Short-Safe Compromise Path for Mobile Robot Navigation In A Dynamic Unknown Environment

Sardjono Trihatmo
R.A. Jarvis
Intelligent Robotic Research Center
Monash University, Wellington Road, Clayton
Victoria 3168, Australia
Sardjono@inn.bppt.go.id
Ray.Jarvis@eng.monash.edu.au

Abstract

This paper presents a path planning method for a mobile robot in an environment that is unknown and can change. This method relies on continually sensing the local environment. Robot and personnel safety is a dominant and essential requirement for this path planning method. A new method for obtaining safe directions in a tessellated map of the environment is introduced. Linear combination of vectors is used to obtain the path that is a compromise between the safest and the shortest path. The path enables a mobile robot to reach its goal directly and safely.

1. Introduction

Successful navigation of mobile robots requires planning a path that enables the robot to reach a predefined goal without collision with obstacles. Two path planning methods that are widely used in robotic research are the A* search method [Nilsson, 1971 & Jarvis, 1983] and the Distance Transform (DT) method [Jarvis, 1993]. The A* search algorithm determines the shortest path among polyhedral obstacles. The Distance Transform algorithm generates the minimum number of steps, thus the shortest path, in a cell based environmental model.

Although both path-planning algorithms (A* and DT) are designed so that the robot avoids collision, there is a tendency for the robot to be driven close to obstacles. Consequently, there is a risk of collision since perfect motion control and exact data about the environment are difficult to obtain. Zelinsky introduced a safe path transform method in his work [Zelinsky, 1994]. This method, which is an extension to the DT method, generates a path that is safer than that of the basic DT method in a known environment. The randomized potential field method [Barraquand and Latombe, 1991] generates a safe path, but the path tends to stray far from the direct path to the goal. The other problem of the randomized potential field method is the existence of local minima at which the robot may get stuck. Other path planning algorithms such the randomized roadmap [Amato and Wu, 1996] and rapid- exploring random tree [La Valle and Kuffner, 1999] are powerful in a cluttered configuration space and the kinodynamic planning problem, respectively. However, both path planning algorithms do not state the safeness of the path explicitly.

In a dynamic unknown environment, a global path to the goal must be recalculated every time the global environment changes or some new aspect of it is discovered as the robot moves. Otherwise, collisions easily occur when using out of date mapping data as a basis for path planning.

Instead of determining a global path and the necessity to recalculate it every time the environment changes, it is useful and essential to develop a path planning method that relies on continually sensing local environments. This so called local path planning method obtains a local target in a local environment such that the mobile robot can move to this local target safely. At the local target the environmental map will be updated and the next local target found and so on until the goal is reached. It must be assumed that the local environment does not change when the robot is moving to the local target.

Lipton has developed the sophisticated *local next step* path planning method [Lipton, 1997] for navigation amongst polyhedral obstacles which are previously unknown. The local target is obtained by using a combination of the desirability and danger function. However, there is a requirement to find a cul de sac in advance to enable the robot to avoid it.

This paper presents a path planning method for a dynamic unknown environment which is tessellated. This path planning method estimates the shortest global path and obtains the safe local direction in which a mobile robot moves safely in a local environment. The estimation of the shortest global path makes it possible for the robot to reach its goal directly and to escape from a cul de sac without prior knowledge of it.

The ability to obtain safe local directions enables the robot to avoid collision with obstacles although there is inaccuracy in the estimation of the global robot using efficient computational and mathematical methods. The Distance Transform and the linear vector combination are the keys to developing this path planning method.

The next section describes the assumptions about uncertain regions of the environment. The path planning method is presented in detail in section 3. Results of computer simulation are presented and discussed in section 4. And finally, conclusions are made in section 5.

2. Assumption in the Environment Model

This path planning method is devised for a cell-based model of the environment. The environment is projected into a two-dimensional tessellated space which represents the robot's workspace, as presented in the work of Murray and Jennings [Murray and Jennings, 1997]. The cells in the workspace are either occupied or free and they represent obstacles and free space, respectively. When using a sensor that has a limited sensor range, there is a part of the environment that can not be viewed by the sensor, as can be shown in Figure 1. This unknown region is assumed to be occupied to obtain the safest direction in a local environment. This assumption is regarded as the pessimistic assumption (Figure 1c). By contrast, this region must be assumed to be free in order to estimate the shortest global path. This assumption of the unknown region is regarded as the optimistic assumption (Figure 1d).

