A Study on Static and Dynamic Path Planning for Autonomous Vehicle

Jae-Kook Lim
jklim@mn.waseda.ac.jp
Graduate School of Science and Engineering,
Waseda University

Eric Y. Shim
eshim@isl.melco.co.jp
Information Technology R&D Center
Mitsubishi Electric Corp

Teruo Takahashi
taka12@mn.waseda.ac.jp
Institute of Asia-Pacific Studies,
Waseda University

Abstract

The purpose of this research is to examine the applicability of real-time control for AVs and efficiency improvements by reducing calculation iterations. This paper describes a path-decision method of a mobile Autonomous Vehicle, AV, to search paths in an unknown environment by using a fixed obstacle information. Assuming multiple AVs are operated for their own tasks, there are chances of conflict resolution such as sharing the same path which could lead to the risk of a collision. This research represents some ways of negotiating the conflict resolution by generating the cooperative actions among AVs.

1 Introduction

It is required to have higher flexibility for the next generation manufacturing system to respond to frequent changes of product or manufacturing process instantaneously. Intelligent material handling system is a good example of the conveyance system which can connect processing, assembling, and inspection phases of the higher flexible manufacturing system.

One of the most important factor of intelligent material handling system is Autonomous Vehicle (AV). When an AV is working for a given job, it is required to have ability which can plan for its own job. For the case of a simple environment, human operator can give the job after carefully consider the environment. However, for the case of a complicated environment or unknown environment, it is impossible for human operator to order AV to complete its job. In this case, AV should recognize the environment and accomplish the path planning itself. Optimization of a path planning is very important. However, it is very difficult to optimize path planning in a complicated environment, also the cost of calculation is expanded.

Since we have to consider that the movement environment changes dynamically according to the movement of several mobile robots, it is very difficult to optimize path planning in advance. Therefore, the dynamic path planning was emphasized in Kato's research[1].

Kant et.al. have proposed an algorithm which can predict other moving robot's path by knowing other's moving pattern in advance, and recalculate its own path to avoid collision[2]. Other than that, similar researches have done by Inoue group and Asama et.al[3] - [4]. However, these researches could not solve some problems such as sudden increase of calculation amount when the number of robot is increased.

Also, other researches proposed other algorithms such as using a communicate means to avoid collision, and using traffic rules and information of opposite robot to avoid collision[5] - [9].

Lim et.al. also have proposed an algorithm which can construct path planning automatically using free space structure based on formal cell-decomposition method[10]. Also, Lim et.al. have proposed another algorithm which can solve many conflict resolutions using negotiation method[11] - [12].

In this research, we are proposing two path generation algorithms, static path generation and dynamic path generation. Static path generation is an algorithm which can find shortest path using the cell-decomposition method. Dynamic path generation is an algorithm which can find temporary moving path to avoid any conflict resolution among the AVs. We also propose an algorithm that can effectively reconstruct collision avoidance activity using dynamic negotiation means such as task of movement, announce time for specific task start, and movement loss rate according to the dynamically changed environment.

2 Static Path Generation

In this section, an algorithm which generates moving path by building gates on the entrance and on the exit of a cell which was structured on the free space using cell-decomposition method[10], as shown in figure 1 is introduced. The gate of the free space indicates the situation of a cell either if an AV can pass cell-zone or not by

IEEE International Workshop on Robot and Human Communication
0-7803-4076-0/97/ $10.00 © 1997 IEEE

closing/opening the gate.

Static path generation algorithm is composed by the following three phases.

