COORDINATING THE MOTION OF MOBILE ROBOTS USING CELLULAR NEURAL NETWORK

Received 14th June 2007; accepted 5th September 2007.

Barbara Siemiątkowska

Abstract:

This paper presents a method for motion planning for a group of mobile robots. The goal of the group is to move through an environment and to reach a destination while maintaining the desired formation. The map of the environment is represented as a grid of cells. A state of each cell is determined. It can be free, occupied by the obstacle, occupied by a robot. The trajectories of the robots are planned using the modification of diffusion method. The algorithm is implemented using Cellular Neural Network. This kind of implementation allows of efficient path planning and to solve conflicts between robots. Computer simulations were preformed in order to proof the efficiency of the approach.

Keywords: *neural networks, mobile robot, mapping, navigation, multi-robots systems, cellular neural network*

1. Introduction

Multi-robots systems have become very popular. It is motivated by development of communication and computation. There are complex tasks that are difficult or inappropriate for a single robot to deal with, but a team of cooperating robots could accomplish them easily. Examples include automated transport system, formation flight control, and exploration of an environment. Arkin [2] divided cooperation tasks into following groups of applications:

1. Grazing, in this kind of tasks an environment should be swept by robots' sensors. Vacuuming and search-and-rescue operations are examples of grazing problems.
2. Consuming, the tasks for the robot is to find and carry objects placed in their environment.
3. Traffic control.
4. Moving in a formation, where the robots form some geometric pattern and maintain it. The formation flight control is the example of the application.

When the group of mobile robots operates in a limited environment avoiding collisions and blockages is the biggest problem, which should be solved, so an efficient path-planning algorithm is the essential part of all multi-robots systems.

Path planning and the pattern formation methods are divided into two groups: centralized and decentralized approach.

In the centralized approach [1][6][7] the central unit plans the paths for the robots. Based on global information the commands are sent to the vehicles. It is assumed that there is a communication channel between the robots and the central unit. The cost of the system increases with the number of robots and it is not robust to failures.

An alternative method is to use decentralized path planning strategies [4][3][10][11]. In this approach each robot generates its path based on local information. Usually gradient methods are used in order to control a robot [5][12][15]. This approach is less costly and more robust that centralized one but it suffers from local minima problems and low precision of pattern formation. This paper considers the problem of path planning for a team of robots. It is assumed that the robots share the same goal. Two kinds of behaviours have been implemented: robots are moving trough the goal independently or in desired formation. The method described in this paper is a decoupled approach to coordinated path planning. It applies the modification of diffusion method presented in [14]. The same method is used for global path planning, pattern formation and for solving collisions avoidance problems. No additional rule based system is required to solve conflicts between robots. The method is implemented using Cellular Neural Network.

2. Cellular Neural Network

The cellular Neural Network was introduced by Chua [8] in 1988. CNN is a single-layer network defined on regular lattices. The dynamics is described by a system of $N \times M$ differential equations:

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{k,l \in N_{i,j}} a_{kl}^{ij} f(x_{kl}(t)) + \sum_{k,l \in N_{i,j}} b_{kl}^{ij} f(u_{kl}(t)) + I \quad (1)$$

where x_{ij} denotes the state of a cell c_{ij} and $ij \in N \times M$. N_{ij} denotes the neighbourhood of a cell c_{ij}, a_{kl}^{ij} is an interconnection weight between cells c_{kl} and c_{ij}, b_{kl}^{ij} is the feed forward template parameter, u_{ij} - an input signal, and I is bias term, f is a linear saturation function. The dynamic of discrete-time CNN is described by equation:

$$x_{ij}(t) = \sum_{k,l \in N_{i,j}} a_{kl}^{ij} f(x_{kl}(t)) + \sum_{k,l \in N_{i,j}} b_{kl}^{ij} f(u_{kl}(t)) + I \quad (2)$$

Chua extended the definition of CNN in 1997 [9]. It is assumed that it consists of cells that interact locally. This type of CNN can be view as a generalization of cellular automata. The neurons can be modelled as locally connected finite states machines. The state of a cell depends on the states of the neighbouring cells, values of input signals, and values of templates. The neurons are usually arranged in rectangular network. Each neuron communicates with nearest neighbours. CNNs are widely used for image processing and patterns recognition but it can be used for path planning.

