6

Non-Rigid Obstacle Avoidance for Mobile Robots

Junghee Park and Jeong S. Choi
Korea Military Academy
Korea

1. Introduction

In mobile robotics, obstacle avoidance problem has been studied as a classical issue. In practical applications, a mobile robot should move to a goal in the environment where obstacles coexist. It is necessary for a mobile robot to safely arrive at its goal without being collided with the obstacles. A variety of obstacle avoidance algorithms have been developed for a long time. These algorithms have tried to resolve the collisions with different kinds of obstacles. This chapter classifies the types of obstacles and presents characteristics of each type. We focus on the new obstacle type that has been less studied before, and apply representative avoidance strategies to solve this new type of obstacle avoidance problem. The results of different strategies would be compared and evaluated in this chapter.

2. Classification of obstacles

This section summarizes previous points of view on obstacles, and presents a new angle on obstacles for advanced robot navigation. A robotic system often suffers from severe damage as result of a physical collision with obstacles. Thus, there is no doubt that robots should have a perfect competency to avoid all kinds of obstacles. In robotics, an obstacle is generally defined as a physical object that is in the way or that makes it hard for robot to move freely in a space, and thereby classified into *static and moving obstacles*: the former are sometimes subdivided into stationary (or fixed) and movable ones of which position can be changed by force from a robot. Also, moving robots are often separated from moving obstacles, since their motion can be changed by themselves.
On the basis of this definition, various strategies have been presented to resolve anticipated physical collisions systematically. Since the majority of strategies focus on generating immediate reaction to environments, obstacles are considered static for an instant although they are moving in most previous studies. Recently, some studies formalized way to explicitly consider current velocity of moving obstacles for coping with real collisions caused by the static assumption. This chapter views obstacles from a different standpoint and introduces a concept of *non-rigid obstacles*, which differs from the conformable obstacle in that the concept covers various physical entities as well as the shape (or boundary) of obstacle. Fig. 1 shows these views on obstacles and details are discussed below.

2.1 Rigid obstacles

Early studies on robot motion assumed that a robot is the only moving object in the workspace and all obstacles are fixed and distributed in a workspace (Latombe, 1991). Under the

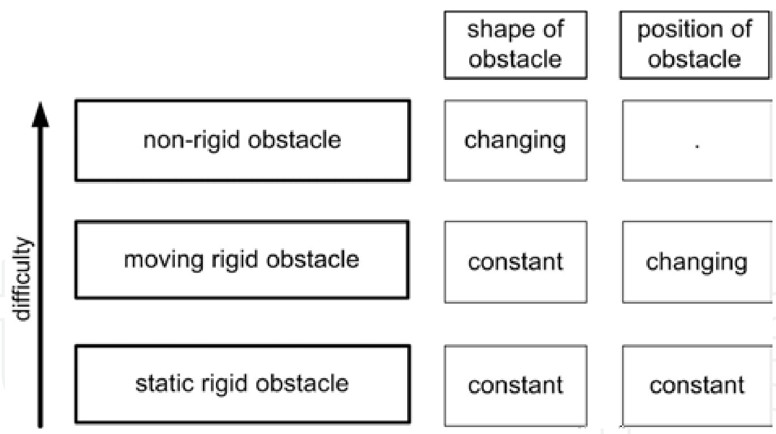


Fig. 1. Classification of obstacles

assumption, obstacle avoidance is automatically solved if we solve the global path planning problem in which the robot is assumed to have complete knowledge of the environment and generates a full path from its start to goal. The famous examples include the following approaches: probabilistic road map (Kavraki et al., 1996), cell decomposition (Boissonnat & Yvinec, 1998), grid cells (Hee & Jr., 2009), and rapidly-exploring randomizing tree (LaValle, 1998). To realize the approaches, several systematic tools for path generation were developed: visibility graph, Voronoi diagram, random sampling method, gradient method, node-edge graph, and mathematical programming (Ge & McCarthy, 1990). The final objective of the approaches is to find the global optimal (or shortest) one in the path candidates given by an optimization algorithm, such as Dijkstra and A* method.

Some studies relaxed the assumption of complete knowledge about environments to take into consideration the case that there is no device to provide the complete knowledge. Accordingly, the studies focused on following a detour (or local path) to avoid a detected static obstacle, instead of finding the shortest path to its goal. As a result, the obstacle avoidance is separated from the global path planning, which implies that it can be performed by on-line planning. The simplest strategy for the static obstacle avoidance with a limited sensing range may be the bug algorithm in which the robot is controlled to simply follow the contour of an obstacle in the robot's way (Lumelsky & Skewis, 1990). As with the bug algorithm, other avoidance strategies involving such obstacles create an immediate reaction to nearby environments for robot navigation. The popular methods include vector field histogram (Borenstein & Koren, 1991), nearness diagram (Minguez & Montano, 2004), fuzzy rules (Lee & Wang, 1994), force (or potential) fields (Wang et al., 2006), and dynamic window (Fox et al., 1997). In fact, these tools have been widely used for even the following moving obstacle avoidance problem by virtue of their simplicity in formulation and low operational speed of robots which permits the assumption that moving obstacles can be seen as static ones for a short time.

Moving obstacle avoidance differs from the static case in that an irreversible dimension - time - should be added into problem formulation. Like the static obstacle avoidance problem, the solution for the moving case can be computed globally or locally according to the assumption of the complete knowledge of environments. The configuration-time (CT) space presented in fig. 2 and 3 is a well-known tool for global problem solving. The CT space is a three-dimensional space in which a time axis is added into a xy-plane where a robot is moving. Static obstacles are represented as a straight pillar as shown in fig. 2 when they are represented with respect to CT space. On the other hands, if a moving obstacle is modeled as a circle, its representation with respect to CT space is an oblique cylinder as shown in fig. 3,

