Multi-robot Coordination by using Cellular Neural Networks

I. Gavrilut, A. Gacsadi,

Electronics Department,
University of Oradea, Str. Universitatii, No. 1, 410087, Oradea, Romania
gavrilut@uoradea.ro

Abstract—**Vision-based algorithms for multi-robot coordination, are presented in this paper. Cellular Neural Networks (CNNs) processing techniques are used for real time motion planning of the robots. The CNN methods are considered an advantageous solution for image processing in autonomous mobile robots guidance.**

I. INTRODUCTION

The advances in technology have facilitated the use of mobile robots in order to carry out a large variety of tasks Recently, the interest has shifted from the design and deployment of few, rather complex and expensive robots towards the design and use of a larger number of robots, which are simple, relatively inexpensive, but capable together of performing complex activities [1].

Multi-robot systems can provide several advantages over single-robot systems: robustness, flexibility and efficiency among others. To benefit from these potential aspects the robots must cooperate to carry out a common mission. It is well known that several problems have to be solved to archive this aim. Some tasks can be carried out by a single robot but if two or more robots can cooperate, the task can be archive faster.

Tasks used in the study of multi-robot control include foraging, which involves searching and retrieving items from a given area, box-pushing, which moves an object between two locations, and formation marching, where robots move while maintaining a fixed pattern.

Much of the literature on distributed control algorithms for autonomous mobile robots has concentrated on two basic tasks, called gathering and convergence [2]. Gathering requires the robots to occupy a single point within finite time, regardless of their initial configuration. Convergence is the closely related task in which the robots are required to converge to a single point, rather than reach it.

In [3] four type of formations are considered: line, column, diamond and wedge. The task for each robot is to simultaneously move to a goal location, avoiding the obstacles, avoiding collision with the other robots and maintaining a formation position. Each robot computes its proper position in the formation, relative to the locations of the other robots.

Three techniques, for robot position determination within the formation, have been identified relative to the formation center, to the leader and to a neighbor.

Before the robot moving the path planning must be completed. A central supervisor can do this but in many cases, the corrections of the path, based on sensorial information obtained online, are required. The most frequently used sensors for mobile robots are the visual sensors such as cameras and range sensors such as laser, IR, or 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.

Cellular neural networks [4],[5] which have very short image processing time have been considered an advantageous solution for images processing in autonomous mobile robots guidance, [6],[7],[8]. The choice of CNNs is based on the possibility of their hardware implementation on a single VLSI chip [9].

II. CELLULAR NEURAL NETWORKS

A cellular neural network (CNN - ***C****ellular* ***N****eural* ***N****etwork* [4]) 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 [4],[5] has a two-dimensional rectangular structure composed from identical, nonlinear analog circuits (cells), is presented in 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 [5]. 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:

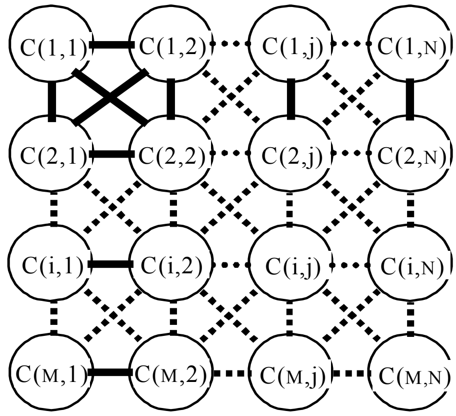

Figure 1. A basic two-dimensional cellular neural network with M rows and N columns.

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

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; $A_{ij,kl}$ 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 in the relation (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 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.

III. GATHERING METHOD FOR MOBILE ROBOTS BY USING CNN

We are considering a system of autonomous mobile robots that are able to freely move in a two-dimensional work space. The robots must be gathering so that the sum of the trajectories will be minimum. For example, if a robot has a central position this robot will keep its location and the other robots will move toward it. There are also situations when two or more robots can be considered as having a central position.
In the algorithm, which will be presented here, the environment image is considered discretized with the same resolution as CNN. Every robot is represented by an image containing a single black pixel, which indicates the robot position.

A. Distance between robots evaluation

Distance evaluation between two robots can be realized by determining the neighborhood of a robot with the minimum radius, which includes the pixel representing the other robot.