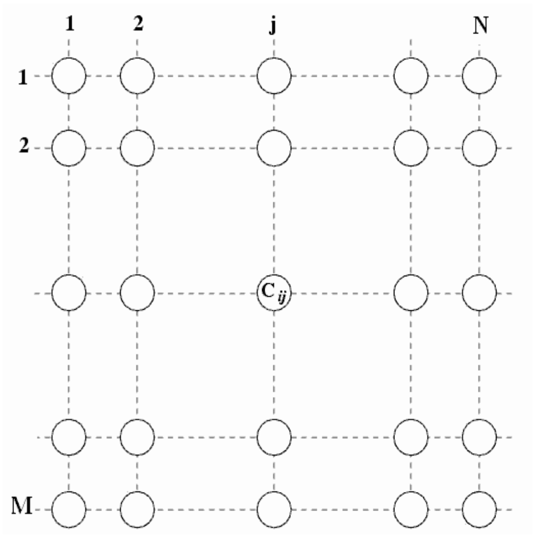


Fig. 1. Cellular Neural Network.

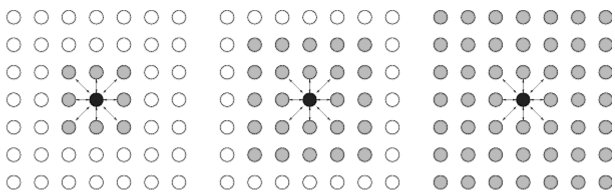


Fig. 2. The neighbourhood of the cell.

3. Path planning for a single robot

CNN, which allows path planning for a single robot, can be described as follows:

The robot's environment is divided into $N \times M$ rectangular areas. The map of an environment is represented as a 2D $N \times M$ CNN, each cell represents the corresponding area of the environment. The algorithm consists of following stages:

3.1. Initialisation

Weights of connection between corresponding cells are computed using following formulae:

$$a_{kl}^{ij} = dist\,(p_{ij}, p_{kl}) \tag{3}$$

where $dist\,(p_{ij}, p_{kl})$ is the distance between areas p_{ij} and p_{kl}, which are represented by cells c_{ij} and c_{kl}.

Initial values of CNN's cells equal:

$$x_{ij}(0) = \begin{cases} G >> 0 & if \quad cell\ is\ the\ goal \\ 0 & in \quad other\ \ cases \end{cases} \tag{4}$$

The initial values of inputs signals u_{ij} are computed as follows:

$$u_{ij}(t) = \begin{cases} 1 & if \quad the\ cell\ is\ free \\ 0 & if \quad the\ cell\ is\ occupied \end{cases} \tag{5}$$

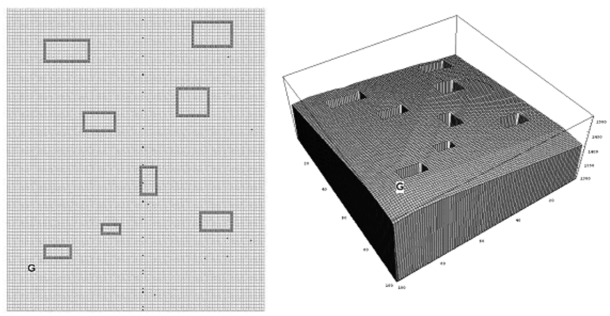
3.2. Diffusion process

$$x_{ij}(t+1) = \begin{cases} u_{ij}(t) \cdot max_{kl \in N_{ij}} (x_{kl}(t) - a_{ij}^{kl}) & if\ \ x_{ij}(t) < G \\ u_{ij}(t) \cdot G & if\ \ x_{ij}(t) = G \end{cases} \tag{6}$$

where N_{ij} is neighbourhood of the cell c_{ij}, a_{ij}^{kl} is the weight between c_{ij}, and c_{kl}. If an environment is static the process is continued until stability, that until:

$$\forall ij \ \ x_{ij}(t) = x_{ij}(t+1) \tag{7}$$

In a dynamic environment the diffusion process is performed continually. If c_{ij} corresponds to the current position of the robot, then the motion direction is indicated by that of neighbouring cells which value x_{ij} is the largest.