Phase 1 : cell-decomposition and build gates
 Step 1 : cell-decomposition
 Step 2 : build gates
Phase 2 : establish network
 Step 1 : establish connectivity graph among cells
 Step 2 : build moving path network
Phase 3 : find the shortest path
 Step 1 : evaluate every moving path
 Step 2 : calculate the distance of each candidate path
 Step 3 : find the shortest path

In the first phase, as shown in figure 1, slice free space by connecting a line from an obstacle's edge to another obstacle or wall of free space. Build gates on this sliced line, and give an identification number for each gate constructed. In the second phase, establish connectivity graph among cells to build a moving path network from the start position(position where a job is) to the end position(goal). Then, evaluate every moving path according to the evaluation factors. There are two big evaluation factors which are the width of moving path to see if AV can pass the path, and existence of a temporary obstacle such as broken AV or AV blocking the way by working on its own job on the moving path. After the evaluation, we can obtain candidate path for the shortest path. Calculate the distance of each candidate path[11]. After this, we can obtain the shortest path among the all candidate paths.

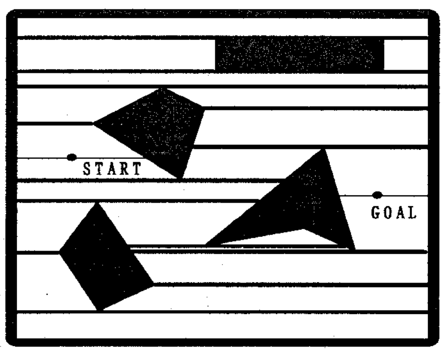

Figure 1 Slicing of Free Space by Cell-Decomposition

2.1 Management of Environmental Information Using Gate Concept

Figure 2 shows various type of gates which have been established on the cell's entrance and exit. The number of gates of free space is determined by obstacle's appearance and number of obstacles.

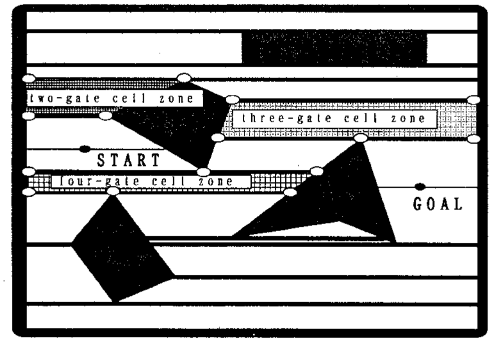

Figure 2 Build Gates on each Cell

Figure 3 shows some examples of how gates are used by one or two AVs in the case of four gate cell-zone which has four gates in a cell-zone as shown in the lower-left cell-zone of Figure 2. Bold lines of Figure 3 is the gates which the AVs are passing through.

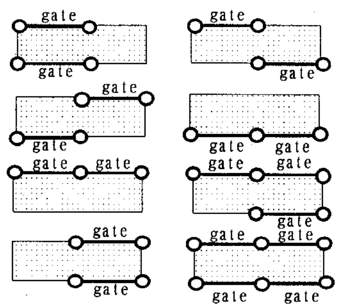

Figure 3 Various Kinds of Gate

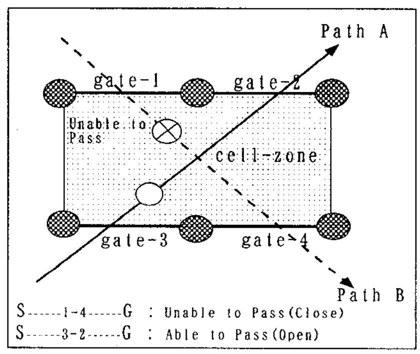

Figure 4 Management of Gate Information

Figure 4 shows a detailed example of a four gate cell-zone and shows how this algorithm manages environmental information using gate concept. Path A shows a case that there is no problem of passing the cell-zone or there is no affection of passing the cell-zone even though there are some problems which were described as evaluation factors before. In this case, open the gates, and register the gates as a possible movement path candidate which can be used either to find the shortest path or to calculate for dynamic path planning. In contrast to path A, path B shows a case of an inadequate path. Whenever it is inadequate to pass a cell-zone, the gate is closed until the cause of inadequatacy to pass is cleared in the free space.

3. Collision Avoidance

This section describes some collision avoidance algorithms as shown in Figure 5.