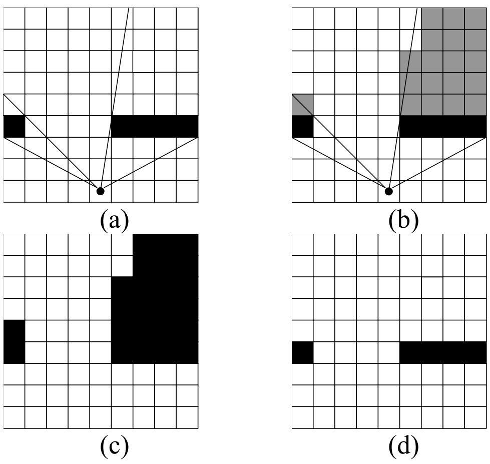


Figure 1. The environment model.
The robot's position is represented by a filled circle. A segment describes the robot's view. Black cells represent obstacles. Empty cells are free areas. Gray cells represent the uncertain region. Since the environment is tessellated, a cell is position. Since this path planning method requires only the information about local environments, the use of a limited range sensor such a stereo vision system, is adequate.

The development of this method is highly motivated by the need to obtain a robust solution for the real world uncertain if 50% or more of its parts contain unknown regions.

3. Path Planning

This path planning method determines local targets that enable the robot to reach its goal. The direction to a local target must fulfill two requirements. First, the direction must enable a mobile robot to avoid collision with obstacles. Secondly, the direction must enable the robot to reach its goal directly. Therefore, there is a need to obtain a safe direction in a local environment which is also consistent with the current estimation of the shortest global path to the goal.

In order to obtain such a direction, we define a set of local vectors which represents all possible directions to a local target. Two local vectors, which are the safest local vector and the shortest local vector, must be determined. The safest local vector represents the safest local direction and the shortest local vector represents the direction in which the robot moves most consistently with the current estimation of the shortest global. A linear combination of these two vectors determines the direction that is a compromise between the safest local direction and the direction of the shortest global path.

3.1. Local Coordinates System

To have a better view of the relationship between the robot and its environment, a local coordinate system $(u, v) \in \Re^2$ is established, as shown in Figure 2.

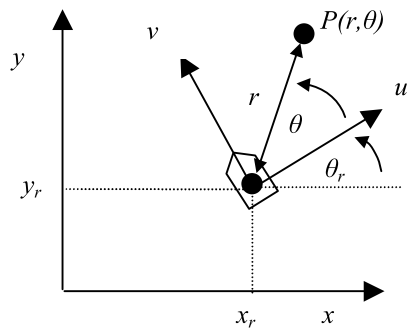


Figure 2. Local coordinate system (u, v).

The center of this coordinate system is the reference point of the robot $(x_r, y_r), x_r \in x, y_r \in y$, where $(x, y) \in \Re^2$ is the world coordinate system. The orientation axis of the robot is aligned to the vertical axis of the local coordinate system (v-axis). The angle

θ_r is the angle between the horizontal axis of the local coordinate system (u-axis) and of the world coordinate system (x-axis). Since the path planning method needs to obtain direction, it is useful to transform the Cartesian coordinate system (u,v) into a polar coordinate system $(r,\theta) \in \Re^2$, where $u = \cos(\theta)$ and $v = r\sin(\theta)$. The relation between polar coordinates (r,θ) and the world Cartesian coordinates (x,y) is given as:

$$r = \sqrt{(x - x_r)^2 + (y - y_r)^2} \qquad (3.1.1)$$

$$\theta = \tan^{-1}\left(\frac{y - y_r}{x - x_r}\right) - \theta_r \qquad (3.1.2)$$

3.2. Local vectors

Figure 3 shows robot's workspace in the local coordinate system. The forward view angle of the robot, α, is represented by a segment of a circle.

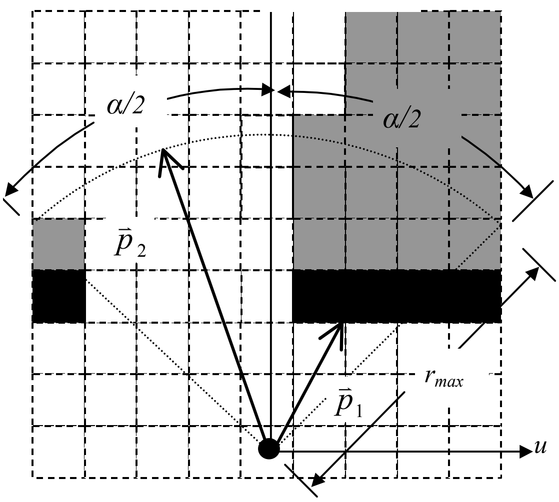


Figure 3. Local vectors representing the directions.

