# A Family of Skeletons for Motion Planning and Geometric Reasoning Applications

Ata A. Eftekharian and Horea T. Ilieş*

Department of Mechanical Engineering

University of Connecticut

August 18, 2011

**Abstract**

The task of planning a path between two spatial configurations of an artifact moving among obstacles is an important problem in practically all geometrically-intensive applications. Despite the ubiquity of the problem, the existing approaches make specific limiting assumptions about the geometry and mobility of the obstacles, or those of the environment in which the motion of the artifact takes place.

We present a strategy to construct a family of paths, or roadmaps, for two- and three-dimensional solids moving in an evolving environment that can undergo drastic topological changes. Our approach is based on a potent paradigm for constructing geometric skeletons that relies on constructive representations of shapes with R-functions that operate on real-valued half-spaces as logic operations. We describe a *family of skeletons* that have the same homotopy as that of the environment and contains the medial axis as a special case. Importantly, our skeletons can be designed so that they are "attracted to" or "repulsed by" prescribed spatial sites of the environment. Moreover, the R-function formulation suggests the new concept of a *medial zone*, which can be thought of as a 'thick' skeleton with significant applications for motion planning and other geometric reasoning applications. Our approach can handle problems in which the environment is not fully known *a priori*, and intrinsically supports local and parallel skeleton computations for domains with rigid or evolving boundaries. Furthermore, our path planning algorithm can be implemented in any commercial geometric kernel, and has attractive

*Corresponding author.

computational properties. The capability of the proposed technique are explored through several examples designed to simulate highly dynamic environments.

## 1 Introduction

Motion planning is ubiquitous in many geometrically-intensive applications involving moving objects such as robotics, virtual reality, computer graphics, computer aided design and manufacturing, computer-aided surgery, computational biology and many others. It heavily relies on spatial information about the environment in which the navigation takes place, and typically focuses on generating a 'useful' motion that takes a moving object, a robot, or a vehicle, between two prescribed configurations subject to some constraints such as minimum length of the path, collision-free trajectories, intermediate sites, and so on[1]. One of the most useful and common abstractions of this problem is to transform the motion planning of an object in the Euclidean space into the path planning of a point in the configuration space (or C-space) of the object [Latombe, 1991].

Consider a moving object leaving from an initial position and moving in the Euclidean space $\mathbb{E}^d$ $(d = 2, 3)$ towards a prescribed final destination. Such a vehicle can be a mobile robot, an autonomous vehicle, a camera flying through a scene or collection of spatial entities[2], a mechanical part that is being assembled or transported and so on. The object may be constrained by intermediate sites that must be either approached or avoided during the motion, such as a fueling or a loading/unloading station or a sightseeing location. The environment itself can be static or dynamic, case in which the obstacles themselves or the boundary of the environment are changing during the motion. The task is to find a path from start to finish that satisfies the intermediate site constraints and is subjected to other specific constraints as illustrated in Figure 1. In its most general form, finding a path in such an environment is a hard problem. Consequently, the existing approaches address simplified versions of this problem.

A large amount of research has been devoted to motion planning in *static* environments with both exact and approximate methods. The existing approaches for finding the motion, which cannot be easily extended to dynamic environments, can be classified into:

[1]Motion planning has been recently extended to include differential constraints that restrict allowable velocities during the motion, and use sampling-based planning as discussed, for example, in [LaValle, 2006].

[2]See [Snavely et al., 2008] for a very interesting application.

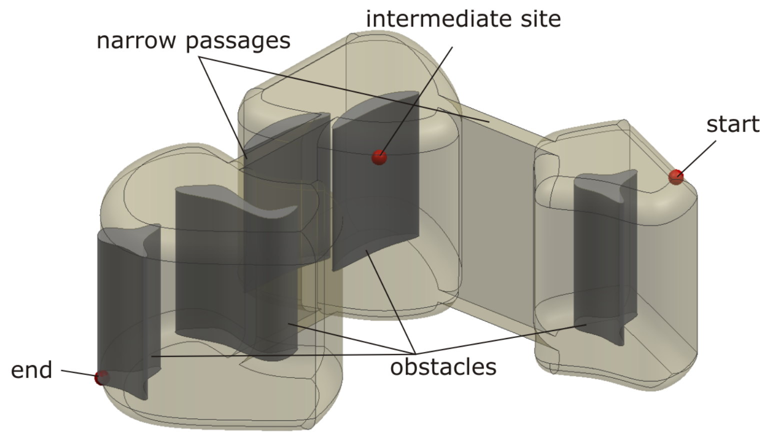


Figure 1: An object moving in an evolving environment with one intermediate site that must be approached during the motion.

---

- *roadmap methods* that plan curves in either the configuration space of the moving object or in the Euclidean space in which the object moves. There has been some success in constructing three-dimensional configuration spaces (see for example [Sacks et al., 1999]), but computing C-spaces explicitly remains a difficult problem, which is exponential in the number of dimensions of the C-space [Latombe, 1991]. Consequently, computing roadmaps in the Euclidean space has remained a strong area of research. Current motion planning algorithms that compute roadmaps rely on visibility graphs [Jiang et al., 1999, Neus M., 2005], Voronoi diagrams [de León S. and Sossa A., 1998, Wilmarth et al., 1999, Foskey et al., 2001, Garber and Lin, 2002, Lee and Choset, 2005, Geraerts and Overmars, 2007], silhouette curves of semi-algebraic sets [Canny, 1993], or probabilistic roadmap planners [Kavraki et al., 1996];

- *potential field methods* that introduce artificial attractive/repulsive field in the environment can conceptually handle both static and moving obstacles [Rimon and Koditschek, 1992, Xidias et al., 2007]. These methods cannot take into consideration the exact geometry of the obstacles, and therefore they cannot be applied to those problems in which the moving object is moving in close proximity with the obstacles. More importantly, potential field methods introduce local minima of the potential field that can "trap" the moving object;

- approaches based on *cell decompositions*, which are inherently approximate. These methods perform a disjoint decomposition of the free space, construct the connectivity graph connecting these cells, and perform graph search algorithms to select a collision free path [Choset, 2000, Lingelbach, 2004].