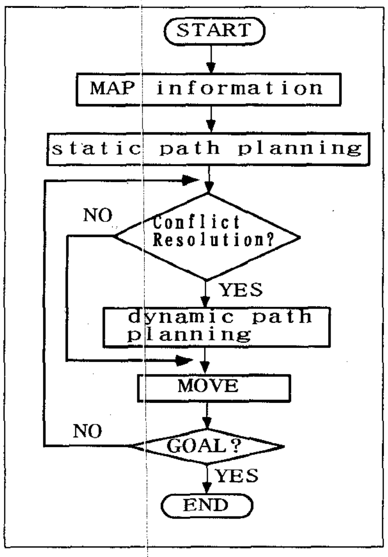

Figure 5 An Algorithm of the Proposed Method

3.1 Priority

An AV is able to negotiate with other AV by using priority method whenever they have any conflict resolution and a special case called movement loss rate. There are two kinds of priorities that we have given to this system. The first priority is the importance of task of movement. We have divided the importance of task of movement into three categories and given priority for each of them. Among these categories, the most important priority is given to an AV which conveys material according to their task. One less priority is given to an AV which is on the way of going to task destination. The lowest priority is given to an AV which is on the way back to the initial position where they were before the task called AV station. AV recognizes the status of their current task such as working on their job, on the way to task destination, or on the way back to the AV station. Then, AV assigns its own priority according to the status of their task.

Second priority is the announce time for specific task start. AVs always declare the task time right before they start their task, and this announce time is used to check the priority to dissolve the conflict resolution. When two AVs have a conflict resolution, they check their task priority first. However, if their task priority is the same, they check the task announce time. Whoever announced their task time first wins, and the loser will take avoidance activity.

3.2 Movement Loss Rate

There is a special case which two AVs are sharing a same path, and the path is very narrow that only one AV can go through at one time as shown in figure 6. That is whoever loses the negotiation by priority described in section 3.1, the AV should take a dynamic path generation step, and should take the temporary path even though they might have to take back very inefficiently. In this case, it would save total performance of the system if they could negotiate again not only by priority but the distance they travel without considering the priority.

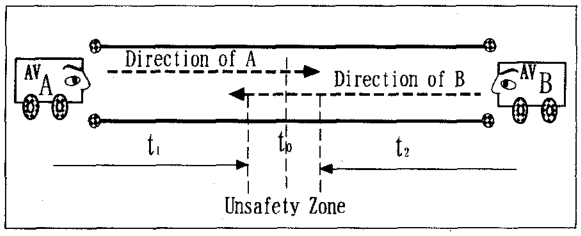

Figure 6 A Special Case for Negotiation

Assuming that AV A has higher priority, and AV A and AV B happened to be in a special case described above. In this situation, they need to negotiate again to accomplish better performance of the system using the following equations.

$$F_1 = D_a + D_b \tag{1}$$

where D_a is the distance from the start position to the goal of AV A, and D_b is the same distance of AV B.

$$F_2 = F_1 + \alpha_A \tag{2}$$

$$F_3 = F_1 + \alpha_B \tag{3}$$

where α_A is the distance of go around of AV A assuming that AV A has less priority, and α_B is the distance of go around of AV B assuming that AV B has less priority.

Now, it is necessary to compare Eq. (2) and Eq. (3) to obtain the movement loss rate. According to the movement loss rate we obtained, we need to decide whether we should keep the priority or we should ignore the priority if and only if we can have better performance of the system by comparing with a arbitrary delta value.

$$\frac{F_2}{F_3}\begin{cases} \geq 1; & \text{keep the priority} \\ < 1; & \text{ignore the priority} \end{cases} \tag{4}$$

If the output of Eq. (4) is greater than or equal to 1, we could not achieve any better performance of the system even by considering the special case. In this case, just keep the priority and perform the avoidance activity as planned by priority. However, if the output of Eq. (4) is less than 1, we could achieve better performance of the system by considering the special case. In this case, we need to check the output again to give another chance for AV A's priority. Here, we give an arbitrary delta to check how much movement loss rate we have accomplished . If the output is less than or equal to the delta, we ignore the priority of AV A and let AV B take advantage by dominating the path just for once. Otherwise, if the output of Eq. (4) is greater than the delta, we just follow the priority since we could not accomplish the movement loss rate as much as we have expected by giving an arbitrary delta. As we change the value of delta, we can control the importance of the priority even in the situation of movement loss rate negotiation. For example, if the value of delta is close enough to 1, it ignores priority and tries to reroute the path of AV, most of cases. However, if delta is very low value, we give more weight to priority than just a little time saving on the movement of AV.