The radius r_{max} of the circle represents a distance within which the environmental data from a range sensor are still reliable. Within this segment each direction $\theta; (\frac{\pi}{2} + \frac{\alpha}{2}) \le \theta \le (\frac{\pi}{2} - \frac{\alpha}{2}), \theta, \alpha \in \Re$, is represented by a vector $\vec{p}$. The direction of $\vec{p}$ is defined as $\angle\vec{p} = \theta$. The magnitude $|\vec{p}|$ is either r_{max} [$|\vec{p}|_2$ in Figure 3] or a distance to a position in which the robot begins to hit obstacle [$|\vec{p}|_1$ in Figure 3].

3.3. The shortest local vector

The DT shortest global path needs to be estimated to obtain the shortest local vector. In a locally known environment the DT path is obtained by making an optimistic assumption in the environmental model [see section 2].

In the DT path planning method, every empty cell in the robot's workspace has a value that represents a cell count distance from a given goal. The goal cell has zero distance from itself. The distance values of empty cells are obtained by using the DT algorithm [Jarvis, 1993]. Then, the shortest global path, and thus the minimum steps which the robot must take from any starting point to the goal is obtained by using a steepest decent tracking algorithm [Jarvis, 1993]. Figure 4a. shows the DT path planning method.

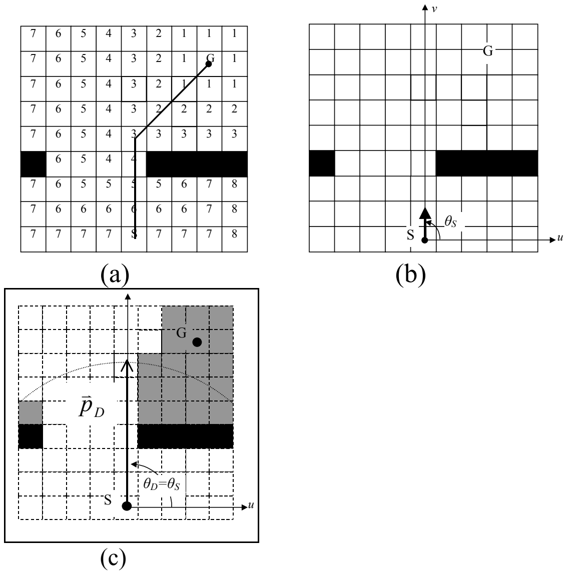


Figure 4. The Distance Transform and the shortest direction. G = goal, S = starting point.

(a) The shortest global path is obtained by making an optimistic assumption.

(b) The direction of the first step, θ_S, that the robot needs to do according to the steepest descent algorithm.

(c) The shortest local vector $\vec{p}_D$.

The steepest descent algorithm determines the first step that a mobile robot needs to do from the starting point. The direction of this step is represented by the angle θ_S, as can be seen in Figure 4b.

If there is a local vector $\vec{p}_D$ so that its direction is the same as the direction of the step $\angle\vec{p}_D = \theta_D = \theta_S$, then $\vec{p}_D$ is the shortest local vector (Figure 4c).

3.4. The safest local vector

A local vector [see Figure 3] determines how far a mobile robot can move in a direction before it hits something. However it does not state, how safe every position in the direction is. A safe position is determined by its minimum distance to the boundary of the closest obstacle.

A method to obtain the minimum distance of a position to the boundary of the closest obstacle is presented in the work of Zelinsky [Zelinsky, 1994]. The method that is called obstacle transform, modifies the DT algorithm in a way that the boundary of obstacles (occupied cells) have zero distance from themselves and the distance value of an empty cell represents its cell count distance from the boundary of the closest obstacle. Figure 5a shows the obstacle transform.

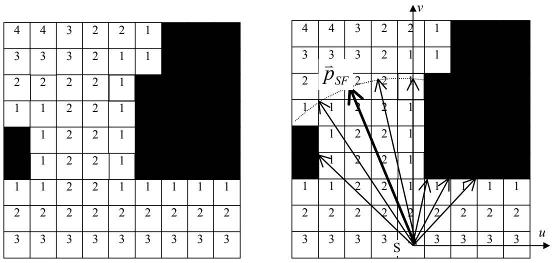


Figure 5.
(a) The obstacle transform.
(b) The safest local vector $\vec{p}_{SF}$ (bold arrow).

The larger the distance value of a cell, the farther the cell is from the nearest obstacle, and thus the safer is the position for the robot. A direction is safe if in this direction a mobile robot moves and steps on cells which have a large distance value. If distance values in a direction are added together, then their total implies the following statement:
One direction is safer than another one if its total of the distance values up to some distance limit is larger than the total of the other one. The distance limit is the maximum sensor range within which the environmental data is still reliable.

Up to the end of this subsection, the method to obtain the safest local vector, and thus the safest local direction is presented.

