Path shortening and smoothing of grid-based path planning with consideration of obstacles

Masatomo KANEHARA, Satoshi KAGAMI, James.J Kuffner, Simon Thompson, Hiroshi MIZOGUHI

***Abstract*— This paper describes a path smoothing method that results in paths with clothoidal curvature and which includes a safety margin from obstacles. The method modifies grid based A^* results, and tries to push the given path away from obstacles. A technique to take into account the initial direction of robot is also shown.**

I. INTRODUTION

Mobile robots need paths that are smooth and have safety margins. Potential field methods and grid-based A^*[1] are used for path plannning for mobile robot. Path created by potential field methods are smooth. But, this planning method does not guarantee completeness. Because, the potential fields can have local-minimum. On the other hand, paths created by grid-based A^* is guaranteed completness and optimality. But, the path is not good for robots because, it is restricted by grid resolution and can get too close to obstacles. In order to solve this problem, a polynomial approximation method[2] was proposed. But, it needs too much calculation costs and sometimes cannot solve the polynomial approximation equation. The other smoothing method[3][4] was proposed. The results path is smooth. But, the path will be winding because the smoothed path is based on the grid-based A^* results. Steering Set Path Planning[5] can create smooth paths. However, it needs too much calculation cost. $FieldD^*$[6][7] performs planning and smoothing at the same time, and results a path smoother than grid-based A^*. But, the paths don't have safety margins. And, it is not smooth theoretically.

Drawing bigger obstacles than real obstacles in the map, all path planning algorithms can make paths having safety margins. But, these paths are not guaranteed completeness because, it block narrow path.

In order to move safely and speedy, we propose a path smoothing method that results in paths with clothoidal curvature and which include a safety margin from obstacles. The method modifies grid based A* results, and tries to push the given path away from obstacles. A technique to take into account the initial direction of the robot is also shown.

In the 2nd section, we describe an argorithm of path shortening and it's results. In the 3rd section, we describe an argorithm of path shortening and including safety margines from obstacles and it's results. In the 4th section, we describe an argorithm of smoothing with clothoidal curvature and taking into account the initial direction of robot.

The authors are with Digital Human Research Center, National Institute of Advanced Industrial Science and Technology,AIST Waterfront 3F Digital Human Research Center 2-41-6 Aomi Koto-ku,Tokyo,135-0064,JAPAN (e-mail:kanehara.masa@aist.go.jp).

M.Kanehara, S.Kagami and H.Mizoguchi are with Department of Mechanical Engineering, Faculty of Science and Technology,Tokyo University of Science,2641 Yamazaki, Noda-City, Chiba 278-8510, JAPAN.

J.J Kuffner is Carnegie Mellon University, Pittsburgh, PA ,USA

II. PATH SHORTENING OF GRID-BASED A^* RESULTS

A. Path Shortening with 2 Node sampling

We choose 2 Nodes from grid-based A^* results, and short cut one Node to the other Node. If there are some obstacles between the 2-Nodes, the shortening fails. Repeasing this procedure, we shorten the grid-based A^* path.

Path shortening with 2 nodes sampling from grid-based A^* results may not make an optimal path. But, the shortened path has less corners than grid-based A^* results, and we can also cancel the path shortening.

B. How to choose the 2-Nodes

We choose the 2-Nodes from grid-based A^* results by all play all. At n^{th} step, we try to shorten node n^{th} to node N^{th} from N to $n^{th}+1$ until we can shorten. Also, we repeat the method at n^{th} from 1 to $N-1$. N is the path Nodes. That is why the maximum calculation number of times are shown in Eq. 1. Eq. 1 means that as a map grows large, the calculation number of grows large. Sometimes, the calculation will not finish.

$$(N-1)+(N-2)+\cdots+1=\frac{1}{2}N(N-1) \tag{1}$$

Any further, we try to shorten n^{th} and $n^{th}+2$ Node from 1 to $N-2$ until we cannot shorten. The maximum number of calculation in this way becomes Eq. 2.

