Pergamon

Pattern Recognition, Vol. 31, No. 8, pp. 1127–1138, 1998

0031-3203/98 $19.00 + 0.00

PII: S0031-3203(97)00141-6

A MORPHOLOGICAL APPROACH TO SHORTEST PATH PLANNING FOR ROTATING OBJECTS†

SOO-CHANG PEI,‡,* CHIN-LUN LAI‡ and FRANK Y. SHIH§

‡Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.
§Department of Computer and Information Science, New Jersey Institute of Technology, Newark, NJ 07102, U.S.A.

(*Received* 1 *August* 1996; *in revised form* 31 *October* 1997)

Abstract—This paper presents a mathematical morphology based approach to find the shortest path of an arbitrarily shaped moving object with rotations amid obstacles of arbitrary shapes. By utilizing the proposed rotational morphology along with the continuous turning-angle constraint, the representation framework of the motion planning problem becomes simple and practical. Two vehicle models with different driving and turning systems are used in solving the path planning problem and their strategies are presented. Simulation results show that our algorithm works successfully in various conditions. © 1998 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Motion planning   Mathematical morphology   Rotational morphology
Continuous turning angle   Distance transformation

1. INTRODUCTION

Modern manufacturing and other high technology fields of robotics and artificial intelligence have stimulated considerable interests in the motion planning or the shortest path finding problems.[1] The main objective is to plan a collision-free path (preferably an optimal path) for a moving object in a workspace populated with obstacles. Given an initial position and orientation for a robot and a target position and orientation, a path is searched from starting to ending positions that is collision-free and satisfies some criteria such as minimal cost and continuous turning angles.

There are several theoretical approaches in optimal path planning, and each has its own particular advantages and drawbacks.[2] In general, two classes are used. One is via configuration space,[3–5] wherein the geometry of the object that is to be moved through a space filled with obstacles is essentially set theoretically added to the obstacle in order to create a new space. The space created by removing the enlarged obstacles is called the configuration space. In the context of the configuration space, the path planning problem is reduced to that of finding optimal paths of a reference point in the object from an initial to ending position. This approach appears in the application of using the traditional mathematical morphology in path planning for non-rotating objects.[6–8]

The second approach uses computational geometry to search the free space directly without transforming the workspace into configuration space.[9,10] This method allows robot movement with rotations, so that it is close to real-world applications. However, it becomes very complicated if obstacles gather closely. To make such algorithms tractable, some limitations, such as polyhedral assumption, must be added to simplify the problems.[3]

The goal of this paper is to use a novel morphological approach to solve the motion planning problem, thus the computational complexity, compared to computational geometry methods, can be reduced while allowing rotations for moving objects. To take the rotation movement into consideration in modeling the moving object, a new technique, named rotational mathematical morphology, is proposed to process both the orientation and geometric information, thus to overcome the disadvantages of the conventional morphology which is limited to non-rotating moving objects. All the properties of the traditional morphology[11,12] are inherited in the rotational morphology. The shape decomposition property and the parallel processing architecture can be used to speed up the computations.

This paper is organized as follows. In Section 2, the relationships between the shortest path-finding and the rotational morphology are described. Section 3 gives the definitions of rotational morphology and its properties, as well as its practical applications. In

* Author to whom correspondence should be addressed.

† This work was supported by the National Science Council in Taiwan, R.O.C. and by the New Jersey Institute of Technology, Newark, NJ, U.S.A.

Section 4, distance transformation and the proposed algorithm for finding the shortest collision-free path amid obstacles using the rotational morphology is described. Experimental results and discussions for two moving object models and other considerations are provided in Section 5. Finally, conclusions are made in Section 6.

2. RELATIONSHIPS BETWEEN SHORTEST PATH PLANNING AND MATHEMATICAL MORPHOLOGY

In order to solve the shortest path finding problem by using mathematical morphology, their relationships are first explored.

Assume that an object of arbitrary shape moves from a starting point to an ending point in a finite region with arbitrarily shaped obstacles in it. To relate this model to mathematical morphology, the finite region consists of a free space set with values equal to ones and a obstacle set with values equal to zeros, and the moving object is modeled as a structuring element. Thus, the shortest path finding problem is equivalent to applying a morphological erosion to the free space, followed by a distance transformation on the region with the grown obstacles excluded, and then tracing back the resultant distance map from the target point to its neighbors with the minimum distance until the starting point is reached.

Drawback of using the traditional mathematical morphology to solve the path planning problem is due to the fixed direction of the structuring element being used throughout the processes. A fixed directional movement results in a worse path which is much longer than the optimal path of real-world applications. To model the practical moving actions of a car-like robot by incorporating rotations, the rotational morphology is proposed such that the structuring element can be rotated simultaneously during translations.

Figure 1a shows the result of a shortest path finding algorithm[7] for a non-rotating object. By incorporating rotations into the motion of a moving object, Fig. 1b gives a more realistic solution to the shortest path finding algorithm.

3. ROTATIONAL MATHEMATICAL MORPHOLOGY

3.1. Definitions

Traditional morphological dilation and erosion perform vector additions or subtractions by translation of a structuring element along an object.[11,12] Since the structuring element is fixed in its orientation during processes, these operations cannot deal with the complex variation of directional features in an image. Moreover, their applications to the sweep motion involving both translation and rotation are limited. In this section, we propose the rotational mathematical morphology for the purpose of an enhanced theoretical framework with a wider range of practical applications.

Assume that the full 360° angles are equally divided into N parts (or directions). Let B_i denote the rotation of a structuring element B by the degree of $360i/N$, where $i = 0, 1, \ldots, N-1$, and let $\mathbf{y}$ denote 2D coordinates. Let A and B be a binary image and a binary structuring element, respectively. The rotational dilation of A by B is defined as

$$\begin{aligned}(A \tilde{\oplus} B)(y) &= [A \oplus B_{N-1}, A \oplus B_{N-2}, \ldots, A \oplus B_1, A \oplus B_0](\mathbf{y}) \\ &= [P_{N-1}, P_{N-2}, \ldots, P_1, P_0](\mathbf{y}) \\ &= \mathbf{P}(\mathbf{y}) \qquad (1)\end{aligned}$$

where $\mathbf{P}$ is a N-bit codeword matrix.

