

## Research Article

# Path Planning with Obstacle Avoidance Based on Normalized $R$ -Functions

Songqiao Tao  and Juan Tan

*School of Mechanic and Electronic Engineering, Wuhan Technical College of Communications, Wuhan 430065, China*

Correspondence should be addressed to Songqiao Tao; [taosongqiao@163.com](mailto:taosongqiao@163.com)

Received 30 June 2018; Accepted 17 September 2018; Published 4 October 2018

Academic Editor: Gordon R. Pennock

Copyright © 2018 Songqiao Tao and Juan Tan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Existing methods for path planning with obstacle avoidance need to check having the interference between a moving part and an obstacle at iteration and even to calculate their shortest distance in the case of given motion parameters. Besides, the tasks like collision-checking and minimum-distance calculating themselves are complicated and time-consuming. Rigorous mathematical analysis might be a practical way for dealing with the above-mentioned problems. An  $R$ -function is a real-valued function whose properties are fully determined by corresponding attributes of their parameters, which is usually applied to express a geometrical object. Thus, a signed distance function based on  $R$ -functions is created to represent whether two objects intervene and their level of intervention or separation. As the signed function is continuous and differentiable, the gradient information of the objective function guides a moving part to avoid its obstacles and to approach its target position rapidly. Therefore, a path planning approach with obstacle avoidance based on normalized  $R$ -functions is proposed in this paper. A discrete convex hull approach is adopted to solve the problem that  $R$ -function is inappropriate to represent a geometric object with some curves or surfaces, and pendent points and edges are generated in Boolean operations. Besides, a normalized approach ensures accuracy calculation of signed distance function. Experimental results have shown that the presented approach is a feasible way for path planning with obstacle avoidance.

## 1. Introduction

In recent years, technologies of path planning with obstacle avoidance have received extensive attention in the fields of robotics, computer simulation, and game animation. In the product design process, path planning with obstacle avoidance is an essential element of the simulations of product assemblies and disassemblies. Path planning with obstacle avoidance is in essence an optimization problem, which usually introduces the evaluation criteria (such as the shortest path or the shortest planning time) to plan an optimal path with obstacle avoidance from a given initial position to the destination in high-interference environments [1].

To avoid obstacles, existing approaches for path planning are necessary to frequently detect whether two objects intervene or to estimate the shortest path between a moving part and an obstacle at the given location. Therefore, there is not a unified form for planning path with obstacle avoidance. Moreover, collision-checking and minimum-distance

estimating themselves are complex and time-consuming tasks. To deal with the above-mentioned problems, rigorous mathematical analysis might be a practical way for path planning with obstacle avoidance.

The attributes of an  $R$ -function are fully determined by corresponding attributes of their argument. Thus, the sign of a real function is fully determined by the sign of its argument.  $R$ -function is usually applied to represent a geometric object [2]. If  $g$  is an  $R$ -function that expresses a geometric object, we can use a signed distance function  $g(x) \leq 0$  to describe interference constraints between this geometric object and a space point  $x$ . Thus,  $g(x) < 0$ ,  $g(x) = 0$ , or  $g(x) > 0$ , respectively, represents that space point  $x$  is the internal, the boundary, or the external of the geometric object.

Based on the above-mentioned analysis, a path planning method with obstacle avoidance based on normalized  $R$ -functions is presented in this paper. Firstly, Boolean operations based on  $R$ -functions are introduced to generate an implicit function for expressing a geometric object. Secondly,

a discrete convex hull method is adopted to solve problems that R-function is inappropriate to represent a geometric object with some curves or surfaces, and there are some pendent points and edges in Boolean operations. Then, a normalized method is introduced to transform the implicit function into the  $m$ -th order normalization function so that the calculation of signed distance function is accurate. Finally, planning paths with obstacle avoidance for a single-point and a polygonal region are conducted to validate the presented approach. Experimental results have shown that the proposed method is a feasible way for path planning with obstacle avoidance.

The rest of this paper is organized as follows. First, related works are introduced in Section 2. Then, the theory of R-function is presented in detail in Section 3. In Section 4, a discrete-convex hull method for generating CSG models of geometric objects is introduced. Section 5 adopts the normalization method to transform  $g$  into the  $m$ -th order normalization function. And path planning with obstacle avoidance based on the normalization function is given in Section 6. Following this section, experimental results applying the proposed method are presented in Section 7. Finally, the paper ends up with some conclusions in Section 8.

## 2. Related Works

Path planning with obstacle avoidance has been extensively applied for the robot navigation. For a robot, path planning carries out a series of translational motions and rotational motions from an initial position to the target position while obstacles are avoided in its working environment [3]. Most of path planning methods with obstacle avoidance can be classified into three categories: potential field method (PFM), sampling-based method, and heuristic-based algorithm.

Artificial PFM appeared earlier in [4], which is simple and highly efficient. However, disadvantages like local minima, nonsmooth movement, and oscillations also appear in artificial PFM. To escape from the location of the local minima, Kim [5] presented a new artificial potential function for path planning. Lee et al. [6] proposed new potential function with random force algorithm using PFM for solving the risks of a robot-obstacle collision and local minima trap. To overcome artificial PFM with slow progress and system instability in the presence of obstacles and narrow passages, a modified Newton's approach was applied to potential field [7]. Sfei et al. [8] proposed a new formula of repelling potential for reducing oscillations and avoiding collisions when the target approaches obstacles. To solve problems where a robot only follows the destination while obstacles are rejected in conventional artificial PFMs, Cosio et al. [9] proposed a path planning method with obstacle avoidance based on improved artificial PFM and genetic algorithm. Yin et al. [10] and Zhang et al. [11] adopted modified PFMs to solve path planning problem in dynamic environments.