In this way, we try to shorten a map whose size is $4[m]\times 4[m]$. We get the result shown in Fig. 1. This time, we use a grid resolution of $1[grid]=10[cm]$.

$$\frac{N}{2}+\frac{N}{2^2}+\frac{N}{2^3}+\cdots+\frac{N}{2^m}=2N(m\to\infty) \tag{2}$$

But, the second of choosing the number of samplings 2-Noses fail to shorten the path on corners. Also, the second method gets different results(Fig. 1). In order to get shortest path, we choose 2-Nodes using the following procedures.

1) Shortening of straight line parts
2) Shortening of corner parts

In summary, we shorten the path with out corner parts at first. After that, we shorten the corner parts. We can get the ideal path.

In many cases, grid-based A^* results has obstacles neighboring the cells such as paths near corners, wall-side and narrow-path.

A node at a corner is likely to have 1 or 2 occupied cells in the 8-neighboring cells. A node at wall-side is likely to have 3 occupied cells in the 8-neighboring cells. A node at

1-4244-0991-8/07/$25.00/©2007 IEEE

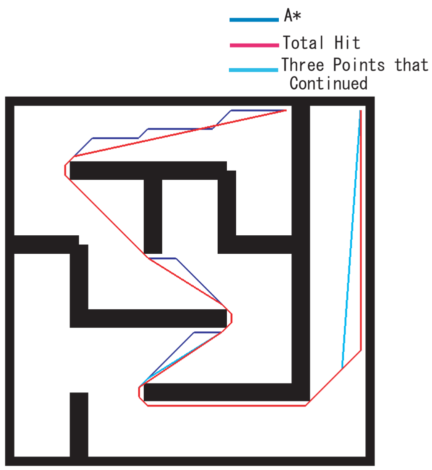


Fig. 1. Comparison Total Hit and The Other

narrow-path is likely to have over 3 occupied cells in the 8-neighboring cells. This way is heuristic, but we use this way to judge whether the $n+1$ Node is a corner or not.

This judging method will not affect the number of calculation times, because the methods changes only a turn of shortening. We name this shortening method the *PruningMethod*.

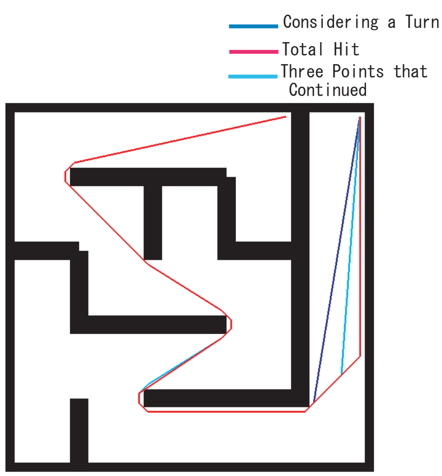


Fig. 2. Comparison Considering a turn and others

C. Comparison with FieldA*

We call *FieldD** on a static map *FieldA**. We compare *PruningMethod* with A^* and *FieldA**, example Fig.3. *PathLength* and *AverageDirectionChange* as is shown in Table. I. *PruningMethod* is effective at reducty *PathLength* and *AverageDirectionChange* compared with *FieldA**. In particularly, the *AverageDirectionChange* reduced dramatically. For path length compared with A^*, *FieldA** decreases 2.2% of the path length, while *PruningPath* decreases 3.7%. You can see that path lengths has little decrease. For *AverageDirectionChange*, *FieldA** decreases about 39% of the *AverageDirectionChange*, while *PruningMethod* decreases about 90%. *AverageDirectionChange* was expressed as in Fig. 4. *AverageDirectionChange* is the sum of direction change from start to goal divided by the *PathLength*. It give the direction change per unit length.