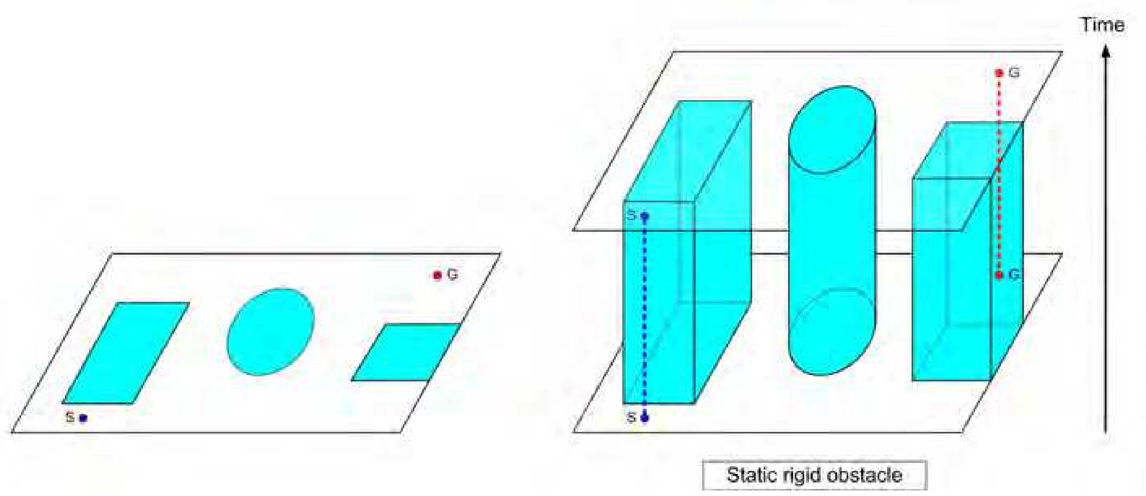


Fig. 2. A configuration-time space of static rigid obstacles

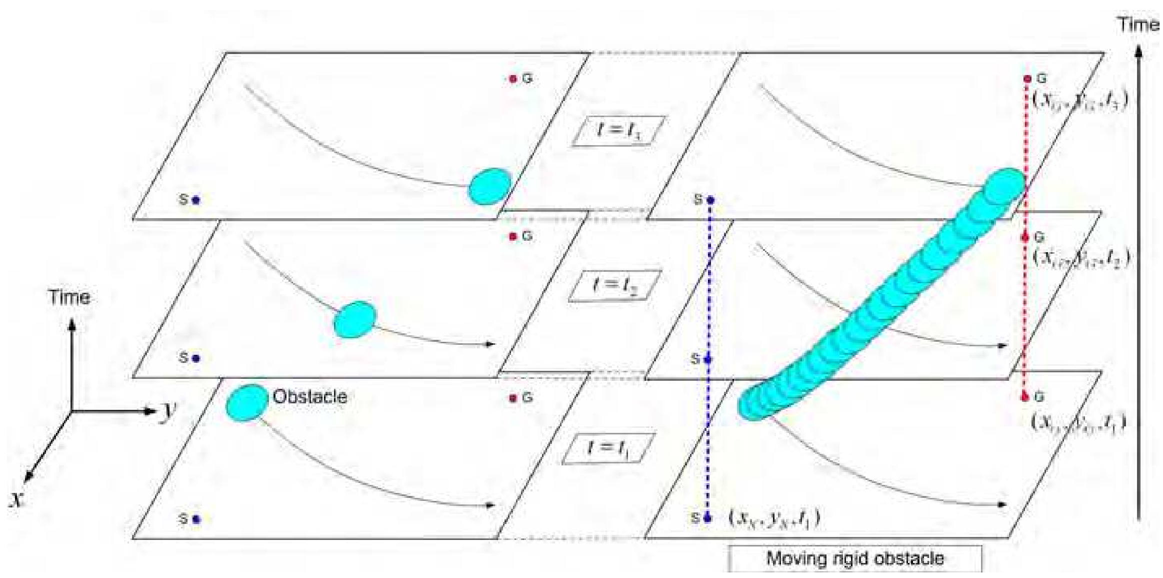


Fig. 3. A configuration-time space of moving rigid obstacles

because the position of the obstacle is varied with time along its path. These representations indicate that the problem of global moving obstacle avoidance is reduced to finding a 3-D geometric path obeying two conditions: safety and time efficiency. Sampling, combinatorial, and velocity-tuning methods have been introduced to solve the geometric problem (Lavalle, 2006).

Local moving obstacle avoidance is split into two groups, according to whether or not obstacles can be seen static for a short time, as mentioned above. Several recent papers argued that the problem can be made more realistic by explicitly considering the robot's velocity and acceleration, and included the constraints to problem formulation. The collision representation tools based on the consideration include the velocity obstacle (VO) (Fiorini & Shiller, 1993), inevitable collision states (ICS) (Martinez-Gomez & Fraichard, 2009) and the triangular collision object (TCO) (Choi et al., 2010). The major difference is that object shapes representing potential collisions are determined by obstacle's velocity, unlike the tools for the static obstacle avoidance problem. In the benchmarking (Martinez-Gomez & Fraichard, 2009), it was shown that the method using the VO is superior to the methods for static obstacle avoidance because it takes into account objects' future behavior, and suggested that the future motion of moving obstacles should explicitly been deal with. So far, we have briefly reviewed conventional rigid obstacle types and avoidance strategies which are popularly used in robotics. Now turn our attention to a new type, non-rigid obstacle in the following section.

2.2 Non-rigid obstacles

In this section, we explain a new obstacle type, non-rigid obstacle. This concept emerges as a consequence of the need for non-physical collisions, such as observation. More specifically, robots only need to have considered a single physical element, obstacle's boundary, for safe navigation to date, but they now have to consider other elements, such as agent's visibility, for smart navigation. Thus, to accommodate such a need, we here define a new term, *non-rigid obstacle*, which is an abstract object of which boundary is changeable and determined by an element involving an obstacle.

We explain this concept, non-rigid obstacle, through a moving observer with omnidirectional view. In a infiltration mission, the robot should covertly trespass on the environment without being detected by sentries. As a reverse example, in a security mission, the most important thing is that robots are not observed (or detected) by the intruder until it reaches the intruder (Park et al., 2010). In this way, the robot can covertly capture or besiege the intruder who would try to run away from the robot when detecting the approaching robot. It causes that robots can closely approach the intruder without being detected, and the intruder cannot recognize himself being besieged until the robots construct a perfect siege. In these cases, the robot should avoid the intruder's field of view during the navigation, rather than its physical boundary, which is called *stealth navigation*. This signifies that the proposed concept of non-rigid obstacle could be exceptionally helpful in planning an advanced robot motion.

The visible area determined by the field is a changeable element, especially in a cluttered indoor environment. So, we can regard the area as an abstract non-rigid obstacle that robots should avoid. This kind of obstacle can be generated by any element the obstacle has, which is a remarkable extension of the conventional concept on obstacles. Fig. 4 illustrates an example of the changes of visible areas with time traveled (or z-axis) in a CT space. As shown in the figure, the shape (or boundary) of visible area is greatly varied with the observer's motion which is a set of position parameterized by time. For this reason, it is needed to develop a new strategy to avoid the highly changeable obstacle, which will be discussed in detail in Section 3. If the robot succeeds in the avoidance with a plausible strategy, it gives a completely new value to a robotic system.

3. Avoidance strategies for non-rigid obstacle avoidance

As we mentioned in section 2, researches on obstacle avoidance problem have been focused on dealing with rigid (static or moving) obstacles. On the other hand, researches on avoidance strategies of non-rigid obstacles scarcely exist except grid-based methods (Marzouqi & Jarvis, 2006; Teng et al., 1993) a few decades ago. These methods were based on the grid map which approximates the environment to spatially sampled space. In this grid map, all possible movements of the robot were taken into consideration to find the optimal(fastest) motion of the robot to the goal. This brought about high burden of computation time. In addition, the arrival time of the planned motion and the computation time highly depended on the grid size.

