2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Experience Mixed the Modified Artificial Potential Field Method

Sijing Wang, Huasong Min

Abstract—**How to find a safe and collision-free path in unstructured environments is always an important issue in mobile robotics. This paper proposed a new path planning method that exploited past experience for obstacle avoidance with a modified artificial potential field, which could help the robot avoid collisions with obstacles effectively and find the optimal path from the start to the goal. This algorithm uses case-based reasoning to obtain the available prior information of the current environment. By retrieving the past cases and adapting to the changes of the environment to solve the problem. The experiments show that this method greatly improves the performance of the robot in terms of time and distance of the path taken from the start to the target.**

I. INTRODUCTION

Path planning is always an important issue in mobile robotics, which is to find an optimal collision-free path from a starting point to a target in an uncertain environment [1].

In order to realize the navigation of an autonomous robot independently, many researchers have proposed different path planning algorithms. Those algorithms can be divided into three categories: traditional methods, intelligent methods and other methods. Traditional methods include visibility graph method, A* search algorithm, artificial potential field method and so on. Visibility graph method [2] can guarantee to find a shortest path, but it has more complicated search process and lower search efficiency, while the environment changes, the model has to be reconstructed; A* search algorithm [3] is a heuristic search algorithm widely used for static problem to search the shortest path quickly, but the global information of the environment was required to be known in advance; Artificial Potential Fields (APF) method [4] is the most common method in dynamic environments, however, its inherent defects are obvious, including considerable problems in local minima and oscillations in a narrow channel. Based on the traditional methods, some researchers presented their improved methods. In 2004, a fast dynamic visibility graph (DVG) [5] was proposed for constructing a reduced roadmap among convex polygonal obstacles. It decreases the computing time and is more adaptive for dealing with multi-target path planning. In 2010, Cheng Piying proposed a novel and improved path planning algorithm called "D++" [6][7], which combines Dijkstra's algorithm with the idea of a senor-based method. The detective range reduces the size of the searching space of Dijkstra's algorithm. However, the limitation of IR sensor will have a considerable effect on the results of the algorithm for a real robot system. Based on APF method, Charifa.S [8] presented an adaptive boundary following algorithm to avoid the local minima, while the calculation is complex. When the size of obstacles is large, it needs to spend a lot of computing time.

Except for the traditional methods described above, there are many artificial intelligence algorithms existing for path planning, such as neural networks [9]-[11], fuzzy logic [12] and genetic algorithm [13][14]. Those algorithms can achieve avoiding obstacles autonomously and result in optimal global path planning, but they are relatively complex and time-consuming. Sometimes they are not very useful for real time applications.

In fact, one of the main issues for mobile robot to navigate in unknown environments is the lack of a priori information about them. If the environmental information could be acquired before path-planning, it would contribute to guiding the robot from the start to the goal. Case-based reasoning (CBR) is such a technique that can reuse previous experience to obtain the suitable information to solve the current problem. Actually, CBR has been used in robotics to perform different tasks for some years. In 2003, Maarja Kruusma [15] presented a global navigation strategy, learning and adaptation by means of CBR permit the robot to choose routes that are less risky to follow and lead faster to the goal based on previous experience. It can improve the performance of the robot in a difficult uncertain environment, while is not efficient in simple dynamic environments. Raquel and Ramon [16] in 2004 integrated a CBR agent into an existing multi-agent navigation system which can prevent the robot from getting blocked in certain situations in real time, however, the path is not optimal. Jaroslav [17] described a path planning method based on CBR and combined with graph algorithms in the environment represented by a rectangular grid. It is applicable to the large-scale environments, saving computing time and traveling distance, nevertheless, the process is too complex to reduce the efficiency of the algorithm.