4. Dynamic Path Generation

This section describes the dynamic path generation algorithm while AV is working on its task. This section divides environment where AV is working on into three patterns, and guide AV when they move to complete their task using these three pattern information. These three patterns are free space, a passage with regular size, and a passage with irregular size.

4.1 Free Space

AV judges the possibility of conflict resolution by sensoring the speed and direction of movement of other AV using following equations.

$$X_A(t) = V_{A0}\cos\theta t + \frac{1}{2}a_A\cos\theta t^2 \tag{5}$$

$$X_B(t) = V_{B0} - \frac{1}{2}a_B\cos\theta t^2 \tag{6}$$

Using Eq. (5) and Eq. (6), if following Eq. (7) or Eq. (8) is satisfied, we conclude that two AVs have conflict resolution.

$$X_A(t) = X_B(t) \tag{7}$$

$$X_B(t) - dx \le X_4 \le X_A(t) + dx \tag{8}$$

When Eq.(7) or Eq.(8) holds, two AVs should start negotiate using priority. Since this is not the special case that we have introduced in above section 3.2, we are not considering movement loss rate for this case.

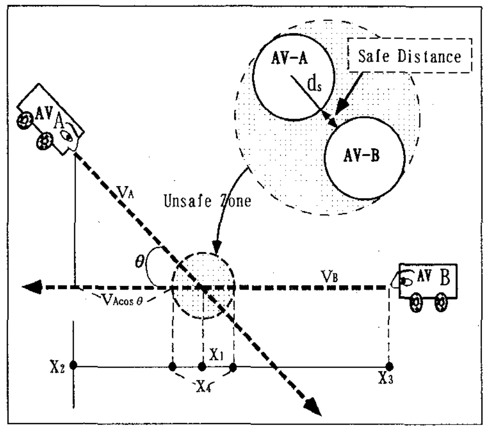

Figure 7 Avoidance of Moving Obstacle Free Space

4.2 A Passage with Regular Size

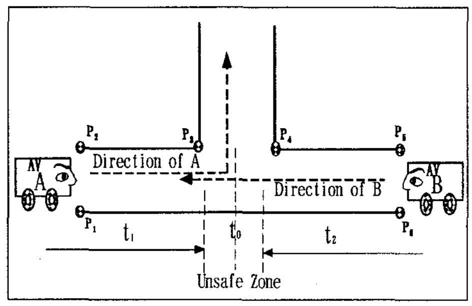

Figure 8 Avoidance of Moving Obstacle Regular Passage Size

Figure 8 shows a passage with a regular size which there are possibility that two AVs can negotiate and avoid the conflict resolution. When an AV enters a passage shown in figure 8, they calculate the size of the passage using 6 points,P1 through P6, and use the result of the collision avoidance movement.

In this case, there are three actions that the AV can take. If the size of the passage is wide enough for two AVs to enter at the same time, the AV which has lower priority go around the other AV in the same passage. Another case is that the width of the passage is not wide enough for two AVs, the AV which has lower priority changes its speed and turns to where it should go. In figure 8, AV-A should turn left in the middle of unsafe zone. They figure out that the conflict resolution exist before they enter the unsafe zone, then AV-A speeds up and turn left before AV-B enters to unsafe zone. The last case is that these two AVs cannot negotiate since AV-A cannot speed up, and the width of the passage is not wide enough for two AVs to enter. In this case, they need to negotiate again according to the contents of section 3.2 to improve the total performance of the system.

4.3 A Passage with Irregular Size

Figure 9 shows a passage with irregular size. In this case, AVs need to calculate if an AV which has lower priority can go around the other one at a certain point within the same passage using following equations.

