Paper: jc12-4-3386:2008/6/9

An Integrated Algorithm for Autonomous Navigation of a Mobile Robot in an Unknown Environment

Lee Gim Hee* **and Marcelo H. Ang Jr.****

*CML, DSO National Laboratories
20 Science Park Drive, Singapore 118230
Email: lgimhee@dso.org.sg
**Department of Mechanical Engineering, National University of Singapore
9 Engineering Drive 1, Singapore 117576
Email: mpeangh@nus.edu.sg
[Received ; accepted]

Global path planning algorithms are good in planning an optimal path in a known environment, but would fail in an unknown environment and when reacting to dynamic and unforeseen obstacles. Conversely, local navigation algorithms perform well in reacting to dynamic and unforeseen obstacles but are susceptible to local minima failures. A hybrid integration of both the global path planning and local navigation algorithms would allow a mobile robot to find an optimal path and react to any dynamic and unforeseen obstacles during an operation. However, the hybrid method requires the robot to possess full or partial prior information of the environment for path planning and would fail in a totally unknown environment. The integrated algorithm proposed and implemented in this paper incorporates an autonomous exploration technique into the hybrid method. The algorithm gives a mobile robot the ability to plan an optimal path and does online collision avoidance in a totally unknown environment.

Keywords:

1. Introduction

Autonomous navigation of mobile robots is continuously gaining importance particularly in the military for surveillance as well as in industry for inspection and material handling tasks. Another emerging market with enormous potential is mobile entertainment robots.

Three competencies are identified for a mobile robot to navigate autonomously. First, the mobile robot must be able to stay away from hazards such as obstacles or operating conditions dangerous to itself and it must not pose any risk to humans in its vicinity. Second, the robot must possess the capability of planning its path so that it will always be able to travel from a starting point to a given goal. Third, the first two competencies should be accomplished with minimal human intervention even when the mobile robot is operating in an unknown environment.

Many algorithms such as the potential field, vector field histogram, roadmap, cell decomposition and navigation function [1–5, 7] were developed over the years for the autonomy of mobile robots. However, these algorithms are not capable of giving the mobile robot all the three competencies in one single framework. For example the potential field and the vector field histogram algorithms [5, 7], which are collectively known as the local navigation methods, give a mobile robot the capability of online collision avoidance but the algorithms fail to provide a mobile robot the capability of planning its own path. The other algorithms such as the roadmap, cell decomposition and navigation function [1–4], which are collectively known as the global path planning methods, give a mobile robot path planning capability only if the information of the environment is surveyed and provided by humans. Moreover, the global path planning methods are not capable of doing online collision avoidance.

Another group of algorithms suggest a combination of the local navigation and global path planning methods which aim to combine the advantages from both the local and global methods, and to also eliminate some of their weaknesses [1, 9]. An example is the hybrid navigation algorithm [9] which combines the navigation function with the potential field method. This algorithm is able to generate an optimal path in a partially or fully known environment and also does online collision avoidance. However, the hybrid navigation algorithm would fail in totally unknown environments.

This paper presents a robust navigation algorithm that gives the mobile robot all the three competencies in one single framework. This paper first discusses some of the existing algorithms and their limitations particularly the navigation functions, potential field method and the hybrid navigation algorithm. It also proposes our integrated algorithm that combines the frontier-based approach, which is an autonomous exploration technique, into the hybrid navigation method, to achieve all the three competencies for autonomous navigation. The integrated algorithm is simulated in a virtual environment and also implemented on the Nomad XR4000 mobile robot.

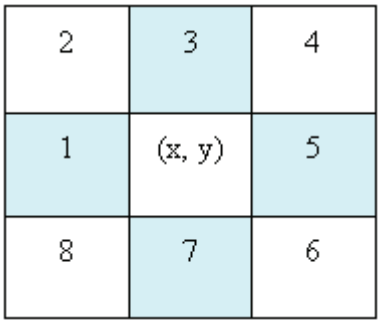

Fig. 1. The shaded cells are the 1-Neighbor of the cell (x, y) and the number shows the priority of the neighboring cells.

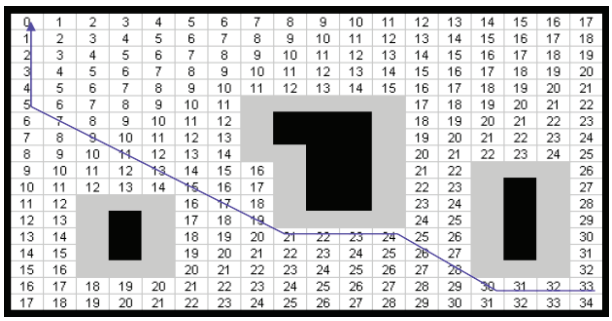

Fig. 2. Path generated by the navigation function.