TABLE I
COMPARISON RESULT

	PathLength[m]	AverageDirectionChange[rad/m]
$A*$	11.9	0.697
*FieldA**	11.7	0.428
PruningPath	11.5	0.068

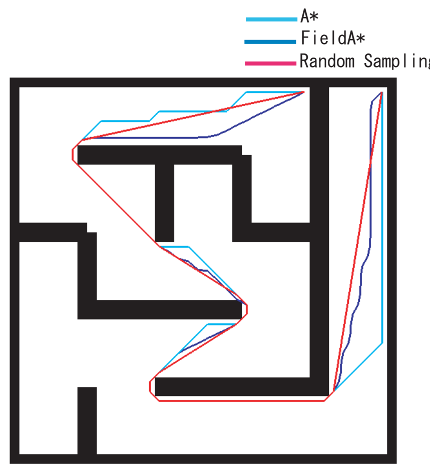
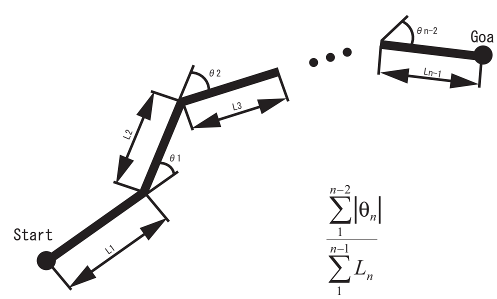


Fig. 3. Conparison Suggestion Technique and FieldA*



$$\frac{\sum_{1}^{n-2} |\theta_n|}{\sum_{1}^{n-1} L_n}$$

Fig. 4. Average Direction Change

III. THE PATH HAS SAFETY MARGINES FROM OBSTACLES

The path created by the method introduced in the previous section is neighboring obstacles. We describe a method in order to make safety margins.

At first, we make a *GradationMap* which is drawn with the gray scall that is proportionate to distance from nearest obstacles. We try to push the grid-based A^* results from obstacles using the *GradationMap*. After that, we revise the path and shorten. In this way, we get a path that has safety margins and is shortened.

A. How to make a Gradation Map

We convert a binary map(Fig. 5(a)) which shows obstacles whether freespace, into a *GradationMap*iFig. 5(b) which is drawn with a color that is proportionate to distance from obstacles. Darker is near obstacles, the lighter is further from obstacles.

We push the path grid-based A^* away from obstacles and then shoten with a $GradationMap$.

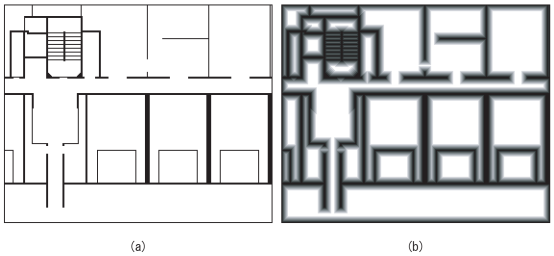


Fig. 5. Floor map and Gradation map

B. How to move the Nodes with GradationMap

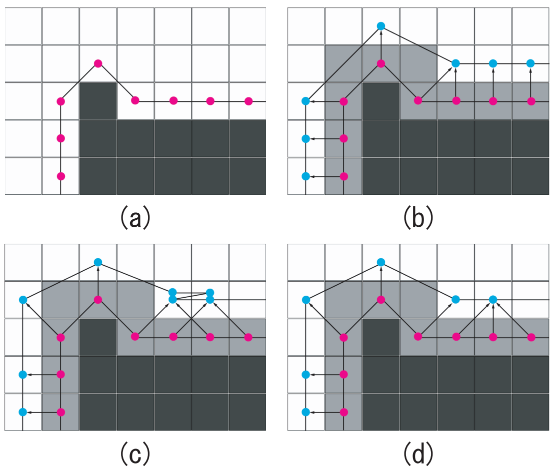


Fig. 6. The method of moving nodes

grid-based A^* results(Fig. 6(a)) planned by the map(Fig. 5(a)) had neighboring cells that contain obstacles. That is why we try to push nodes from dark color to light color with a $GradationMap$ (Fig. 5(b)). Beforehand, we decide that the quantity of movement. We get a path which includes safety margins. In case narrow paths, pushing no cells behinds lighter gradation causes a centering effect.

If we refer to only the $GradationMap$, the path approaches obstacles and some Nodes are zigzag (Fig. 5(b)).