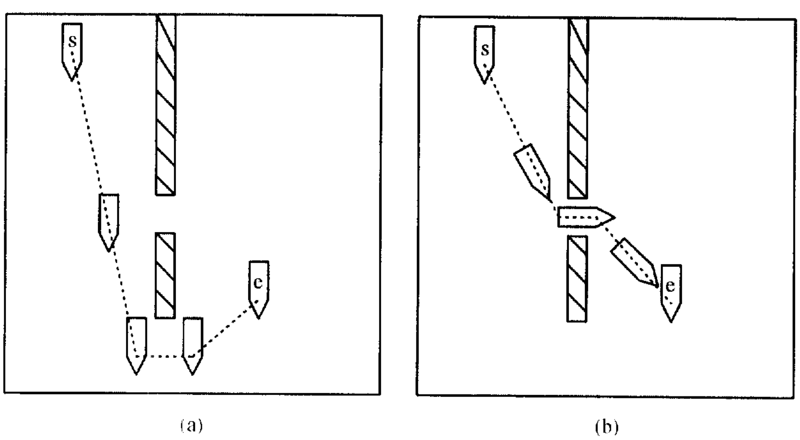


Fig. 1. Results of a path-planning algorithm for (a) non-rotating and (b) rotating vehicle models, where "s" and "e" denote the starting and ending points, respectively, and the slashed regions denote the obstacles set. Note that the resultant path in (a) is longer than the path in (b).

Similarly, the rotational erosion of A by B is defined as

$$(A \tilde{\ominus} B)(\mathbf{y}) = [A \ominus B_{N-1}, A \ominus B_{N-2}, \ldots, A \ominus B_1, A \ominus B_0](\mathbf{y}). \quad (2)$$

Figure 2 shows an example of a rotational erosion. Compared to the conventional mathematical morphology, the rotational morphology provides not only the geometric shape but also the orientation information. In other words, the results through rotational morphological operations indicate which direction of the structuring element is included or excluded in a local object region. Since the rotational morphology is equivalent to executing N binary morphology modules as shown in Fig. 3, it can be computed in parallel and its computational complexity remains the same as in binary morphology.

Note that $N = 8$ is often used in digital image processing applications. Therefore, $\mathbf{P} = [P_7, P_6, \ldots, P_1, P_0]_2$. That is, $\mathbf{P}$ is a 8-bit codeword with a base 2. If all the bits are summed up, it ranges from 0 to 8. After normalization, it ranges from 0 to 1 whose value possesses a fuzzy interpretation. Let us consider a fuzzy rotational erosion. For value 0, it indicates that the structuring element cannot be completely contained in the object region in any direction. For value 1, it indicates that the structuring element is entirely located within the object region for all orientations. Therefore, the value can be interpreted as the degree of the structuring element's orientations that can be fitted into the object region. Readers should note that if the structuring element is symmetric with respect to all directions such as a circle, the fuzzy rotational morphology is equivalent to the binary morphology.

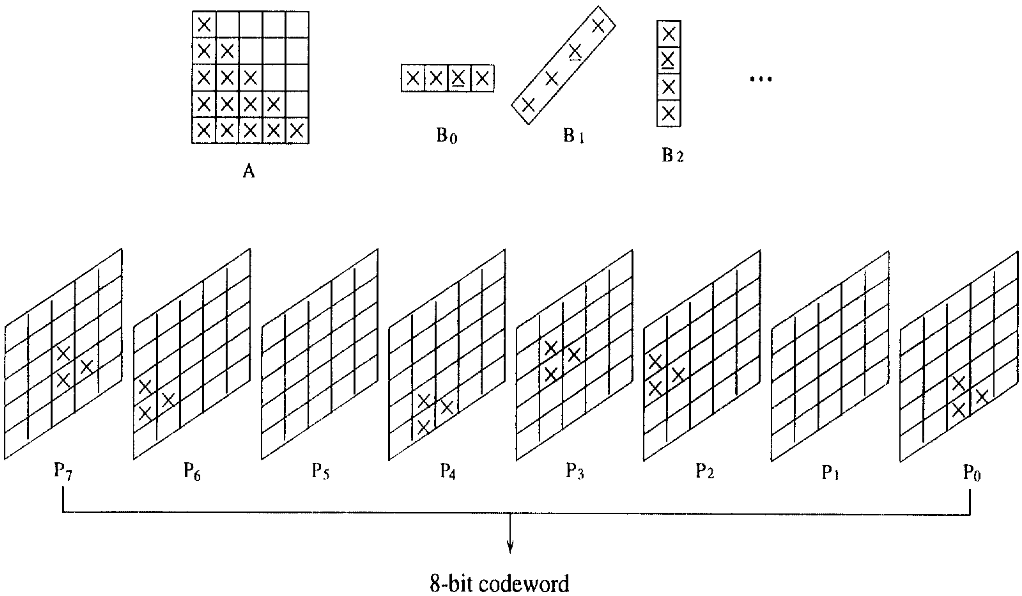


Fig. 2. An example of a rotational erosion operation of A by B.

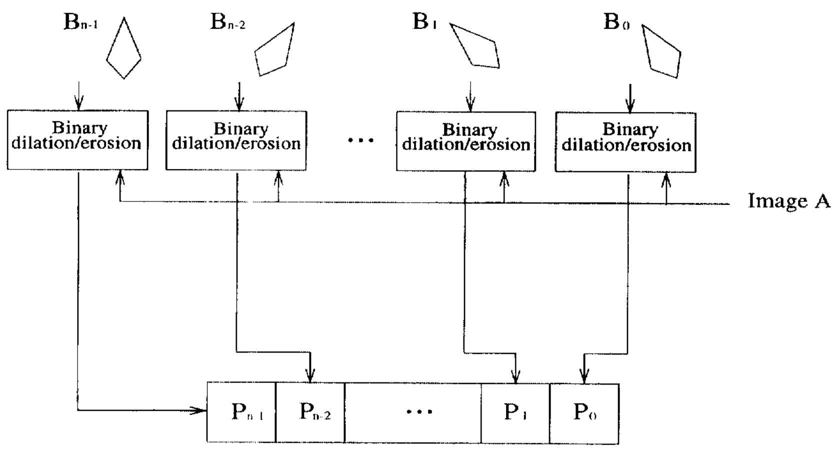


Fig. 3. The parallel processing structure of the rotational morphology.