As can be seen in figure 5b, a set of local vectors is defined as

$$\mathbf{P} = \{\vec{p}_1, \vec{p}_2 ..., \vec{p}_n\} \qquad n \in \aleph \qquad (3.4.1)$$

Beginning from the starting point S, a series of equal steps are taken in each direction. A set of cells which are visited in each step in a direction of $\vec{p}_k$, is defined as:

$$\mathbf{C} = \{c_{\vec{p}_k,i}, ..., c_{\vec{p}_k,i_{\max}}\} \quad i_{\max} = \frac{|\vec{p}_k|}{l} \qquad (3.4.2)$$

where $c_{\vec{p}_k,i}$ is the cell that is visited in i^{th} step in the direction of $\vec{p}_k$. and l is the length of the step. The maximum length of the step is the minimum distance from the center of gravity (COG) of a cell to the COG of any of its neighbor cells. This ensures either stepping on the same cell or on any of the neighbor cells.
If $n(c_{\vec{p}_k,i})$ is the value of the cell $c_{\vec{p}_k,i}$, then the total of the values in the direction of $\vec{p}_k$ is defined as:

$$N(\vec{p}_k) = \sum_{i=1}^{i_{\max}} n(c_{\vec{p}_k,i}) . \qquad (3.4.3)$$

The safest local direction is a local direction in which the total of distance values is the largest. This direction is represented by the safest local vector $\vec{p}_{SF}$ that is defined as

$$\vec{p}_{SF} : N(\vec{p}_{SF}) = \max\{N(\vec{p}_k)\}, \vec{p}_k \in \mathbf{P} \quad (3.4.4)$$

Figure 6 shows the totals of distance values in some discrete directions which are described in Figure 5b.

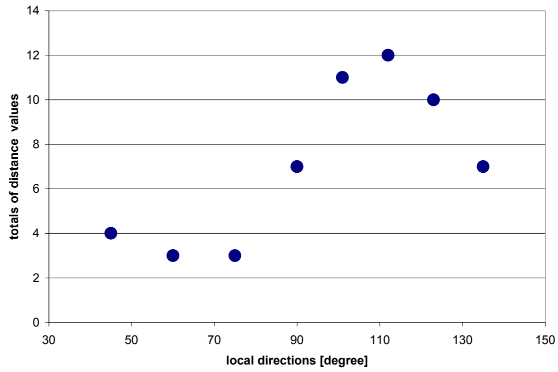


Figure 6. The totals of distance values.
The maximum total is in the direction of 112 degree.

3.5. Local target

The direction to the local target is determined by the direction of a vector that is a linear combination of the shortest and the safest local vector.

$\vec{p}_{SF}$ and $\vec{p}_D$ are the safest local vector and the shortest local vector, as shown in Figure 4c and 5b respectively. The unit vectors $\vec{u}_{SF}$ and $\vec{u}_D$ are determined as:

$$\vec{u}_{SF} = \frac{1}{|\vec{p}_{SF}|} \vec{p}_{SF} \qquad (3.5.1)$$

and

$$\vec{u}_D = \frac{1}{|\vec{p}_D|} \vec{p}_D . \qquad (3.5.2)$$

A scalar α is defined as the weighting factor. If vector $\vec{P}$ is the result of the linear combination of vectors $\vec{u}_{SF}$ and $\vec{u}_D$, which is performed as:

$$\vec{P} = \alpha.\vec{u}_{SF} + (1-\alpha).\vec{u}_D \qquad (3.5.3)$$

$$\alpha \subset \Re, \alpha \in [0...1],$$

then the direction to the local target θ_{target} is determined by the direction of $\vec{P}$.

$$\theta_{target} = \angle \vec{P} . \quad (3.5.4)$$

If there is a local vector $\vec{p}_L$ that has same direction as the direction of $\vec{P}$

$$\angle \vec{p}_L = \angle \vec{P} , \quad (3.5.5)$$

then the distance of the local target d_{target} must fulfill the following requirement:

$$d_{target} < \left| \vec{p}_L \right| \quad (3.5.6)$$

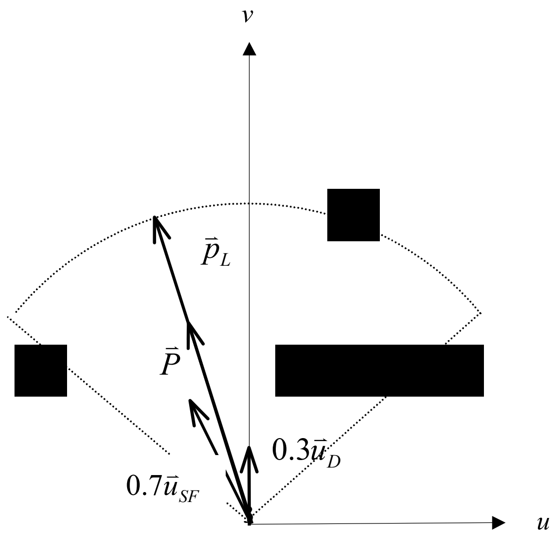


Figure 7. The direction of the local target $\vec{p}_L$.
The weighting factor α is 0.7.