Since then, any other strategies have not been developed for non-rigid obstacle avoidance. Even though other methods have not been suggested, we can apply rigid obstacle avoidance

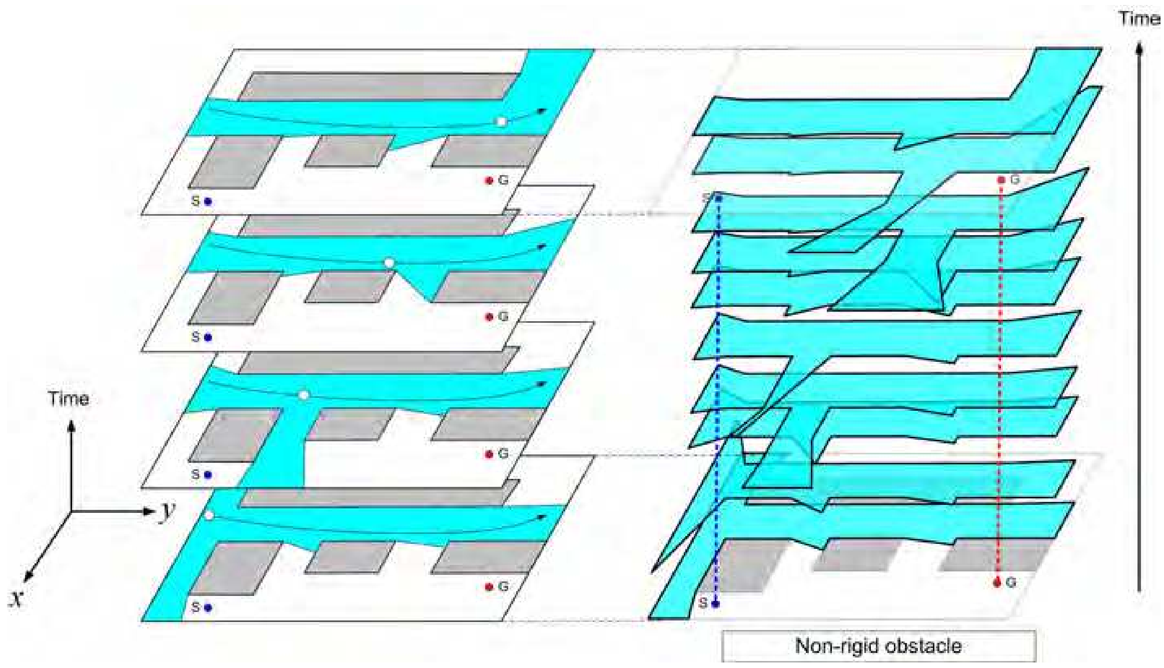


Fig. 4. A configuration-time space of non-rigid obstacles

algorithms to the problem of non-rigid obstacle avoidance. On a big scale, these algorithms can be classified into four categories: reactive method, grid-based method, randomization method, and path-velocity decomposed method. We will explain this classification of algorithms and how they can be applied to non-rigid obstacle avoidance.

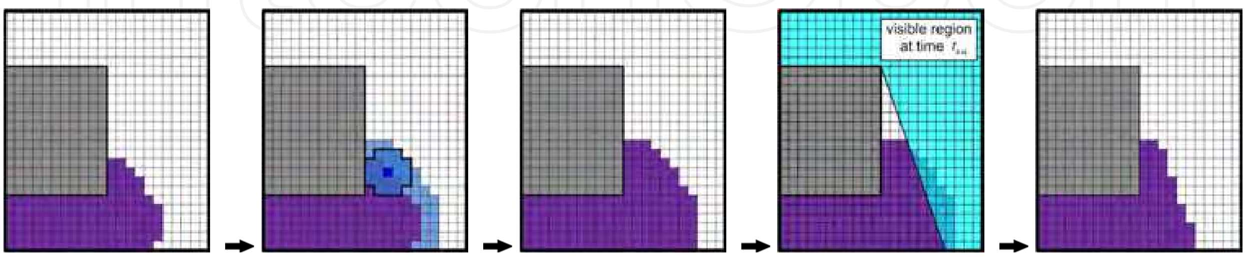
3.1 Reactive method

Reactive methods utilize nearby environment information for robot navigation, i.e., the robot has a sensing capability with a limited coverage. In the case where the robot is sent to a unknown region for exploration or reconnaissance, it is reasonable to assume that information of the environment is not known beforehand. In addition, if the robot has no communication connection with other sensors on the environment, then it has to rely on the nearby information acquired from its own sensor. In this case, the robot should find a suitable motion by making full use of given information in order to attain two purposes: avoiding obstacles and moving to its goal. In other words, it can be widely used when the information of the environment is limited and not deterministic, whereas other methods, which will be explained in following sections, are based on the assumption that the information of environment is known in advance.

This reactive method can also be called short-term planning because it plans a few next steps of robot motion. The planner focuses on creation of immediate respondence of the robot in order to avoid obstacles and move to the goal. On a rough view, the reactive method yields instantaneous moving direction by synthesizing attractive attribute to the goal and repulsive attribute from obstacles. In non-rigid obstacle avoidance, the robot is planned to be repulsive from non-rigid obstacles. As mentioned in section 2.1, there have been many researches to find reactive respondence of the robot, such as vector field histogram(VFH) (Borenstein & Koren, 1991), force field (Wang et al., 2006), dynamic window (Fox et al., 1997), and behavior-based robotics (Arkin, 1985). In this chapter, we implemented one of representative reactive methods, vector field histogram, for comparison. It constructs a polar histogram, which contains nearby environment information, and finds suitable detour direction towards its goal.

3.2 Grid-based method: (Teng et al., 1993)

Grid-based methods use spatially sampled environment map as mentioned earlier. This algorithm has a merit that all possible robot motion can be considered in the grid map at the expense of increasing computation time. Therefore it can guarantee optimal solution in the grid map. That is, this method can have the completeness even though it is limited to the given grid map which depends on the shape and the size of grids.



(a) Safely reachable region at time t_k

(b) Each boundary point is dilated in amount of the movable distance of the robot in one time step

(c) Reachable region at time t_{k+1}, which is dilated from the safely reachable region at time t_k

(d) The dilated region should be trimmed in order to avoid visibility from the observer

(e) Safely reachable region at time t_{k+1}

Fig. 5. Construction procedure of safely reachable region(purple) in grid-based method

Differently from other methods, this grid-based algorithm was designated for solving this non-rigid obstacle avoidance and introduced in (Teng et al., 1993). In this algorithm, the concept of the *safely reachable region* was introduced and used. It represents the region that the robot is able to arrive without being detected by the observer at each time step. The planner sequentially constructs safely reachable region of each time step until the last safely reachable region contains the final goal. Fig. 5 represents the procedure for obtaining safely reachable region from a previous one. Let the purple region of fig. 5(a) be the safely reachable region at time t_k. This region is dilated to the reachable region from it one time step later, which is shown in fig. 5(c). This dilated region is trimmed by the visible area of the observer at time t_{k+1}. Then it becomes the region that the robot can arrive without being detected by the observer at time t_{k+1}, which is shown in fig. 5(e). In this way, all possible safely reachable grid points were calculated at each time, and the shortest time for arrival is yielded.

3.3 Randomization method: modified RRT

In order to avoid a high computational burden of brute-force technique, randomization methods have been developed in robot motion planning. In these methods, free configurations are randomly selected and discriminated whether they can be via configurations that the robot is able to safely pass by. These kinds of algorithms have been developed for a long time, such as probabilistic road map(RPM) (Kavraki et al., 1996) and rapidly-exploring randomizing tree(RRT) (LaValle, 1998). These randomization methods could be usefully adopted for solving non-rigid obstacle avoidance problem in order to prevent burgeoning computational burden. In this chapter, we have modified RRT algorithm in order to cope with non-rigid obstacles which change its shape while time passes. The modified RRT algorithm for solving non-rigid obstacle avoidance problem is described in table 1.