Using P1 and P4 in figure 9, calculate an equation of a segment to check the width of passage at each point.

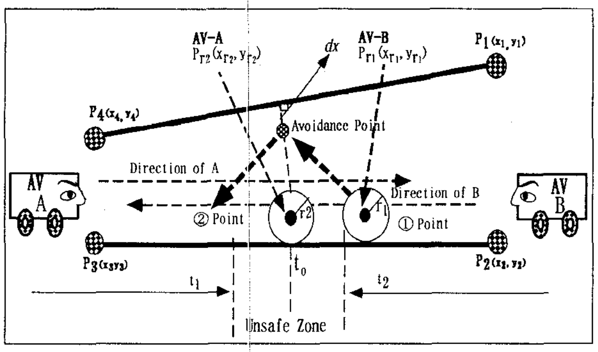

Figure 9 Avoidance of Moving Obstacle Irregular Passage Size

$$y - y_1 = \frac{y_4 - y_1}{x_4 - x_1}(x - x_1) \text{ , where } x_4 \neq x_1 \quad (9)$$

By using Eq. (9) we can derive an equation of a segment from P1 to P4. This equation let us to get the value of a and b as shown in Eq. (10) and Eq. (11).

$$a = \frac{y_4 - y_1}{x_4 - x_1} \quad (10)$$

$$b = y_1 - \frac{y_4 - y_1}{x_4 - x_1}x_1 \quad (11)$$

where $y = ax + b$.

Now, we can derive Eq. (12) using the position of AV-A at t_0 which presented as $p_{r2}(x_{r2}, y_{r2})$.

$$dx = \frac{|ax_{r2} - y_{r2} + b|}{\sqrt{a^2 + 1}} \quad (12)$$

After calculate Eq.(12), we can derive either of following equations. Eq. (13) shows that it is impossible for AV to pass another AV since the width of the passage is not wide enough. In this case, we let the AVs to negotiate again using movement loss rate method described in section 3.2. However, if we could derive Eq. (14), it is possible that the AV which has lower priority can go around the other one within the passage as shown in figure 9. In this case, AV-B which should go around in figure 9, needs to speed up as much as $\sqrt{2}v_0$ since it should travel more than what it should have.

$$dx - r_2 \leq 2r_1 + \alpha \quad (13)$$

$$dx - r_2 > 2r_1 + a \quad (14)$$

5. Simulation

We made a simulator to prove effectiveness of the algorithms we have proposed above. Figure 10 shows the initial environment frame shot.

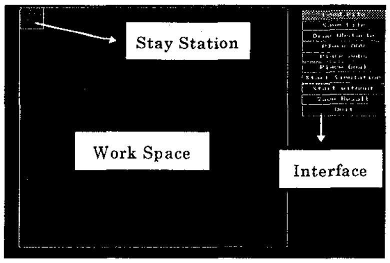

Figure 10 Initial Environment of System

We assumed and set the following environmental conditions in the simulator.

- ●Size of environment is 600 X 600
- ●Obstacles are either triangles or rectangle
- ●Size of an AV is 30 for its diameter
- ●Speed of an AV is 25 unit/sec
- ●Each AV generations its own static path
- ●Each AV can sense overall situation of system
- ●Initial static obstacle information, task of each AV, overall information of the system is given by the user in advance
- ●Each AV recognizes its location in the system
- ●Each AV updates its own priority itself

5.1 Result of Simulation

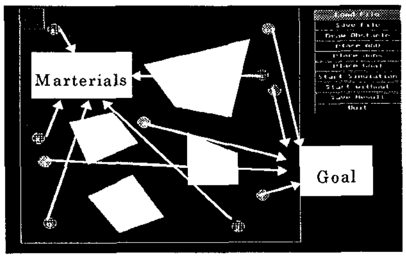

Figure 11 Schematic View of the Simulation Environment

Figure 11 to Figrure 13 show an example of three AVs working on their own task. Figure 11 shows the initial position of obstacles, materials, and goals. Following scenes were taken when the time was 30 seconds after started and every AV has finished their task. This example shows three AVs are working on its own task. White lines drawn are the path that AV was taken from a material to their goal.