The all-directional dilation, based on the intersection of all bits of the codeword, is defined as

$$(A \check{\oplus} B)(\mathbf{y}) = \bigwedge_{i=0}^{N-1} P_i(\mathbf{y}), \tag{3}$$

while the all-directional erosion is defined as

$$(A \check{\ominus} B)(\mathbf{y}) = \bigvee_{i=0}^{N-1} P_i(\mathbf{y}). \tag{4}$$

Since the all-directional dilation represents growth of obstacles, positions with all bits equal to one represent absolute collision-free space where no obstacles grow there. Note that because the all-directional erosion is used to indicate the collision-free positions in path finding such that the moving vehicle can be contained in the free space for at least one orientation, it is defined as a union of the codeword bits.

3.2. *Properties*

All the properties of binary morphology are also true in the rotational morphology since it is composed of N binary morphological operations. Some properties used in our shortest path finding algorithm are described below.

Property 1. The rotational morphology is translation invariant, e.g.,

$$(A \tilde{\oplus} B_z) = (A \tilde{\oplus} B)_z, \tag{5}$$

where $\mathbf{z}$ denotes any 2D coordinates. The rotational morphology owns the duality and decomposition property as follows.

Property 2.

$$A^c \tilde{\oplus} \bar{B} = (A \tilde{\ominus} B)^c \tag{6}$$

where A^c represents the complement of set A and $\bar{B}$ represents the symmetric set of B with respect to the origin.

Property 3. For $B = C \cup D$

$$A \tilde{\oplus} B = (A \tilde{\oplus} C) \cup (A \tilde{\oplus} D), \tag{7}$$

$$A \tilde{\ominus} B = (A \tilde{\ominus} C) \cap (A \tilde{\ominus} D). \tag{8}$$

In finding the collision-free path, Property 2 provides a significantly efficient way in finding the free position (FP) set. That is, computing the obstacle regions saves more time than computing the entire FP set. Property 1 states that the solution will not be affected by translation, while Property 3 can be used for reducing the computational complexity by decomposing the structuring element into the union of small subsets.

3.3. *Applications*

The rotational morphology can be applied to various image processing applications such that their orientation information need be taken into account. Except the motion planning application described in this paper, some other practical applications are available.

One example of those applications is presented in the edge-linking problem.[13] In practice of edge detection, many noisy edge pixels and broken gaps along an edge contour often appear. Edge-detection algorithms are typically followed by linking. A new morphology-based edge linking algorithm is proposed there to link up the broken edges by extending edge legs in their directions. Length of edge legs is a variable related to the edge curvature. An adaptive morphology mechanism is built as a flexible and efficient tool to deal with object shapes by varying the size and orientation of the structuring element according to their local geometric properties.

4. THE SHORTEST PATH PLANNING ALGORITHM

Due to the non-rotating movement of the conventional morphological operations, the path determination algorithm in reference (7) cannot really generate the shortest path in practical cases. By using the rotational morphological operations, the rotation movement can be modeled to fit real-world applications and the optimal path can be acquired. The shortest collision-free path finding algorithm can be divided into two major parts and are described as follows.

4.1. *Distance transformation*

Distance transformation is a transform operation that converts a binary image with feature and non-feature pixels into a gray-level image whose pixel values represent corresponding distance to the nearest feature elements. In many applications of digital image processing such as expanding/shrinking, it is essential to measure the distance between image pixels. Distance transformation is a specific example of a more general cost function transformation. Since the contents of a distance map reveals the shortest distance to the referred points, tracing the distance map from a specified point to a referred point will result in a shortest path between the specified and the referred points.

The distance transformation used in this paper corresponds to the propagation distance transformation.[14] Advantage of using such distance transformation is that distances for each pixels are computed gradually, like a fire burn, and its behavior is independent of the convexity properties of the domain. Thus, transform with background (obstacles in a support region) is allowed. In this paper, recursive distance transformation with (3,4)-distance metric is used and the following procedures compute the distance map $\mathrm{D}(a,s)$ for all $a \in Z$. In this procedure, d[a] denotes an array representing the current transformation value at pixel a. Definitions corresponding to all variables can be found in reference (14).

1. Set d[s] = REFERENCE, where s denotes the starting point; set d[a] = UNTRANS, for all $a \in Z - s$, where UNTRANS is the largest positive integer available and REFERENCE = UNTRANS-1. If a rectangular array representation is used then set background points (obstacles set) = BACKGROUND, which is an arbitrary negative integer, so that non-domain points may be avoided.
2. Apply procedure **propagate (s)**
3. When the algorithm terminates, $D(a,s)$ = d[a] for all $a \in Z$.

Procedure **propagate()** modifies pixel values and, recursively, the pixel neighbor values. It replaces d[a] by the minimum of its original value and the minimum of the value of d[ ] at a neighbor plus the distance from that neighbor. Whenever d[a] is changed, **propagate()** is applied recursively to all neighbors of a. The **propagate()** procedure is expressed in pseudo-C as follows:

```
propagate(s)
pixel a;
{
    distance nd;
    extern distance d[ ] ;
    if (d[a] = = REFERENCE)
        nd = 0;
    else
        nd = min(d[N_i(a)] + d_i|N_i(a) ∈ Z);
    if (nd < d[a]){
        d[a] = nd;
        for (all neighbors i|N_i(a) ∈ Z)
        propagate(N_i(a) )
        }
}
```

4.2. Algorithm description

Assume we have a working support Z, an obstacles set O, an "L"-shape car set B, a starting point s and an end point e, as shown in Fig. 4a. The candidate space is denoted as $A = Z \setminus O$. Our goal is to find a collision-free