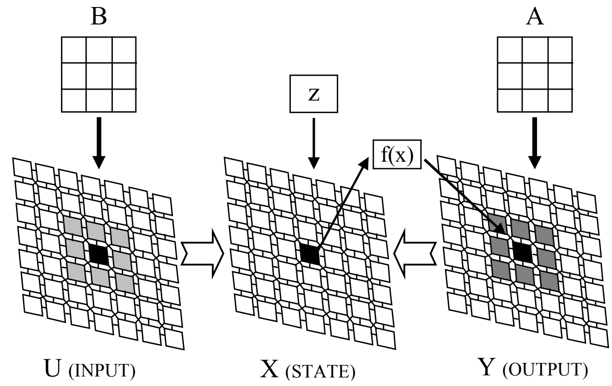

Figure 2. Signals processing with a standard cellular neural network having templates of 3×3 dimensions.

For n mobile robots, each having the position coordinates in the image given by (i, j), i. e. ($R1(i_1, j_1)$, $R2(i_2, j_2)$ … $Rn(i_n, j_n)$), the neighborhoods with radius r for a robot Rm are given by the following relations:

$$V_{Rm,r}(i_m, j_m) = \{C(k_m, l_m) \mid \max\{|k_m - i_m|, |l_m - j_m|\} \le r\},\ m \in [1, n] \quad (3)$$

If we suppose that the distance between the robot R1 and the other robots will be determined first, a neighborhood around this robot, with a radius $r = 1$ will be generated. Then, the radius is increased to $r = r + 1$ and so on, until all pixels representing the other robots will be overlapped ($r = r_1$).
The distance (pixels number) between the robot Rp and the robot Rm is given then by the relation:

$$\begin{aligned} &\text{if } (i_m, j_m) \subset V_{Rp,r}(i_p, j_p) \text{ and } (i_m, j_m) \not\in V_{Rp,r-1}(i_p, j_p) \\ &\text{then} \quad r_{pm} = r \quad \text{for } \forall\, m, p \in [1, n],\ m \ne p \text{ and } r \in [1, r_p] \end{aligned} \quad (4)$$

The algorithm was simplified taking into account the reciprocity propriety:

$$r_{pm} = r_{mp} \quad \text{for } \forall\ p \in [1, n] \text{ and } m \in [1, n],\ p \ne m. \quad (5)$$

The neighborhood of each pixel can be obtained by using the template DILATION, given by the following relations [10]:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad z = 8 \quad (6)$$

For example, in the case of four mobile robots (R1, R2, R3, R4), the neighborhoods of R1, which successively overlap the pixels corresponding to the other robots, are shown in Fig. 3. In a similar way, the neighborhoods of R2, R3 and R4 can be obtained.
The distance determination procedure can be done in a prescribed order or in an increasing order.

1) Distances evaluation in a prescribed order

The neighborhood with radius $r = 1$ of the robot R1 is generated first (see Fig. 4). In the next step, the overlapping with the pixel representing R2 is verified.

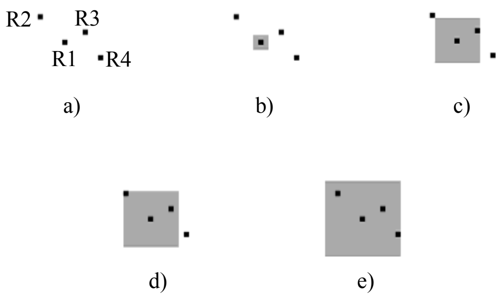

Figure 3. The R1 neigborhoods; a) the robots positions; b) r =1; c) r =4 (R3 is overlap); d) r =5 (R3 and R2 are overlap); e) r =7 (R3, R2 and R4 are overlap).

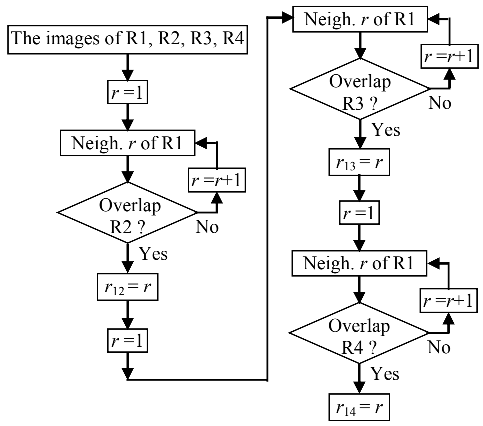

Figure 4. The flowchart for distance evaluation in a prescribed order.