An approximately time-optimal trajectory is proposed in [van den Berg and Overmars, 2005] that uses a precomputed roadmap for the static part of the scene to determine the collision-free trajectory among the moving obstacles. This approach requires the complete environment and motions to be known ahead of time. Other approaches that incorporate into the formulation time as the extra dimension have been discussed in [Ishikawa, 1991, Fraichard, 1993]. More general approaches to dynamic motion planning couple a pre-computed probabilistic road map and a cell decomposition to dynamically determine which cells of the roadmap are affected by the motion of the obstacles [Kallman and Mataric, 2004]. These methods produce successful paths when the dynamic subset of the environment is much smaller than the whole environment, and when changes to the pre-computed roadmap are fairly minor. Motion planning in dynamic environments in which both moving obstacles and target are moving with known velocities has been addressed in [Masehian and Katebi, 2007]. Because of the assumption of known velocity vectors, this approach cannot handle the case when the number, type, and motion of the moving obstacles are not known in advance.

A fast motion planning algorithm in a 2D dynamic environment based on Generalized Voronoi Diagrams (GVD) has been described in [Hoff III et al., 2000]. The Voronoi decomposition is computed for the prescribed "sites" – the obstacles of the domain. For each pixel in the domain they compute the closest site and shortest distance to that site. All pixels closest to the same site get assigned the same color so that different Voronoi cells will have different colors. The algorithm has been implemented on the GPU which allows the computation of the Voronoi decomposition at each time step.

## 1.1 Main Contributions of Our Work

In this paper we present a strategy to develop a family of paths, or roadmaps, for 2D and 3D solids[3] moving in highly dynamic environments that can undergo drastic topological changes. The main contribution of this paper is the definition of a *class of skeletons* that have the same homotopy as that of the environment and contains the medial axis as a special case. Importantly, our skeletons

[3]These solids are closed, bounded, regular and semi-analytic sets [Requicha, 1980].

can be designed so that they are *attracted* to or *repulsed* by prescribed sites in the environment, while the attraction and repulsion are controlled by a set of degrees of freedom (see section 2.4). This can effectively alter the planned path to accommodate specific geometric constraints such as approaching a re-fueling station or a sightseeing location. Furthermore, we exploit the concept of a medial zone that we introduced in [Eftekharian and Ilieş, 2010] to control the subset of the environment in which the roadmaps are being computed, and we show that, for the same domain, the resulting roadmaps are shorter and smoother than those obtained based solely on skeletons.

Once we construct the corresponding skeletons and medial zones, we use the established Dijsktra's algorithm to compute shortest paths of moving points along the skeletons or within the medial zones. Furthermore, the properties of R-functions afford a conservative motion planning of solid objects within the same formulation via level sets. The approach presented in this paper can handle problems in which one or more objects move in environments that are not fully known in advance, and intrinsically supports local and parallel computations of skeleton and medial zones (and hence path planning) for domains with rigid or evolving boundaries. Our approach can be implemented in any commercial geometric kernel, and has attractive computational properties.

## 2 Problem formulation

In this paper we focus on evolving spatial environments with obstacles that can move as well as merge/split. We assume that the initial and final position (i.e., points in the environment) of the path to be planned are known, and that two lists of obstacles that contain the "attractive" obstacles $\mathcal{A}$ as well as "repulsive" obstacles $\mathcal{R}$ are provided along with positive parameters $\lambda \in \mathbb{R}^+$ that quantify the extent by which each obstacle will attract or repulse the path to be computed (see section 2.4). Each site that must be approached is provided as a point in the environment. The task that we investigate in this paper is to compute the shortest path along the skeleton or within the medial zones that connects the prescribed initial and final positions subject to the intermediate constraints, or output a message that one does not exist for the prescribed environment.

## 2.1 Summary of the Approach

Our approach is illustrated in Figure 2 and consists of five, not necessarily sequential, stages:

1. Construct an exact or approximate distance function over the given semi-analytic domain.

2. Adjust the distance function according to the "attractive" and "repulsive" factors for obstacles prescribed in $\mathcal{A}$ and $\mathcal{R}$.

3. Extract the skeleton of the domain as the subset of points where the continuous distance function is non-differentiable. Note that our construction provides an explicit mapping between each half-space defining the environment and each branch of the skeleton, which can play a critical role in local skeleton computations.

4. Alternatively, compute the medial zones as "thick" versions of the skeletons;

5. Compute the shortest collision free path along the skeleton or inside the medial zones subject to the intermediate sites that must be visited.

Steps 1 and 3 above are summarized in section 2.2 (see also [Eftekharian and Ilieş, 2009]); step 2 is discussed in section (2.3), while the medial zones are summarized in section 3 and described in detail in [Eftekharian and Ilieş, 2010]. Section 4 illustrates the usefulness of our approach for several domains that simulate highly dynamic, topologically evolving domains.