The weighting factor α determines the priority of the safeness or directness of the path. For small values of α, direct paths are preferred. As α increases, then the robot tends to travel in safe directions. Figure 7 shows the direction of the local target that is determined by using a linear combination of vectors.

3.6. Escaping from a cul de sac

There is a situation where the shortest local vector $\vec{p}_D$ can not be obtained. That is the case if $\theta_S < (\frac{\pi}{2} - \frac{\alpha}{2})$ and $\theta_S > (\frac{\pi}{2} + \frac{\alpha}{2})$, where θ_S is the direction of the shortest global path at the starting point $\theta_S \rightarrow r_S = f(\theta_S) = 0$.

This situation indicates to the robot that there are two possibilities. First, the robot could be moving in the direction that is not consistent with the current estimation of the shortest global path (Figure 8a). Secondly, the robot could be entering a cul de sac (Figure 8b). In order to generates a direct path to the goal, the mobile robot must turn to the direction of the shortest global path θ_S but must not move forward, since a new local environment has not been previously explored.

$$\theta_{target} = \theta_S \quad (3.6.1)$$

$$d_{target} = 0 \quad (3.6.2)$$

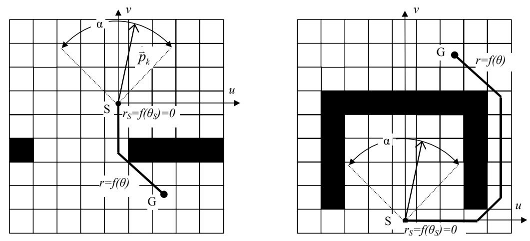


Figure 8.
(a) The robot is about to move in an 'undesired' direction.
(b) The robot is about to enter a cul de sac.

3.7. Threshold for the shortest local direction

The idea of a linear combination of vectors in the path planning is motivated by a consideration that in many cases the shortest local direction leads the robot close to obstacles. However, it does not mean that the shortest local direction is definitely not safe. There is a possibility that the total distance values in the shortest local direction do not make a big difference to the total in the safest local direction. Therefore, determining a safety threshold *δ*, $0 < \delta < 1, \delta \in \Re$ for local directions is beneficial for directness of the path to the goal.

If the comparison between the total of distance values in the shortest local direction $N(\vec{p}_D)$ and in the safest local direction $N(\vec{p}_{SF})$ is above the threshold *δ*,

$$\frac{N(\vec{p}_D)}{N(\vec{p}_{SF})} > \delta \quad (3.7.1)$$

then the shortest local direction is safe for the robot to move. The weighting factor α is set to be zero.

$$\theta_{target} = \angle \vec{p}_D \quad (3.7.2)$$

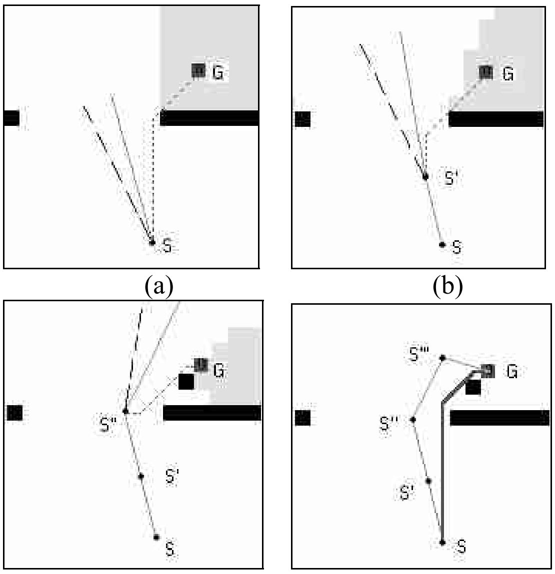
$$d_{target} < \left| \vec{p}_D \right| \quad (3.7.3)$$

As a result, the mobile robot moves in the shortest local direction.

4. Results and Discussion

This section presents some results of computer simulation. The robot is represented by an object point. If the physical extent of the robot must be considered, then the obstacles must be grown, as introduced by Jarvis in his paper [Jarvis, 1983]. The weighting factor α is 0.6 when it is not specifically mentioned.

In Figure 9a, a mobile robot needs to go from the starting point S to the destination G in an unknown environment. An obstacle near the goal is initially not in view. The maximum length of a robot step is four times the cell size. After the second step S'' (Figure 9c), the obstacle near the goal is in view. The robot avoids this obstacle and finally, the goal is reached (Figure 9d).