In conclusion, CBR technique can improve the performance of path planning for mobile robots in complicated environments, while sometimes in simple cases, the effect is not well. In this paper, we proposed a novel path planning method for mobile robots. We called it "EMMAPF". Case-based reasoning is utilized as a basic method to store previous experience which is used to obtain an effective solution to solve the current problem, in addition, a modified artificial potential field (MAPF) is combined as a real-time obstacle avoidance method. No matter in simple or complicated environments, this method always guides the robot with an optimal path. This paper is organized as follows: In Section II, a detail description of "EMMAPF" method is represented, including the modified artificial potential field

Sijing Wang is a master candidate of Computer Science in Wuhan University of Science and Technology, Wuhan, 430065, China, (e-mail: WsjDream@gmail.com)

Huasong Min is a professor in Wuhan University of Science and Technology, Wuhan, 430065,China. (e-mail: mhuasong@wust.edu.cn). He is the corresponding author.

method and CBR technique. In Section Ⅲ, the experiments and analysis results are reported in simulation. Finally, we summary some conclusions and plan the work to do in the future.

II. "EMMAPF" METHOD

Our path planning method mixes past experience with the modified artificial potential field method, not only can help the robot to avoid the obstacles autonomously, but also can save the computing time and traveling distance to obtain an optimal path.

The whole procedure of "EMMAPF" method is shown as follows.

When the robot encounters the obstacle
{ Retrieve the similar case from the casebase by using **nearest neighbor method**.
If there is a similar case/are similar cases in the casebase
{ Return the number **Num** of the similar case/cases. If **Num** is greater than one. Find the most similar case **Cp** by computing the distance **D(R,G)** between the robot and the reset Goal.
Accept the solution **Sp** of **Cp**
{ Reset the transferring direction and two turning goals, modify the turning goals if necessary.}
Calculate the cost r(Pp). It stands for the speed of the robot using CBR to avoid the obstacle.
}
Else
{ Invoke the "MAPF" method to solve the current problem. Find the solution **Sm** to avoid the obstacle**.**
Calculate the cost r(Pp). It stands for the speed of the robot using "MAPF" method to avoid the obstacle.
}
Case learning
{ **If** there is no similar case in the case base.
{ Store the new case, its solution, and **r(Pp)** }
Else
{ **If** no modify the transferring goals
If r(Pp) < r(Pp^{old})
{ Update **r(Pp)** }
Else
{ Store a new case, its solution, and **r(Pp)** }
}
}
}

This pseudo codes describe the process of "EMMAPF" method, when the robot encounters the obstacles during driving, CBR will retrieve the most similar cases from the case base. If the returned solutions are more than one, select the closest case, and then reset the transferring direction. Else if no retrieving case returns, the system will invoke the "MAPF" method. Finally, evaluate the proposed solution by the cost r(Pp), and update the system by learning from the case base. Thus in CBR, learning and adaptation are through accumulation of cases.

A. "MAPF" method

The traditional artificial potential field method is commonly used in real-time obstacle avoidance and smooth path for mobile robots. It's particularly attractive because of its elegant mathematical analysis and simplicity. However, this method exists the local minima problem, and no optimization process is involved. Liu [18] has proposed a concept of virtual obstacle to solve this problem. The virtual obstacle is located around local minimums to repel the mobile robot. We make some improvements based on this approach, modify the potential field force function, and set the virtual obstacle according to the shapes of obstacles detected by the sensor. The steps are shown as follows.

- Attractive force function

$$F_{act}(X) = -grad[U_{act}(X)] = k(X - X_{goal}) \tag{1}$$

Where, k is the attractive factor, $(X - X_{goal})$ represents the distance between the robot and the goal.

- Repulsive force function

$$F_{rep} = \eta\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2}(X - X_{goal})^2 \tag{2}$$

$$F_{rep1} = \eta\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2}(X - X_{goal})^n \tag{3}$$

$$F_{rep2} = \frac{n}{2}\eta\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)^2(X - X_{goal})^{n-1} \tag{4}$$

$$F_{rep1} = \begin{cases} F_{rep1} + F_{rep2} & \rho \le \rho_0 \\ 0 & \rho > \rho_0 \end{cases} \tag{5}$$

Where, ρ is the shortest distance between the position of the robot and one obstacle, ρ_0 is the radius of obstacle sphere of influence. η represents the repulsive factor. n is a real number greater than zero. Contrast with the original potential field function, it introduces the relative distance between the target and the robot $(X - X_{goal})$, ensuring the whole potential field minimum only in the target X_{goal}.