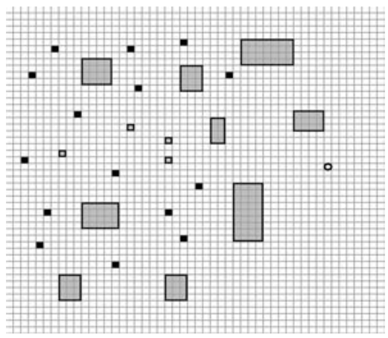


(a) (b)

Fig. 3. The environment and its diffusion map.

Figures 3a) – 3b) present the idea of path planning. Figure 3a) presents the occupancy grid based map of an environment. Black squares represent obstacles; G represents the position of the goal. Figure 3b) presents the result of diffusion algorithm. Coordinate z presents the value x_{ij}. The value of the cell, which cannot be traversed, equals 0.

(a)



(b)

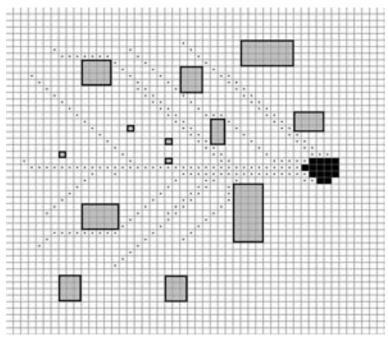


Fig. 4. Path planning for a single goal.

4. Path planning for a team of robots

Values stored in the diffusion array depend on an environment and the position of the goal. They do not depend on positions of the robot. That means that if the goal is common for the team of robots, each robot can generate its own collision free path based on the same, common diffusion array. In order to avoid collision between robots each cell receives additional input signal $v_{ij}(t)$.

$$v_{ij}(t) = \begin{cases} G & \text{the cell is the goal} \\ 0 & \text{another cases} \end{cases} \quad (8)$$

and equation (5) is replaced by:

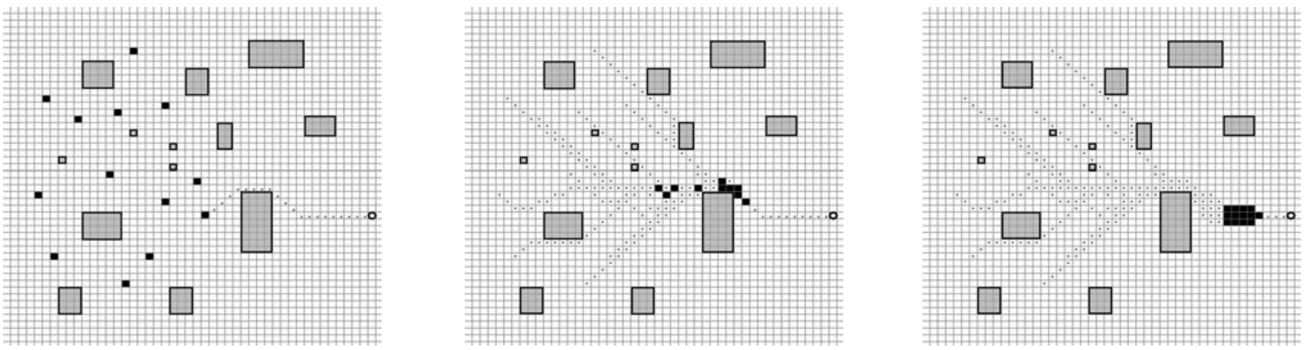
$$u_{ij}(t) = \begin{cases} 1 & if \quad \text{the cell is free} \\ 0 & if \quad \text{the cell is occupied by an obstacle} \\ -1 & if \quad \text{the cell is occupied by a robot} \end{cases} \quad (8)$$

$$x_{ij}(t+1) = u_{ij}(t) \cdot max(v_{ij}(t), max_{kl\in N_{ij}}(|x_{kl}(t)| - a_{ij}^{kl})) \quad (10)$$

When the cell *in* is occupied by the robot the corresponding value u_{ij} is temporally set to -1, this cause the change of diffusion array and this cell cannot be accessed by another vehicle. When the robot leaves the cell c_{ij}. the original value of u_{ij} is restored then the corresponding cell becomes traversable again.