2. Existing Works and Their Limitations

2.1. Global Path Planner – Navigation Functions

Global path planning means finding a safe path from the initial position of the robot to the goal in a known environment. There are many existing approaches that attempt to solve the problem. One of them is based on the navigation function where the path planning problem is transformed to a steepest descent problem from the initial position to the goal.

The algorithm based on a navigation function discretizes the workspace W of the robot into retangloid grid cells gC. In addition a fixed frame F_W is embedded in the workspace of the robot. The position of individual gC can thus be specified as cartesian coordinates with respect to F_W. Each gC is either free or occupied space. The subsets of gC in free and occupied space are denoted by gC_{free} and $gC_{occupied}$ respectively. The "wave-front expansion" [1] algorithm is used to compute the navigation function. The algorithm is practical, easy to implement and robust.

The path that is generated by the navigation function is computed in four steps. First, the coordinates of the free cells at the border of any obstacles in the workspace, denoted by gC_{border}, are extracted and stored into a First-In-First-Out (FIFO) list L_B. gC_{border} is defined as any gC_{free} on the tessellation of gC which has at least one $gC_{occupied}$ as its neighbor. Second, the unsafe regions are computed from the gC_{border}. The unsafe region includes all the cells that are less than α distance away from the border of the obstacle. The unsafe regions are then filled up with occupied cells. This will prevent the generated path from getting too close to the obstacles. Third, the navigation function values N are computed and applied to the rest of gC_{free} using the "wave-front expansion" algorithm. Fourth, a minimum length path can thus be generated following the steepest descent of N. The coordinates of all the grid cells that constitute the path are stored in a List L_p.

The pseudo code for the computation of the path generated by the navigation function is as follows:

Step 1-Extracting free cells at border of obstacle cells
1. begin
2. for every gC_{free} in the map do
3. if there exist a $gC_{occupied}$ neighbor then
4. begin
5. insert coordinates of gC_{free} to end of L_B;
6. end;
7. end;

Step 2-Computing the unsafe regions
1. begin
2. for every element i in L_B, where $i = 1, 2, \ldots$ do
3. for every neighbor of the cells in L_{Bi} do
4. if the neighbor is gC_{free} and $< \alpha$ distance then
5. begin
6. neighbor $\leftarrow$ occupied;
7. insert coordinate of neighbor at the end of L_B;
8. end;
9. end;

Step 3-Computing the navigation function values N
1. begin
2. for every gC_{free} in the map do
3. $N \leftarrow M$ (large number)
4. N for goal cell $\leftarrow 0$;
5. insert coordinates the of goal cell to the end of L_0;
6. for every element i in L_0, where $i = 1, 2, \ldots$ do
7. for every 1-neighbor cells of L_{0i} do
8. if $N = M$ then
9. begin
10. $N \leftarrow N$ of $L_{0i} + 1$;
11. insert coordinate of 1-neighbor at end of L_0;
12. end;
13. end;

Step 4 - Generating the optimal path L_p
1. begin
2. temp $\leftarrow$ cell coordinates of robot currently occupying;
3. insert temp to the end of L_p;
4. while N for temp $\neq 0$ do
5. temp $\leftarrow$ coordinate of min(neighboring cells of temp);
6. insert temp to the end of L_p;
7. end;

Figure 1 shows the definition of 1-Neighbor and the priority of the neighboring cells. The min function returns the coordinate of the neighboring cell of (x, y) that has the lowest N value. However, in cases where there are more than 1 neighboring cells having the same lowest N value, the cell with the highest priority will be chosen.

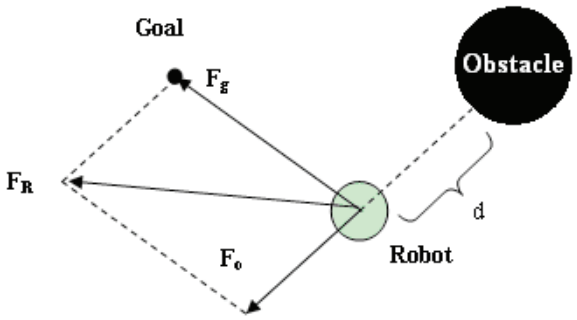

Fig. 3. Robot's motion influenced by potential field.

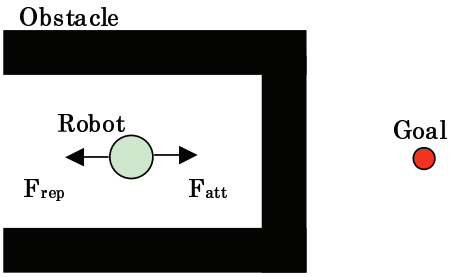

Fig. 4. An example of a local minimum from the potential field method.