The advantage of sampling-based path planning methods (SPPMs) has abilities to deal with planning problems in complex and/or time critical real world. At present, most influential SPPMs are probabilistic road-maps (PRMs) and rapidly exploring random trees (RRTs) [12]. Although PRMs and

RRTs both randomly sample connecting points from their state space, their graphs connecting points are generated in different ways [13]. More work related to SPPMs can be found in [14]. PRMs have better performance for path planning in high-dimensional state space. As curves or straight lines are used for constructing PRMs, a robot may go anywhere in its free space. Visibility graph [15–17] and Voronoi diagram [18–20] are two typical road-map approaches, which have achieved better results with dramatically various types of roads. RRT algorithms randomly build a space-filling tree to efficiently handle problems with obstacles and differential constraints by searching nonconvex and high-dimensional spaces [21]. Kothari et al. [22] proposed a RRT method to handle path planning problem of unmanned aerial vehicles with uncertainty. The probability of constraint violation and uncertain dynamic obstacles are, respectively, limited by a chance constraint and its extended framework. To further improve search efficiency, Bi-RRT algorithms with bidirectional search have appeared [23, 24]. Two RRTs, respectively, root at the initial and the target state, and algorithms seek states that are “common” for two trees. In [24], a Bi-RRT\* approach with the intelligence is adopted to handle path planning problem with complex cluttered environments.

A complete overview to heuristic-based algorithms for path planning, which can be found in [3].

## 3. Theory of R-Functions

*3.1. Terminology Definition.* The sign of an  $R$ -function is completely determined by the sign of its variable and has nothing to do with the value of its variable. For example, a function  $g_1 = x_1 x_2 x_3$  is negative, if the number of its negative variables is odd or positive otherwise. In general, the domain of variable not just has positive or negative but has the form  $\geq a$  or  $< a$  ( $a$  is a real number). In a practical application, the real domain is usually divided into  $(-\infty, -0]$  and  $[+0, +\infty)$ . Then, the definition of an  $R$ -function can be expressed as follows.

*Definition 1 (R-function  $y$ ).* Give a function  $S(x)$  with real parameter  $x$ , which has the forms

$$S(x) = \begin{cases} 0 & \text{if } x \leq -0 \\ 1 & \text{if } x \geq +0. \end{cases} \quad (1)$$

A real function  $y = g(x_1, \dots, x_n)$  is an  $R$ -function, if there is a logic function  $Y = G(X_1, \dots, X_n)$  which makes the equation  $S[f(x_1, \dots, x_n)] = G[S(x_1), \dots, S(x_n)]$  true. Here, the function  $Y$  is called  $g$ 's adjoint function. Some  $R$ -functions have the same adjoint function, such as functions

$$\begin{aligned} g_1 &= x_1 x_2 x_3, \\ g_2 &= x_1 + x_2 + \sqrt{x_1 x_2 + x_1^2 + x_2^2}. \end{aligned} \quad (2)$$

A group of  $R$ -functions with the same adjoint function is called an  $R$ -function branch. Key characteristics of an  $R$ -function are given in the following section (more characteristics can be found in [2]).

- (1) The set of  $R$ -functions has closure properties under a composition operator. Thus, a function can be obtained by compositing some  $R$ -functions, and it is also an  $R$ -function.
- (2) If  $g_1$  and  $g_2$  are  $R$ -functions from the same branch, then the function  $g_1 + g_2$  is an  $R$ -function belonging to the same branch.

3.2. *Sufficiency Completeness.* The characteristics of an  $R$ -function show that a complex  $R$ -function may be generated from simple  $R$ -functions. If there is a group of  $R$ -functions, which can generate  $R$ -functions belonging to any branch by using composition operators, we define this group of  $R$ -functions as a function system with sufficiency completeness.

Rvachev [25] presented some useful  $R$ -functions with sufficiency completeness. Here, we give a  $R_a$ -function system for basic logic operation, such as “negation” ( $\neg$ ), “conjunction” ( $\wedge$ ), or “disjunction” ( $\vee$ ).

$$\neg x = -x;$$

$$x_1 \wedge_a x_2 = \frac{1}{1+a} \left( x_1 + x_2 - \sqrt{x_1^2 + x_2^2 - 2ax_1x_2} \right); \quad (3)$$

$$x_1 \vee_a x_2 = \frac{1}{1+a} \left( x_1 + x_2 + \sqrt{x_1^2 + x_2^2 - 2ax_1x_2} \right).$$

Here,  $a(x_1, x_2) \in [-1, 1]$ . If  $a=0$  or  $a=1$ , we call a  $R_a$  system an  $R_0$  system or an  $R_1$  system. For function systems  $R_a^m$  and  $R_p$ , their logical operations can be defined as follows:

$$\begin{aligned} x_1 \wedge_a^m x_2 &= (x_1 \wedge_a x_2) (x_1^2 + x_2^2)^{m/2}; \\ x_1 \vee_a x_2 &= (x_1 \vee_a x_2) (x_1^2 + x_2^2)^{m/2}; \\ x_1 \wedge_p x_2 &= x_1 + x_2 - (x_1^{2p} + x_2^{2p})^{1/2p}; \\ x_1 \vee_p x_2 &= x_1 + x_2 + (x_1^{2p} + x_2^{2p})^{1/2p}. \end{aligned} \quad (4)$$

And their “ $\neg$ ” operations have the same handling rule with  $R_a$ -function system.

Although the above-mentioned  $R_0$ -function system and  $R_1$ -function system have the same logic function, their differential characteristics are various. For instance,  $R_0$ -function system is nondifferentiable at the origin of coordinate while other points are differentiable; but  $R_1$ -function system is nondifferentiable at the point  $x_1=x_2$  or  $x_1=-x_2$ ; and  $R_a^m$ -function system has  $m$ -order differential operators at the origin of coordinate while other points have infinite differential operators. Differential characteristics of  $R_a$ -function system have great influence in practical applications.  $R_a^m$ -function system has better differential characteristics while the calculation of  $R_0$ -function system is simple.