That is why we push node from dark color cell checking 8-neighbors depth of intensity. The algorithm is finding the maximum value in the neighborhood as expressed by Eq.3. The value is found by multiply the Manhattan distance from obstacles by the depth of color in all 8-neighbors. We push the Node to the biggest value cell. For example, before push to the Node and the 8-neighbors are Fig. 7. We can find the biggest value which included by No.6 with Eq.3. We push the Node to No.6. The intensity of occupied cells color is D, the intensity of freespace is 0.

$$\sum_{n=1}^{7} Manhat\left(N[n], N[m]\right) \times D_n \tag{3}$$

We restrict the distance of Nodes so that the Nodes is not to be far. Using this method, We can get a path(Fig. 6(c)) which has safety margins, but we cannot solve the problem that some nodes are moved irregularly.

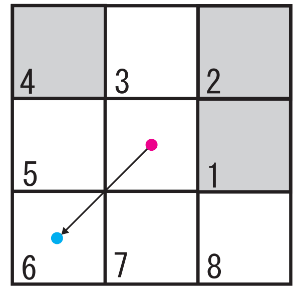


Fig. 7. Movement example

C. Revision of the path

If we push the Nodes refering to only the $GradationMap$, the path include Nodes that zigzag and is too meandering. That is why, we often push the nodes as shorten the path and not to move in darker Cell. We can get the path Fig. 6(d) which includes no nodes which replaced zigzag.

D. Path shorten

We can get paths which have safety margins by the methods, but we cannot solve the problem that the path has too many meanderings.That is why, we shorten[8] the path with 2-Nodes sampling. We choose 3 consecutive nodes and, join by the head and the tail. If the path between the head and tail nodes approaches obstacles or moves to a darker cell, we cannot shorten the path. By repeating the method, we can get the shortened path. We call the resulting path an $AvoidingPath$.

E. Evaluation

We make a simulation to push the Nodes 3$[gridcells]$ from obstacles in the map($23[m] \times 28[m]$). $1[gridcells]$ is $10[cm]$. The resultin path is shown in Fig. 8. The evaluation of the path is Table III. In comparison with A^* results, the path length has little difference. But, the $AverageDirectionChange$ decreases remarkably. Also, the path has minimum safety margins of $3[gridcells]$ from obstacles. In this way, the path becomes safe path.

We can pre-compute the $GradationMap$. That is why we get the $theGradationMap$ with no cost. The cost of pushing nodes is in proportion to the product of the path Nodes and some grid to push. The number of calculations to shorten is given by Eq. (2).

TABLE II

RESULT OF NON CURVE PATH

	PathLngth[m]	AverageDirection Change[rad/m]	MinimumDistance from obstacles[grid]
A^*	52.5	0.56	0
$AvoidingPath$	49.0	0.08	3

IV. PATH SMOOTHING

The curvature of the path created by method of previous section jumps up from 0 to $infinity$. It means that the robot needs to rotate on the spot. The robot cannot move smoothly without stopping and starting. We try to smooth the path by curved path.

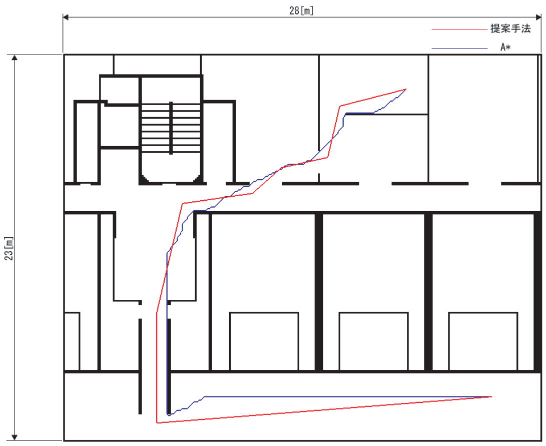


Fig. 8. non Curve path

A. Selection of curve lines

It is desirable to move to robot so that the change of curvature is smooth. There are various curve lines: polynomial curve, spline curve, arc and clothoid curve. However, the change of curvature of arc and clothoid curves change according to a linar equation. Clothoid curves are expressed