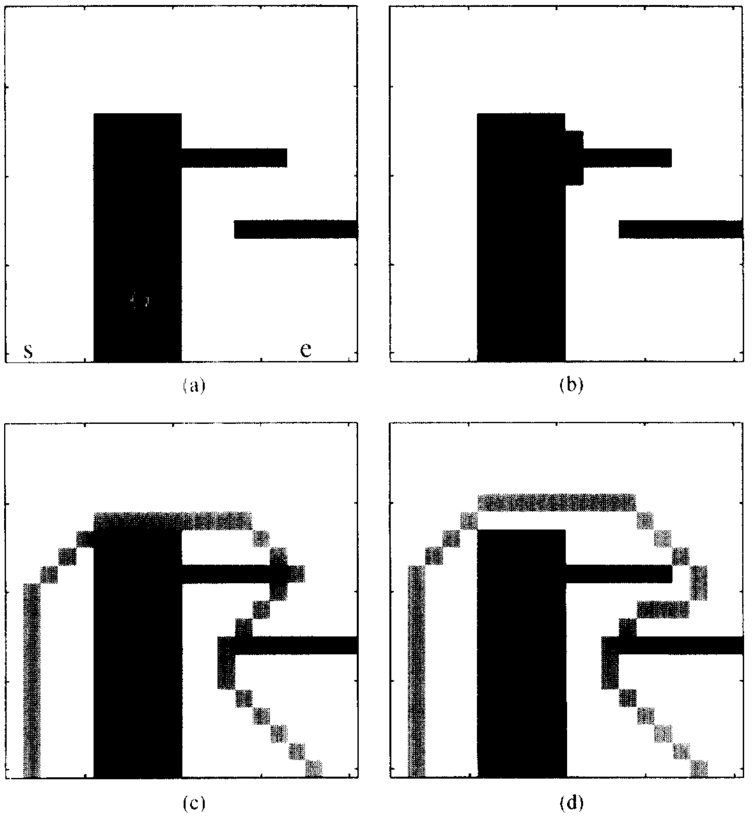


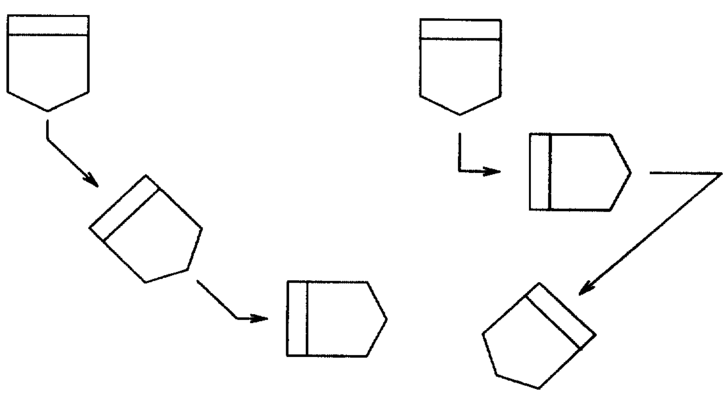
Fig. 4. (a) The original free position set, denoted as white region, and the obstacle set O, denoted as black region. (b) Result after applying an all-directional erosion. Note that the free position set is shrunk. (c) A collision-free position set attaching "s" and "e" points. (d) The practical result of our path planning algorithm.

path for B to move from s to e inside the set Z. To find the shortest avoidance-free path among obstacles, the candidate space A first undertakes an all-directional morphological erosion, i.e. $A \check{\ominus} B$. The resultant set $\check{A}$, shown in Fig. 4b, is called a "free-position" space and represents all the positions that the car set B can be contained for at least one orientation. After the free-position space is computed, a propagated distance transformation with (3,4)-distance metric described above is applied on Z with the constraint set $\check{O} = Z \setminus \check{A}$ to compute the distance map. Thus, a shortest path can be found by tracing the distance map with a pre-defined rule, choose the nearest distance in this example, from s to e as shown in Fig. 4c.

However, the shortest path obtained above (candidate solution) may be no longer valid if the practical case of smooth turning transition of car-like vehicle is considered. For each point the vehicle moves through, if its direction is not contained in the orientation information matrix $\tilde{A} = A \tilde{\ominus} B$ or its direction violates the smooth turning-angle limitation, the path is said to be impractical and this shortest path is not really a path, but just a free position space. To find the practical collision-free path, those points which disobey the above constraint will be set as the obstacles and the above steps should be repeated until a new path, which all its path points obey these constraint, is found. Figure 4d shows a practical path after the continuous turning-angle constraint is applied. The one-pixel upwards displacement of the horizontal upper part of the path is added to escape the problem of turning a 90° angle occurred in Fig. 4c. Figure 5 shows the example paths with continuous and discontinuous turning angle, respectively. After a satisfactory path is found, the directional information is recorded by chain-coding and the coordinate of the starting point and the orientation chain codes are sent as the final solution. The shortest path planning algorithm is described in the procedures below.

Step 1. Given sets Z, O, B, e and s, which are defined previously, set a candidate space as $A = Z \setminus O$.
Step 2. For A, do the rotational morphological erosion, $\tilde{A} = A \tilde{\ominus} B$, and an all-directional morphological erosion, $\check{A} = A \check{\ominus} B$, to find the directional information matrix $\tilde{A}$ and the free positions set $\check{A}$, respectively. The new obstacles set now becomes $\check{O} = Z \setminus \check{A}$ where $\setminus$ denotes a set subtraction.
Step 3. Apply the propagation distance transform (PDT) on Z with the obstacles constraint set $\check{O}$ to build the distance map.
Step 4. Routing the distance map from s to e with a pre-defined rule to find out one candidate path.
Step 5. For each point in the path, record its directional information by the chain-coding technique and trace the chain code with the directional information matrix $\tilde{A}$. If there exists one point p that its direction is not contained in $\tilde{A}$ or its direction does not satisfy the smooth turning-angle constraint, add that point to the obstacles set, i.e. $\check{O} = \{\check{O} \cup p\}$, and repeat steps 3–5.
Step 6. If the path connects s and e, i.e. a successful path exists, send the coordinates of the starting point and the orientation chain codes as the final solution. Otherwise, there is no solution to the current field Z. The entire process can be illustrated by tracing each step of Fig. 4.