$\mathbf{s}_{init} \leftarrow (t_{init}, \mathbf{q}_{init})$;
G.init($\mathbf{s}_{init}$);
while
    $\mathbf{q}_{rand} \leftarrow$ RAND_FREE_CONF();
    $\mathbf{s}_{near} = (t_{near}, \mathbf{q}_{near}) \leftarrow$ NEAREST_VERTEX($\mathbf{q}_{rand}$,G);
    $v_{rand} \leftarrow$ RAND_VEL();
    if CHECK_SAFETY($\mathbf{s}_{near}$,$\mathbf{q}_{rand}$,v_{rand}) then
        $t_{new} \leftarrow$ NEW_TIME($\mathbf{s}_{near}$,$\mathbf{q}_{rand}$,v_{rand});
        $\mathbf{s}_{new} \leftarrow (t_{new}, \mathbf{q}_{rand})$;
        G.add_vertex($\mathbf{s}_{new}$);
        G.add_edge($\mathbf{s}_{near}$,$\mathbf{s}_{new}$);
        $v_{g\_rand} \leftarrow$ RAND_VEL();
        if CHECK_SAFETY($\mathbf{s}_{new}$,$\mathbf{q}_{goal}$,v_{g_rand}) then
            $t_{arrival} \leftarrow$ NEW_TIME($\mathbf{s}_{new}$,$\mathbf{q}_{goal}$,v_{g_rand});
            $\mathbf{s}_{goal} \leftarrow (t_{arrival}, \mathbf{q}_{goal})$;
            G.add_vertex($\mathbf{s}_{goal}$);
            G.add_edge($\mathbf{s}_{new}$,$\mathbf{s}_{goal}$);
            break;
        end if;
    end if;
end while;

Table 1. Construction procedure of rapidly-exploring randomizing tree G for non-rigid obstacle avoidance

In the algorithm, the tree G is constructed with randomly selected vertexes. Each vertex $\mathbf{s}$ consists of time t and configuration $\mathbf{q}$, i.e., $\mathbf{s} = (t, \mathbf{q})$. In non-rigid obstacle avoidance problem, the time t when the robot passes by the configuration $\mathbf{q}$ should be coupled with $\mathbf{q}$ as a vertex. Thus, the time t also should be selected randomly. In this algorithm, t is calculated as the time at which the robot is able to arrive at $\mathbf{q}$ with the randomly selected velocity. Each randomly selected vertex becomes component of the tree G, and the tree G is continuously extended until new selected vertex is able to be linked with the goal vertex. After construction of tree G is completed, the connection from the start vertex to goal vertex is determined as the robot's movement.

In table 1, RAND_FREE_CONF() returns randomly selected free configuration. NEAREST_VERTEX($\mathbf{q}$,G) returns the vertex $\mathbf{s}_r = (t_r, \mathbf{q}_r)$ in G, which has the minimum distance from $\mathbf{q}$ to $\mathbf{q}_r$. RAND_VEL() is the function that yield randomly selected velocity v which should satisfy the robot's physical constraints, i.e., $0 \leq v \leq v_{max}$. In this procedure, the probability distribution function $f_V(v)$ for selecting v was designed as follows because it is beneficial for the robot to move as fast as possible.

$$f_V(v) = \frac{2v}{v_{max}^2}, \;\; 0 \leq v \leq v_{max}$$

CHECK_SAFETY($\mathbf{s}_1$,$\mathbf{q}_2$,v) returns whether it is possible that the robot is able to safely move from the configuration $\mathbf{q}_1$ of vertex $\mathbf{s}_1$ to $\mathbf{q}_2$ with the velocity v or not. NEW_TIME($\mathbf{s}_1$,$\mathbf{q}_2$,v) returns the time t_r taken for the robot to move from $\mathbf{q}_1$ of $\mathbf{s}_1$ to $\mathbf{q}_2$ with the speed v. In this chapter, the configuration $\mathbf{q}$ of the robot is two dimensional point $\mathbf{p}$ because the robot is assumed to be holonomic and has circular shape. In this case, returned time t_r is calculated as follows.

$$t_r = t_1 + \frac{\| \mathbf{p}_2 - \mathbf{p}_1 \|}{v}$$

3.4 Path-velocity decomposed method

Lastly, we can consider the path-velocity decomposed(PVD) method. This concept was firstly introduced in (Lee & Lee, 1987) and has been developed for resolving collisions of multiple robots in (Akella & Hurchinson, 2002; Lavalle & Hurchinson, 1998). This method decomposes the robot motion into the path, which is a geometric specification of a curve in the configuration space, and the velocity profile, which is a series of velocities according to the time following the path. In other words, the PVD method firstly plans the path, and adjusts the velocity of the robot following this path. It reduces computation burden by not guaranteeing the safety condition when planning the path at the cost of optimality. This safety condition is considered only in planning of velocity profile. If we predefine the geometric road map (topological graph) on the environment, we can easily plan the robot's path on the road map and find the velocity profile that guarantees safely movement of the robot following the path. This method used for solving moving rigid obstacle avoidance also can be applied to non-rigid obstacle avoidance problem without difficulty.

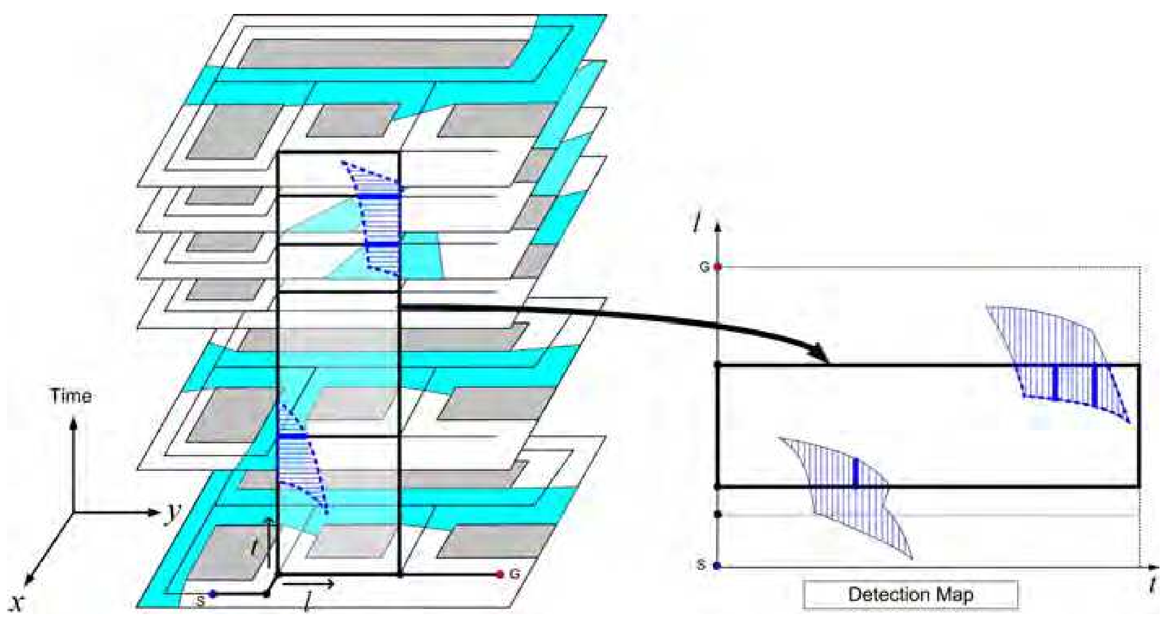


Fig. 6. Construction of the detection map

(Park et al., 2010) has proposed the concept of detection map. The detection map is a two-dimensional(time and point on the path) map that represents whether each time and point on the path is detected by the observer or not. It depends on the path chosen on the road map. Fig. 6 shows the construction of the detection map on the predetermined path. When the detected intervals on the path are accumulated according to the time, the detection map is constructed. In the detection map, the shaped regions represents the time and points on the path that are detected by the observer, which is called detection region. We can draw the robot's trajectory on the detection map without intersecting these detection regions in order to make the robot safely move to its goal. After the detection map is constructed, the safely velocity profile can be found in a shorter time because moving direction is restricted to one dimension.

