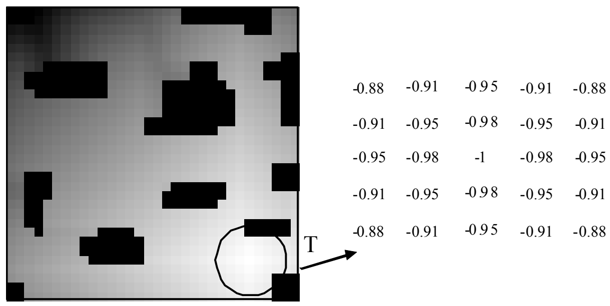

Fig. 5. Evaluation of the distance in between the target and the free points from the workspace.

3.3 Template for optimal direction choosing

Having indicated the robot position by one black pixel, their neighborhood with radius r = 1 can be easily obtained using the DILATION template, above mentioned.

After a same logical operation on that image, a mask image is obtained which will be overlapped on the wave image. The resulting image represents the wave (Fig. 6a) only in the robot neighborhood (Fig 6b). Further, it must be chosen a pixel, which has the minimal value, in this example the pixel value is 0.63.

For that, it can be design a template, which applied on image shown in Fig. 6b, will be indicate the future position of the robot, but in another configuration this it's not good for that. The bias value must be the same, changed according with the new configuration, where the robot is situated.

From this reason, in our algorithm the same processing operations will be made on the image presented in Fig. 6a, so that by applying the same

ISBN: 978-1-61804-023-7

template on the image from Fig. 6b it will be obtained an image which indicates next position of robot.

Through CNN processing an image having all pixels at minimal value (in our example 0,63) is created. If that image is subtracted from the image presented in Fig. 6a, an image having pixel values around the robot position (Fig. 6c) or in robot neighborhood (Fig. 6d) is obtained.

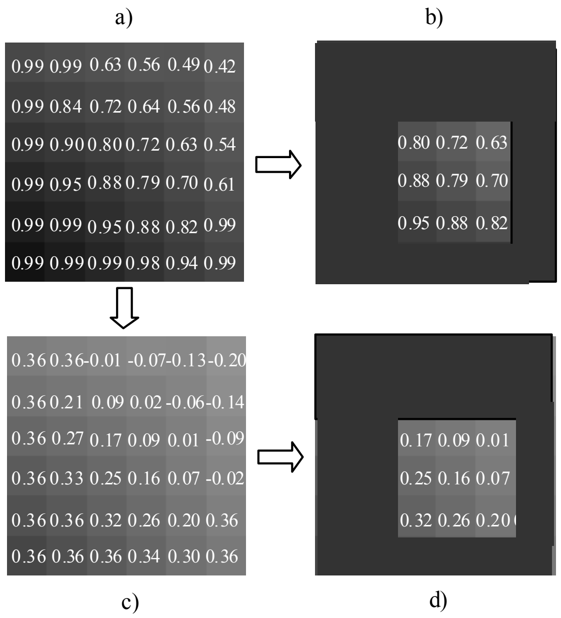

Fig. 6. The resulting image represent the wave (pixels values): a),b) around the current position on the path after applying EXPLORE (0,79); c),d) after the minimal value (0,63) from b), was subtracted.

The resulting image has approximately same values around the robot current position, respectively of robot position in its trajectory toward target. So, having an adequate template applied through a mask image on the image like in Fig. 6d, an image with the future position is obtained, for all position from free space where the robot can be situated on the its trajectory.

The template form can be like in (7):

$$A=\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}\quad B=\begin{pmatrix}0&b_1&0\\b_1&b_2&b_1\\0&b_1&0\end{pmatrix}\quad z=c \tag{7}$$

By applying this template, named PATH, on the image from Fig. 6b, the future value of every pixel depends on the current value of pixel as well as on the pixels values from its neighborhood.

The state equation corresponding to the PATH template has the following form:

$$\dot{x}_{ij}(t)=\sum_{C_{kl}\in N_1} b_{ij,kl}u_{kl}+c \tag{8}$$

From the relations (7) and (8) the particularized state equation is obtained:

$$\begin{aligned}\dot{x}_{ij}(t)&=b_1\cdot u_{i-1,j}+b_1\cdot u_{i+1,j}+b_1\cdot u_{i,j-1}+\\&+b_1\cdot u_{i,j+1}+b_2\cdot u_{ij}+c\end{aligned} \tag{9}$$

The binary image, representing the robot neighborhood with radius r=1 of the current position is used like a mask image for that processing step. In this way only the pixels around the robot position will be modified by applying this template. The output image of the network must contain a single black pixel that indicates the future position of the robot.