## 2.2 Preliminaries

Geometric skeletons are fundamental concepts in practically all geometrically intensive areas of science and engineering, such as automated finite element meshing, shape manipulation, recognition, and comparison, dimensional reduction in design and analysis, robotic surgery, and a variety of path and motion planning in commercial and defense applications. More recently, skeletons have been used to explore the fundamental geometric problems of folding and unfolding that are the abstraction of some of the most important open problems in science today, such as protein folding, as well as packing and sheet metal bending. There are multiple ways to define a skeleton of a given set, and a variety of definitions have been proposed for different applications. Introduced by Blum

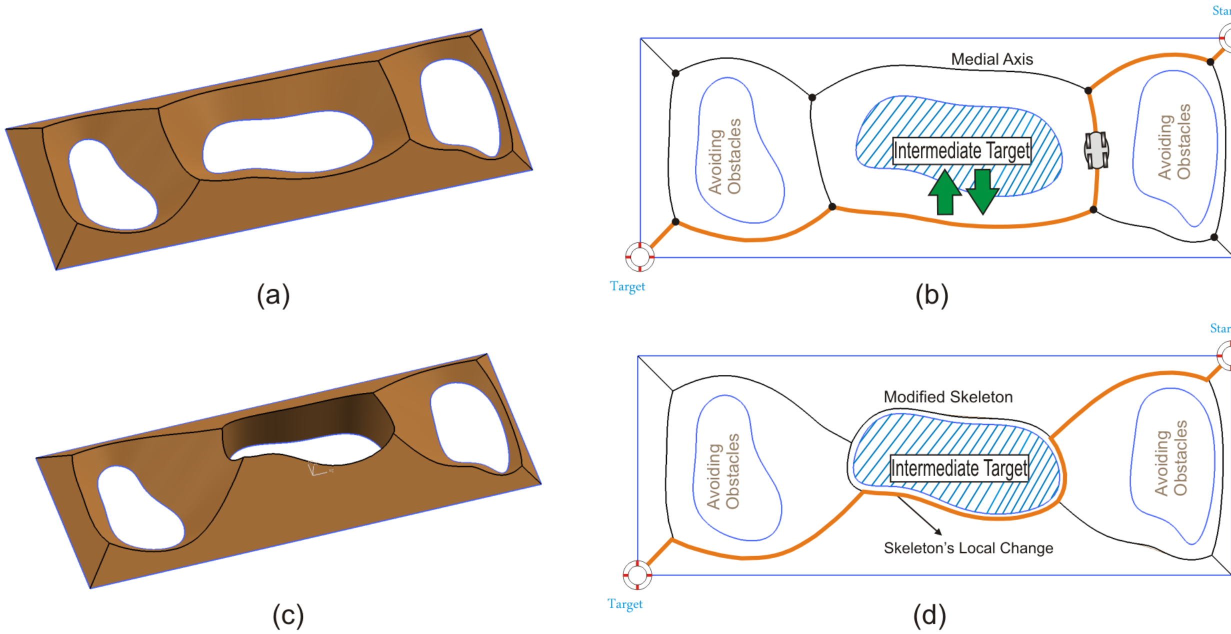


Figure 2: (a) the 3D distance function of the domain constructed with R-functions; (b) the medial axis of the domain and the shortest path that follows the medial axis of the domain; (c) the modified skeleton attracted to the intermediate target and the resulting shortest path.

---

[Blum, 1967] as a tool for image analysis, the medial axis has become one of the mainstream geometric concepts due to the fact that it provides a compact representation of the geometric features of a shape and its topology. The medial axis captures the connectivity of the shape, has a lower dimension than the space itself, and is closely related to the distance function constructed over the same domain.

The concept of medial axis has been described with the help of the fire grass concept (see for example [Attali et al., 2008]) as follows: if a fire starts from all points of a planar curve at the same time and moves with constant velocities in all directions in the same plane, then the medial axis is the locus of points where the fire (in fact a moving front) meets itself. The concept can be extended to $k$-dimensional geometric shapes in $\mathbb{R}^k$, case in which the medial axis becomes a set of dimension $k-1$. Intuitively, the points on the medial axis are equally distant to at least two points of the boundary of the domain. Since its formulation, the medial axis has been used as alternative solid modeling representation [Shaham et al., 2004], as well as in many other applications such as shape matching and reconstruction [van Eede et al., 2006, Liu et al., 1998, Goh, 2008, Sebastian et al., 2004, Siddiqi et al., 1999, Damon, 2005], dimensional reduction in boundary value problems [Suresh, 2003], representation and classification of 2D shapes [Sherbrooke et al., 1995, Pizer et al., 2003, Shah, 2005], human vision [Kimia and Tamrakar, 2002], pattern analysis and shape recognition [Bookstein, 1979, Blum and Nagel, 1978], mesh generation [Sampl, 2000, Sampl, 2001, Quadros et al., 2004], and so on.

The mathematical properties of medial axis are fairly well understood [Attali et al., 2008, Chazal and Soufflet, 2004, Choi et al., 1997]. However, the practical uses of these skeletons are limited by their notorious computational difficulties, and their instability under small perturbations of the shape. Although it is possible, in principle, to compute the medial axis exactly for general semi-algebraic/analytic sets, we do not seem to have any algorithms for doing so. The most advanced algorithms make significant simplifying assumptions on the geometry of these sets by focusing on planar or piecewise linear shapes. Algebraic planar curve segments whose bisectors admit rational parametrizations are examined in [Ramamurthy and Farouki, 1999], while exact computation of medial axis for polyhedra is described in [Culver et al., 2004]. The reasons behind these restrictions become apparent once the algebraic difficulties in computing MA are examined [Attali et al., 2008].