3.3. *R-Function Representation for a Geometric Object.* Boolean operations (such as “intersection” and “union”) between basic primitives can be used to express a CSG model [26]. Then, a geometric object  $B$  has the following form:

$$B = G [B_1, B_2, \dots, B_m]. \quad (5)$$

Here,  $B_i (i=1, 2, \dots, m)$  is a basic primitive; and  $G$  is a regular Boolean function whose operations have forms  $\cap^*$ ,  $\cup^*$ , and  $-^*$ . If we use logical operators  $\wedge$ ,  $\vee$ , and  $\neg$  to displace Boolean operations,  $G$  will become a logical function. Combined with the definition of  $R$ -function, a geometric object  $B$  can be expressed as follows:

$$G [S (B_1), S (B_2), \dots, S (B_m)] = 1 \quad (6)$$

If we define the signed distance function  $g_i(x)$  for a basic primitive  $B_i$  as

$$B_i : g_i(x) \geq 0 \quad (i = 1, 2, \dots, m), \quad (7)$$

(6) can be rewritten as follows:

$$G [S (g_1), S (g_2), \dots, S (g_m)] = 1. \quad (8)$$

Given a continuous function  $g(x)$  and its adjoint function  $G(X_1, \dots, X_n)$ , a geometric object  $B$  in (6) can be expressed as the following form [16]:

$$g (g_1, g_2, \dots, g_m) \geq 0. \quad (9)$$

In the above formula,  $g_i (i=1, 2, \dots, m)$  is a real continuous function while  $g$  is an implicit function of a geometric object. In the following, we give an example to illustrate the definition of implicit function  $g$ . In Figure 1(a), basic regions  $B_1$  and  $B_2$ , respectively, have the following equations:

$$B_1 : g_1(x, y) = 4 - x^2 \geq 0; \quad (10)$$

$$B_2 : g_2(x, y) = 1 - y^2 \geq 0. \quad (11)$$

It can be seen from Figure 1(a), rectangular region  $B$  has logic function  $B=B_1 \wedge B_2$ . Then, using  $R_0$ -function system to represent region  $B$  has the following implicit function  $g$ :

$$g = g_1 \wedge_0 g_2 = g_1 + g_2 - \sqrt{g_1^2 + g_2^2}. \quad (12)$$

If  $g_1$  in (10) and  $g_2$  in (11) are plugged into (12), (2) has

$$g = 5 - x^2 - y^2 - \sqrt{x^4 + y^4 - 8x^2 - 2y^2 + 17}. \quad (13)$$

#### 4. Discrete-Convex Hull Method

In Section 3.3, using  $R$ -function to express a geometric object is based on its CSG model. However, a geometric object usually has various CSG representations. For example, rectangular region  $B$  in Figure 1(a) has logic function  $B=B_1 \vee B_2$  in Figure 1(b). CSG representations usually have various logic functions and implicit functions, but they have the same logic functionality. In fact, it is an intractable task to automatically generate a CSG representation for a complex object. Even so, existing algorithms can automatically generate CSG representation for polygon regions and polyhedrons. In [27], a polygon region is automatically divided into some Sidechains with the intersection of convex hull vertices. Then, the intersection of the half-space defined by Sidechains can be adopted to express a polygon region while the half-space

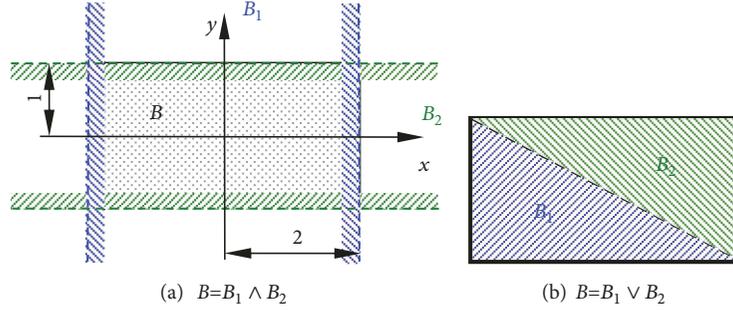


FIGURE 1: The rectangular region  $B$  and its logic function  $g$ .

is represented by the intersection or the union of the edges. If the intersection of two edges is a concave, the half-space is the union; otherwise, the half-space is the intersection. For instance, polygonal region  $P$  in Figure 2 has logic function  $P=C_1 \wedge C_2 \wedge C_3 \wedge C_4$ . Here,  $C_1=e_1 \wedge e_2 \vee e_3$ ,  $C_2=e_4$ ,  $C_3=e_5 \vee e_6$ , and  $C_4=e_7 \wedge e_8$ . Besides, convex hull method has been applied to polyhedrons in a three-dimensional space [28, 29].

However, the above-mentioned method is unsuitable for a geometric object with some curves or surfaces. Then, the surface is approximately discretized into planes while the curve is approximately discretized into straight lines. Thus, plane region  $P$  in Figure 3(a) has logic function  $P=e_1 \wedge e_2 \wedge e_3 \wedge e_4$  (see Figure 3(b)) and  $P=e_1 \wedge (e_{21} \wedge e_{22} \wedge e_{23} \wedge e_{24}) \wedge e_3 \wedge e_4$  (see Figure 3(c)). In Figure 3(c), the curve  $e_2$  is discretized into straight lines  $e_{21}$ ,  $e_{22}$ ,  $e_{23}$ , and  $e_{24}$ .

The above-mentioned approach for dealing with a geometric object is called a discrete-convex hull method. If the geometric object is a polygon without a curve or a polyhedron without a surface, the implicit function is accurate; otherwise, the implicit function approximately expresses the geometric object.

Discrete-convex hull method not only automatically generates CSG representation for a geometric object but also solves the problem of Point Membership Classification (PMC) [30]. PMC problem might lead to the implicit function with 0 value for a nongeometric object on its boundary. There are the following two aspects. First, irregular Boolean intersection  $\cap$  for a basic primitive might produce pendent points, pendent edges, or pendent planes (see Figure 4). As the operation  $\wedge$  for an  $R$ -function is irregular, the implicit function of a geometric object may have pendent points on its outside part. Second, rectangular region  $B$  has logic function  $B=B_1 \vee B_2$  in geometry (see Figure 1(b)). But for mathematical analysis, the operation  $\vee$  of an  $R$ -function cannot ensure that the implicit function has 0 value on  $B$ 's boundary. As PMC exists, the implicit function cannot define the interference constraint between a geometric object and a space point. Fortunately, a discrete-convex hull method avoids the above-mentioned problems. The reasons are that two adjacent lines (planes) of a polygon (polyhedron) just intersect in one point (straight line), and this point (straight line) must belong to the boundary of this polygon (polyhedron). For the operation Boolean union, the discrete-convex hull method will not produce the PMC problem.