- Resultant force

$$F_{all} = F_{act} + \sum_{1}^{m} F_{rep} \tag{6}$$

m is the number of the obstacles. When the robot enters into the local minimum area, the shape of the obstacle will be detected by the sensor, according to the situation the virtual obstacle is set in one side of the dead area, thus generating the additional repulsion F_{add} to compel the robot to escape from the trap. F_{add} is defined the same as F_{rep}.

- Total force

$$F_{total} = F_{all} + F_{add} \tag{7}$$

The total force will help the robot to escape from the local minima. While the robot breaks away from the sphere of obstacle's influence, that is, two grids around the obstacles,

the virtual obstacle will be canceled, the robot then moves in the resultant direction.

B. CBR technique

Case-based reasoning (CBR) is a self-learning method which helps to solve the current problems by re-using past cases and experiences to adapt to the changes in the environment. This method is based on the concept that similar problems have similar solutions. Three steps can represent a simplified description of our CBR technique: retrieving the most similar case or cases, revising the proposed solution, learning and retaining the new case for future problem solving. Figure 1 is the flow diagram of CBR technique.

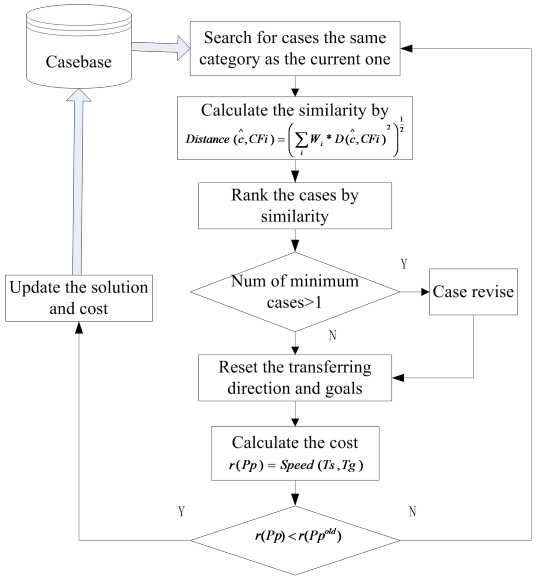


Figure 1. Flow diagram of CBR technique

(a) Case Representation

When exploiting CBR technique to plan path for mobile robots, the first fundamental decision to make is how to establish a case base. Cases can be stored in many different representational formats [19], such as framework representation, object-oriented representation and predicate representation. Because of its features of applicability, summarizing and reasoning, we choose framework as the representational format. A classical case base usually includes characteristic of the problem and the relevant solution that was used on an earlier occasion when a similar situation was encountered. A case can be described as follows:

$$CaseBase =< C1, C2, C3, ..., Ci > \quad (8)$$

$$C =< F, S > \quad (9)$$

Where, F represents the features of the obstacle; S represents the relevant solution.

$$F =< caseID, obstacleCategory, leftDistance, rightDistance, angle > \quad (10)$$

Where:

(1) case ID: identifier of the obstacle.

(2) obstacleCategory: the category of the obstacle's front side where the sensor could scan, such as line, triangle.

(3) leftDistance/rightDistance: the length of the obstacle.

(4) angle: the global angle of the obstacle/the intersection angle for the obstacle.

$$S =< resetDirection, resetGoal, nResetGoal > \quad (11)$$

(5) resetDirection: the transferring direction for the robot.

(6) resetGoal: the first turning point for the obstacle.

(7) nResetGoal: while the robot arrives at the resetGoal, change the direction to the nResetGoal to escape the influential area.

(b) Cost Function

In order to measure the cost of problem solving, all the cases are characterized by a cost function to calculate the robot consumption in obstacle avoidance. It depends on the mean traveling speed of the robot from encountering the obstacles **Ts** to escaping from the influence of the obstacles **Tg**. The faster the robot can avoid the obstacles, the better the solution is.

$$r(Pp) = Speed(Ts, Tg) = D(Ts, Tg)/(Tg - Ts) \quad (12)$$

When the robot encounters an obstacle, r(Pp) will be generated to accumulate the distance and time of each step until the robot gets rid of the influence of the obstacle. A solution is selected with a lower cost.

(c) Case Retrieval

Case retrieval is always the key part in CBR technology and directly impacts on the overall performance of the algorithm. The most commonly investigated retrieval methods by far, are the k-nearest neighbors [20], artificial neural networks [21], knowledge-guided approaches and inductive index approaches [22].