Figure 2 shows a path generated by the navigation function. The black cells are the obstacles and the grey cells are the unsafe regions.

The navigation function requires the surrounding of the robot to be known and static. A continuous free path can thus be found in advance by analyzing the connectivity of the free space. A continuous free path always exists with the method. However any changes in the environment could invalidate this since N needs to be recomputed when there are changes in the environment. Furthermore, complete information of the environment is required to compute the navigation function. Hence, the navigation function is usually not suitable for navigation in an initially unknown environment with dynamic obstacles.

2.2. Local Navigation Method – Potential Field Method

The potential field method [7] is perhaps the most widely used algorithm for autonomous navigation of mobile robots. The robot is represented as a particle in the configuration space q moving under the influence of an artificial potential produced by the goal configuration q_{goal} and the scalar distance to the obstacles. Typically the goal generates an attractive potential,

$$U_g(q) = \frac{1}{2} K_g (q - q_g)^T (q - q_g) \quad \ldots\ldots \quad (1)$$

which pulls the robot towards the goal. The obstacles produce repulsive potential,

$$U_o(q) = \begin{cases} \frac{1}{2} K_o \left(\frac{1}{d} - \frac{1}{d_o} \right)^2 & \text{if } d < d_o \\ 0 & \text{otherwise} \end{cases} \quad \ldots \quad (2)$$

which pushes the robot away. K_g and K_o are the respective gains of the attractive and repulsive potential. d is the scalar distance between the robot and the obstacle. The repulsive potential will only have effect on the robot when its moves to a distance which is lesser than d_0. This implies that d_0 is the minimum safe distance from the obstacle that the robot tries to maintain.

The negated gradient of the potential field gives the artificial force acting on the robot.

$$F(q) = -\nabla U(q) \quad \ldots\ldots\ldots\ldots \quad (3)$$

Figure 3 shows the attractive force

$$F_g(q) = -K_g (q - q_g) \quad \ldots\ldots\ldots\ldots \quad (4)$$

that is generated from the goal and the repulsive force

$$F_o(q) = \begin{cases} K_o \left(\frac{1}{d} - \frac{1}{d_o} \right) \frac{1}{d^2} & \text{if } d < do \\ 0 & \text{otherwise} \end{cases} \quad \ldots \quad (5)$$

that is generated from the obstacle. F_R is the resultant of both the repulsive and attractive force. At every position, the direction of this force is considered as the most promising direction of motion for the robot.

$$F_R(q) = F_g(q) + F_o(q) \quad \ldots\ldots\ldots \quad (6)$$

The potential field method does not include an initial processing step aimed at capturing the connectivity of the free space in a concise representation. Instead, it integrates online sensory data into motion generating process, which is otherwise known as reactive control method. Hence a prior knowledge of the environment is not needed. At any instant in time, the path is determined based on the contents of the immediate surrounding of the robot. This allows the robot to avoid any dynamic obstacles in its vicinity.

The potential field method is basically a steepest descent optimization method. This renders the mobile robot to be susceptible to *local minima* [6]. **Fig. 4** shows an example of local minimum in the potential field method. It occurs when the attractive and the repulsive forces cancel out each other. The robot will be immobilized when it falls into a local minimum, and loses the capability to reach its goal.

2.3. Hybrid Method – Hybrid Navigation Algorithm

Figure 5 describes the hybrid navigation algorithm and its results [9]. The robot first computes the path joining its current position to the goal using the navigation function. It then places a circle with an empirical radius centered at its current position. The cell that corresponds to the intersection of the circle with the navigation function path is referred to as the attraction point. The attraction point is the cell with the lowest N if there is more than one intersection.