Therefore, the implicit function of a geometric object based on discrete-convex hull method has 0 value on the object's boundary. And the internal point and external point, respectively, have a value  $< 0$  or  $> 0$ . Then, the sign of the function determines whether a space point and a geometric object have the interference. If  $g(x)$  represents an implicit function of a geometric object,  $g(x) \leq 0$  shows that there is no interference between the space point  $x$  and the object.

## 5. Normalized R-Function

The implicit function based on the above-mentioned approach cannot accurately calculate the signed distance between a space point and an object's boundary. For instance, the function has value of -0.84 at the point (1, 0) for the rectangular region given by the implicit function equation (13) in Section 3. But the minimum signed distance between the point (1, 0) and the boundary of the rectangular region is -1.0 in practices.

Let  $L$  be the boundary of a geometric object and  $p$  be a space point; the signed distance  $\rho(p, L)$  between the point  $p$  and the boundary  $L$  can be expressed as follows:

$$\rho(p, L) \equiv \inf \rho(p, q), \quad q \in L. \quad (14)$$

where  $\rho(p, q)$  is the signed distance between the point  $p$  and the point  $q$ ; if  $p \in G$ , the signed distance  $\rho(p, L)$  is "+" (there is the interference between the point  $p$  and the point  $q$ ); otherwise, it is "-". If all space points satisfy the condition  $g(p) = \rho(p, L)$ , the function  $g$  is called  $G$ 's normal function. For example, let  $(a, b)$  and  $r$ , respectively, be the centre and radius of a circle; the normal function  $g$  of this circular region has the following form:

$$g = r - \sqrt{(x-a)^2 + (y-b)^2}. \quad (15)$$

As most of geometric objects have some nonsmooth adjacent surfaces, their normal functions usually do not exist. Thus, we introduce the concept of normalized function for a geometric object. A similar normal function  $\bar{g}$  with essential

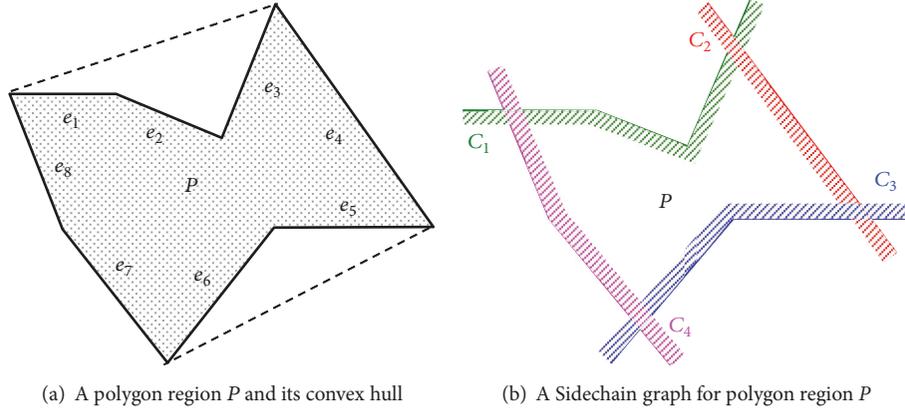
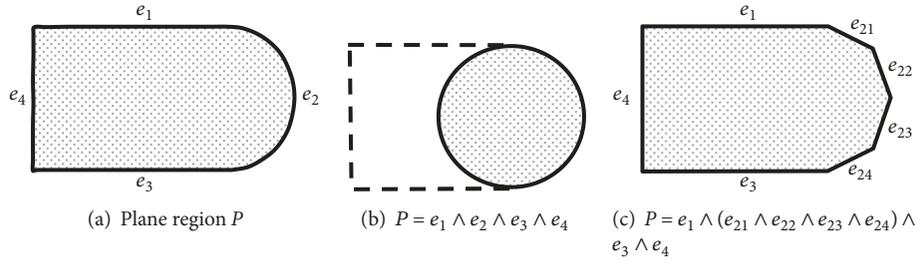
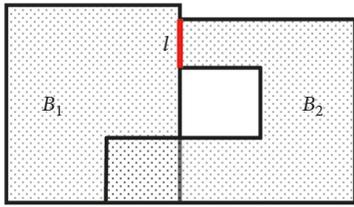


FIGURE 2: Convex hull method for a polygon region.

FIGURE 3: Plane region  $P$  with a curve is discretized.FIGURE 4: Irregular intersection  $B_1 \cap B_2$  causes pendent edge  $l$ .

differential attributes for a complex object is introduced, which satisfies the following conditions:

$$\begin{aligned} \bar{g}|_q &= 0, \\ \frac{\partial \bar{g}}{\partial v}|_q &= 1, \\ \frac{\partial^k \bar{g}}{\partial v^k}|_q &= 0 \quad (k = 1, 2, \dots, m), \end{aligned} \quad (16)$$

where  $q$  is a regular point on the boundary  $L$  and  $v$  is the normal unit vector of the point  $q$ . Then,  $\bar{g}$  is the  $m$ -order normalized function of a geometric object. There are very small differences between normalized function near the boundary and signed distance function. And normalized function far from the boundary is gradually, continuously, and smoothly similar to signed distance function. In general, many  $R$ -functions have the above-mentioned normalized

properties. For an  $R$ -function without normalized properties, the following method can be adopted to normalize.

If the function  $g \in C^m$ , a point  $q$  on the boundary  $L$  satisfies the following conditions:

$$\begin{aligned} g|_q &= 0, \\ \frac{\partial g}{\partial v}|_q &> 0. \end{aligned} \quad (17)$$

Then,  $g$  can use the following formula to normalize into the first-order form.

$$\bar{g} \equiv g (g^2 + \|\nabla g\|^2)^{-1/2} \in C^{m-1}. \quad (18)$$

Moreover, higher-order normalized method for the function  $g$  can be found in [31].  $M$ -order normalized functions  $g_1$  and  $g_2$  have the same logical operations  $g_1 \vee g_2$  and  $g_1 \wedge g_2$  with  $R$  function. Figure 5 lists the distribution of first-order normalized function (it is also called similar signed distanced field;  $R_p$ -function ( $p=2$ ) system is adopted) for the polygonal region in Figure 2.