The properties of the rotational morphology described in Section 3.2 are used to simplify the implementation of the algorithm. Using the duality property, the rotational erosion of A by B can be replaced by the rotational dilation of O by B to reduce the computational complexity since in most cases the obstacles set is far less than the candidate set. Another way to reduce the complexity is to compute those boundary points of O, denoted as ∂O, but not the entire obstacles set O.[6]. The shape decomposition technique described in equations (7) and (8) is adopted



(a) (b)

Fig. 5. Two examples for illustrating the conditions of (a) obeying, and (b) violating the continuous turning-angle constraint when the moving object is a car-like model.

to further speed up the computation of the free positions of a moving object.

The significant advantage of using morphological tools to solve the path planning problem is that the complex geometric problem can be solved by set operations. By the way, growing the obstacles is equivalent to shrinking the vehicle to a single point, thus simplifying the spatial planning problem.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, several complicated examples based on two vehicle models with size 30×30 field "Z" are examined to validate our system.

Two types of the vehicle models are used in simulating the real-world robot motions. One type is with driving-and-turning wheels located at the center of the vehicle body (center-turning), while the other type, like a bus, is with turning wheels being located at the front of the vehicle body (front turning). Considering the motion behavior of these two models, the former permits rotations at a fixed point while the other leads rotations and translations simultaneously, thus they produce different paths. An important constraint for the front-turning vehicles is the rotating angle. That is, if the turning angle is not continuous, as an example shown in Fig. 5b, it is not practical for a car-like vehicle to move across obstacles without collision. Figure 6 shows an example of the resultant paths obtained based on the two vehicle models. The reader should note that for the center-turning model, since the moving objects can rotate by arbitrary angles without translation, the continuous turning-angle constraint used in the proposed algorithm can be removed, and thus a more efficient path could be obtained. However, due to the mechanism limitation of the front-turning model, this constraint must be considered. Figures 7–10 show the simulation results for these two models with variant vehicle shapes and target points. It is observed that the proposed algorithm works successfully for various situations.

Some other constraints can be placed into the proposed algorithm to fit in variant situations. For some applications, it is needed to assign the orientation of the vehicle at the start point as well as at the end point. For example, it is necessary to drive a car in an assigned direction as well as park it in a fixed direction. These constraints are trivial to the proposed algorithm since it is only needed to put an additional

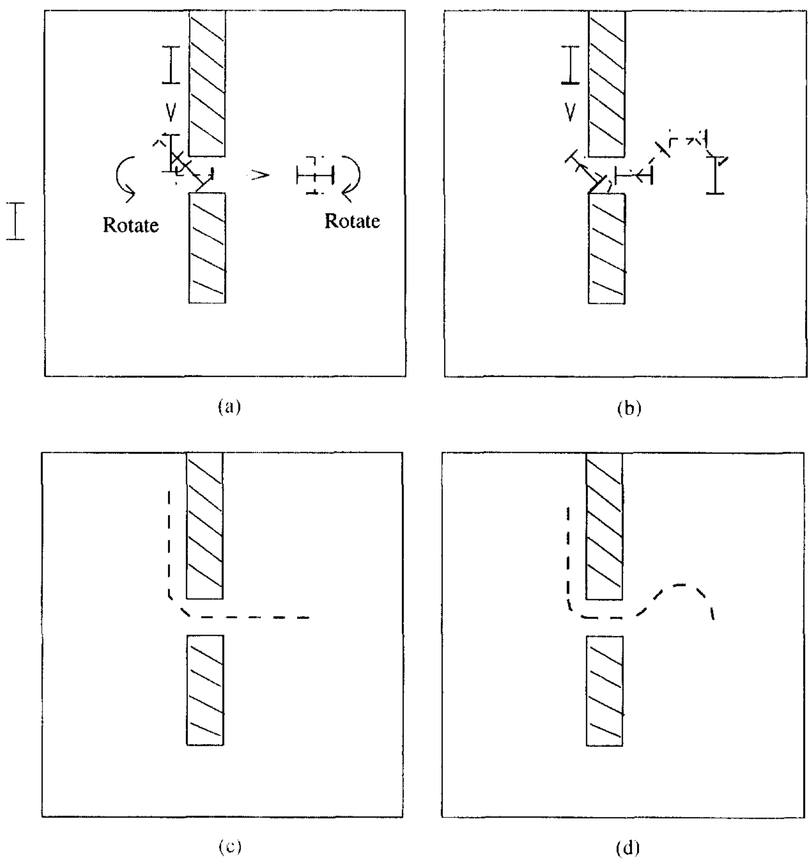


Fig. 6. Resultant paths of adopting (a)–(c) the center-turning and (b)–(d) front-turning vehicle models in the proposed algorithm.

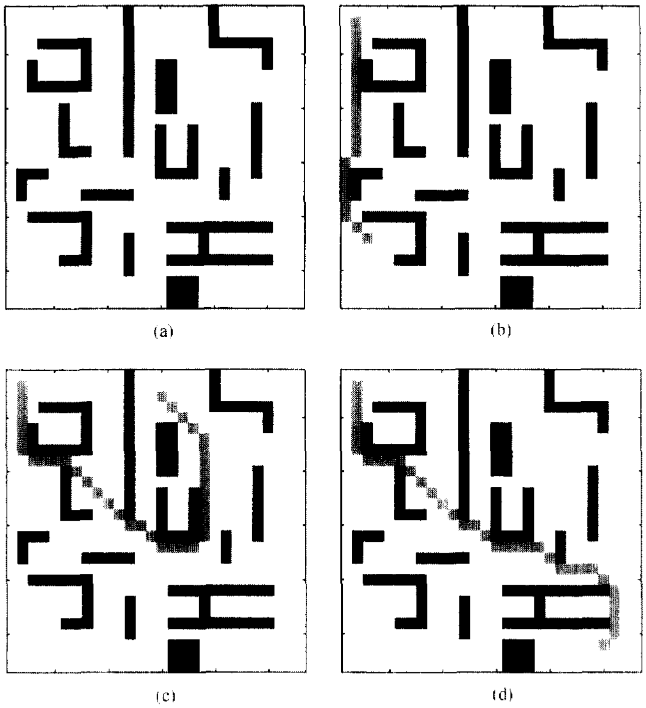


Fig. 7. Simulation results of the proposed algorithm by using the "|" shaped front-turning vehicle model: (a) original work space and resultant paths when end points are located at (b) [22,3], (c) [3,15] and (d) [26,25].