The planning method of the velocity profile can be selected in a various way. (Park et al., 2010) applied the grid-based method to this velocity profile planning. The concept in (Teng et al., 1993), which constructs safely reachable region by dilation and trimming in order to find the optimal solution on the grid map, is adopted to solve the similar problem with the one-dimensional moving direction. The points on path is sampled with uniform interval and examined whether it is safely reachable by the robot for each discretized time step. This grid-based method also finds the velocity profile that has the shortest arrival time. We note that this optimality is valid only when the predetermined path could not be modified by the planner.

The randomization method could be proposed in velocity profile planning. Without considering all possible safe movement of the robot, we can find the robot's safe trajectory by randomly selecting via points on the detection map. In this chapter, we applied the modified RRT method explained in section 3.3 t velocity profile planning. In this application, the one-dimensional value is randomly selected as via point that the robot would pass. The RRT would be constructed on the detection map without intersecting detection regions. The detection map makes it easy to discriminate whether tree link is safe or not.

The method of selecting path on the road map also could be suitably adopted. (Park et al., 2010) has selected the path that has the shortest moving distance. If the safe velocity profile did not exist on the selected path, the next shortest path was examined in consecutive order. However, it often led to a situation that not a few path had to be examined because this method completely ignored the safety condition on the path planning step. We proposed the method to select the path highly possible to have safe trajectory of the robot. We designed the cost of each link on the road map as follows. Let $E_{i,j}$ be the link between node i and node j on the road map.

$$cost(E_{i,j}) = length(E_{i,j}) * (detection\_ratio(E_{i,j}))$$

$$\text{where } detection\_ratio(E_{i,j}) = \frac{area\ of\ detection\ region\ on\ the\ detection\ map\ of\ E_{i,j}}{area\ of\ whole\ detection\ map\ of\ E_{i,j}}$$

Compared to the method which considered only path distance, this cost function makes the path planner to consider not only distance but also safely condition. If we select the path that has the least cost on the road map, this path would be relatively less detected by the observer. If a safe velocity profile is not found on the selected path, then the other path which has next least cost could be found and examined again.

4. Simulation results

We have conducted a computer simulation on the environment in fig. 7. In this simulation, the visible area of the observer is considered as a example of the non-rigid obstacle. The robot is planned to move stealthy from the observer. The observer is patrolling on the environment following a specific path with the speed of 2.0m/s. It is assumed that the robot has a maximum speed of 4.0m/s and it can change its speed and its moving direction instantaneously. We assumed that the information of the environment and the non-rigid obstacle is known to the

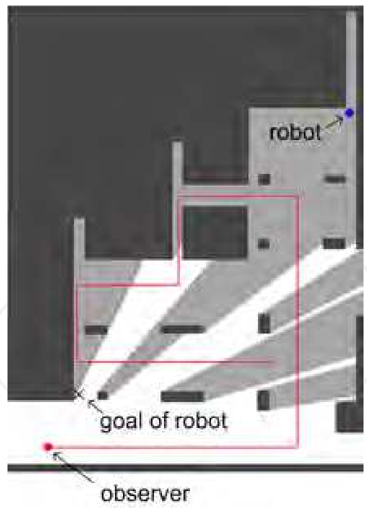


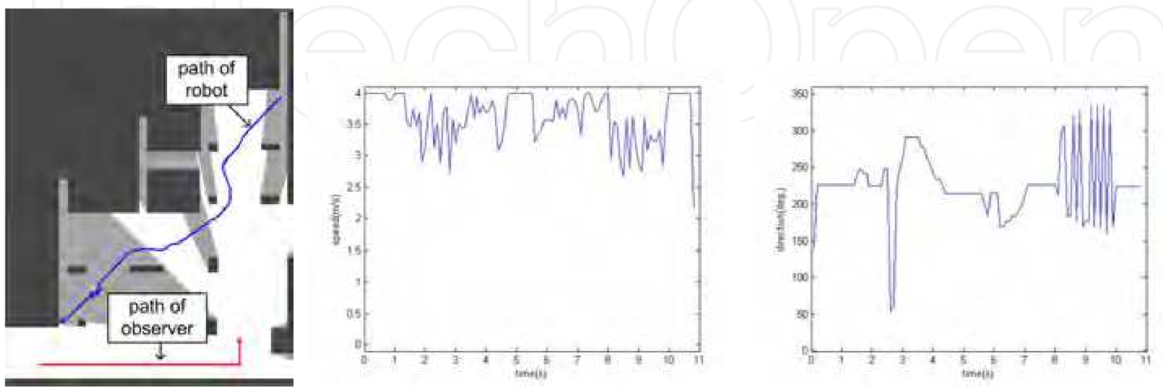
Fig. 7. Simulation environment (27.5m x 36.5m) and the patrol path of the observer

robot. This computer simulation is conducted on a commercial computer with 3.34GHz CPU and 3.00GB RAM.
Table 2 shows the simulation results according to algorithms from reactive method to path-velocity decomposed method. In table 2, the *exposure rate* means a ratio of the time interval that the robot is detected by the observer to the whole arrival time. In addition, we define *safety* as the distance between the robot and the point which is contained in the visible area of the observer and is the closest to the robot. When the robot is contained in the visible area of the observer, i.e., the robot is detected by the observer, the safety value is set to zero. In table 2, the average value of safety during the operation time is revealed. Also, the computation times of each method are shown in table in order to analyze practical implementation feasibility of each method.

	Reactive (VFH)	Grid-based (Teng et al., 1993)	Randomization (Modified RRT)	Path-velocity decomposed Grid-based (Park et al., 2010)	Path-velocity decomposed Randomization (Modified RRT)
Arrival time	10.8s	10.4s	20.1s	16.2s	16.6s
Exposure rate	3.0%	0.0%	0.0%	0.0%	0.0%
Average safety	2.97m	2.88m	3.14m	3.32m	3.40m
Computation time	0.01s	33.67s	17.15s	path planning: 3.70s 0.01s	path planning: 3.70s 0.01s