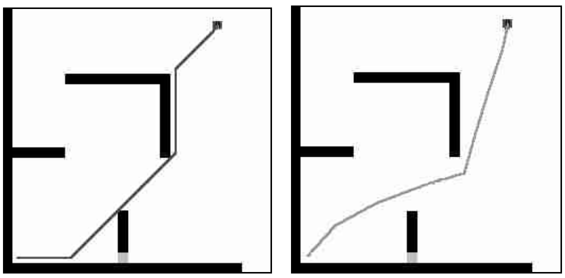


(a) (b) (c) (d)

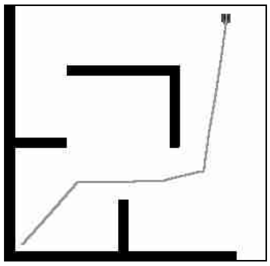
Figure 9. Sequential path planning.
Figure 9a, b, and c: The safest local direction (dashed line). The direction of the local target (solid line). The shortest the shortest global path at the current robot position (the dotted line).
Figure 9d: The compromise path (single line). The shortest global path (bold line).

It is shown in Figure 9d that the compromise path has safe distances to obstacles. If the path is compared with the shortest global path, under the assumption that the global environment is known, the compromise path is safer than the shortest global path.

Figure 10 shows the influence of the weighting factor α to safety of the path. For α =0, the robot goes directly to the goal. As α increases, the robot tends to find positions which are far from obstacles.



(a) α =0 (b) α =0.5



(c) α =1

Figure 10. Influence of the weighting factor to safety of the path.

Figure 11 shows how the robot is able to escape from a series of cul de sac and reaches its goal without prior knowledge about the existence of the cul de sac.

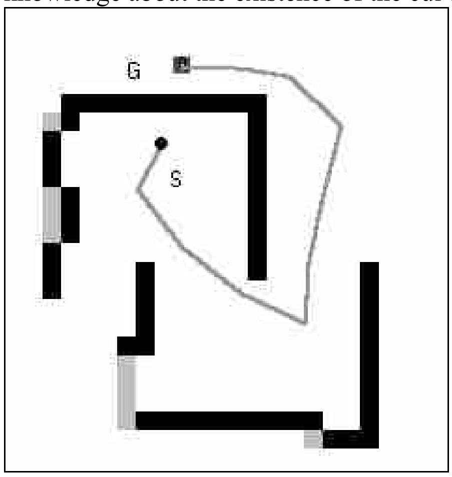
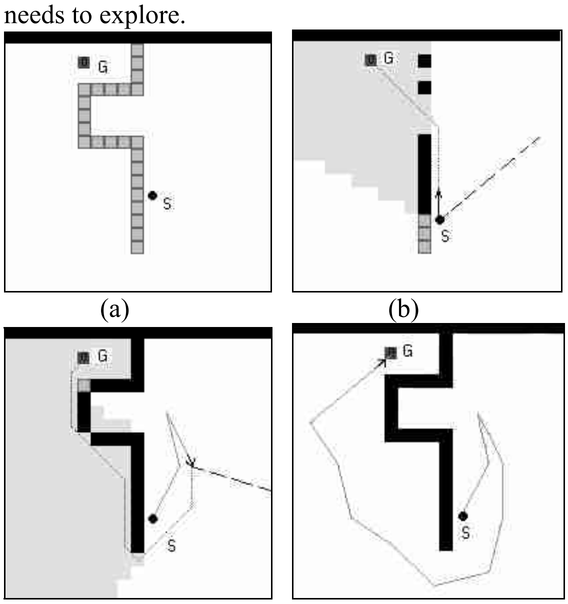


Figure 11. Escaping from the cul de sac.

Figure 12a shows the global environment that the robot needs to explore.

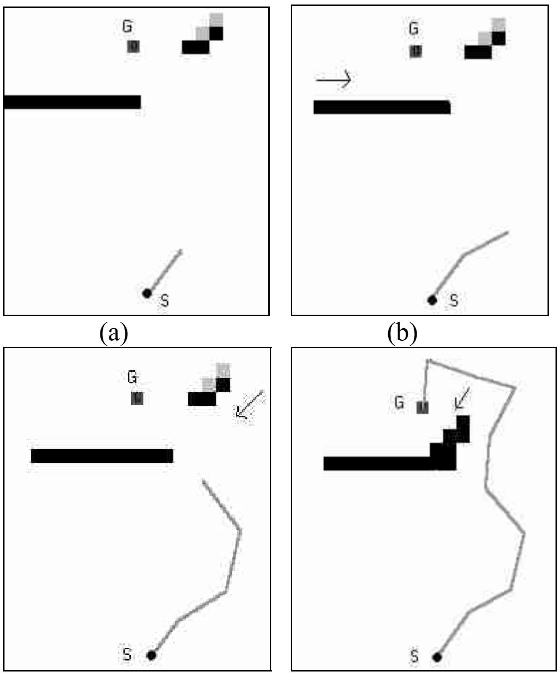


(a) (b) (c) (d)