Figure 4a) presents the idea of path planning method for robots that have been assigned to get the same target. Obstacles are presented as a grey squares. The black circle presents the position of the goal. The dotted lines presents the paths planned for each robot independently, but based on the common diffusion array. Figure 4a) presents the initial positions of the robots (black squares). Figure 4b) presents the final positions of the robots. The robots surround the goal. This configuration is obtained using CNN and no additional rule based system is required. The main disadvantage of value iteration methods is that the generated path is not smooth. In the paper [13] the method, which allows of computing optimal controls for the robot, based on a diffusion array is presented.

4.1. Swarm of robots



a) b) c)

Fig. 5. Stages of the swarm forming.

The next task of the robot is to reach the goal and to form a swarm. The task is divided into following stages:

- The path to the goal is planned for each robot, the robot which path to the goal is the shortest one, starts to be a leader of a team. The width of a path is also taken into account. CNN based methods that allow path planning for different constrains are presented in [16].
- The leader follows the planned path.

The cell placed after the leader is the goal for another robots. The equation (8) is replaced by the equation (11).

$$v_{ij}(t) = \begin{cases} G >> 0 & if \ c_{ij} \text{ is behind the leader} \\ 0 & \text{in other cases} \end{cases} \quad (11)$$

The equation (9) is replaced by the equation (12).

$$u_{ij}(t) = \begin{cases} 1 & if \quad \text{the cell is free} \\ 0 & if \quad \text{the c is occupied or ahead of the leader} \\ -1 & if \quad \text{the cell is occupied by a robot} \end{cases} \quad (12)$$

Figures 5a) - 5c) present the stages of path execution by a swarm of robots. Figure 5a) presents the initial positions of the robots (black squares) and the position of the goal (black circle). Dotted line presents the path planned for the leader. Figures 5b), 5c) present stages of the swarm forming. The final locations of the robots are different than in the case when robots are moving to the goal independently.

4.2. Moving goal

If the task for the robot is to surround the moving target the equation (11) is replaced by the equation (13)

$$v_{ij}(t) = \begin{cases} G >> 0 & if \quad c_{ij} \in N_r(goal) \\ 0 & in \quad \text{other cases} \end{cases} \quad (13)$$

The equation (13) is applied for a new position of the goal. In order to force the robots to surround the goal equations (12) is replaced by:

$$u_{ij}(t) = \begin{cases} 1 & \text{the cell is free} \\ 0 & \text{the } c_{ij} \text{ is occupied by an obstacle or by a robot} \end{cases} \quad (14)$$

5. Moving in formation

The next task of the robot is to move trough the goal in desired formation.

The task is divided into two stages:

- The path to the goal is planned; the robot which path to the goal is the shortest one starts to be a leader of a team.
- The leader follows the planned path.
- Based on the position of the robot and the desired formation the set S of cells that robots should occupy is computed. The distance $dist_{ij}$ between the leader and c_{ij} is computed for each cell of the set S. The set of values $dist_{ij}$ is static and depends on the formation not the position of robots.

(a) (b) (c) (d)