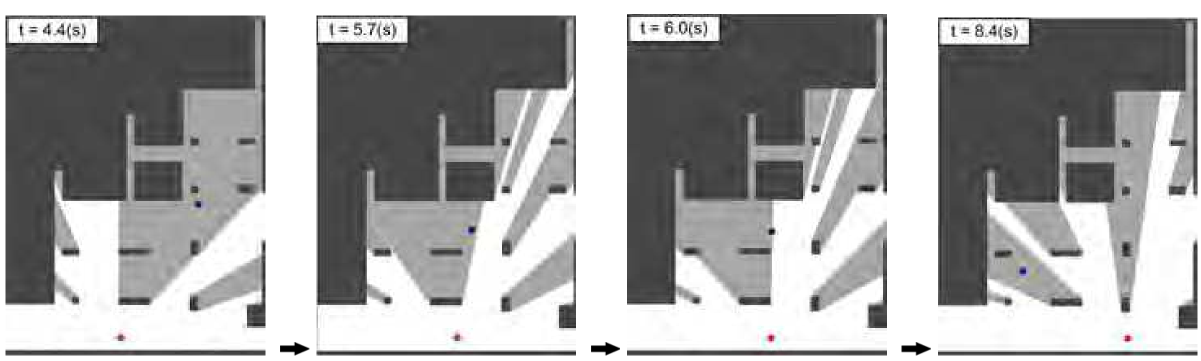
Table 2. Simulation results

In table 2, the exposure rate had non-zero value only when the reactive method was applied. This is because the reactive method does not utilize whole information of visible areas of the observer. It assumes that the robot is able to sense only nearby information of environment and visible area. In this simulation, the robot's sensing coverage is assumed to be omnidirectional with 3.0m radius. Therefore, it is possible that the robot is placed in the situation that it is too late for avoiding the visible area that emerges suddenly near by. At the cost of this dangerous situation, the reactive method does require a negligible amount of the computation time. More detailed results are shown in fig. 8 and fig. 9. As we can see in fig. 8(b), and 8(c), the robot's speed and direction were changed extremely for a whole operation time. In fig. 9, the robot detects the non-rigid obstacle in the rear of itself at time 5.7(s), nevertheless, it could not avoid the obstacle due to the lack of speed, and it collided with the obstacle at time 6.0(s). We note that this chapter assumes that non-rigid obstacle is also detected by the sensor of the robot in order to apply the reactive method for rigid obstacle avoidance to non-rigid obstacle avoidance problem. In real application, some non-rigid obstacle such as observer's field of view could not be detected by the sensor. As this result shows, the robot has a difficulty in stealth navigation when it has a limited sensing capability of the environment.



(a) path (b) speed (c) direction

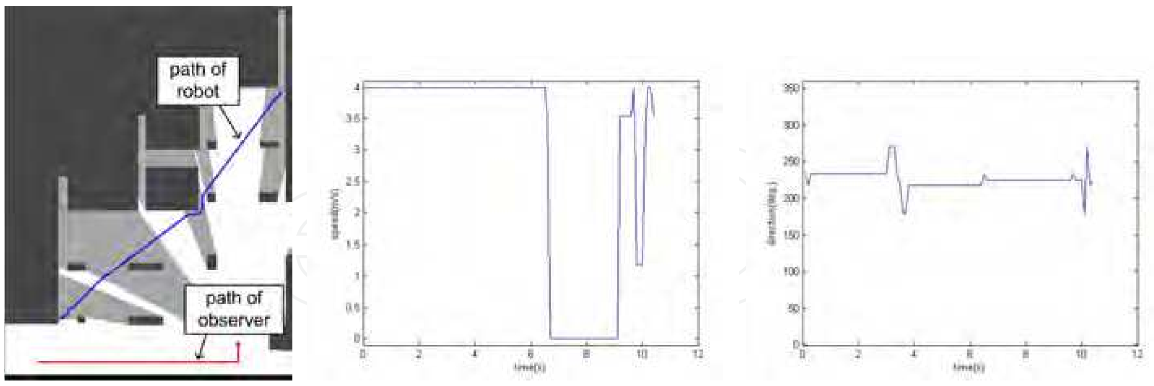
Fig. 8. The reactive method (VFH) results



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

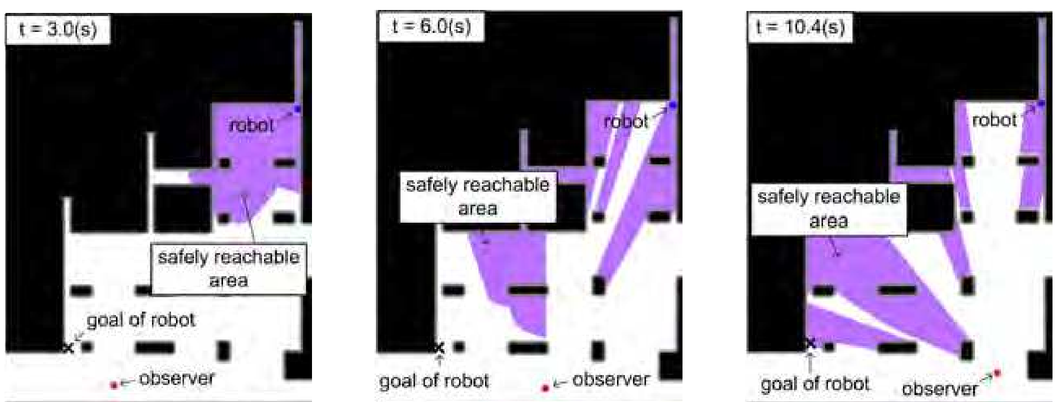
Fig. 9. The movement of observer(red) and robot(blue) in the reactive method result

From the results in table 2, we can find that the arrival time was the least when the grid-based method was applied. This is because the grid-based method considers all possible motions of the robot and finds the fastest motion among them, i.e., it guarantees the optimality and completeness. However, these are valid only on given grid map. That is, if the grid map is set in a different way, the robot's fastest motion and the arrival time would become different. Since it examines all grid cells whether they are possibly reached safely by the robot at each time, it needs high computational burden. In table 2, the computation time of the grid-based method is higher than that of any other methods. More detailed results are shown in fig. 10. The speed and direction of the robot were not changed extremely. Between time 6.8(s) and 9.1(s), the robot remained stationary in the shaded region until the obstacle was away from itself. The path in fig. 10(a) shows that the robot was able to almost directly traverse the environment to the goal without being detected by the observer. The performance in path length and arrival time is the best in the grid-based method. Fig. 11 shows safely reachable regions changing in accordance with the passed time.



(a) path (b) speed (c) direction

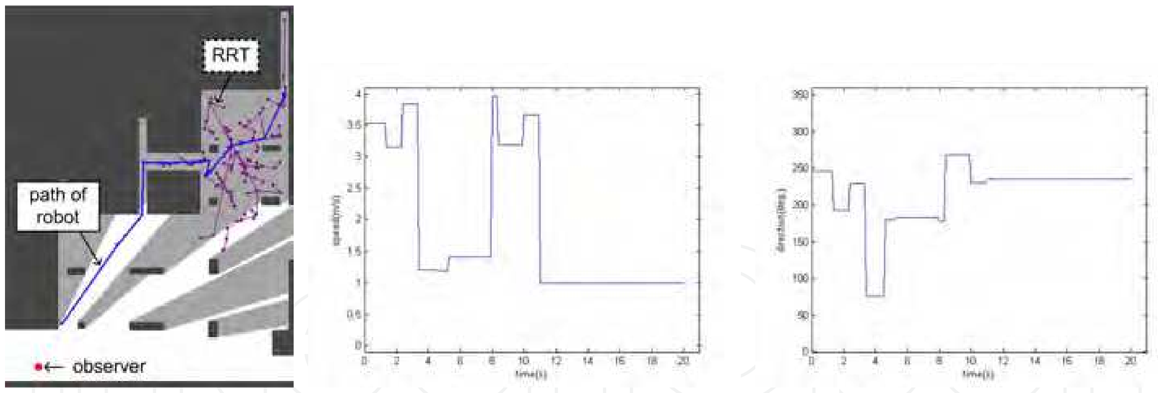
Fig. 10. The grid-based method results(grid size is $64cm^2$/cell)