Consequently, these difficulties promoted a variety of algorithms that approximate the complex shape by a set bounded by a piecewise linear boundary for which the medial axis can be computed exactly, followed by the extraction of the medial axis of the approximating shape – the so called pruning step. The computed medial axis has to be post-processed in order to eliminate the branches that appear in the medial axis due to the shape approximation. The main approaches to approximately compute the medial axis for piecewise linear shapes rely on: Voronoi/Delaunay decompositions of space [Brandt and Algazi, 1992, Amenta et al., 2001, Dey et al., 2003] where the medial axis is the Voronoi graph defined by a piecewise linear approximation of the shape [Attali et al., 2008]; solutions of partial differential equations (such as diffusion or Hamilton-Jacobi equations) [Siddiqi et al., 2002, Du and Qin, 2004]; and level set methods [Kimmel et al., 1995, Gomes and Faugeras, 2000]. A recent method to compute the medial axis for curved planar sets [Cao and Liu, 2008] is iteratively tracing the Frenet frames for pairs of boundary curves that bound a closed planar domain. This algorithm does not generalize to 3D space and its stability appears to be problematic in 2D due to its iterative marching along the boundary curves.

### 2.2.1 R-functions as Logic Operators on Real-Valued Half-spaces

For any closed subset $\Omega$ of $\mathbb{E}^d$, one can construct a $C^\infty$ function that vanishes on the boundary $\partial\Omega$ of $\Omega$. It is known that such shapes are in fact semi-analytic sets of points that can be constructed as a finite Boolean combination of real analytic functions $f_i \geq 0$. This, in turn, suggests an approach to construct a $C^n$ function over a semi-analytic subset $\Omega$ of $\mathbb{E}^d$ by subdividing the boundary of $\Omega$ in primitive half-spaces $f_i$, followed by a combination of $f_i$ into a single predicate using the standard Boolean logic operators AND, OR or NOT [Shapiro, 2007].

R-functions have been invented in the 60's by V.L. Rvachev, who called these functions "logically charged functions". These functions provide the means to construct a $C^n$ function over a domain defined by primitive half-spaces. The main contribution of the theory of R-functions to the topic of this paper is to replace these logical operations by real-valued functions, which generates an implicit representation for any semi-analytic set $\Omega$. One important feature of these real valued functions is that their sign is *completely* determined by the sign of their arguments, and is independent of any of their magnitudes.

There are many systems of sufficiently complete R-functions. One such system is known as principal system of R-functions

$$R_\alpha(\Delta): \qquad \frac{1}{1+\alpha}\left(x_1 + x_2 \pm \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2},\right) \tag{1}$$

where $(+)$ and $(-)$ signs correspond to the R-conjunction $(x_1 \vee_\alpha x_2)$ and R-disjunction $(x_1 \wedge_\alpha x_2)$ respectively of two real variables $x_1$ and $x_2$. By varying the value of $\alpha$, we obtain different systems of R-functions. In particular, by setting $\alpha = 1$ in equation (1) we obtain the $R_1(\Delta)$ system of R-functions, while a value $\alpha = 0$ in equation (1) would result in the $R_0(\Delta)$ system (see Figure 3).

Importantly, when expression under the square root in equation (1) becomes zero, the corresponding implicit function becomes non-differentiable. Since the medial axis of a semi-analytic domain corresponds to the points where the distance function is non-differentiable, we will be looking for points of the implicit function where the expression under the square root vanishes.

Figure 3 shows the difference between the implicit functions constructed over a polygonal domain with the $R_\alpha(\Delta)$ system for three values of $\alpha$. One can see that for $\alpha \neq 1$, the resulting functions are differentiable (except at the origin - not shown). On the other hand, the implicit function corresponding to $R_1(\Delta)$ shown in Figure 3(c) is made of piecewise planar patches that intersect at

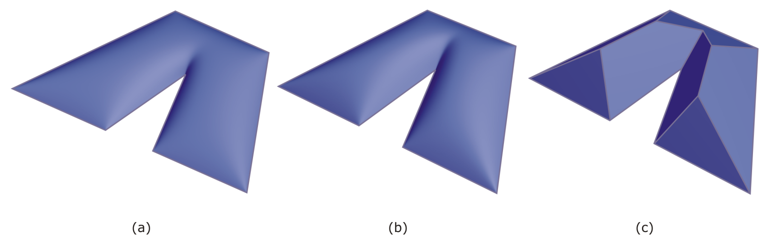


Figure 3: Three implicit representations of a polygonal domain that correspond to (a) $\alpha = 0$; (b) $\alpha = 0.5$; and (c) $\alpha = 1$.

---

piecewise linear edges. In fact, by projecting the edges of the R-function surface on the plane of the polygon we obtain a skeleton of the polygon. But how do we construct the function, and which skeleton do we obtain?

#### 2.2.2 Constructing Boolean Expressions

Conceptually, the problem of constructing a Boolean expression for a domain bounded by half-spaces is the same as that of converting a Boundary Representation (B-rep) into a Constructive Solid Geometry (CSG) representation in Solid Modeling.

Several efficient algorithms for performing the B-rep to CSG conversion for polygons are known [Tor and Middleditch, 1984, Dobkin et al., 1988]. However, the complexity of the problem quickly escalates with the increase in the complexity of the boundary of the domain. Separating the boundary into primitive pieces is no longer sufficient to construct the Boolean expression of the domain, because additional half-spaces need to be introduced – the so-called separating half-spaces. Determining a sufficient set of separating half-spaces is the critical step of any such algorithm, but the problem is not well understood in general. Solutions exist for planar domains bounded by linear and curved edges that are subsets of convex curves [Shapiro, 2001], and solids bounded by linear or quadric surfaces [Shapiro and Vossler, 1993].