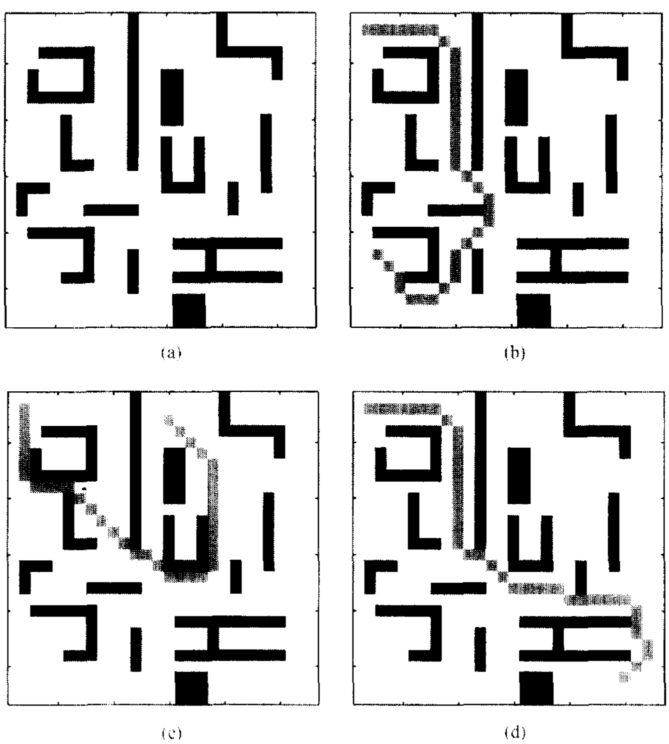


Fig. 8. Simulation results of the proposed algorithm by using the "T"-shaped front-turning vehicle model: (a) original work space and resultant paths when end points are located at (b) [22,3], (c) [3,15] and (d) [26,25].

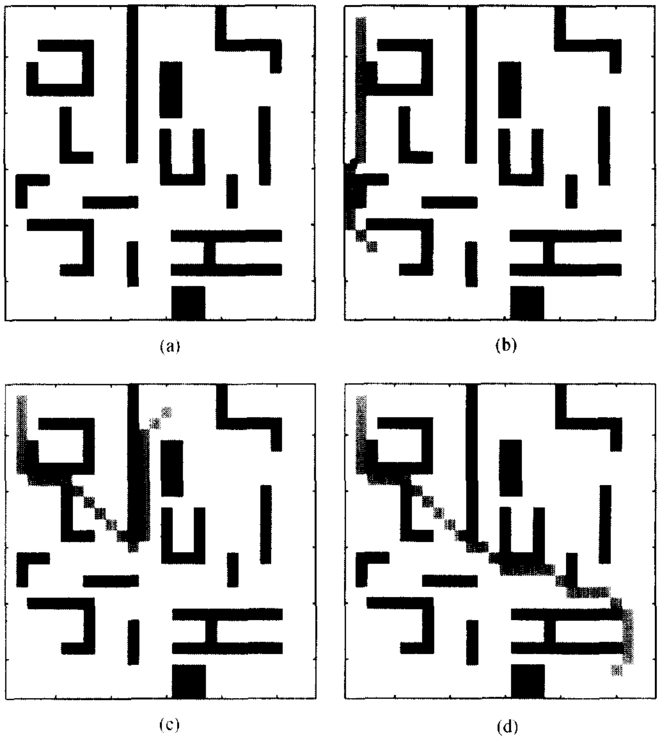


Fig. 9. Simulation results of the proposed algorithm by using the "|"-shaped center-turning vehicle model: (a) original work space and resultant paths when end points are located at (b) [22,3], (c) [3,15] and (d) [26,25].

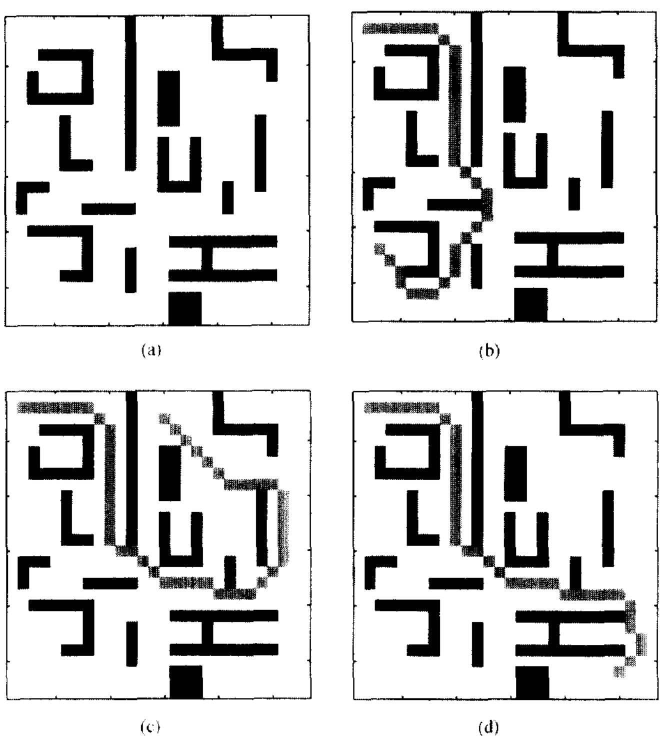


Fig. 10. Simulation results of the proposed algorithm by using the "T"-shaped center-turning vehicle model: (a) original work space and resultant paths when end points are located at (b) [22,3], (c) [3,15] and (d) [26,25].