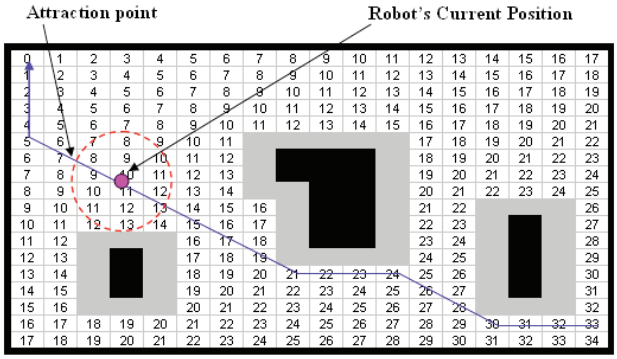

Fig. 5. Illustration of the hybrid navigation algorithm

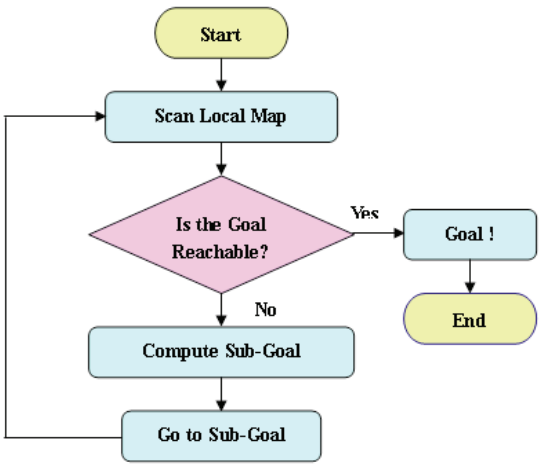

Fig. 6. The integrated algorithm.

The robot advances towards the attraction point using the potential field method and the circle moves along with the robot which will cause the attraction point to change. As a result, the robot is always chasing after a dynamic attraction point which will progress towards the goal along the local minima free navigation function path. The radius of the circle is made larger to intersect the navigation function path in cases where no intersections are found. The radius of the circle is reduced to smaller than the distance between the robot and its goal when this distance becomes smaller than the radius of the circle. This is to make sure that N of the next intersection will be smaller than the current N.

The hybrid navigation algorithm has the advantage of eliminating *local minima* failures and at the same time doing online collision avoidance. However, it requires the environment to be fully known for the search of an optimal navigation function path to the goal. The algorithm will fail in a fully unknown environment. It also does not possess any capability to re-plan the navigation function path during an operation. Therefore any major changes to the environment could cause failure in the algorithm.

3. Our Integrated Algorithm

Our integrated algorithm gives a mobile robot all the competencies needed to achieve autonomous navigation in one single framework. The algorithm modifies the frontier-based exploration method [8], which was originally used for map building, into a path planning algorithm. The modified frontier-based exploration method is then combined with the hybrid navigation algorithm into a single framework.

Figure 6 shows an overview of the integrated algorithm. The mobile robot will first build a local map of its surrounding. It then decides whether the goal is reachable. It will advance towards the goal if it is reachable, or compute another sub-goal if it is not reachable. The robot will build another local map, which will be added onto the previous local maps to form a larger map of the environment, after it has reached the sub-goal. This process goes on until the goal is reached. The following discusses the algorithm in detail.

3.1. Local Map Building

It is impossible for a mobile robot to plan a path that allows it to navigate safely from a starting position to a given goal in an unknown environment. It is therefore imperative for the robot to have the ability in acquiring the knowledge of its surrounding so that it would be possible for it to do path planning. A mobile robot can gain knowledge of an unknown world via sensors. Sensors provide local information about the environment in the vicinity of the robot. As the robot moves, information of the environment is updated in a process referred to as map building.

In many algorithms, the map building and path planning processes are decoupled. The robot has to acquire a map of its surrounding before it could plan a path. For example, in the frontier-based approached [8], a mobile robot has to complete an exploration mapping of the entire environment prior to the path planning process. This is inefficient because extra computation time and memory is needed for the exploration mapping of the whole environment. In many cases, a substantial part of the map is not utilized in the path planning process.

Our integrated algorithm does not attempt to build the map of the whole environment for path planning. Instead, the algorithm seeks to build local maps which are essential for path planning. Local map is the perceived view of a robot at its existing position. The mobile robot starts navigation by building a local map at its starting position. Next, it does further local mappings at the sub-goals (see Section 3.3 for the definition of sub-goals). The resulting map is therefore a concatenation of the local maps which the robot had built at the starting position and the respective sub-goals.

The occupancy grid mapping algorithm, which is described in detail in [10], is employed to build the local map. The occupancy map is chosen because it represents the environment as a tessellation of retangloid grid cells gC, which is similar to the navigation function that

(a) (b)

Fig. 7. (a) The goal is reachable because it is in the gC_{free} region (b) The goal is unreachable because it is in the $gC_{unknown}$ region.