Note that the problem of B-rep to CSG conversion is well defined by using the natural half-spaces of the domain, but a CSG expression may not exist for a given set of half-spaces. This necessary and sufficient condition for the existence of a CSG expression is formally described by the 'Describability Theorem' in [Shapiro and Vossler, 1991]. However, if a *canonical* CSG expression exists for a given set of half-spaces bounding a specific domain, then this canonical CSG expression is unique. Observe that we do not require the canonical CSG expression, since all CSG expressions that are valid for a given domain will describe the *same* set of points.

**Planar Domains**

The Boolean set representation of a polygonal domain can be computed based on the Convex Deficiency Tree [Dobkin et al., 1988], which treats each polygon as its convex hull *minus* a finite number of concavities. Note that the polygon is a closed set, so the subtraction of concavities must necessarily be regularized (i.e., one must use regularized Boolean operations). Finally, we perform a syntactic substitution to replace the union and intersection with the R-disjunction and R-conjunction given in equation (1), which results in an R-function expression whose zero set is the original planar domain. By following this procedure, we obtain an R-function expression that corresponds to the exact distance function for any convex planar domain polygon, and an approximate distance function for a concave domain. Note that such a distance function is obtained from the principal system of R-functions given in equation (1) by setting the value of $\alpha = 1$, i.e., the $R_1(\Delta)$ system. Values of $\alpha < 1$ correspond to implicit functions over the same domain that have established differential properties [Shapiro, 2007]. These approximate distance functions can be converted into an exact distance function by introducing additional half-spaces at the concave vertices of the domain [Eftekharian and Ilieş, 2009]. Specifically, each concave vertex requires a conical half-space with a half angle of $\pi/4$ and two separating/trimming half-spaces that are normal to the boundary curves that are incident at the concave vertex.

**3D domains**

It is important to note that the construction algorithms for simple polygons can be extended to some other point sets, such as curved polygons [Shapiro, 2001], 3D polyhedra, and more general 3D solids [Shapiro, 1991, Buchele and Crawford, 2003]. However, converting the resulting approximate distance function into an 'exact' distance function for such domains requires the computation of additional halfspaces as discussed above, which can only be computed approximately in many situations. The alternative that we take in this paper is to compute distance functions to each face bounding the 3D domain, and combine them with the R-disjunction as described in more detail in [Eftekharian and Ilieş, 2010].

#### 2.2.3 Adding/Removing Obstacles

Obstacles in the environment can be represented as holes/voids of the domain. One of the main advantages of our approach to compute skeletons based on R-functions is that obstacles can be easily added to the R-function expression representing $\Omega$ (see also [Eftekharian and Ilieş, 2009]. The R-function expression corresponding to the obstacle is subtracted from the R-function expression defining the boundary of $\Omega$ according to the R-subtraction[4]:

$$R_1(\Delta) = \frac{1}{1+\alpha}\left(f_1 - f_2 - \sqrt{f_1^2 + f_2^2 + 2f_1 f_2}\right) \tag{2}$$

where $f_1$ is the function describing the outer domain and $f_2$ represents the hole. Figure 4 shows how obstacles can be added to the main space. Figures 4 (a-c) illustrate this process in which two obstacles are sequentially added to the environment, then one of them is rotated. The corresponding distance functions are illustrated in Figures 4(d-f).

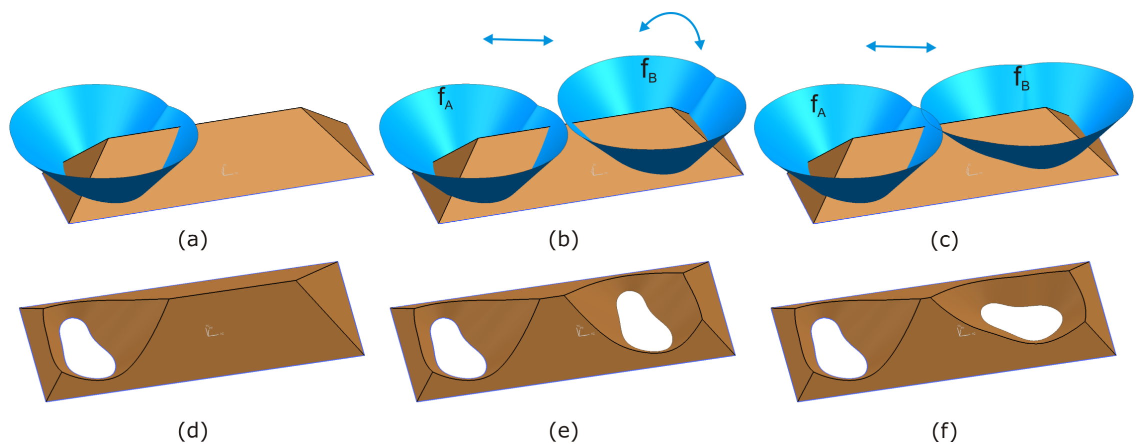


Figure 4: Obstacles can be added to (or removed from) the environment. Distance functions to individual obstacles are illustrated in (a)-(c). The resulting distance functions of the domain are shown in (d)-(f).

---

One important property of the R-functions is that their sign is independent of the magnitude of each half-space involved in the R-function expression, and depends only on the sign of each half-

[4]Recall that the subtraction of two point sets $A$ and $B$ is usually defined in terms of the Boolean operations as $A - B = A \cap B^c$, where $B^c$ is the complement of $B$, which implies that for solids one must use regularized set operations.

space. Consequently, holes can be easily relocated and re-sized to perform, for example, parametric or topology optimization [Luo et al., 2007, Chen et al., 2007]. These modifications exploit the connection between implicit functions defined with R-functions and level sets so that the boundary of the domain defined by the R-function expression is the zero level set of the corresponding implicit function, and moving boundaries of $\Omega$ can be handled as easily [Eftekharian and Ilieş, 2009].