(a) safely reachable region at time t=3.0(s) (b) safely reachable region at time t=6.0(s) (c) safely reachable region at time t=10.4(s)

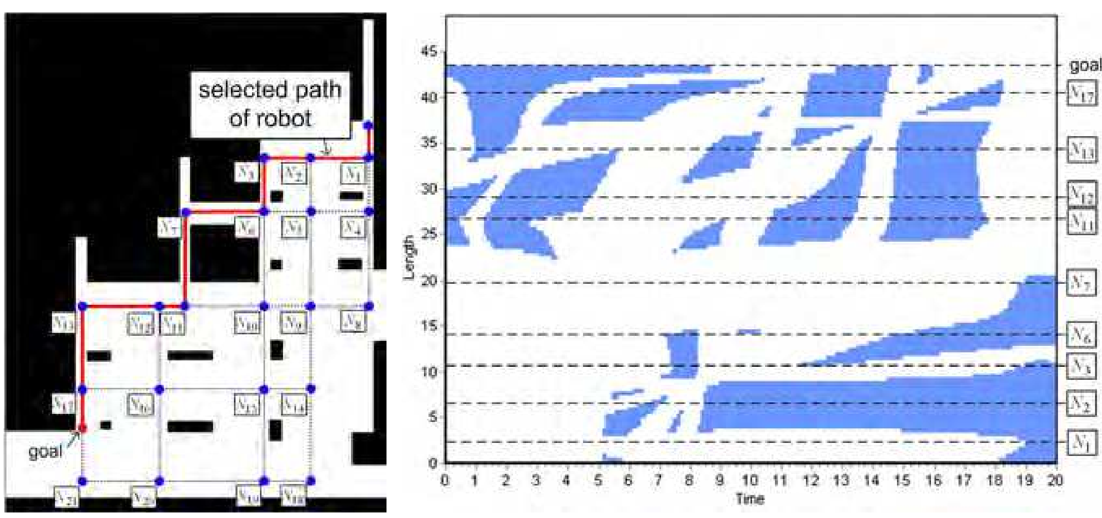
Fig. 11. Safely reachable regions of the grid-based method in fig. 10

The result of randomization method in table 2 has the lowest performance in the arrival time. This result is not deterministic and can be changed in each trial. There is possibility that relatively better performance can be yielded, but it could not be guaranteed. This method does not purpose on greatly reducing the arrival time, but it concerns a practical computation time. In table 2, modified RRT algorithm needed not a low computation time. This is because it needs relatively much time to check that the tree link candidate is safe for the movement of the robot. In order to reduce the computation time in the randomized method, it is necessary to develop a specialized randomization method to solve the non-rigid obstacle avoidance problem. Fig. 12 shows the result of modified RRT method. In fig. 12(a) and 12(c), the path of the robot was planned as zigzagged shape because via points are selected randomly. As shown in fig. 12(b), the segments between via points have respectively uniform speeds which are also selected randomly.
Lastly, we implemented the PVD method in the same environment. The road map of the environment was predefined, and the path with the least cost on the road map was selected as shown in fig. 13(a). We can see that the path with the least cost is selected as series of



(a) path (b) speed (c) direction

Fig. 12. The randomized method (modified RRT) results

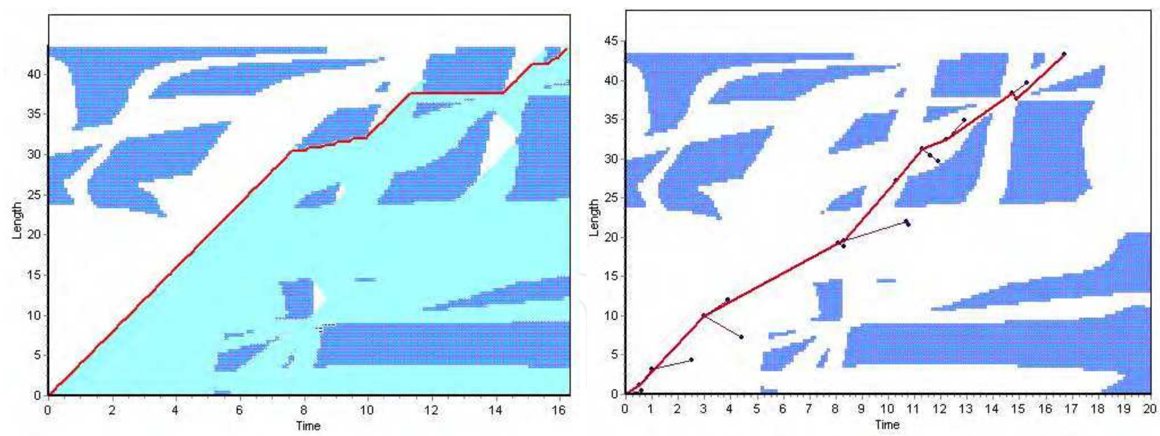


(a) road map on the environment and selected path (b) detection map on the selected path

Fig. 13. The road map and the detection map

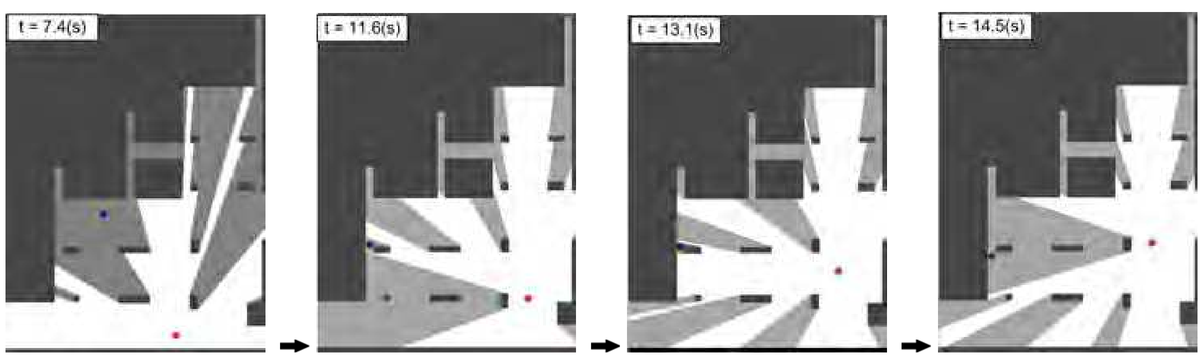
edges that are in the vicinity of walls, because they have relatively small detection ratios. The detection map corresponding to the selected path was constructed as shown in fig. 13(b). In the detection map, the colored regions, which are detection regions, are the set of time and point that is detected by the observer on the determined path. The trajectory of the robot following this path could be planned on the detection map without traversing detection regions.

As shown in table 2, the arrival time of the grid-based result in PVD method was larger than that of the grid-based method. This is because the PVD method restricts the path to be selected on the predefined roadmap, not among whole possible movement. Therefore, the PVD method has the lower performance than the grid-based method even though it also finds the optimal solution on the determined path. However, its computation time is much smaller than the grid-based method because it reduces the dimension of the grid map. This planned result is shown in fig. 14(a). The movement of the observer and the robot in this result is shown in fig. 15. In fig. 15(b) and 15(c), we can see that the robot remains stationary because



(a) planned trajectory of the grid-based method (b) planned trajectory of the randomization method

Fig. 14. The trajectory results of two detail velocity profile planning algorithm in the path-velocity decomposed(PVD) method



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

Fig. 15. The movement of observer(red) and robot(blue) in the grid-based result of path-velocity decomposed method

the observer's field of view obstructs the front and the rear of the robot. In fig. 15(d), the robot started to move after the observer's field of view is away from itself.
In the case of the randomization result in PVD method in table 2, it not only needs much lower computation time but also yields the better performance than the modified RRT methods. The randomization methods cannot guarantee the optimality, and its performance is on a case by case basis. Therefore, it is possible that the case restricting path in PVD method has the shorter arrival time than the case having whole opportunity of path selection. This planned result is shown in fig. 14(b).