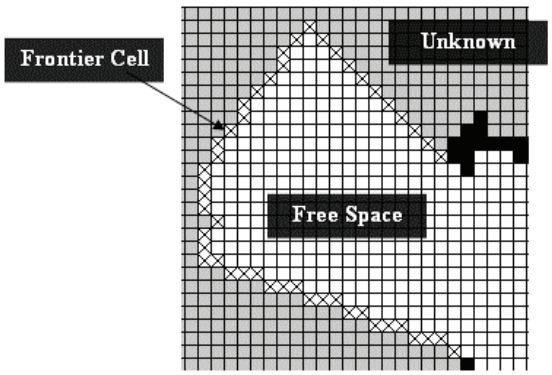

Fig. 8. A frontier region made up of a group of adjacent frontier cells.

we use in the integrated algorithm (described in Sections 3.3, 3.4 and 3.5). The grid cells are classified into three categories namely $gC_{unknown}$ which represents the unexplored cells, gC_{free} which represents the unoccupied cells and $gC_{occupied}$ which represents cells that are occupied by obstacles. The classification into either one of the three classes depends on the certainty values that each cell holds.

3.2. Goal Reachability

After the robot has finished a local map building process, it will determine whether the goal is reachable based on the map. The goal is reachable if it is in the gC_{free} region, and is not reachable if it is in the $gC_{unknown}$ region. **Fig. 7(a)** shows an example of a reachable goal in the gC_{free} region (white region) and **7(b)** shows an example of an unreachable goal in the $gC_{unknown}$ region (grey region).

3.3. Computation of Sub-Goal

The robot will compute a sub-goal if the goal is unreachable. This is done in three steps. First, compute the path that joins the robot's current position and the goal using the navigation function. The unknown cells are taken to be free space in the computation of the navigation function. Second, all the frontiers in the map are computed. The boundary of free space and unknown region is known as the frontier. A frontier is made up of a group of adjacent frontier cells. The frontier cell is defined as any gC_{free} cell on the map with at least two $gC_{unknown}$ cells as its immediate neighbor. The total number of frontier cells that make up a frontier must be larger than the size of the robot to make that frontier valid. **Fig. 8** shows an example of a valid frontier. Third, the frontier that intersects the navigation function path will be selected and its centroid chosen as the sub-goal. The sub-goal is an effective point in bringing the robot closest to its goal. The coordinates $({}^{j}x_c, {}^{j}y_c)$ of the centroid can be calculated using Eqs. (7) and (8).

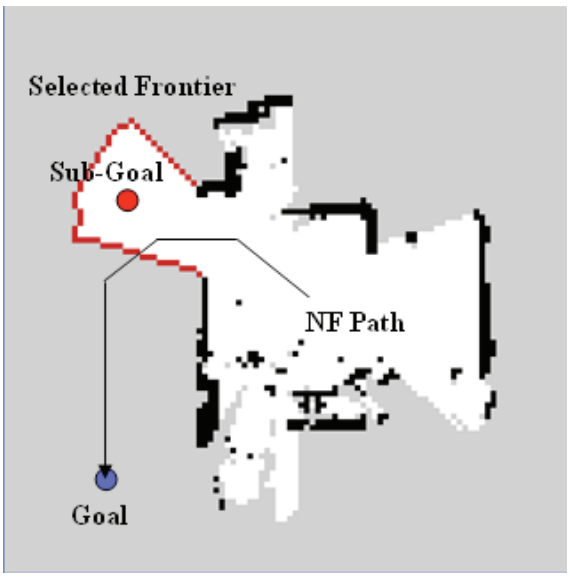

Fig. 9. Sub-goal is the centroid of the frontier that intersects the navigation function (NF) path.

$$ {}^{j}x_c = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i \quad \ldots\ldots\ldots\ldots\ldots\ldots (7) $$

$$ {}^{j}y_c = \frac{1}{n_j}\sum_{i=1}^{n_j} y_i \quad \ldots\ldots\ldots\ldots\ldots\ldots (8) $$

where,

n_j = number of frontier cells in the selected frontier j
x_i = x coordinate of i^{th} frontier cell in frontier j
y_i = y coordinate of i^{th} frontier cell in frontier j

Figure 9 shows an example of the computation of the sub-goal. The centroid of the frontier that intersects the navigation function path is selected as the sub-goal.

3.4. Reaching for the Sub-Goal

After computing the sub-goal, the robot will compute another navigation function path from its current position to the sub-goal. It will then move towards it using the

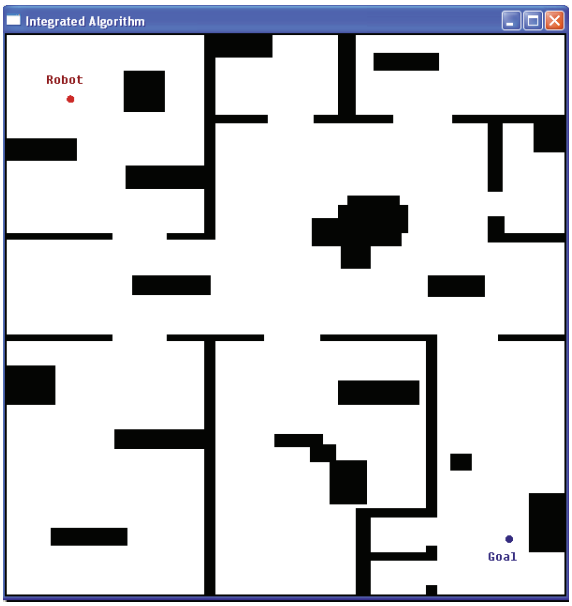

Fig. 10. Map of the simulation environment.

hybrid navigation algorithm. The robot will build another local map when it has reached the sub-goal. This local map is added onto the previous local map to form a larger map.