The generation of normalized function of a geometric object can be summarized below.

Firstly, a discrete-convex hull method is adopted to generate the implicit function  $g$  using the following three steps:

- (1) A geometric object is transformed into a polygon (polyhedral). Then, a half-space implicit function that points to the interior of a geometric object is created based on its polygon (polyhedral) and edges (planes).

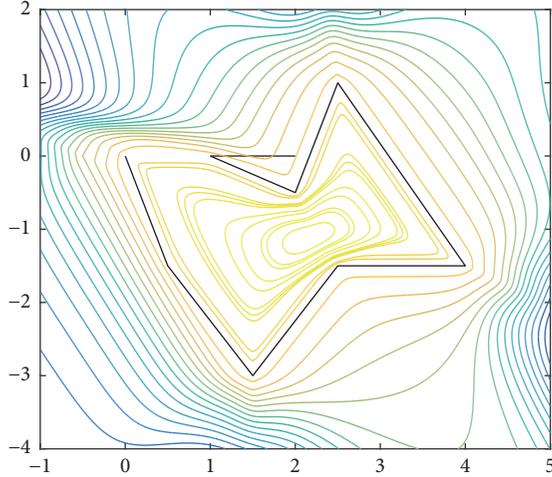


FIGURE 5: The normalized function and its similar signed distance field.

- (2) A convex hull method is introduced to generate CSG representation and logic function for this geometric object.
- (3) The logic function, the implicit function of an edge (plane), and appropriate  $R$ -function system are adopted to construct the implicit function  $g$  for whole geometric object.

Secondly, the implicit function  $g$  is normalized into  $\bar{g}$ .

## 6. Path Planning with Obstacle Avoidance

In obstacle environments, path planning with obstacle avoidance is to find an obstacle avoidance path for a moving part from a given initial posture to the final target posture. It is an optimization problem with noninterference between moving parts, noninterference between a moving part and a barrier, or kinematics constraints. For simplicity, we only discuss path planning with a moving part and noninterference between a moving part and a barrier in this paper. For the environments with  $n$  obstacles, path planning with obstacle avoidance based on typical optimization can be expressed as follows:

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & g_i(x) \leq 0 \quad i = 1, 2, \dots, n. \end{aligned} \quad (19)$$

In the above formula,  $x$  is a motion variable and  $g_i(x) \leq 0$  represents noninterference between a moving part and the  $i$ -th obstacle and  $f(x)$  is an objective function.

For existing path planning approaches with obstacle avoidance based on numerical analysis, most of interference constraints cannot be explicitly defined using the above-mentioned inequality. Thus, two procedures are adopted to solve these problems. First, the interference checking is used to qualitatively determine whether there is interference between a moving part and an obstacle. Second, if it is noninterference, the minimum distance between a moving part and an obstacle and gradient direction of motion parameters are quantitatively calculated, which are used to

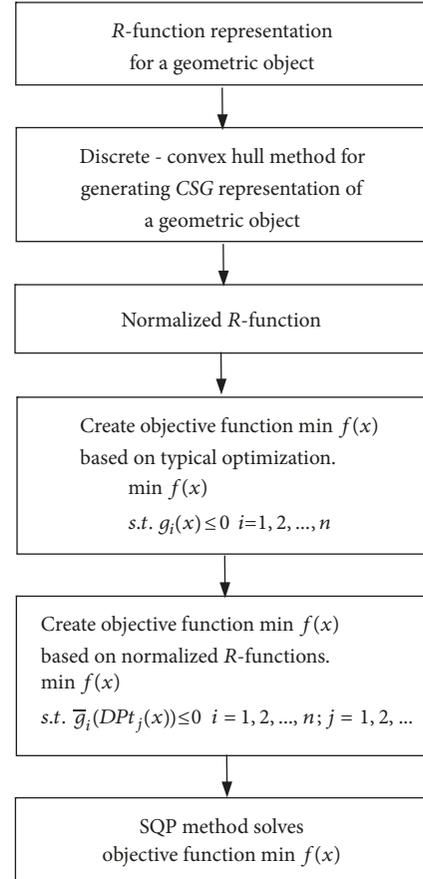


FIGURE 6: The flow diagram of the proposed method.

guide next iteration. Although existing algorithms can deal with them, they usually have heavy computing burden.

Fortunately, normalized  $R$ -function of a geometric object is not only applied to represent interference constraints between two objects, but also there are essential differential properties. Then, (19) can be rewritten as the following forms:

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & \bar{g}_i(DPt_j(x)) \leq 0 \\ & (i = 1, 2, \dots, n; j = 1, 2, \dots). \end{aligned} \quad (20)$$

Here,  $\bar{g}_i$  and  $DPt_j$ , respectively, express normalized  $R$ -function of the  $i$ -th obstacle and a discrete point of moving part  $M$ 's surface.

For path planning with obstacle avoidance, we must ensure that planning paths have noninterference between a moving part and a barrier. Then, it requires that each iteration point will not default for solving (20). The reason is that interference constraint conditions cannot be strictly ensured for some artificial potential field methods [11]. Therefore, a feasible SQP method in [32] is adopted to solve (20). SQP is a local optimum algorithm based on the gradient with the superliner convergence.

Based on the above analysis in Sections 3.3, 4, 5, and 6, the flow diagram of the proposed method is given in Figure 6.