5. Conclusion

In this chapter, we classified the type of obstacles into three categories: static rigid obstacles, moving rigid obstacles, and non-rigid obstacles. However, most researches of obstacle avoidance have tried to solve the rigid obstacles. In reality, the non-rigid obstacle avoidance problem is often emerged and needed to be solved, such as stealth navigation. We adopted the representative algorithms of classic rigid obstacle avoidance, and applied to non-rigid obstacle avoidance problem. We examined how the inherent characteristics of each algorithms are revealed in non-rigid obstacle avoidance and evaluated the performance. The result in the reactive method informed us that limited information of the environment and obstacle greatly hinders the robot from avoiding non-rigid obstacles. The grid-based method could find the optimal safe motion of the robot at the cost of high computation burden. The randomization method gave priority to reducing computational burden over improving the performance. Finally, the path-velocity decomposed(PVD) method greatly reduced computation burden by abandoning the diversity of path selection. This method does not cause great loss in arrival time. This method makes the robot avoid the obstacle by adjusting its speed. In velocity profile planning, several detail methods, such as the grid-based method and the randomization method, can be applied with practical computation time. When complete information of the environment and the obstacle is given, this PVD method could be efficiently used.

6. References

Akella, S. & Hurchinson, S. (2002). Coordinating the motions of multiple robots with specified trajectories, *Proceedings of IEEE International Conference of Robotics and Automation*, New Orleans, LA, pp. 3337–3344.

Arkin, R. C. (1985). *Behavior-based Robotics*, MIT Press.

Boissonnat, J. D. & Yvinec, M. (1998). *Algorithm Geometry*, Cambridge University Press.

Borenstein, J. & Koren, Y. (1991). The vector field histogram - fast obstacle avoidance for mobile robot, *Journal of Robotics and Automation* Vol. 7(No. 3): 278–288.

Choi, J. S., Eoh, G., Kim, J., Yoon, Y., Park, J. & Lee, B. H. (2010). Analytic collision anticipation technology considering agents' future behavior, *Proceedings of IEEE International Conference on Robotics and Systems*, pp. 1656–1661.

Fiorini, P. & Shiller, Z. (1993). Motion planning in dynamic environments using the relative velocity paradigm, *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. Vol. 1, pp. 560–566.

Fox, D., Burgard, W. & Thrun, S. (1997). The dynamic window approach to collision avoidance, *IEEE Robotics and Automation Magazine* Vol. 4(No. 1): 23–33.

Ge, Q. J. & McCarthy, J. M. (1990). An algebraic formulation of configuration-space obstacles for special robots, *Proceedings of International Conference on Robotics and Automation*, pp. 1542–1547.

Hee, L. & Jr., M. H. A. (2009). Mobile robots navigation, mapping, and localization part i, *Encyclopedia of Artificial Intelligence* pp. 1072–1079.

Kavraki, L. E., Svestka, P., Latombe, J. C. & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration space, *IEEE Transactions on Robotics and Automation* Vol. 12(No. 4): 566–580.

Latombe, J. C. (1991). *Robot Motion Planning*, Kluwer academic publishers.

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning, *Technical report 98-11, Computer Science Department, Iowa State University* .

Lavalle, S. M. (2006). *Planning Algorithms*, Cambridge University Press.

Lavalle, S. M. & Hurchinson, S. A. (1998). Optimal motion planning for multiple robots having independent goals, *IEEE Transactions on Robotics and Automation* Vol. 14(No. 6): 912–925.

Lee, B. H. & Lee, C. S. G. (1987). Collision-free motion planning of two robots, *IEEE Transactions on Systems Man and Cybernetics* Vol. 17(No. 1): 21–31.

Lee, P. S. & Wang, L. L. (1994). Collision avoidance by fuzzy logic for agv navigation, *Journal of Robotic Systems* Vol. 11(No. 8): 743–760.

Lumelsky, V. & Skewis, T. (1990). Incorporating range sensing in the robot navigation function, *IEEE Transactions on Systems Man and Cybernetics* Vol. 20: 1058–1068.

Martinez-Gomez, L. & Fraichard, T. (2009). Collision avoidance in dynamic environments: an ics-based solution and its comparative evaluation, *Proceedings of IEEE International Conference on Robotics and Automation*, Kobe, pp. 100–105.

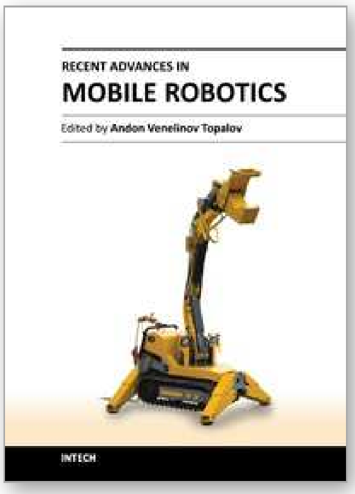
Marzouqi, M. S. & Jarvis, R. A. (2006). New visibility-based path-planning approach for covert robotics navigation, *IEEE Transactions on Systems Man and Cybernetics* Vol. 24(No. 6): 759–773.

Minguez, J. & Montano, L. (2004). Nearness diagram navigation: collision avoidance in troublesome scenarios, *IEEE Transactions on Robots and Automation* Vol. 20(No. 1): 45–59.

Park, J., Choi, J. S., Kim, J., Ji, S.-H. & Lee, B. H. (2010). Long-term stealth navigation in a security zone where the movement of the invader is monitored, *International Journal of Control, Automation and Systems* Vol. 8(No. 3): 604–614.

Teng, Y. A., Dementhon, D. & Davis, L. S. (1993). Stealth terrain navigation, *IEEE Transactions on Systems Man and Cybernetics* Vol. 23(No. 1): 96–113.

Wang, D., Liu, D. & Dissanayake, G. (2006). A variable speed force field method for multi-robot collaboration, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 2697–2702.



Recent Advances in Mobile Robotics
Edited by Dr. Andon Topalov

ISBN 978-953-307-909-7
Hard cover, 452 pages
Publisher InTech
Published online 14, December, 2011
Published in print edition December, 2011

Mobile robots are the focus of a great deal of current research in robotics. Mobile robotics is a young, multidisciplinary field involving knowledge from many areas, including electrical, electronic and mechanical engineering, computer, cognitive and social sciences. Being engaged in the design of automated systems, it lies at the intersection of artificial intelligence, computational vision, and robotics. Thanks to the numerous researchers sharing their goals, visions and results within the community, mobile robotics is becoming a very rich and stimulating area. The book Recent Advances in Mobile Robotics addresses the topic by integrating contributions from many researchers around the globe. It emphasizes the computational methods of programming mobile robots, rather than the methods of constructing the hardware. Its content reflects different complementary aspects of theory and practice, which have recently taken place. We believe that it will serve as a valuable handbook to those who work in research and development of mobile robots.