3.5. Reaching for the Goal

As mentioned earlier, the robot checks for the goal reachability after it finishes building a local map. The robot will move towards the goal using the hybrid navigation algorithm when it detects that the goal is in the gC_{free} region.

4. Simulation and Implementation Rusults

A simulation of the integrated algorithm was done in the environment shown in **Fig. 10**. The white regions represent gC_{free} cells and the black regions represent $gC_{occupied}$ cells. The robot was placed in an initial arbitrary position and commanded to move to a given goal. Note that the robot does not have any initial knowledge of the environment. The robot is equipped with range sensors all around but with limited range. **Fig. 11(a)** to **11(h)** are the snap-shots of the robot's perception of the environment as it advances towards the goal. The grey regions represent the $gC_{unknown}$ cells.

Figure 11(a) shows that the robot was initially in a completely unknown environment. The robot gains knowledge of its surrounding by building a local map as shown in **Fig. 11(b)**. In this case, the robot finds that the goal is not reachable and hence the navigation function path connecting the current position of the robot is computed. Next, the centroid of the frontier that intersects the navigation function path is taken as the sub-goal. The robot then computes another navigation function path

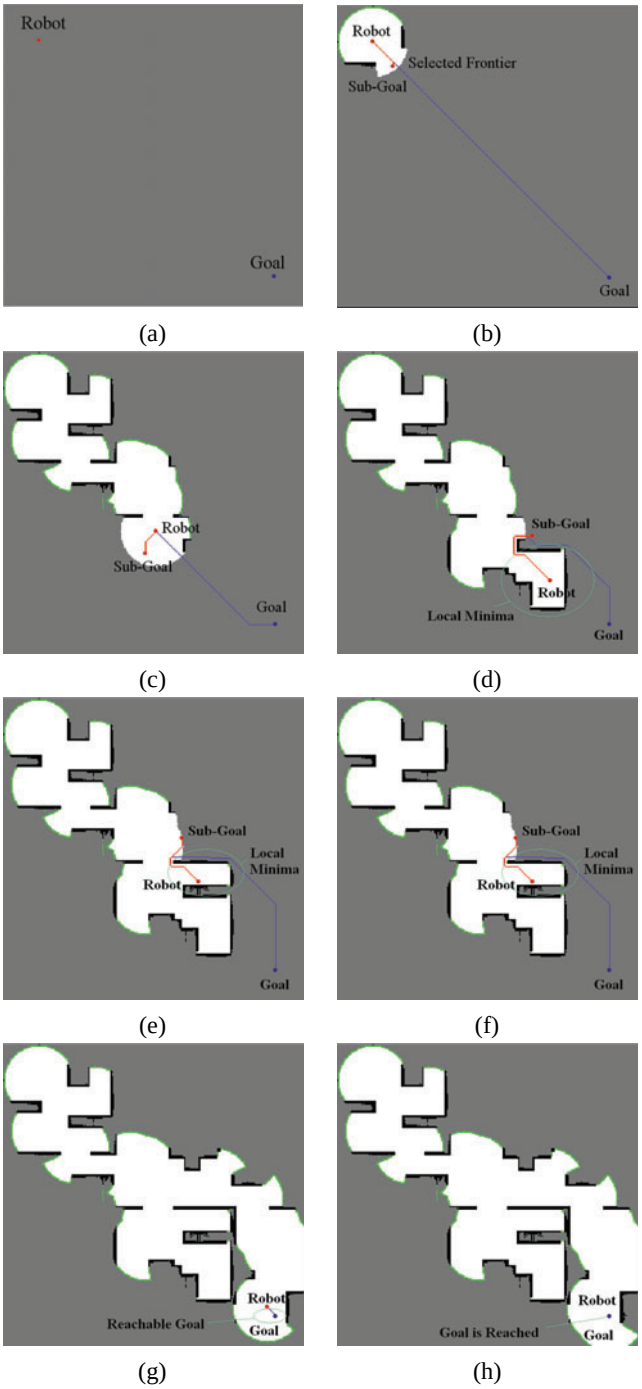

Fig. 11. Snap shots of the robot's perception in the simulated environment. The red line represents the navigation function path connecting the robot to the goal. The blue line represents the navigation function path connecting the robot to the sub-goal. Note that the frontier cells are represented by the green cells.

that connects its current position and the sub-goal. Finally, it advances to the sub-goal via the hybrid navigation method.