Figure 12. Recovering the path to the goal.
The safest local path (dash line). The shortest global path from the current robot position (dot line). The compromise path (solid line). Robot orientation (arrow)

In Figure 12b the robot tends to go in an undesired direction because the environment is only partially known and because of the optimistic assumption. In

Figure 12c, the robot obtains a direction error after more information of the environment has been received. Finally, in Figure 12d the robot is able to recover the path to the goal.



(a) (b)

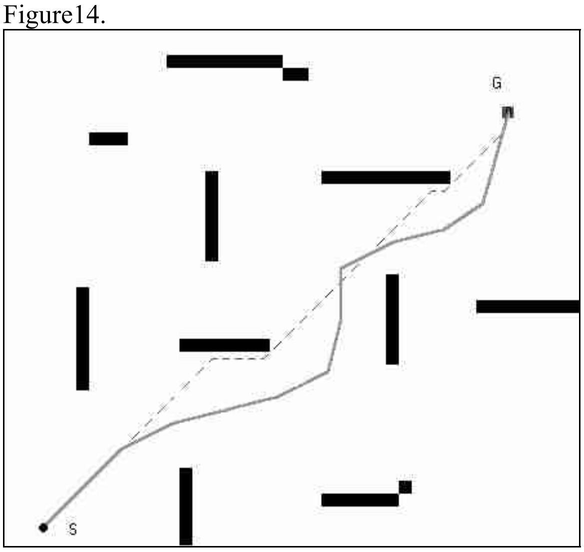
(c) (d)

Figure 13. Moving obstacles

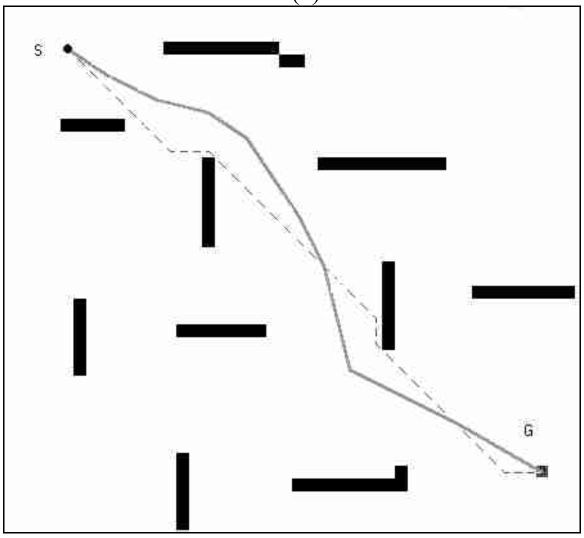
Figure 13 shows the ability of the robot to cope with changes of the environment. A moving obstacle (Figure 13b) causes the deviation of the path. In Figure 13c another obstacle is moving so that the direct path to the goal is or will be soon be blocked. Figure 13d show the ability of the robot to find an alternative way to the goal.

Figure 14a,b,c,and d show the compromise path and the shortest global path in complex environments. It is shown that the compromise path (solid line) is safer than the shortest global path (dashed line).

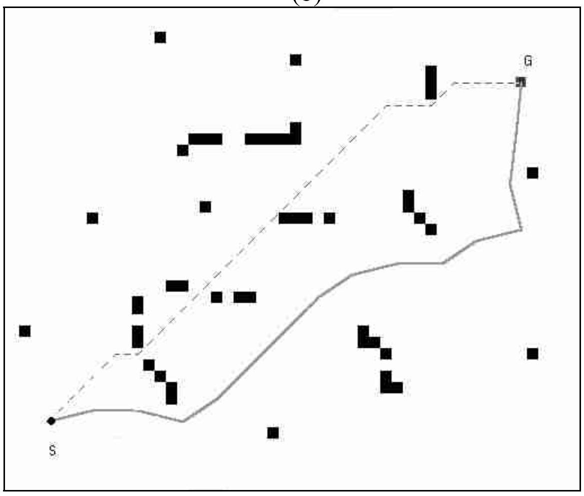
Figure14.