Because of the number of cases to be searched in this case base is small, we choose k-nearest neighbors as our retrieval method, by computing the distance between the current case and each case in the case base ,the closest one will be found as the best solution.

In many situations, different features have different levels of importance and contribution to the success of a case, so a reasonable weight needs to be set to reduce the searching space for the case retriever. Here, we assign the weightings according to the relative importance of functions.

For a given case,

$$F =< caseID, obstacleCategory, leftDistance, rightDistance, angle > \quad (13)$$

The retrieval process carries through the following steps:

Step1. Search the cases which have the same category with the present one in the case base, $C_i \in CF$.

Step2. Calculate the degree of similarity between the present case $\hat{C}$ with each case in CF. A similarity formula can be defined as

$$Distance(\hat{C}, CFi) = \left(\sum_i W_i * D\left(\hat{C}, CF_i\right)^2 \right)^{\frac{1}{2}} \quad (14)$$

$$D\left(\hat{C},CFi\right)=abs(\hat{C}-CFi) \tag{15}$$

Where, (14) is the most common type of distance measure called Euclidean distance.

Step3. Rank the cases by similarity measure calculated in Step2.

Step4. Choose the most similar case.

$$Distance=\min(Distance(\hat{C},CFi),i\in(1,size(CF))) \tag{16}$$

(d) Case Adaptation

When the case retrieval finished, we need to adapt the solution to fulfill the needs of the current case. If the returned similar cases are more than one, compute the length according to the positions of the robot and the first reset goal to obtain the closest one. If necessary, modify the solution. The following pseudo code of case adaptation is shown as follows:

```
num=number(retrieved case/cases);
count =1;
If (num>1) Then
    For i=1 to num
        Dist(i)=sqrt((Robot.x-resetGoal(i).x)^2+
                     (Robot.y-resetGoal(i).y)^2);
        If Dist(i)<Dist(count)
            count=i;
            i=i+1;
        End
    End
     Return case(i);
Else
     Return case(1);
End
If  Dist(count)>10
   Modify the two turning goals according to the current case.
End
   Update r(Pp).
```

(e) Case Learning and Case Retaining

Case learning takes an important part in CBR system, it involves the policies and techniques of adding, deleting and updating cases in a CBR system in order to guarantee its ongoing effectiveness and performance. Here, case learning is measured by the cost introduced in II-B-(b), cases with a lower cost can play a positive role in the decision making.

If there is no similar case in the case base during case retrieval, the system will invoke the "MAPF" method to solve the current problem. The new case, its relevant solution, and r(Pp) will be retained into the case base. It not only increases the coverage of the case base, but also reduces the distance between an input vector and the closest stored vector.

In addition, after case adaptation, the solution may be modified in order to adapt to the current environment.

If r(Pp) < r(Pp^{old}), update the cost to have a better learning process. What's more, if the size of the case base is greater than 30, some useless cases will be rejected to keep the efficiency of the system.

III. EXPERIMENT

This section is designed to compare the performance of the robot with "MAPF" method and "EMMAPF" method. We have demonstrated some simulations to verify the efficiency and accuracy of our method. In order to facilitate our actual test, the robot is depicted as a hexagon in the simulation. First of all, the kinematical structure of the robot is described in the following subsection.

A. *Kinematical Model*

The experiments are implemented by using the three omni-wheel robot. It has a 32-bit ARM Cortex-M3 microprocessor. The omni-wheel robot is shown as Figure 2.

Figure 2. Omni-wheel robot

Kinematical equation can be obtained from its kinematical diagram described in Figure 3.

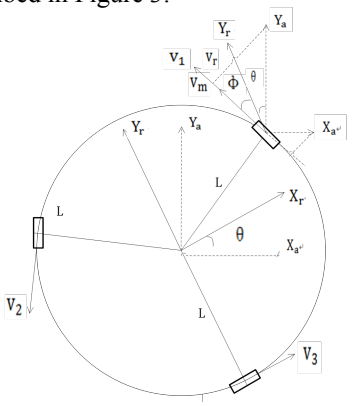


Figure 3. kinematical diagram

Where, $[\ X_a \quad Y_a\ ]^T$ is the world coordinates, $[\ X_r \quad Y_r\ ]^T$ represents a moving frame with respect to the center of the robot, θ is the rotation angle which is positive in the counterclockwise direction. L is the distance between the robot and the center of each wheel. $[V_1,V_2,V_3]^T$ stands for the translational velocity of each wheel.