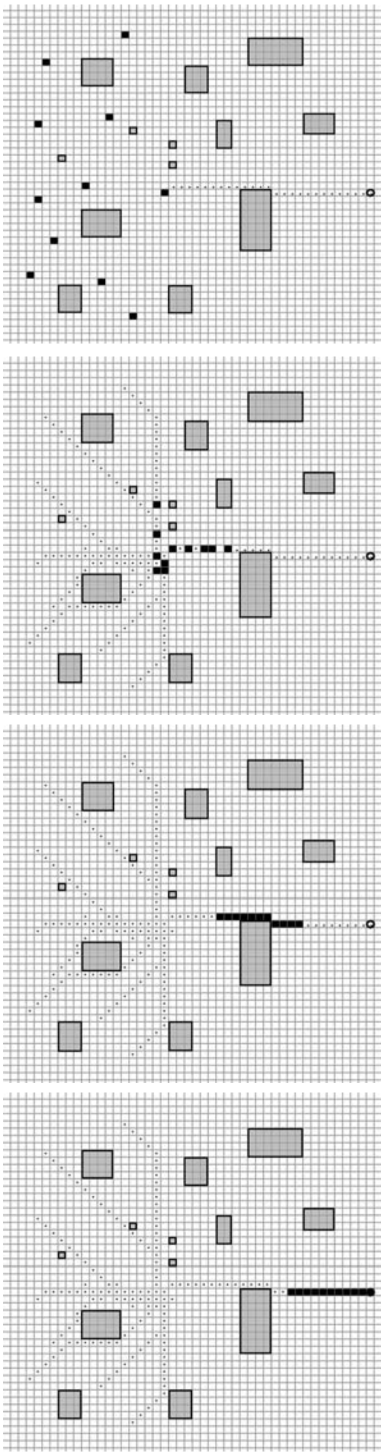


Fig. 6. Chain formation

The equation (13) is replaced by the equation (15)

$$v_{ij}(t) = \begin{cases} G - \varepsilon \cdot dist_{ij} & if \quad c_{ij} \in S \\ 0 & in \quad other \ cases \end{cases} \tag{15}$$

ε is a scaling factor. It is assumed that:

$$\forall c_{ij} \in S \quad \varepsilon \cdot dist_{ij} << a_{ij}^{kl} \tag{16}$$

Figures 6a) - 6d) present the stages of creating chain formation. Figure 6a) presents the initial locations of the robots. Dotted rectangles present the path of the leader. Figure 6b) - 6c) present intermediate stages of the team. Figure 6d) present the result of the team formation.

(a) (b) (c) (d)

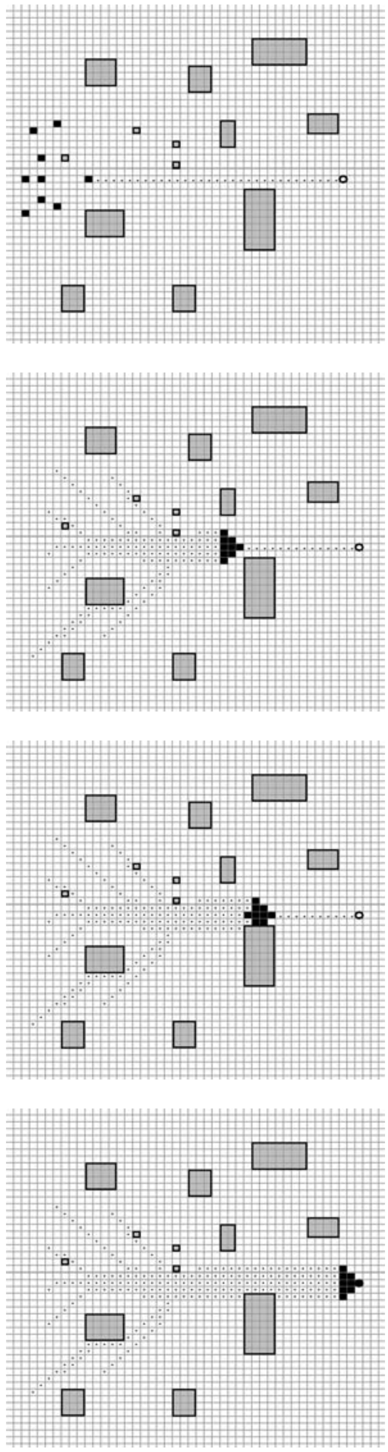


Fig. 7. Triangular formation.

Figures 7a) - 7d) present the stages of creating triangular formation. Figure 7a) presents the initial locations of the robots and the goal. Dotted rectangles present the path of the leader. Figure 7b) - 7c) present intermediate stages. If the desired formation is impossible to be kept (figure 7c) the robots change theirs configuration automatically. Figure 7d) presents the final locations of the robots.