### 2.3 From Distance Functions to Collision-Free Trajectories

Constructing the R-function expression described above results in the (exact in 2D and approximate in 3D) distance function of any bounded, regular, closed and semi-analytic domain. The ridges of the distance function contain the points where the distance function is non-differentiable, and therefore, belong to the medial axis of the domain.

In [Eftekharian and Ilieş, 2009] we explored two strategies for extracting the ridges of the distance function. Specifically,

- for the planar case, if the distance functions $H_i$ corresponding to each individual half-space $f_i$ are constructed in a commercial geometric kernel as a standard 3-dimensional NURBS surface, then the distance function corresponding to domain $\Omega$ can be obtained by combining the individual $H_i$ according to the Boolean expression developed in the previous step. This outputs individual curve segments/branches of the medial axis.

- alternatively, one can take advantage of the fact that the distance function is not differentiable at the ridge points. If one assumes that the first derivative of a given continuous function is large (it reaches a local extremum) wherever the second derivative has a zero crossing, one can use Laplacian based methods for ridge detection, which are widely used in image analysis and computer vision. This leads to a numerical procedure that outputs points of the medial axis, which needs to be followed by a segmentation of these points into branches of the medial axis. One practical way to perform the medial axis segmentation (or branching) is to determine the half-spaces $H_i$ that evaluate to zero for each point of the medial axis. The advantage of this approach is that it can be applied to both 2D and 3D path planning problems.

Regardless of the ridge extraction algorithm being employed, by constructing a single function describing the boundary of each obstacle, we can significantly reduce the number of branches of the medial axis, which will significantly reduce the computational cost of the path planning algorithm. This is illustrated in Figure 5 which shows a polygonal domain and its exact distance function in (a). The rectangular obstacle is bounded by four half-spaces $H_1 \dots H_4$, to which we add trimmed conical half-spaces $C_1 \dots C_4$. These half-spaces can be treated individually or collectively during the segmentation, as mentioned above. Note that even though we use linear half-spaces for illustrative purposes, the approach remains valid for all semi-analytic half-spaces.

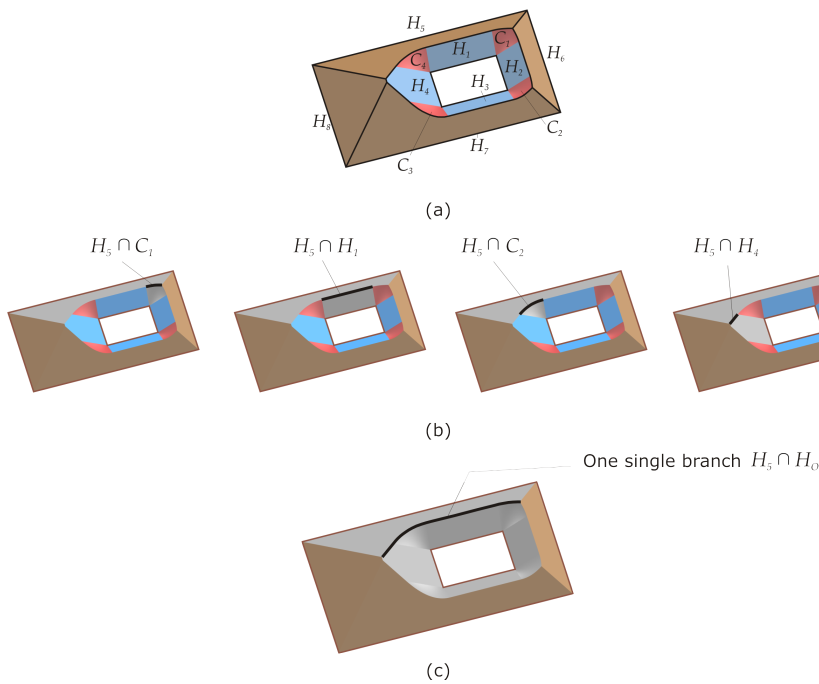


Figure 5: Combining the half-spaces defining the obstacles into a single function reduces the number of branches of the medial axis.

---

If each individual half-space bounding the obstacle is treated individually during the segmentation, then one half-space defining the outer boundary of the environment, say $H_5$, will generate four separate branches of the medial axis as shown in Figure 5(b). Alternatively, if we construct one single Boolean expression $H_O$ for the boundary of the obstacle as:

$$H_O = (H_1 \cup H_2 \cup H_3 \cup H_4) \cup (C_1 \cup C_2 \cup C_3 \cup C_4) \tag{3}$$

then the segmentation will effectively combine the four individual branches of the medial axis into one single branch (see Figure 5(c)). Repeating this procedure for the remaining outer boundary of

$\Omega$, we will generate the remaining branches of the medial axis according to:

$$\begin{aligned}\Omega =& [(H_1 \cup H_2 \cup H_3 \cup H_4) \cup (C_1 \cup C_2 \cup C_3 \cup C_4)] \cap \\ & \cap (H_5 \cap H_6 \cap H_7 \cap H_8)\end{aligned} \tag{4}$$

and replace the Boolean operations with the R-conjunction and R-disjunction given in equation (1).

Figure 6 illustrates the impact of the two approaches on the number of branches of the medial axis for a simple planar domain. In this example, by combining the half-spaces bounding each obstacle as described above, we reduce the number of branches of the medial axis from 30 to 14.