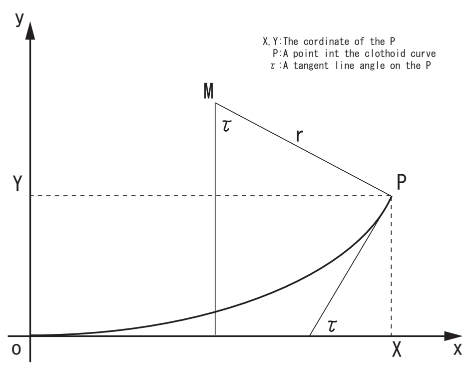


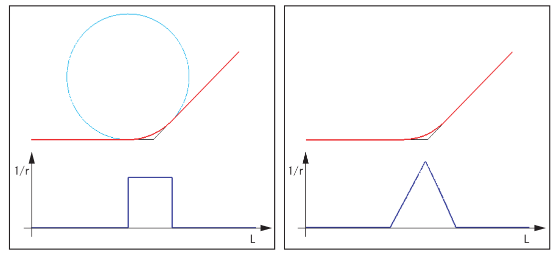
Fig. 9. Explanation of Clothoid Curve

by $R \times L = A^2$(A is fixed number, clothoid parameter). The radius of curvatures and a curve lengths are in inverse proportion. If the robot moves at a constant speed, the change of curvature is constant. A coordinate and radius in an arbitrary point P of clothoid curve in Fig. 9 expressed by Eq.4. In order to express the coordinate and radius, we needs parameter τ(A tangent line angle on the P)

$$\begin{aligned} X &= \frac{A}{\sqrt{2}} \int_0^{\tau} \frac{\cos\tau}{\sqrt{\tau}} d\tau \\ &= A\sqrt{2\tau}\left(1 - \frac{\tau^2}{2! \times 5} + \frac{\tau^4}{4! \times 9} - \frac{\tau^6}{6! \times 13} + \cdots\right) \\ Y &= \frac{A}{\sqrt{2}} \int_0^{\tau} \frac{\sin\tau}{\sqrt{\tau}} d\tau \\ &= A\frac{\sqrt{2\tau^3}}{3}\left(1 - \frac{3\tau^2}{3! \times 5} + \frac{3\tau^4}{5! \times 11} - \frac{3\tau^6}{7! \times 15} + \cdots\right) \\ r &= A/\sqrt{2\tau} \end{aligned} \quad (4)$$

If we smooth with only an Arc(Fig. 10(a)) at the intersection of strait lines, the robot cannot move smoothly, because the changes of curvature are discontinuous. Robots would need quick steering operation. To trace such a path, which is very dangerous, not only fot tracking errors, but also make a sudden centrifugal force.

If we smooth with only a clothoid curve (Fig. 10(b)), the path has no discontinuous change of curvature. But, the minimum turning radius shrink. That is why we smooth with the arc and clothoid curves (Fig. 11).



(a) (b)

Fig. 10. Sample of Circle Cueve and Clothoid Curve

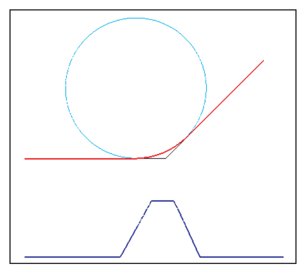


Fig. 11. Sample of Clothoid Curve with Circle Curve

B. Smoothing algorithm

In curvature of a road, there should be a constraint $R_{min}/3 < A < R_{min}$ considering safety. And, it is said that we should be $R_{min} < 2A$ if we can do it. We decide $A = 2/3R_{min}$ in order to get clothoid curve uniformally. In order to