The robot builds another local map when it reaches the sub-goal. This local map is added onto the previous maps to form a larger map. **Fig. 11(c)** to **Fig. 11(h)** shows the gradual increment of the map by as the robot advances

Fig. 12. Snapshots of the trajectories and acquired maps of the Nomad XR4000 mobile robot during the implementation of the integrated algorithm. The blue lines represent the navigation function (NF) path from the robot to the respective sub-goals and red lines represent the actual trajectories taken by the robot.

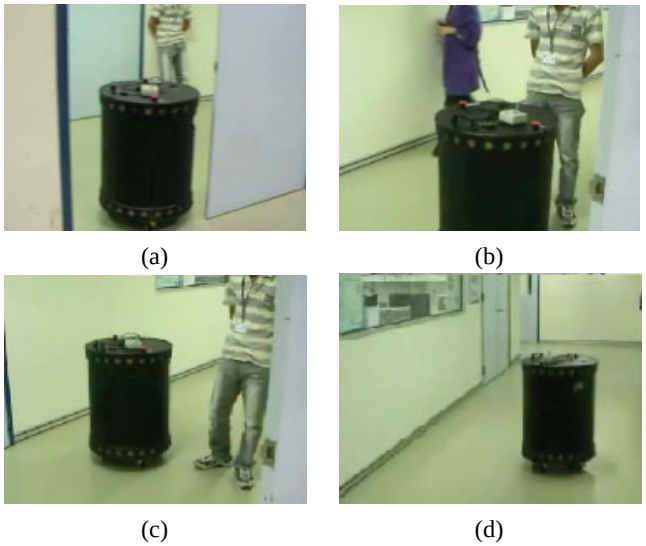

Fig. 13. Implementation of the integrated algorithm on the Nomad XR4000 robot (a) Robot moving through narrow doorway (b) Robot encounters an unforeseen obstacle (human) (c) Robot successfully avoids the unforeseen obstacle (d) Robot at the goal.

nearer to the goal. Note that the robot does not attempt to map the whole environment. Instead, it will map out only the necessary portions of the environment that is sufficient for it to find the goal.

Fig. 11(d) and **11(e)** show two possible cases of *local minima* if the robot navigates using only the potential field method. With the integrated algorithm, the robot finds a subgoal using the *local minima* free navigation function path. The robot is therefore able to escape from any possible local *local minimum*. The goal is within reachability in **Fig. 11(g)**. Here, the robot will not compute any sub-goal. It will compute a navigation function path that connects its current position with the goal and advances towards the goal using the hybrid navigation algorithm. Finally the algorithm terminates at **Fig. 11(h)** when the goal is reached.

Our integrated algorithm was successfully implemented on the Nomad XR4000 mobile robot. The robot is equipped with a SICK laser range finder that gives it an 180^o view of the environment. **Fig. 12(a)** to **12(f)** are snapshots of the trajectories and acquired maps of the Nomad XR4000 robot during the implementation of our integrated algorithm. The robot starts from an initially unknown environment and navigates towards the goal. **Fig. 12(a)** shows the first local map acquired by the robot. The robot does not find the goal within the gC_{free} region, hence the first sub-goal is determined and the robot advances towards it via the hybrid navigation method. **Fig. 12(b)** shows the trajectory of the robot as it moves through a narrow doorway towards the first sub-goal using the hybrid navigation algorithm. Notice that the circle for finding the attraction point becomes equal to the distance between the robot and its goal as the robot moves closer towards to goal. **Fig. 13(a)** shows the robot moving through the narrow doorway. The robot reaches its first sub-goal in **Fig. 12(c)** and builds another local map which is added onto the previous local map to form a larger map of the environment. The robot does not find the goal to be within the gC_{free} region and hence computed another sub-goal. **Fig. 12(c)** also shows a local minimum if the robot navigates to the goal from its initial location using only the potential field method. **Fig. 12(d)** shows the trajectory of the robot as it does an online collision avoidance with an unforeseen obstacle (human). **Fig. 13(b)** shows the encounter of the robot with an unforeseen obstacle (human) and **Fig. 13(c)** shows that the robot has successfully avoided the obstacle and advancing towards the second sub-goal. A third local map is added to the previous map in **Fig. 12(e)** and the robot finally finds the goal to be within the gC_{free} region. It then advances towards the goal and finally reaches the goal in **Fig. 12(f)**. **Fig. 13(d)** shows the robot at its goal.