Two situations are to be considered:

- if $(i_2, j_2) \subset V_{R1,1}(i_1, j_1)$ then $r_{12} = r$;
- if $(i_2, j_2) \not\subset V_{R1,1}(i_1, j_1)$ then the radius r will be incremented ($r = r + 1$) and the process will be repeated.

Finally, r_{12} has the ultimate value assigned for r. Further on, the parameters r_{13} and r_{14} are determined in the same manner.

Taking into account the relation (5), for robots R2, R3 and R4 will be determined only the parameters: r_{23} r_{24} and r_{34}.

2) Distance evaluation in an increasing order

The distance evaluation in an increasing order between R1 and the other robots can be done like in Fig. 5.

In the first step, the neighborhood of the R1 robot is generated, then the overlapping with one or more pixels, representing the other robots (R2, R3 or R4), are verified.

Depending of the possible situations, the values for r_{12}, r_{13} or r_{14} can be determined as follows:

- if $(i_2, j_2) \subset V_{R1,1}(i_1, j_1)$ then $r_{12} = 1$;
- if $(i_3, j_3) \subset V_{R1,1}(i_1, j_1)$ then $r_{13} = 1$;
- if $(i_4, j_4) \subset V_{R1,1}(i_1, j_1)$ then $r_{14} = 1$.

If not, the radius r will be incremented and a neighborhood having radius $r = 2$ is generated.

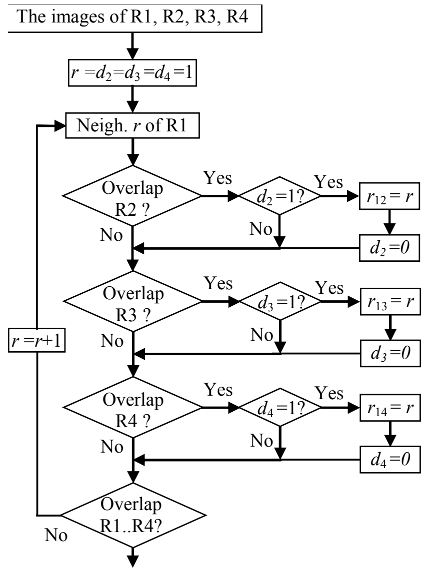

Figure 5. The flowchart for distance evaluation in an increasing order.

The process will be repeating until all the parameters: r_{12}, r_{13} and r_{14} are determined. The algorithm will be continued for the robots R2, R3 and R4, taking into account the relation (5).

B. Gathering of the robots

The proposed method includes, as a first phase, the determinations of the distances between all robots of the collectivity. Then the values N_1, N_2, N_3 and N_4, which indicate how much a robot is close to the other robots are determined, according to the relation (7):

$$\begin{aligned} N_1 &= r_{12} + r_{13} + r_{14} \\ N_2 &= r_{21} + r_{23} + r_{24} \\ N_3 &= r_{31} + r_{32} + r_{34} \\ N_4 &= r_{41} + r_{42} + r_{43} \end{aligned} \tag{7}$$

By comparison of them can be determined the robot or a group of robots which have a central position comparatively with the rest of the robots in the collectivity.

The following possible situations have to be considered in our example:

- there is no robot with a central position ($N_1=N_2=N_3=N_4$);
- a single robot (e.g. R1) has a central position ($N_1<\min(N_2, N_3, N_4)$);
- two robots (e.g. R1 and R2) have a central position ($N_1=N_2<\min(N_3, N_4)$);
- three robots (e.g. R1, R2 and R3) have a central position ($N_1=N_2=N_3<N_4$).

Based upon the same $N_1 \ldots N_4$ values, already determined, the algorithm can be extended for the case of determination of a robot or a group of robots having extreme positions. These robots must move toward the other robots.