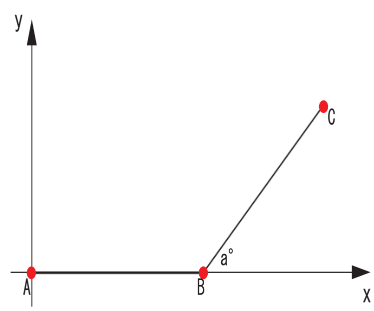


Fig. 12. 3 Nodes and Lines

simplify the smoothing by curve, we put 3-Nodes (A-C) like Fig. 12. Node-A is on the origin, node-B is on the X-axis, and node-C is in the 3rd-quadrant. 2-straight lines intersect on Node-B in Fig. 12. Then, we can smooth with arcs and clothoid curves if a minimum turning radius is decided. This process is as follows:

1) We grow τ from 0 by dt until $r = R_{min}$ as in Eq.4. When r become R_{min}, we decide $t = \tau$. If a is under 2τ, we advance to (3).

2) We draw arc centered on M (Fig. 13) until $\tau = a - 2t$.
3) We draw a clothoid curve that we draw in (1) to become in line(MC).

By the above-mentioned procedure, we can produce a smooth curve from node A to node C. We can make smooth path

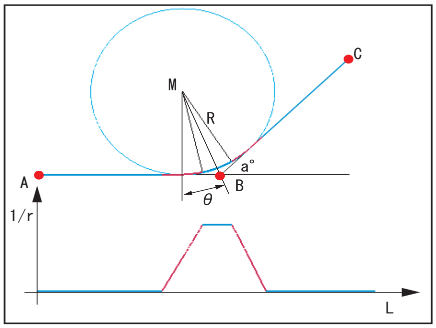


Fig. 13. Reference of SmoothingAlgrithm

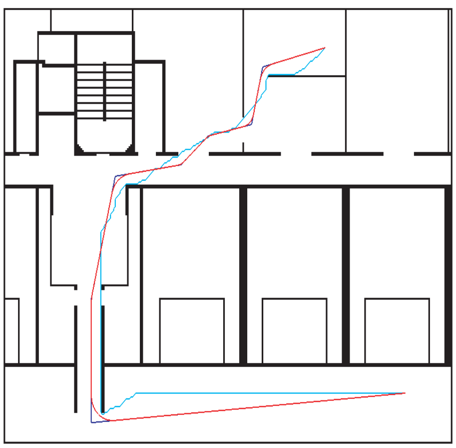


Fig. 14. Smoothed with ClothoidCurve

TABLE III
RESULT OF NON CURVE PATH

	PathLength[m]	AverageDirectionChange[rad/m]
A^*	52.5	0.56
AvoidingPath	49.0	0.08
SuggestionMethod	47.8	0.08

(Fig. 14) by smoothing off the path created by the presented in previous section. The relation of the smoothed path length and curvature is shown in Fig. 15(b). It shows the path's curvature is changing softly. On the other hand, the relation of the no smoothed path length and curvature is expressed Fig. 15(a). It means the path's curvature is changing from 0 to *infinity* suddenly.

C. How to decide a minimum turning radius

It is clear that little curvature is little burden for robot. If we make a minimum turning radius large in order to minimize the curvature, the smoothed path will leave

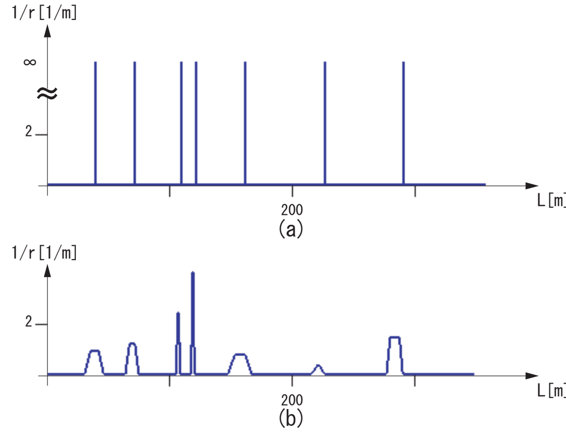


Fig. 15. Curvature of the smoothed path and the other

AvoidingPath. It is very dangerous that the robot hit obstacles. Because, the smoothed path is too close obstacles.