By applying relation (9) to all the pixels from the robot neighborhood, the template elements for PATH must satisfy an inequality system. For the pixel, that represents the future position, the inequality is given by the relation (10) and for the other pixels by the relation (11).

$$b_1\left(u_{i-1,j}+u_{i+1,j}+u_{i,j-1}+u_{i,j+1}\right)+b_2\cdot u_{ij}+c>0 \tag{10}$$

$$b_1\left(u_{i-1,j}+u_{i+1,j}+u_{i,j-1}+u_{i,j+1}\right)+b_2\cdot u_{ij}+c<0 \tag{11}$$

If that inequalities are satisfied, the integrator which is in composition of every cell, will determine a low level at the cell output $y_{ij}(\infty)=-1$, for all neighbor pixels and a high level $y_{ij}(\infty)=+1$ in case of the pixel which indicates the next robot position. Of course, that pixel has every time the lowest value in comparison with all the other pixels from the robot neighborhood in the current position.

After solving the inequalities (10), (11) the parameters values: b_1, b_2 and c, are obtained and the PATH template becomes: (12)

$$A=\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}\quad B=\begin{pmatrix}0&-0.1&0\\-0.1&-8&-0.1\\0&-0.1&0\end{pmatrix}\quad z=0.3$$

4. Simulation results

The experimental results of the path-planning algorithm were obtained using the simulation programs written in assembling language (Extended Analogic Macro Code and Interpreter). In our simulation we used 32×32 images but the algorithm is very easy to be extended to 64×64 pixels or more.

Fig. 7 shows an example of binary image used for simulation of the CNN path planning algorithm and in Fig. 8, an example of full path computing result, for the environment's image (32*32 pixels) (from Fig. 7) is presented.

5. Conclusions

The processing time is relative short by applying the PATH, for choosing of next position, compared with other processing methods CNN, such as applying the SHIFT template.

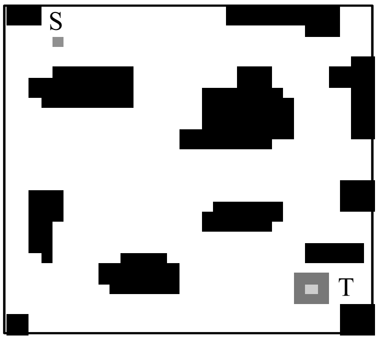

Fig. 7. The binary image of the workspace with obstacles. The Start (S) and Target (T) positions are indicated.

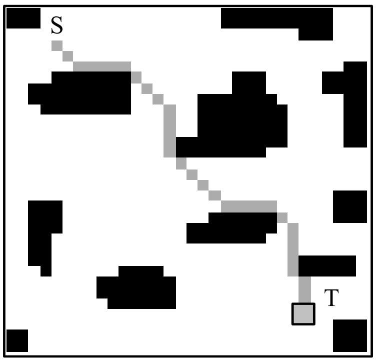

Fig. 8. An example of full path computing result.

The processing time of the step in which the evaluating of distance between free points from the free workspace and the target is made, depends on the distance between the start and the target position because is necessary that wave reaches, through propagation, the start position. If the wave front no reaches the target pixel then the image resolution can be modified.

One of the advantages in case of proposed method is that the wave front is propagated only until when the robot position is reached, so that the propagation time of the wave is minimized. This is very important when the robot is relative close to target.

An important condition for the proposed algorithm to run correctly is that at each captured image, the Start and Target point must be identified and then each of them represented by a pixel.

ACKNOWLEDGMENT

This work was supported by CNCSIS – UEFISCSU, project number PNII – IDEI 668/2008.

References:

[1] M. Ghallab, D. Nau, and P. Traverson, *Automated planning, theory and practice*, Elsevier, Morgan Kauffman, 2004.

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

[3] T. Roska and 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.

[4] V. Kilic, R. Yeniceri and Mustak E. Yalcin, *A New Active Wave Computing Based Real Time Mobile Robot Navigation Algorithm for Dynamic Environment*, 12th Int. Workshop on Cellular Nanoscale Networks and their Applications (CNNA), 2010.

[5] A. Gacsádi, T. Maghiar, V. Tiponuț, *Path planning for a mobile robot in an environment with obstacles using cellular neural network*, Int. Workshop on CNN and their Applications, Frankfurt/Main, Germany, pp. 188-194, 2002.

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

[7] I. Gavriluț, V. Tiponut, A. Gacsádi, *An Integrated Environment for Mobile Robot Navigation based on CNN Images*, Proc. of the WSEAS Int. Conf. on Systems, Crete Island, Greece, pp. 81-86, 2007.

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

[9] H. Kim, H. Song, T. Roska,L.O. Chua, *Optimal Path Finding with Space and Timevariant, Metric Weights via Multi-layer CNN*, Int. Journal of Circuit Theory and Applications, (IJCTA), Vol. 30, pp. 247-270, 2002.

[10] ***, *Software Library for Cellular Wave Computing Engines, Version 3.1*, Cellular Sensory and Wave Computing Lab. of the Comp. and Aut. Research Inst., Hungarian Academy of Sciences and the Jedlik Lab. of the Pázmány P. Catholic Univ., Budapest, 2010.