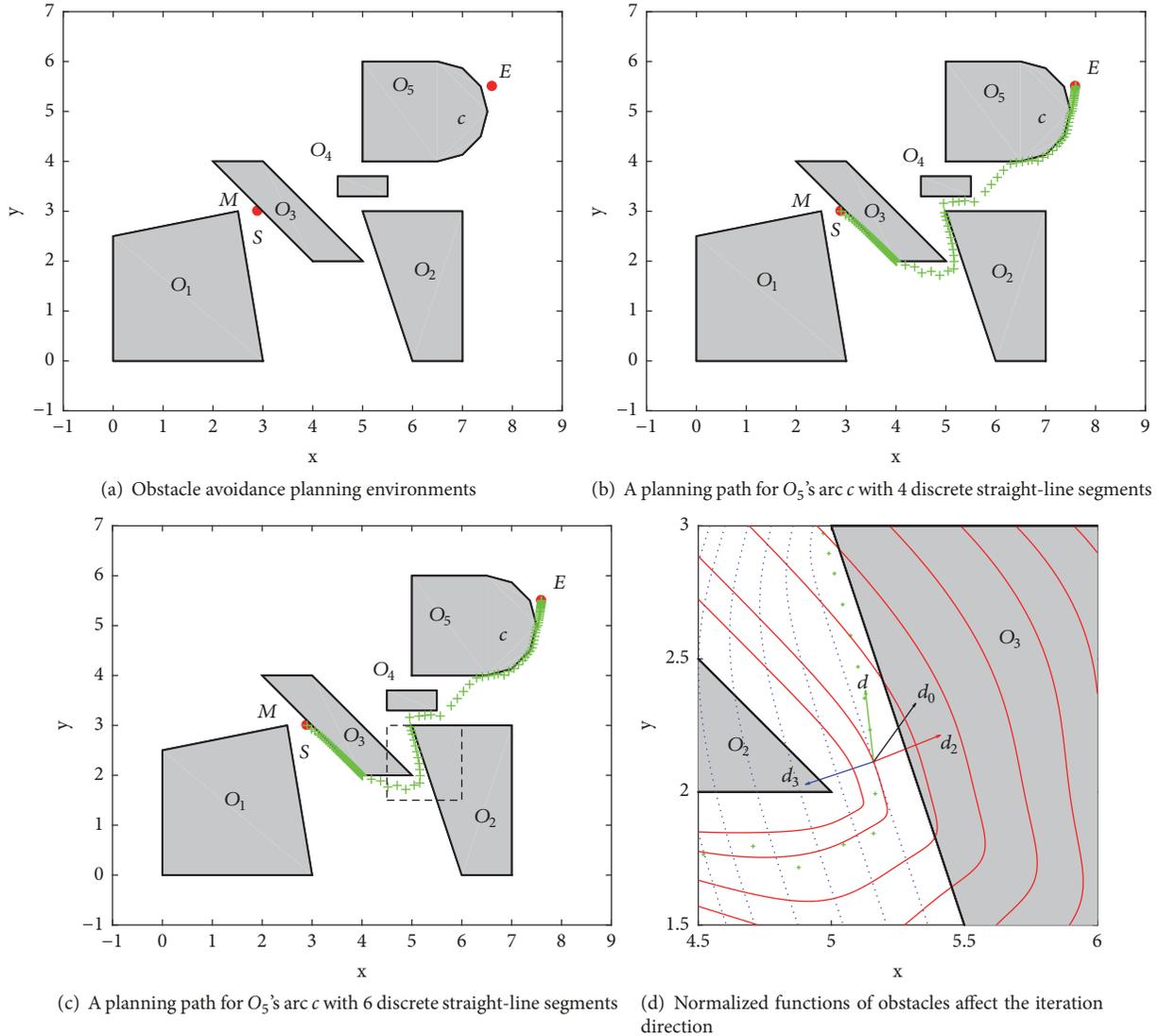


FIGURE 7: Planning path with obstacle avoidance for a single-point moving part.

## 7. Experimental Results

To validate the proposed approach in this paper, some experiments have been performed for path planning with obstacle avoidance on the following three conditions:

- (1) The generation of the implicit function is similar in a two-dimensional space and a three-dimensional space, and the representation of the implicit function is relatively complex in a three-dimensional space. For simplicity, examples are conducted in a two-dimensional space.
- (2) Compromise differential properties and the computational complexity for a normalized function; an  $R_p$ -function system ( $p=2$ ) is adopted to create the implicit function for an obstacle.
- (3) The tests are conducted on a computer with *DELL OPTIPLEX 760* and *Windows XP* and *Matlab* (R2001a).

First, a single-point moving part  $M$  is moved to the target position  $E$  from an initial position  $S$  through the translation movement in  $x$  direction and  $y$  direction, and the obstacles  $O_1, O_2, \dots, O_5$  have been avoided (see Figure 7). Besides, the obstacle  $O_5$  has an arc  $c$ . Then, it is easy to get normalized function of an obstacle and to create interference constraints for (20).

In Figures 7(b) and 7(c),  $O_5$ 's arc  $c$  is, respectively, discretized into 4 and 6 straight-line segments, and their corresponding planning paths are listed. As there is a higher discrete approximation for  $O_5$ 's arc  $c$  in Figure 7(c), its planning path is more reasonable. In general, the curve of an obstacle has more discrete straight-line segments; normalized function is more complex, which will lead to higher computational complexity. The time is spent on planning paths for  $O_5$ 's arc  $c$  with various discrete approximations is given in Table 1.

Figure 7(d) is partial enlargement of dashed boxes in Figure 7(c), which is listed to further illustrate the role of normalized function in path planning. Red and blue plots in

TABLE 1: The time is spent on planning paths for  $O_5$ 's arc  $c$  with various discrete approximations.