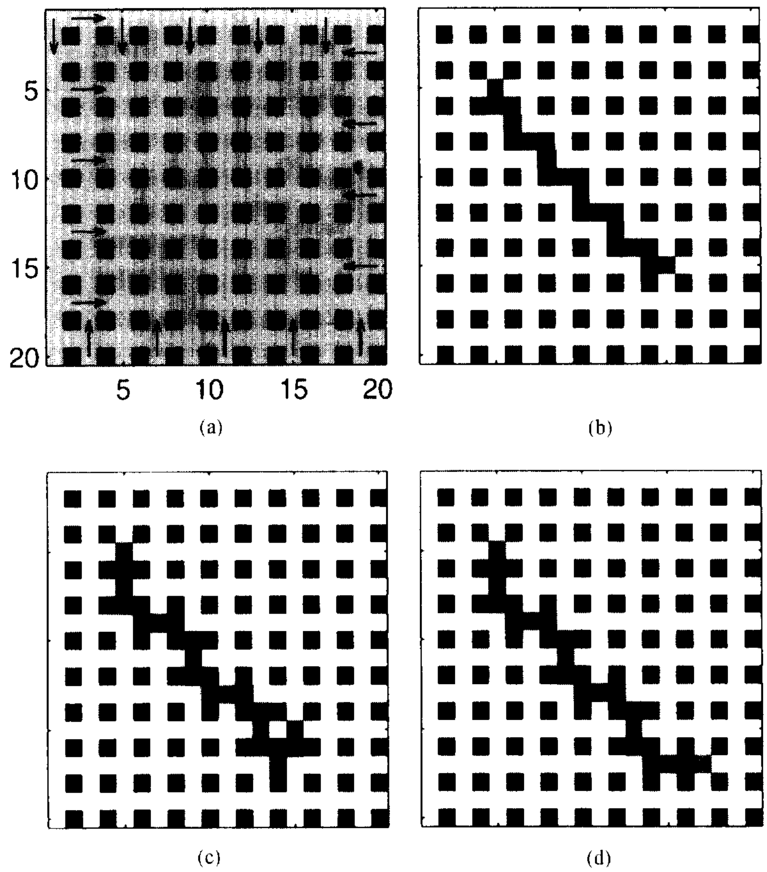
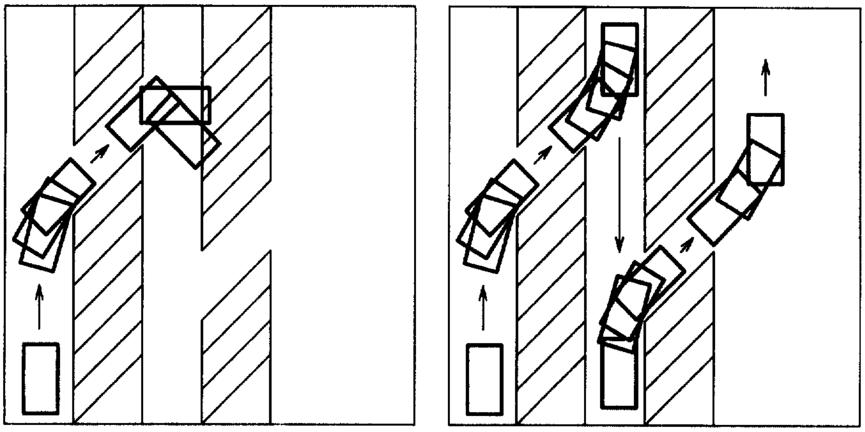


Fig. 11. Resultant path of a "|"-shaped car running in (a) the tested work space when the following conditions are imposed, (b) no restriction on driving direction and (c), (d) moving direction is restricted for each road. Note that the arrows denote the allowed moving direction for that row or column and the gray scale represents the moving paths from start position (5,5) to end positions (15,15), (15,15) and (17,17).

direction constraint into Step 5 of the proposed algorithm. Figure 11 shows an example of such case.

The computational complexity for the proposed algorithm is somewhat difficult to analyze due to the following two reasons. One results from the fact that the complexities of applying the PDT are different for various environments. Another comes from the allocation variety of obstacles as well as starting and target points; thus, the resultant path and the number of process iterations cannot be pre-determined. However, a coarse analysis based on calculating the CPU time, where the platform is a SUN SPARC-II workstation running the MATLAB program, is given as a reference to the complexity. In the proposed algorithm, since the distance transformation occupies near 80–90 percentage of the computational complexity and the rotational morphology can be implemented in parallel, it is reasonable to estimate the complexity by just considering the repetition of the PDT algorithm. For the cases in Figs 7–10, the CPU time for one iteration of PDT algorithm is 22 s, and the number of iterations is about 10–20 depending on the shape of vehicles and obstacles. According to our previous work experience,[15] the proposed algorithm with one iteration PDT could be done within 0.1 s using a DSP96002 micro-processor operated at 40 MHz frequency, thus a practical system is feasible.

In implementing the proposed algorithm for robotics control applications, the algorithm succeeds if both obstacles and moving objects are modeled as sets and the rotational degree is small enough. In spite of that, there still exist limitations on the proposed algorithm as appeared in the previous systems which used configuration space algorithms. One example is that the workspace is fixed and cannot be changed during this moving interval, or the predicted path is not valid and the configuration space needs to be re-computed. This limitation, however, can be overcome if a robust system can take the varying environment and starting points into consideration. An additional restriction comes from the relaxation of the object rotations. This means the object can be rotated to any orientation at any place. This condition will not be applied to the motor vehicle in which its mechanism requires continuous motion. Some other factors that limit the proposed approach to have an optimal solution can be mentioned in the following description.



(a) (b)

Fig. 12. Resultant path when considering (a) forward movement only, and (b) forward and backward movement, into the vehicle model. Note that there is no solution to the condition in (a).

In the proposed system, some other considerations are not included for the sake of simplicity. It is easy to find that only the 2D models (vehicle and obstacle) are considered, thus some minor modifications may be needed for applying this approach to the 3D environment. Another consideration is the rule used in tracing the distance map to find out a candidate path. For example, it results in different paths if horizontal-first or vertical-first rules are used. Sometimes this induces different solutions and thus an optimal path may be lost. Besides, consideration of backward motion of vehicles, which sometimes helps us in finding more heuristic paths as shown in Fig. 12, is neither included in this paper since it adopts complicated geometric methods in solving the smooth path problem.[16,17] Efforts on combining these geometric methods and the morphological tool for solving the path planning problem is still ongoing research work.