After reasoning, Vi of each wheel can be obtained and described in the vector-matrix form as follows.

$$\begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix} = P(\theta)\bullet\begin{bmatrix} \dot{X}_a \\ \dot{y}_a \\ \dot{\theta} \end{bmatrix} \tag{17}$$

Where

$$P(\theta)=\begin{bmatrix} -\sin(\theta+30°) & \cos(\theta+30°) & L \\ -\sin(30°-\theta) & -\cos(30°-\theta) & L \\ \cos\theta & \sin\theta & L \end{bmatrix} \tag{18}$$

In fact, the velocity of each wheel has been given in advance; we can obtain the orientation and speed of the robot through another equation transformed by (14).

$$\begin{bmatrix} \dot{X}_a \\ \dot{y}_a \\ \dot{\theta} \end{bmatrix} = P^{-1}(\theta) \begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix} \tag{19}$$

Where

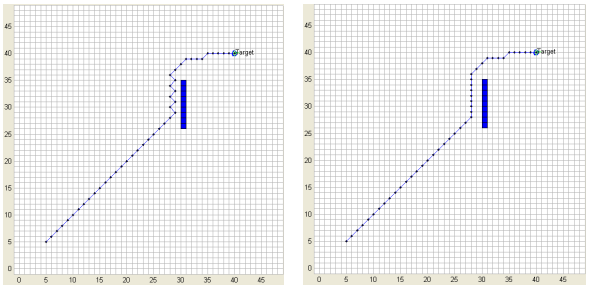
$$P^{-1}(\theta) = \begin{bmatrix} -\frac{2}{3}\sin(\theta+30°) & \frac{2}{3}\sin(\theta-30°) & \frac{2}{3}\cos\theta \\ \frac{2}{3}\cos(\theta+30°) & -\frac{2}{3}\cos(\theta-30°) & \frac{2}{3}\sin\theta \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \tag{20}$$

B. Simulation

Based on kinematical equation described above, experiments are designed to verify the performance of "EMMAPF" method. In simulations, four different scenarios have been designed to run the experiments, including one linear obstacle, one triangle obstacle, simple multi-obstacles and complex multi-obstacles.

To model the partially known environment, a grid-based map is used. The area of each small grid is 1cm*1cm, while the whole grid space is 50cm*50cm, start point (5, 5), and target point (40, 40). The velocity of the robot is a unit grid per second. The distance and time below respectively represent the path length and time consumption from the start point to the target point.

(1) one linear obstacle



(a) (b)

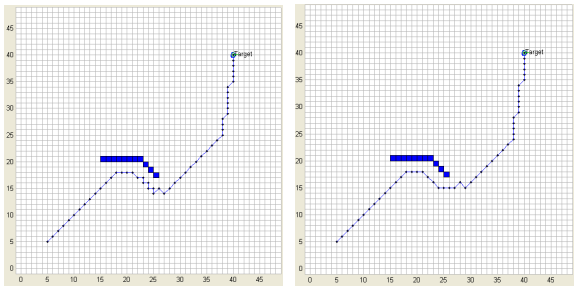
Figure 4. Path with "MAPF" method and "EMMAPF" method in Scenario1.

TABLE I. AVERAGE TIME AND DISTANCE OF THE PATH IN SCENARIO 1.

Scenario 1	time	distance
MAPF	46.703s	62.49cm
EMMAPF	44.578s	59.178cm

As we can see in scenario1, "EMMAPF" method reduces the average time by almost 4.3%, and the distance from the start to the target is reduced by almost 5.3%.

(2) one triangle obstacle



(a) (b)

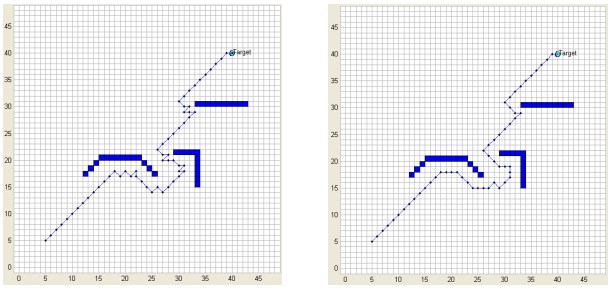
Figure 5. Path with "MAPF" method and "EMMAPF" method in Scenario2.