The number of discrete straight-line segments for arc $c$	4	6	10	30	100
The time is spent on planning paths(s)	0.6262	0.6434	0.6790	0.92846	1.9188

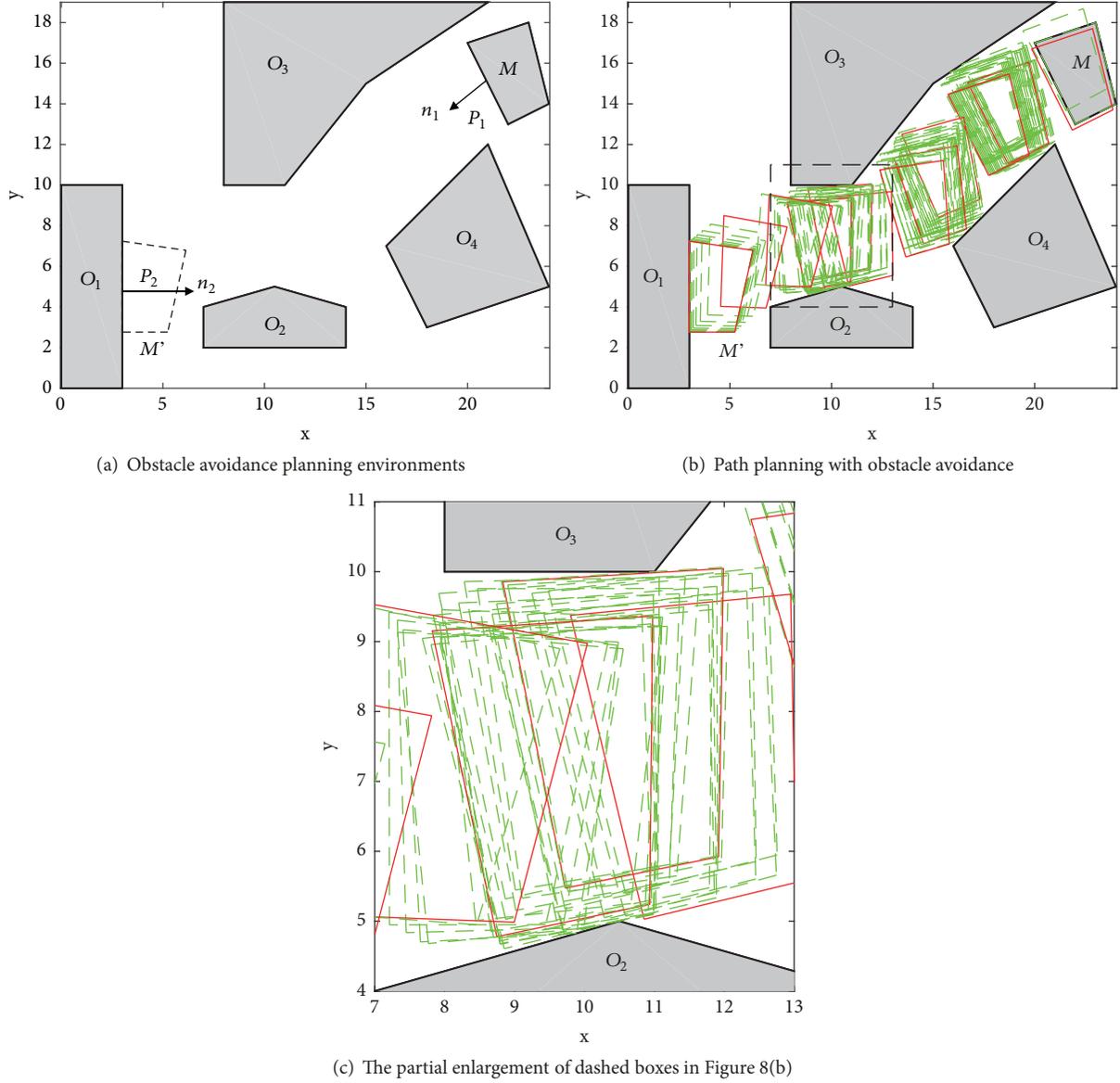


FIGURE 8: Path planning with obstacle avoidance for a polygonal region moving part.

Figure 7(d), respectively, express outer distance field of obstacles  $O_2$  and  $O_3$ . At an iteration point  $I$ ,  $d_2$  and  $d_3$ , respectively, express negative gradient direction of normalized function of obstacles  $O_2$  and  $O_3$ , and  $d_0$  represents negative gradient direction of the objective function. To avoid obstacles  $O_2$  and  $O_3$ ,  $d_2$  and  $d_3$  make final search direction be revised to  $d$  at iteration point  $I$ .

Second, a moving part  $M$  is a polygonal region, which has three degrees of freedom (the translation movement in  $x$  and  $y$  directions, and the rotation motion in  $z$  direction), and  $O_1$ ,  $O_2$ ,  $O_3$ , and  $O_4$  are obstacles (see Figure 8). To

generate interference constraints between a moving part and an obstacle,  $M$ 's boundary is uniformly discretized into 142 points. And  $M'$  is the target posture of a moving part. The translation and rotation motions of a moving part  $M$  make bold edges of  $M$  and the obstacle  $O_1$  are aligned, and their midpoints  $P_1$  and  $P_2$  are coincident in Figure 8(a). Therefore, the objective function for (20) can be expressed as follows:

$$f(x) = W_1 \cos(\langle n_1(x), n_2 \rangle) + W_2 \|P_1(x) - P_2\|^2, \quad (21)$$

where  $n_1(x)$  is a function of external normal vector  $n_1$  of  $M$ 's bold edge for motion parameter  $x$  and  $n_2 \equiv (1 \ 0 \ 0)$  expresses

external normal vector of  $O_1$ 's bold edge and  $P_1(x)$  represents a function of the midpoint  $P_1$  of  $M$ 's bold edge for the parameter  $x$  and  $W_1$  and  $W_2$  describe weight coefficients. As the cosine value of the vectorial angle is smaller ( $\geq -1$  and  $\leq 1$ ), we, respectively, set  $W_1 = 100$  and  $W_2 = 1$ .

Figure 8(b) has listed obstacle avoidance planning paths for a polygonal region  $M$ , and dashed lines represent  $M$ 's posture. To observe clearly, solid lines are adopted to express  $M$ 's typical postures. And Figure 8(c) is partial enlargement of dashed boxes in Figure 8(b). It can be seen from Figure 8, moving part  $M$  successfully avoids obstacles  $O_2$  and  $O_3$  on the basis of posture adjustments. Experimental results have shown that normalized functions of the obstacles can well approximate their signed distanced fields, and their gradient information may guide obstacle avoidance planning paths.

The proposed approach for path planning with obstacle avoidance is based on semianalytic geometry with  $R$ -functions, which has rigorous mathematical analysis. In general, the characteristics of the algorithm are that the results usually have higher accuracy while the processing speed of the algorithm is relatively slow. In particular, the processing time for various instances has greater diversity.