5. Conclusion

The ability to navigate safely from one point to another is the most important part in the development of autonomous mobile robots. A vast number of techniques and methods have been introduced and implemented by many researchers. In this paper, a few methods are examined. They include the navigation function, the potential field method and the hybrid navigation algorithm. The contribution of this paper is the integrated algorithm. This method modifies the frontier-based exploration method, which was originally a map building technique, into a path planning algorithm. In the integrated algorithm, the navigation function and the potential field method are fused together with the modified frontier-based exploration method into one single framework. The algorithm is capable doing path planning in an unknown environment. The navigation function is used to plan the path and hence *local minima* free. The algorithm is also capable of avoiding any dynamic obstacles which were not included in the path planning.

References:

[1] J. -C. Latombe, "Robot Motion Planning," Boston Kluwer Academic Publishers, 1991.

[2] V. Akman, "Unobstructed Shortest Paths in Polyhedral Environments," Lecture Notes in Computer Science, Springer-Verlag, 1987. C.

[3] C. 'Dnliang and C. Yap, "Retraction: A New Approach to Motion Planning," Proceeding of the 15th ACM Symposium on the Theory of Computing, Boston, pp. 207-220, 1983.

[4] J. Schwartz and M. Sharir, "On the Piano Movers' Problem: The Case if a Two-Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers," Communications on Pure and Applied Mathematics, Vol.36, pp. 345-398, 1983.

[5] J. Borenstien and Y. Koren, "The Vector Field Histogram – Fast Obstacle Avoidance For Mobile Robots," IEEE Journal of Robotics and Automation Vol.7, No.3, pp. 278-288, June 1991.

[6] J. Borenstien and Y. Koren, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," Proc. of the IEEE Conf. on Robotics and Automation, Sacramento, California, pp.1398-1404, April 7-12, 1991.

[7] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," Int. Journal of Robotic Research, Vol.5, No.1, pp. 90-98, 1986.

[8] B. Yamauchi, "A Frontier-Based Approach for Autonomous Exploration," Proc. of the IEEE Int. Symposium on the Computational Intelligence in Robotics and Automation, Monterey, CA, pp. 146-151, 1997.

[9] L. C. Wang, L. S. Yong, and M. H. Ang Jr., "Hybrid of Global Path Planning and Local Navigation implemented on a Mobile Robot in Indoor Environment," Proc. of the 2002 IEEE Int. Symposium on Intelligent Control, Vancouver Canada, pp. 821-826, 2002.

[10] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics," The MIT Press, Cambridge Massachusetts, London, England, 2005.

Name:
Lee Gim Hee

Affiliation:
A member of technical staff, with the Defence Science Organization (DSO) National Laboratories, Singapore

Address:
20 Science Park Drive, Singapore 118230

Brief Biographical History:
July 2005 received B.Eng Degree (1st Class Honours) in Mechanical Engineering from the National University of Singapore (NUS)
July 2005- July 2007 continued to pursue his M. Eng degree in Mechanicalf Engineering
A member of technical staff, with the Defence Science Organization (DSO) National Laboratories, Singapore

Main Works:
• the development of autonomous mobile robots, particularly in the development of algorithms for path planning, map building and localisation. development of algorithms for autonomous multi-robot systems in establishing and maintaining wireless sensor networks.

Name:
Marcelo H. Ang, Jr.

Affiliation:
Department of Mechanical Engineering, National University of Singapore

Address:
9 Engineering Drive 1, Singapore 117576

Brief Biographical History:
1981 received the B.S. degrees (*Cum Laude*) in Mechanical Engineering and Industrial Management Engineering from the De La Salle University, Manila, Philippines
1985 received the M.S. degree in Mechanical Engineering from the University of Hawaii at Manoa, Honolulu, Hawaii
1986 received the M.S. degrees in Electrical Engineering from the University of Rochester, Rochester, New York
1988 received the Ph.D. degrees in Electrical Engineering from the University of Rochester, Rochester, New York
1989 joined the Department of Mechanical Engineering of the National University of Singapore, where he is currently an Associate Professor.
heading the Technical Training Division of Intel's Assembly and Test Facility in the Philippines
Assistant Professor of Electrical Engineering at the University of Rochester, New York
the East West Center in Hawaii and at the Massachusetts Institute of Technology

Main Works:
• His research interests span the areas of robotics, mechatronics, automation, computer control, and applications of intelligent systems methodologies

Membership in Academic Societies:
• the Singapore Robot ic Games as its founding chairman