TABLE II. AVERAGE TIME AND DISTANCE OF THE PATH IN SCENARIO 2.

Scenario 2	time	distance
MAPF	53.578s	64.42cm
EMMAPF	51.546s	61.006cm

As it can be seen from TABLE II, the average traveling time of "EMMAPF" method reduces almost by 3.8%, it also minimized the distance 5.3% with respect to "MAPF" method.

(3) simple multi-obstacles



(a) (b)

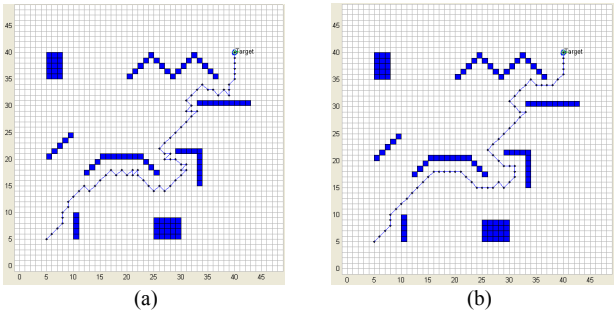
Figure 6. Path with "MAPF" method and "EMMAPF" method in Scenario3.

TABLE III. AVERAGE TIME AND DISTANCE OF THE PATH IN SCENARIO 3.

Scenario 3	time	distance
MAPF	65.423s	77.872cm
EMMAPF	57.078s	69.974cm

Scenario 3 is a little complicated than the former scenarios, using "EMMAPF" method is more effective in the performance of time and distance. The data in TABLE III show that using this method minimized the traveling time 12.8% approximately, and minimized the distance about 10.1%.

(4) complex multi-obstacles



(a) (b)

Figure 7. Path with "MAPF" method and "EMMAPF" method in Scenario4.

TABLE IV. AVERAGE TIME AND DISTANCE OF THE PATH IN SCENARIO 4.

Scenario 4	time	distance
MAPF	73.415s	86.458cm
EMMAPF	63.757s	74.312cm

Table IV and Figure 7 show how much the performance of the path planning system depends on the properties of the environment, it is obvious to see the degree of optimization in Scenario 4 is better than that in Scenario 1, 2 and 3.

C. Result analysis

Based on the experimental results, we can conclude that the path of "EMMAPF" method is more optimal than "MAPF" method. The path generated through "MAPF" method is safe but long, and passes through many unnecessary places. Using "EMMAPF" method to obtain the available prior experience can help the robot to make the best decisions when it encounters the obstacles. The experiments demonstrated that "EMMAPF" method can reduce the collision risk, traversal time and distance obviously.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have proposed "EMMAPF" method to solve the path planning of mobile robot. This method mainly combines past experience with the modified potential field method. By acquiring the environmental information every time before path planning, robot can deal with unknown and dynamic environment in real-time. Therefore, "EMMAPF" method is not only suitable for simple situations, but also complex environments. Compared with the above algorithms introduced in Section Ⅰ, "EMMAPF" method improves the performance of the robot in terms of time and distance of the path. The resulting path is optimal and computing process is simple and fast.

In future, we need to pay more attention to analysis the performance of CBR in real-time, some improvements also need be taken to shorten the time in searching and matching, otherwise time delay in CBR mechanism would lead to the failure of the obstacle avoidance. With the development of "EMMAPF" method, our system will become a real-time system to deal with path planning for mobile robot in the future.

ACKNOWLEDGMENT

This research work is supported by National Natural Science Foundation of China. (Project No. 61175094).

REFERENCES

[1] P Raja*, S Pugazhenthi. "Path Planning for a mobile robot in dynamic environments", International Journal of the Physical Sciences, vol.6 (20), pp. 4721- 4731, 2011.