That is why we compare the smoothed path with R_{min} and *GradationMap*. If the distance from obstacles to the smoothed path was smaller than the value we decide before, we make R_{min} smaller and repeat the sometimes until the distance larger than the value. But, sometimes, we cannot find a smoothed path with a particle map or the value.

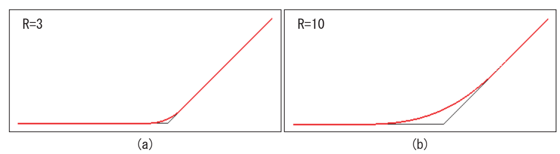


Fig. 16. Smoothing with Circle and ClothoiCurve

D. Comparison of safety margin

We compare grid-based A^* results with smoothed paths while varying the *SafetyMargin* in Fig. 17. These smoothed paths have safety margins and are smoother than grid-based A^* results.

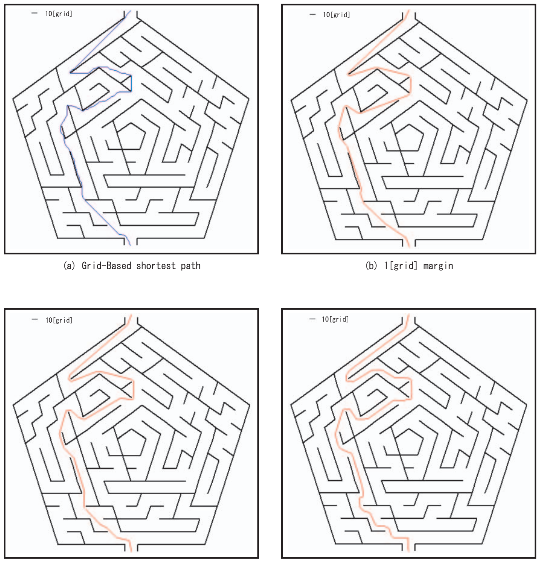
When we assign a small *SafetyMargin*, the smoothed path(Fig. 17(b)) is very similar to the original smoothed path without safety margin. But, the path has little margin from obstacles. If we assign a large *SafetyMargin*, the smoothed path(Fig. 17(c)) has a large safety margin. If we assign too large a *SafetyMargin*, and there is not enough margin in the environment, a path in the center of the obstacles is guaranteed. The smoothed path shown in Fig. 17(d) is in the center of the narrow path.

The above-mentioned issues mean that the suggested method is heuristic, but is robust for the narrow environments presented. The path is smooth and has a margin from obstacles in the complicated map.

The evaluation is expressed in Table III. The path length depends on the *SafetyMargin*. But, the *AverageDirectionChange* is smoller than grid-based A^*.

E. Path smoothing considering initial direction

Presented method has path smoothing from start to goal. But, it is generally the case that the initial direction of robot



(a) Grid-Based shortest path (b) 1[grid] margin (c) 3[grid] margin (d) 6[grid] margin

Fig. 17. Diference of Sefety margin

TABLE IV

PATH DATA OF VARIOUS SAFETY MARGIN

	PathLength[m]	AverageDirection Change[rad/m]
$ShortestMethod(Fig.17)(a)$	724	0.184
$1[grid]$ $Margin(Fig.17)(b)$	705	0.022
$3[grid]$ $Margin(Fig.17)(c)$	736	0.025
$6[grid]$ $Margin(Fig.17)(d)$	774	0.044

and the smoothed path will not be same direction. Robot rotation on the initial spot(Fig. 18) will solve the problem. But, this method is not appleciated way for car-like robots.

In order not to rotate on the initial spot(Fig. 19), we adjust the path from the initial direction of robot.

In the same condition as shown Fig. 14, if the initial direction is facing upward, we can generate the path (Fig. 20) without rotation on the initial spot.