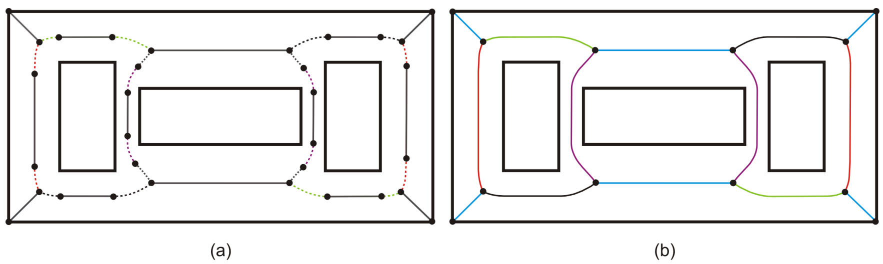

Figure 6: The half-spaces bounding each individual obstacle are combined so that each obstacle will be defined by one single function, which reduces the number of medial axis branches from 30 to 14. Dashed lines in (a) are branches created by intersection of conical half-spaces with the outer half-spaces (see section 2.2.2). The black dots on the medial axis separate the branches of the medial axis.

---

After the medial axis segmentation, the last step in planning a path for a moving point is to perform a graph search algorithm on a weighted graph. Each critical point becomes a vertex in this graph and each branch will be represented as a weighted edge whose weight is proportional to the length of the branch. In this work we used the established Dijkstra's algorithm [Dijkstra, 1959] that has been widely used in motion planning and routing problems due to its simplicity, efficiency and robustness. However, one can use any of the other existing graph search algorithms to generate collision-free paths in this weighted graph.

Note that some domains admit more than one shortest path, and that all standard graph search algorithms can be modified to produce all possible shortest paths. The interested reader is referred to [Takaoka, 2005] as well as to standard textbooks on algorithms such as [Dasgupta et al., 2008].

## 2.4 A Family of Skeletons that Contains the Medial Axis

To summarize the discussion above, we are first constructing the distance function over the domain from individual half-spaces defining the boundary of the domain (both outer boundary and obstacles) as described in section 2.2.2. As a result, we obtain a continuous piecewise smooth surface, whose zero level set is the boundary of the domain. This construction is followed by the extraction of the medial axis as the subset for which the continuous distance function is non-differentiable.

However, by altering the distance functions defined by each half-space $f_i \geq 0$, we can *deform* the medial axis such that the deformed skeleton will be "attracted" or "repulsed" by specific obstacles inside the domain or by specific half-spaces defining the outer boundary of the environment.

In our construction above, we define each half-space of the original domain as an implicit function $f_i \geq 0$. Each half-space $f_i$ in the $d$ dimensional space $\mathbb{E}^d$ induces a higher dimensional half-space in $\mathbb{E}^{d+1}$ denoted by $H_i \geq 0$.

We will describe the concept with the help of a 2D example, but this discussion naturally extends to 3D environments. Consider the example shown in Figure 7, which shows two planar half-spaces $f_1$ and $f_2$, as well as their exact distance functions that are the boundaries of the corresponding 3D half-spaces defined by equation $H_i = 0$, namely

$$\partial H_1 = \{\mathbf{P}(x, y, f_1(x, y)) \in \mathbb{E}^3 \quad | \quad H_1(x, y, f_1(x, y)) = 0\} \tag{5}$$

$$\partial H_2 = \{\mathbf{P}(x, y, f_1(x, y)) \in \mathbb{E}^3 \quad | \quad H_2(x, y, f_2(x, y)) = 0\} \tag{6}$$

Assume that the two surfaces $\partial H_1$ and $\partial H_2$ are the exact distance functions to curves $f_1 = 0$ and $f_2 = 0$, and that point $\mathbf{P}$ is on the intersection curve between $\partial H_1$ and $\partial H_2$ as illustrated in Figure 7. By definition, both $\partial H_1$ and $\partial H_2$ make with the $(x, y)$ plane an angle of $\pi/4$ at the footpoints of $\mathbf{P}$ on $f_1$ and $f_2$, for all points $\mathbf{P} \in \partial H_1 \cap \partial H_2$. Denote these footpoints by $\mathbf{P}_{f1}$ and $\mathbf{P}_{f2}$. Furthermore, assume that $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are two positive real numbers that multiply $H_1$ and $H_2$ respectively such that

$$WF_1 = \lambda_1 H_1(x, y, f(x, y)), \qquad WF_2 = \lambda_2 H_2(x, y, f(x, y))$$

Clearly, the functions $WF_1$, and $WF_2$ are distance functions for curves $f_1(x, y) = 0$ and $f_2(x, y) = 0$ if and only if $\lambda_1 = \lambda_2 = 1$. In this case, by projecting the ridge between $WF_1$,