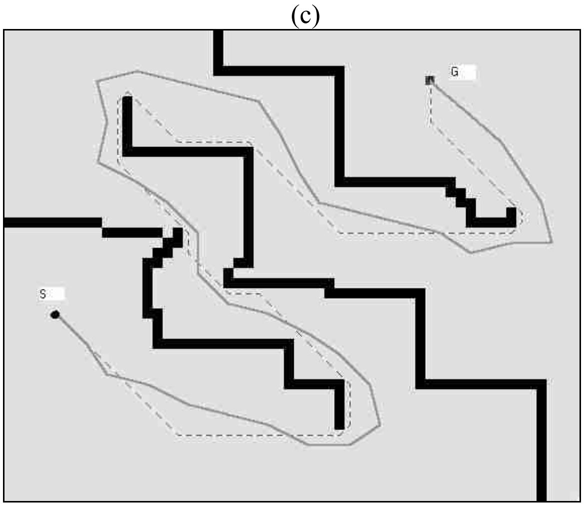
(a)



(b)



(c)



(d)

5. Laboratory Experiment

We conducted a laboratory experiment to prove the path planning algorithm. An odometry was used to estimate the robot position. The odometry reflects the incompleteness of the accuracy of a position estimator.

However, the ability of this path planning algorithm to obtain a safe local direction enables the robot to avoid collision with obstacles.

6. Conclusion

In a dynamic unknown environment, generating the path sequentially is essential in order to cope with changes in the environment. However, since the knowledge of the environment is limited, there is a possibility that the robot may take an undesirable direction or enter a cul de sac. The results of the computer simulation show that this path planning method enables the mobile robot to recover the way to the goal and escape from a cul de sac without a prior knowledge of it.

It has been shown that a feasible path can always be obtained by continually sensing local environments. Therefore, the use of a short range sensor such a stereo vision system, is adequate. Although a long range sensor can acquire a global environment, obstacle avoidance requires primarily the information about the local environment. Therefore, this path planning method does not depend critically on the sensor range.

This local path planning method generates a path that is a compromise between the shortest global path and the safest local path. It is important to note that the optimal path is not always the shortest path. In a real word with dynamic unknown environments, such as in traffic or in a factory, safety is also a high priority while a mobile robot is trying to achieve the destination directly. The path planning method presented in this paper solves the problem in a situation where a compromise between the shortest path and the safest path is needed.

References:

[Amato and Wu, 1996] N. M. Amato, Y. Wu. A randomized roadmap method for path and manipulation planning. In IEEE *Int. Conf. Robotics & Automation*, pages 113-120,1996.

[Barraquand and Latombe, 1991] J. Barraquand, J.-C. Latombe, Robot Motion Planning: A distributed representation approach. *Int. J. Robotics Research*, 10(6):628-649, 1991.

[Barraquand *et al.*,1992] J. Barraquand, B. Langlois B, and J.-C. Latombe, Numerical Potential Field Techniques for Robot Path Planning. *IEEE Transactions on System, Man, Cybernetics*, 22(2), pp. 224-241, April 1992.

[Greenberg, 1998] M.D. Greenberg, *Advanced Engineering Mathematics*, 2nd ed., Upper Saddle River, N.J., Prentice Hall, 1998.

[Jarvis, 1983] R.A. Jarvis, Growing Polyhedral Obstacles for Planning Collision-Free Paths, *The Australian Computer Journal* , Vol. 15, No 3, pp. 560-570, August 1983.

[Jarvis, 1990] R.A. Jarvis, Mobile Robot Navigation, *3rd National Conference on Robotics*, pp 8-17, Melbourne, June 1990.

[Jarvis, 1993] R.A. Jarvis, Distance Transform based Path Planning for Robot Navigation. In Y. F. Zheng, editor, *Recent Trends in Mobile Robots*, volume 11 of World Scientific Series in Robotics and Automated Systems, chapter 1, pages 3-31. World Scientific, Singapore, 1993.

[La Valle and Kuffner, 1999] Steven M. La Valle, James J. Kuffner Jr., Randomized Kinodynamic Planning. In *IEEE Int. Conf. Robotics & Automation*, pages 473-479, May 1999.

[Lipton, 1997] A.J, Lipton Path Planning for an Unreliable Autonomus Mobile Robot in an Unknown Environment, *Australian Joint Conference on Artificial Intelligence*: Workshop on Robotics, Real World AI, Canberra, November 1995.

[Murray and Jennings, 1997] D. Murray, C. Jennings, Stereo Vision based Mapping for a Mobile Robot*, IEEE Conf. On Robotics and Automation*, May 1997.

[Nilsson, 1971] N.J. Nilsson, Problem-Solving Methods in Artificial Intelligence, McGraw-Hill, 1971.

[Zelinsky, 1994] A. Zelinsky, Using Path Transform to Guide the Search for Findpath in 2d, *International Journal of Robotics Research*, 13(4), pp. 315-325, August 1994.