The algorithm has been tested for different kinds of formations: triangle, diamond and extended line. The method does not fail in the case if one of the robot stops working.

6. Conclusion

In this paper, a modified diffusion method is used for designing cooperation between robots. The method is implemented using CNN. It is shown that for different values of initial states of neurons and different values of input signals different behaviours of team of robots are obtained. Using CNN it is possible to attain desired geometric formation of robots and to simulate hunting behaviours. CNN based path planning method does not generate the controls for the robots, but the controls can be computed based on method presented in [13].

Computer simulations have been performed in order to proof efficiency of the method. Experiments with the use of real robots will be performed in a future.

AUTHOR

Barbara Siemiątkowska - PhD works for Warsaw University of Technology, Faculty of Mechatronics, Św. Andrzeja Boboli 8 street, 02-525 Warsaw, Poland and Institute of Fundamental Technological Research, Polish Academy of Sciences, 00-049 Świętokrzyska str., Warsaw, Poland; e-mail: bsiem@ippt.gov.pl.

References

[1] T. Arai and J. Ota, "Motion Planning of multiple mobile robots", *Proc. of the 1992 IEEE International Conference on Intelligent Robots and Systems*, pp. 261-268.

[2] R.C. Arkin, *Behavior-based robotics, MIT Press, Cambridge*, 1998.

[3] K. Azarm and G. Schmidt, "A decentralized approach for the conflict-free motion of multiple mobile robots", *Proc. of IROS*, 1996, pp. 1667-1673.

[4] T. Balch and R.C. Arkin, "Behavior-based formation control for multirobot teams", *IEEE Trans. on Robotics and Automation*, 1998, pp. 926-939.

[5] B. Barraquand, J.C. Langois, J. C. Latombe, "Numerical potential field techniques for robot path planning", *IEEE Trans. on Robotics and Automation*, vol. 22, issue 2, 1992, pp. 224-241.

[6] M. Bennewitz, W. Burgard, S. Thrun, "Optimizing schedules for prioritized path planning of multi-robot systems", *Proc. of the IEEE International Conference on Robotics Automation (ICRA)*, 2000.

[7] Z. Bien and X. Lee, "A minimum-time trajectory planning method for two robots", *IEEE Transactions on Robotics and Automation*, vol. 8, issue 3, 1992, pp. 414-418.

[8] L. Chua and L. Young, "Cellular Neural Network", *IEEE Transaction on Circuit System*, 1990, pp. 500-505.

[9] L. Chua , "CNN A Vision of Complexity", *Int. Journal of Bifurcation and Chaos*, vol.7, 1997, pp. 2219-2425.

[10] K. S. Evans, C. Unsal, J. S. Bay, "A reactive coordination scheme for many-robot system", *IEEE Trans. on Systems*, 1997, pp. 598-610.

[11] Y. Guo and L. E. Parker, "A distributed and Optimal Motion Planning Approach for Multiple Mobile Robots", *Proc. of ICRA*, 2002, pp. 2612-2619.

[12] K. A. Konolige, "Gradient Method for Realtime Robot Control", *IROS*, 2000, pp. 639646.

[13] P. Orgen, N. E. Leonard, "A Convergent Dynamic Window Approach to Obstacle Avoidance", *IEEE Transaction on Robotics*, 2005.

[14] B. Siemiątkowska, "Cellular Neural Network for Path Planning", *Proc. of SIRS*, 2000, pp. 125-130.

[15] C. Warren, "Multiple robot path coordination using artificial potential fields", *Proc. of ICRA*, pp. 500-505, 1990 Addison-Wesley, 1986.

[16] B. Siemiątkowska, R. Chojecki, M. Olszewski, „Zastosowanie wielowarstwowych sieci komórkowych do planowania trasy dla robota mobilnego" [Application of multi-layer cell networks for route's planning of a mobile robot], *Krajowa Konferencja Robotyki*, Poland, 2006.