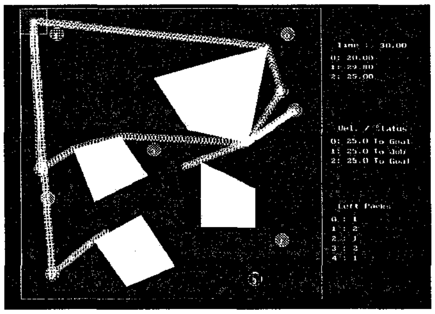

Figure 12 Simulation Result at T = 30 second

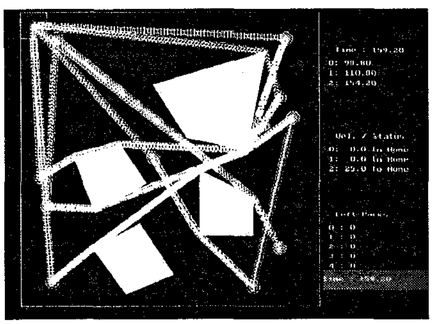

Figure 13 Simulation Result Finished

In this simulation, we have observed that an AV went around another one which has higher priority when they had conflict resolution. Also, there was another case that an AV was waiting for another AV which has higher priority to let the AV take its task first.

6. Conclusions

We have proposed an algorithm which can establish static and dynamic path. Also, some algorithm which can perform collision avoidance method such as priority method, and movement loss rate method are presented. This research covers many situations which can happen in real world model where AV can be applied.

Another interesting subject to study is the delta value which was discussed in section 3.2. Delta value acts like a weight on either priority or better performance by movement loss rate method. However, for the time being, the operator gives the delta value to the system. It should be calculated by each AV according to importance of their task, but not by the operator of the system.

References

[1] S.Kato, "Cooperation of Multiple Robot" JRSJ,No6,785-790,1994

[2] K.Kant and S.Zucker, "Planning collision-free tranjectoriesin time-varying environments" Proc. of Int.Conf. on Robots and Automation,1644-1649,1988

[3] T.Tsubouchi, "Planning and navigation by a mobile robot in the presence of multiple moving obstacle and their velocities" JRSJ,Vol.12,1029-1037,1994

[4] A.Inoue, "On-line Motion Planning of an Autonomous Mobile Robot to Avoid Multiple Moving Obstacles Based on the Prediction of Their Future Trajectories" JRSJ,No.10,444-449,1992

[5] H.Asama, "Collision Avoidance using Communication between Autonomous Mobile Robots" JRSJ,No.14,961-967,1996

[6] S. Prembuti and S. Yuta.; "Consideration on the Cooperation of Multiple Autonomous Mobile Robots" Internatinal Workshop onIntelligent Robots and Systems(IROA'90),59-63,1990

[7] T.Shibata, "Coordinative Behavior by Genetic Algorithm and Fuzzy in Evolutionary Multi-Agent System" IEEE,760-765,1994

[8] J. Wang.; "Deadlock Detection and Resolution in Distributed Robotic Systems" Proc. of JAPAN/USA Symposium on Flexible Automation, 673-677,1992

[9] D.D.Grossman,"Traffic Control of Multiple Robot Vehicles" IEEE J. Robotics and Automation,4(5), 491-497,1988

[10] Jae-Kook Lim and T.Takahashi, " An approach to dynamic path generation based on cell-decomposition" JSME The 4th Transportation and Logistics conferance,1995

[11] Jae-Kook Lim and T.Takahashi, "An approach to the algorithm of an instantaneous avoidance-activity generation based on priority & dangerous degree" JSME conferance'96,1996

[12] Jae-Kook Lim and T.Takahashi, " An approach to reconstruction of moving path based on reconnaissance simulator" JSME The 5th Transportation and Logistics conferance,1996