## 8. Conclusions

A path planning method with obstacle avoidance based on normalized  $R$ -functions is presented in this paper. First, an  $R$ -function is introduced to represent a geometrical object, which ensures that normalized  $R$ -function near the boundary of a geometrical object has little difference with signed distance functions, and other parts are continuously and smoothly similar to signed distance functions. Thus, normalized  $R$ -function can be applied to define interference constraints between a space point  $x$  and a geometrical object. As for the generation of interference constraint between two geometrical objects, this needs to plug discrete points of a geometrical object's boundary into normalized  $R$ -function of another object. Finally, path planning with obstacle avoidance for a single-point and polygonal region is conducted to validate the presented approach. Experimental results have shown that the proposed method is a feasible way for path planning with obstacle avoidance.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by research team of Wuhan Technical College of Communication (no. CX2018B03).

## References

- [1] J. C. Latombe, *Robot Motion Planning*, Springer Science & Business Media, 2012.
- [2] V. Shapiro, *Theory of R-Functions and Applications: A Primer*, Cornell University, 1991.
- [3] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.
- [4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [5] D. H. Kim and S. Shin, "Local path planning using a new artificial potential function composition and its analytical design guidelines," *Advanced Robotics*, vol. 20, no. 1, pp. 115–135, 2006.
- [6] J. Lee, Y. Nam, and S. Hong, "New potential functions with random force algorithms using potential field method," *Journal of Intelligent & Robotic Systems*, vol. 66, no. 3, pp. 303–319, 2012.
- [7] J. Ren, K. A. McIsaac, and R. V. Patel, "Modified Newton's method applied to potential field-based navigation for mobile robots," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 384–391, 2006.
- [8] J. Sfeir, M. Saad, and H. Saliha-Hassane, "An improved Artificial Potential Field approach to real-time mobile robot path planning in an unknown environment," in *Proceedings of the 2011 9th IEEE International Symposium on Robot and Sensors Environments, ROSE 2011*, pp. 208–213, Canada, September 2011.
- [9] F. Arambula Cosío and M. A. Padilla Castañeda, "Autonomous robot navigation using adaptive potential fields," *Mathematical and Computer Modelling*, vol. 40, no. 9–10, pp. 1141–1156, 2004.
- [10] L. Yin, Y. Yin, and C.-J. Lin, "A new potential field method for mobile robot path planning in the dynamic environments," *Asian Journal of Control*, vol. 11, no. 2, pp. 214–225, 2009.
- [11] Q. Zhang, S. Yue, Q. Yin, and Y. Zha, "Dynamic Obstacle-Avoiding Path Planning for Robots Based on Modified Potential Field Method," in *Intelligent Computing Theories and Technology*, vol. 7996 of *Lecture Notes in Computer Science*, pp. 332–342, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [12] J. Lee, O. Kwon, L. Zhang, and S.-E. Yoon, "A selective retraction-based RRT planner for various environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 1002–1011, 2014.
- [13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [14] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: a review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
- [15] T. Siméon, J.-P. Laumond, and C. Nissoux, "Visibility-based probabilistic roadmaps for motion planning," *Advanced Robotics*, vol. 14, no. 6, pp. 477–493, 2000.
- [16] J. Kim, M. Kim, and D. Kim, "Variants of the quantized visibility graph for efficient path planning," *Advanced Robotics*, vol. 25, no. 18, pp. 2341–2360, 2011.
- [17] M. Zimmermann and C. König, "Integration of a visibility graph based path planning method in the ACT/FHS rotorcraft," *CEAS Aeronautical Journal*, vol. 7, no. 3, pp. 391–403, 2016.
- [18] E. Masehian and M. R. Amin-Naseri, "A voronoi diagram-visibility graph-potential field compound algorithm for robot path planning," *Journal of Robotic Systems*, vol. 21, no. 6, pp. 275–300, 2004.

- [19] P. Bhattacharya and M. L. Gavrilova, "Roadmap-based path planning - Using the voronoi diagram for a clearance-based shortest path," *IEEE Robotics and Automation Magazine*, vol. 15, no. 2, pp. 58–66, 2008.
- [20] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerospace Science and Technology*, vol. 16, no. 1, pp. 47–55, 2012.
- [21] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., Computer Science Department, Iowa State University, 1998.
- [22] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 2, pp. 231–253, 2013.
- [23] S. M. LaValle and J. J. Kuffner Jr., "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
- [24] A. H. Qureshi and Y. Ayaz, "Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments," *Robotics and Autonomous Systems*, vol. 68, pp. 1–11, 2015.
- [25] V. L. Rvachev, "On the analytical description of some geometric objects," *Reports of Ukrainian Academy of Sciences*, vol. 153, no. 4, pp. 765–767, 1963.
- [26] B. Mirtich, "V-Clip: Fast and robust polyhedral collision detection," pp. 177–208.
- [27] D. Dobkin, L. Guibas, J. Hershberger, and J. Snoeyink, "An efficient algorithm for finding the CSG representation of a simple polygon," *Algorithmica. An International Journal in Computer Science*, vol. 10, no. 1, pp. 1–23, 1993.
- [28] Y. S. Kim and D. J. Wilde, "A convex decomposition using convex hulls and local cause of its non-convergence," *Journal of Mechanical Design*, vol. 114, no. 3, pp. 459–467, 1992.
- [29] Y. S. Kim and D. J. Wilde, "A convergent convex decomposition of polyhedral objects," *Journal of Mechanical Design*, vol. 114, no. 3, pp. 468–476, 1992.
- [30] V. Shapiro, "Well-formed set representations of solids," *International Journal of Computational Geometry & Applications*, vol. 9, no. 2, pp. 125–150, 1999.
- [31] A. Biswas and V. Shapiro, "Approximate distance fields with non-vanishing gradients," *Graphical Models*, vol. 66, no. 3, pp. 133–159, 2004.
- [32] Z. Zhu and K. Zhang, "A new SQP method of feasible directions for nonlinear programming," *Applied Mathematics and Computation*, vol. 148, no. 1, pp. 121–134, 2004.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