After the robots having a central position are determined, the other robots from the collectivity have to move toward to these robots so that the total length of trajectory tr_{total}, given by relation (8) to be minimum.

$$tr_{total} = tr_{R1} + tr_{R2} + tr_{R3} + tr_{R4} \tag{8}$$

For example, if robot R1 has a central position and robot R2 has an extreme position, R2 will move toward R1 until it enters into the R1's neighborhood with radius r =1 (see Fig. 6).

IV. Testing the Algorithm

The gathering algorithm for mobile robots, using CNN, has been tested using the simulation environment CadetWin [10]. Images having a resolution of 32×32 pixels have been used in all simulations.

In the case of the algorithm used for determination of the robots having a central position, the necessary total time was measured for three variants.

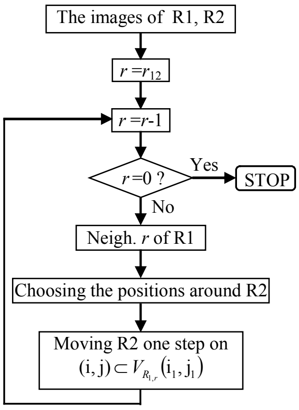

Figure 6. The flowchart for R2 displacement toward R1.

An example of determination of the robots having central and extreme positions respectively, based on the algorithm described above, is presented in Fig. 7. A collectivity of four mobile robots is considered to be arranged like in Fig. 7a.

V. Conclusions

A necessary condition for the proposed algorithms to run correctly is that the robots positions to be clearly identified in the workspace images.

The necessary total time for central and extreme robots determination, depending by the distance evaluation method. In the case of the algorithm used for determination of the robots having a central position, the necessary total time was measured for three variants.

Taking into account the reciprocity propriety, given by (5), the processing time has been reduced up to 26%. If the distance between robots was determined in an increasing order, the time was decreased with 43%.

It should be noted that in the proposed algorithms, the exact geometric position of the robots is not necessary.

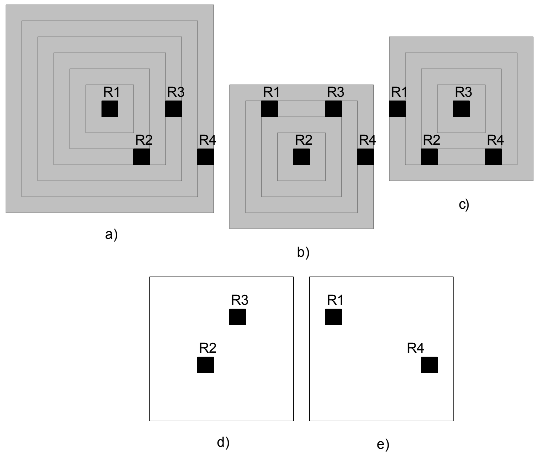

Figure 7. Determining of the robot positions; a) the neighborhoods of R1, b) the neighborhoods of R2, c) the neighborhoods of R3, d) the robots having a central position, e) the robots having an extreme position.

References

[1] Donald D. Dudenhoeffer, Michael P. Jones, "A formation behavior for large-scale micro-robot force deployment", Proc. of the 2000 Winter Simulation Conf., J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds., pp. 972-982, 2000.

[2] R. Cohen, D. Peleg, "Convergence Properties of the Gravitational Algorithm in Asynchronous Robot Systems", Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot, 76100, Israel, march, 2004.

[3] T. Balch, Ronald C. Arkin, "Behavior-Based Formation Control for Multirobot Teams", IEEE Trans. on Robotics and Automation, Vol. 14, No. 5, dec. 1998.

[4] 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.

[5] 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.

[6] I. Gavriluț, A. Gacsádi, L. Țepelea, V. Tiponuț, "Motion planning for two mobile robots in an environment with obstacles by using cellular neural networks", The 7-th Int. Symposium on Signals, Circuits and Systems, (ISSCS 2005), pp. 801-804, Iaşi, 2005.

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

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

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

[10] ***, CadetWin-99, "CNN application development environment and toolkit under Windows", Version 3.0, Analogical and Neural Computing Laboratory, Computer and Automation Institute, Hungarian Academy of Science, Budapest, 1999.