[2] J. Kitzinger. "The Visibility Graph Among Polygonal Obstacles: a Comparision of Algorithms", Publisher, University of New Mexico, 2003

[3] Hart PE, NilssonNJ, Raphael B. "Correction to a formal basis for the heuristic determination of minimum cost paths". SIGART Newslett , vol.37, pp. 28–29, 1972.

[4] Koren Y, Borenstein J. "Potential field methods and their inherent limitations for mobile robot navigation". In: IEEE International conference of robotics and automation, vol.2, pp. 1398–1404,1991.

[5] Huang Han-Pang, Chung Shu-Yun. "Dynamic Visibility Graph for Path Planning". Intelligent Robots and Systems (iRos), pp. 2813-2818, 2004.

[6] Cheng Pi-Ying and Chen Pin-Jyun. "The D++ Algorithm: Real-Time and Collision-Free Path-Planning for Mobile Robot". The 2010 IEEE/RSJ. International conference on intelligent robots and systems (iRos), pp.3611–3616, 2010.

[7] Cheng Pi-Ying, Chen Pin-Jyun. "Navigation of mobile robot by using D++ algorithm". Intel Serv Robotics, vol.5, pp. 229-243, 2012.

[8] Charifa, S. and M. Bikdash. "Adaptive boundary-following algorithm guided by artificial potential field for robot navigation". Proceedings of the IEEE Workshop on Robotic Intelligence in Informationally Structured Space, Nashville, TN, pp. 38-45, 2009.

[9] Roy Glasius, Andrzej Komoda. "Neural Network Dynamic for Path Planning and Obstacle Avoidance". Neural Network, vol.8, No.1, pp. 125-133, 1995.

[10] Simon X. Yang, Max Meng. "Neural Network Approaches to Dynamic Collision-Free Trajectory Generation," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, vol.31, pp. 302-318, 2001.

[11] Q.Hong, Yang S.X, Willms A.R and Y.Zhang, "Real-Time robot path planning based on a modified pulse-coupled neural network model," IEEE Trans. Neural Networks, vol.20, pp. 1724–1739, 2009.

[12] Khaldoun K. Tahboub, Munaf S. N. Al-Din. "A Neuro-Fuzzy Reasoning System for Mobile Robot Navigation". Jordan Journal of Mechanical and Industrial Engineering, vol.3, pp. 77-88, 2009.

[13] M. Gemeinder, M. Gerke. "GA-based path planning for mobile robot systems employing an active search algorithm". Applied Soft Computing, vol.3, pp. 149–158, 2003.

[14] W. Chen, H Qin. "Path planning of mobile robot based on hybrid cascaded genetic algorithm". Intelligent Control and Automation, pp. 501–504, 2011.

[15] Maarja Kruusmaa*. "Global navigation in dynamic environments using case-based reasoning". Autonomous Robots, vol.14, pp. 71-91, 2003.

[16] Raquel Ros, Ramon Lopez de Mantaras. "A CBR system for autonomous robot navigation". Artificial Intelligence Research Institute. C02 (02), 2005

[17] Jaroslav Hodal, Jiri Dvorak. "Using case-based reasoning for mobile robot path planning". Engineering Mechanics, no.3, vol.15, pp. 181-191, 2008.

[18] Liu Chunyang, Cheng Yiqiang, Liu Changan. "Anti-collision path planning for mobile robot based on modified potential field method". Journal of Southeast University, vol.39, pp. 116-120, 2009.

[19] Sankar K. Pal, Simon C.K. Shiu. "Foundations Of Soft Case-based Reasoning, John Wiley: NY". Wiley-Interscience, USA, 2004.

[20] Xiao-Hong Wu, Jian-Jiang Zhou. "Kernel-based fuzzy k-nearest neighbor algorithm, International Conference on Computational Intelligent for Modeling, Control and Automatio". Austria, IEEE Computational Society, vol. 2, pp. 28-30, 2005.

[21] Chang pei-Chann, Lin Jyun-Jie. "Forecasting of manufacturing cost in mobile phone products by case-based reasoning and artificial neural network models". Journal of Intelligent Manufacturing, vol.23, pp. 517-531, 2012.

[22] Kyung-shik Shina, Ingoo Han. "A case-based approach using inductive indexing for corporate bond rating". Decision Support Systems, Vol.32, pp. 41–52, 2001.