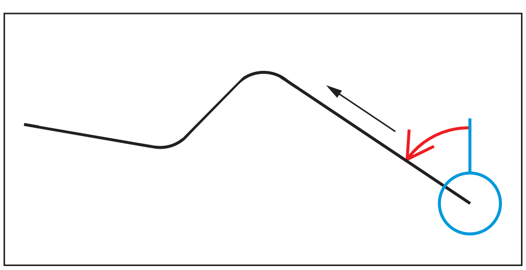


Fig. 18. Rotation on start spot

V. CONCLUSION

In this research, we try to make paths which are safe and smooth. In order to move safely, we ensure the path has safety margins. In order to move smoothly, we ensure the curvature of the path changes continuously.

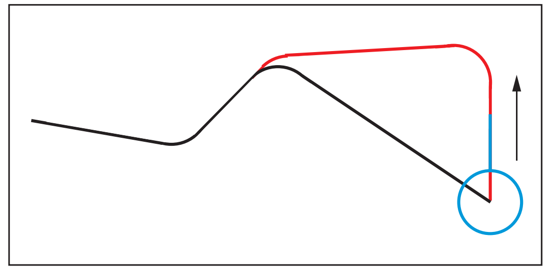


Fig. 19. No rotation of the starting spot

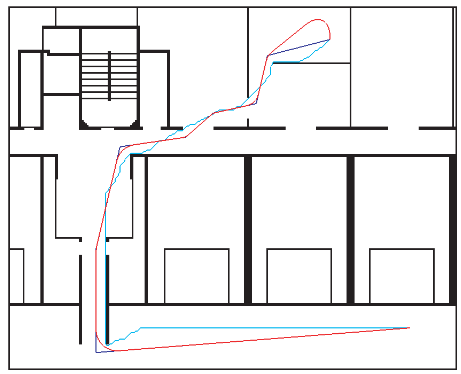


Fig. 20. Path when there is no rotation at the starting spot

Our suggested method pushs the Nodes of the grid-based A^* results away from obstacles and shortens the subsequent path. If there is no space to push, the pushed path is in the center of the obstacles. In order to change the curvature continuously, we smooth the path with arc and clothoid curves. Also, we add a function to the smoothed path so that there is no need to rotate on the initial spot.

That is how we ensure the robot advances smoothly and safely, paths which have safety margins and without the need to rotate on the spot.

REFERENCES

[1] Stuart C. Shapiro : Encyclopedia of artificial intelligenceC1991.

[2] Simon Thompson, Satoshi Kagami: Smooth Trajectory Planning with Obstacle Avoidance for Car-Like Mobile Robots , The 23th Annual Conference of the Robotics Society of Japan.

[3] Koji MAKANAE, “An application of Parametric Curves to Highway Alignment”, Journal of Civil Engineering Information Processing System in 2000, 169-176, 2000.

[4] Li Guiqing, Li Xianmin, Li Hua: “3D Discrete Clothoid Splinesh,(CGIf01,pp321-324)

[5] H. Koyasu and J. Miura . ”Mobile Robot Motion Planning Considering Path Ambiguity of Moving Obstacles,” Proc. the 9th Int. Conf. on Intelligent Autonomous Systems, pp. 85-93, Kashiwa, Japan, Mar. 2006.

[6] D. Ferguson, A. Stentz:The Field D* Algorithm for Improved Path Planning and Replanning in Uniform and Non-Uniform Cost Environments, Robotics Institute, Carnegie Mellon University, June, 2005, CMU-RI-TR-05-19, Pittsburgh, PA.

[7] D. Ferguson, A. Stentz: Field D*: An Interpolation-based Path Planner and Replanner, Proceedings of the International Symposium on Robotics Research (ISRR), October, 2005.

[8] Masatomo KANEHARACSatoshi KAGAMI, James J.Kuffner, Hitoshi Mizoguchi: “Path smoothing technique for a mobile robot path planning,and its comparison whit Field A*”, Robotics and Mechatronics Conference 2006.

[9] Masatomo KANEHARA, Satoshi KAGAMI, James J.Kuffner, Hiroshi Mizoguchi: “Generation of an obstacle evation path with room for a mobile robot.”, The 24th Annual Conference of the Robotics Society of Japan.