6. CONCLUSIONS

A novel morphological algorithm for planning the shortest collision-free path amid obstacles for arbitrary vehicle model by utilizing the rotational morphology is developed in this paper. To take the rotation movement of a real-world vehicle into consideration in modeling the moving object, the rotational morphology is proposed to process both the orientation and geometric information, thus it can overcome the drawback of former algorithms which are functional limited to non-rotating moving objects. Since all the properties of the conventional morphology still hold for the rotational morphology, some powerful techniques can be used to enhance the efficiency of the rotational morphology. Moreover, the rotational morphology can be implemented in parallel, so that its computational complexity remains the same as the standard morphology. On the other hand, due to the tractable complexity of using set operations to deal with the rotational movement, limitations (such as polyhedral assumption) on the geometrical methods, which is used to prevent obtaining intractable algorithms, can be removed. Finally, experimental results demonstrate that the proposed algorithm works successfully in solving the path planning problem for two different vehicle models.

Acknowledgment—The authors wish to thank Ji-Hwei Horng, a member of image lab. of EE department of NTU, R.O.C. for his helpful comments and suggestions.

REFERENCES

1. J. Latombe, *Robot Motion Planning*. Kluwer, Dordrecht (1991).
2. Y. K. Hwang and N. Ahuja, Gross motion planning a survey, *ACM Comput. Surveys* **24**(3), 219–290 (1992).
3. T. Lozano-Perez and M. A. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, *Comm. ACM* **22**(10), 560–570 (1979).
4. T. Lozano-Perez, Spatial planning: a configuration space approach, *IEEE Trans. Comput.* **c-32,** 108–120 (1983).
5. T. Lozano-Perez, A simple motion planning algorithm for general robot manipulators, *IEEE Trans. Robotics Automat.* **RA-3**(3), 224–238 (1987).
6. P. K. Ghosh, A solution of polygon containment, spatial planning, and other related problems using Minkowski operations. *Comput. Vision Graphics Image Process.* **49**, 1–35 (1990).
7. P. L. Lin and S. Chang, A shortest path algorithm for a nonrotating object among obstacles of arbitrary shapes, *IEEE Trans. Systems Man Cybernet.* **23**(3), 825–833 (1993).
8. V. Lumelsky, Algorithmic and complexity issues of robot motion planning in an uncertain environment, *Journal of Complexity* **3**, 146–182 (1987).

9. R. A. Brooks, Solving the find-path problem by good representation of free space, *IEEE Trans. Systems, Man Cybernet.* **13**(3), 190–197 (1983).
10. S. K. Kambhampati and L. S. Davis, Multiresolution path planning for mobile robots, *IEEE Trans. Robotics Automat.* **2**(3), 135–145 (1994).
11. G. Matheron, *Random Sets and Integral Geometry*. Wiley, New York (1975).
12. J. Serra, *Image Analysis and Mathematical Morphology*. Academic, New York (1982).
13. F. Y. Shih, S. C. Pei and J. H. Horng, Edge linking using adaptive morphology, *Pattern Recognition* (submitted).
14. J. Piper and E. Granum, Computing distance trans formation in convex and non-convex domains, *Pattern Recognition* **20**(6), 599–615 (1987).
15. C. L. Lai, The implementation of 8×8 DCT-based codec by using fast algorithms on digital signal processor, Master thesis, EE Department of Tatung Institute of Technology (1993).
16. J. A. Reeds and R. A. Shepp, Optimal paths for a car that goes both forward and backwards, *Pacific J. Math.* **145**(2), 367–393 (1990).
17. J. P. Laumond, P. E. Jacobs, M. Taix and R. M. Murray, A motion planner for nonholonomic mobile robots, *IEEE Trans. Robotics Automat.* **10**(5), 577–593 (1994).

About the Author—SOO-CHANG PEI was born in Soo-Auo, Taiwan in 1949. He received BSEE from National Taiwan University in 1970 and MSEE and Ph.D. from University of California, Santa Barbara in 1972 and 1975, respectively.

He was an engineering officer in the Chinese Navy Shipyard from 1970 to 1971. From 1971 to 1975, he was a research assistant at the University of California, Santa Barbara. He was Professor and Chairman in the EE Department of Tatung Institute of Technology from 1981 to 1983. Presently, he is the Professor and Chairman of EE Department at National Taiwan University. His research interests include digital signal processing, image processing, optical information processing, and laser holography. Dr Pei is a member of the IEEE, Eta Keppa Nu and Optical Society of America.

About the Author—CHIN-LUN LAI was born in Hsin-Chu, Taiwan, in 1969. He received the BSEE and MSEE from Tatung Institute of Technology, Taiwan, in 1991 and 1993, respectively.

He is currently working towards the Ph.D. degree in Electrical Engineering Department at National Taiwan University. His research interests include digital signal processing, image processing, video compression and mathematical morphology.

About the Author—FRANK Y. SHIH received B.S. from National Cheng-Kung University, Taiwan, in 1980, M.S. from the State University of New York at Stony Brook, in 1984, and Ph.D. from Purdue University, West Lafayette, Indiana, in 1987, all in electrical engineering.

He is presently an Associate Professor jointly appointed in the Department of Computer and Information Science and the Department of Electrical and Computer Engineering at New Jersey Institute of Technology. He serves as the director of Computer Vision Laboratory.

Dr Shih is on the Editorial Board of the International Journal of Systems Integration. He is also an associate editor for the International Journal of Information Sciences Applications, and for the International Journal of Pattern Recognition. He has served as a member of several organizing committees for technical conferences and workshops. He was the recipient of the Honorable Mention Winner of the International Pattern Recognition Society Award for Outstanding Contribution. He has received several awards for distinguished research at New Jersey Institute of Technology. He is an IEEE senior member. Dr Shih has published over 95 technical papers in well-known prestigious journals and conferences. His research interests include image processing, computer vision, computer graphics, artificial intelligence, expert systems, robotics, computer architecture, fuzzy logic, and neural networks.