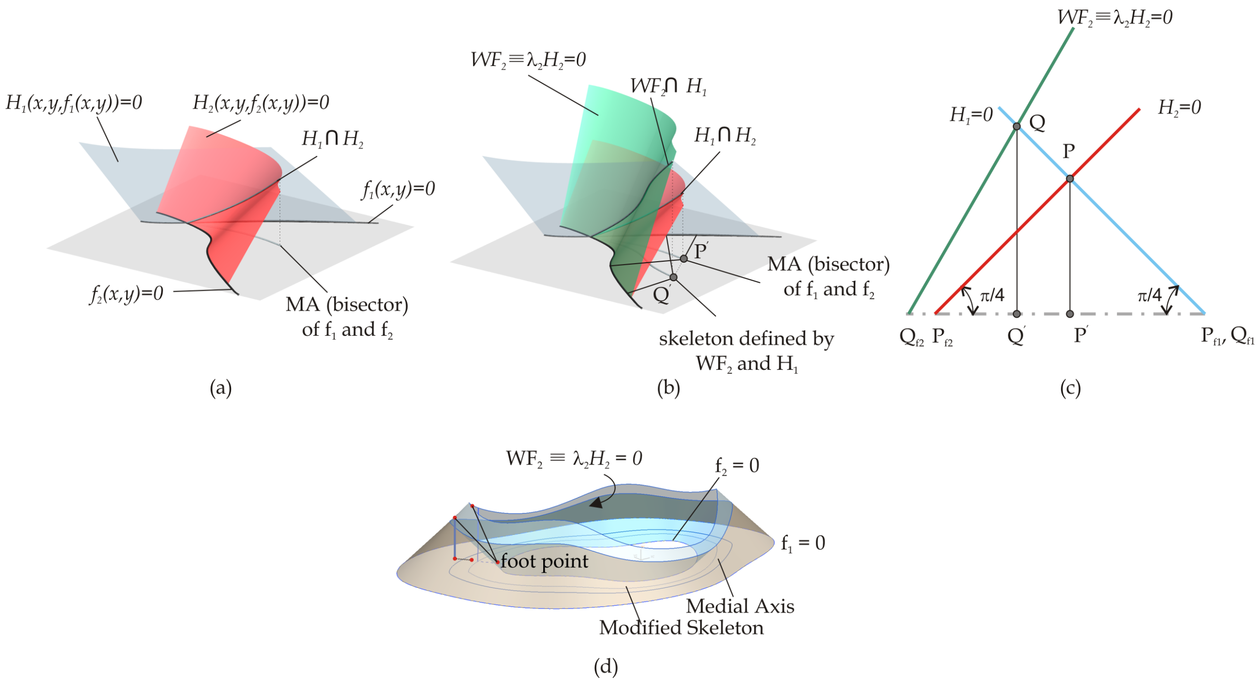


Figure 7: (a) Exact distance functions $H_1 = 0$ and $H_2 = 0$ for two planar curves defined by $f_1(x, y) = 0$ and $f_2(x, y) = 0$; (b) A weighted distance function $WF_2 = \lambda_2 H_2 = 0$ results in a different skeleton than the medial axis; (c) the triangles formed by $\mathbf{P}_i$, $\mathbf{P}_i^{'}$ $(i = 1, 2)$ and the corresponding footpoints flattened on the plane; (d) a single weight $\lambda$ is applied on the distance function defining the obstacle $f_2 = 0$ regardless of how many half-spaces contribute to $f_2 = 0$.

---

and $WF_2$ onto the plane of $f_1$ and $f_2$ we obtain the bisector of $f_1$ and $f_2$, i.e., the medial axis. However, if we choose $\lambda_1 = 1$, but we let $\lambda_2 > 1$ or $\lambda_2 < 1$, the projection of the ridge will move closer to or farther away from $f_2 = 0$ as illustrated in Figure 7(b). Consequently, the reals $\lambda_i$ act as weights and therefore the resulting R-function expression for the domain generates a weighted distance function (denoted here by $WF$). In fact,

$$\theta_i = tan^{-1}(\lambda_i)$$

where $\theta_i$ is the angle formed by $WF_i = 0$ and the plane $(x, y)$ at the footpoint $\mathbf{P}_{fi}$ on $f_i = 0$ as shown in Figure 7(c). The fact that each scalar $\lambda_i$ is constrained to be positive definite implies that the angle $\theta_i \in [0, \frac{\pi}{2}]$.

In other words, scalars $\lambda_i$ act as degrees of freedom controlling the local attraction or repulsion of the skeleton by $f_i$. Furthermore, since each $\theta_i \in [0, \frac{\pi}{2}]$, the branch of the skeleton defined by two half-spaces $f_i \geq 0$ and $f_j \geq 0$ will lie in the region of the plane where both $f_i > 0$ *and* $f_j > 0$, i.e., between $f_i > 0$ *and* $f_j > 0$.

In this paper, we assign a separate scalar $\lambda$ to the distance function that defines each obstacle. In other words, each $\lambda$ will act as a global weight for the corresponding obstacle as shown in Figure 7(d). If the function defining each obstacle of the domain, which is represented by one single R-function expression, is weighted by a single weight $\lambda$, we conjecture that the resulting skeleton of the domain is homeomorphic to the medial axis, which in turn preserves the homotopy of the domain [Lieutier, 2003]. In this case, the class of skeletons controlled by $\theta_i$ will have the same homotopy as that of the domain.

## 3 Medial Zones as Thick Skeletons

We have been using the principal system of R-functions given in equation (1) with a value of $\alpha = 1$. This in turn generates piecewise smooth distance functions as shown in Figure 3(c), and the skeletons are obtained as the non-differentiable points of the distance function as discussed in sections 2.3 and 2.4. However, one important feature of R-functions is that they can generate continuous functions over the domain that are differentiable almost everywhere (see [Shapiro, 2007] and [Shapiro and Vossler, 1991] for a detailed discussion). In fact this is illustrated in Figure 3 for the principal system of R-functions of equation (1), which produces such differentiable functions by simply choosing a value of $\alpha < 1$.

For a given R-function expression corresponding to a given domain we define the *medial zones* as those points of $\mathbb{E}^d$ that belong to the surface constructed with R-functions, and are "near" the crest of the surface[5], as illustrated in Figure 8. The "nearness" can be defined in several different ways, and can be computed via Laplacian, Gaussian curvature or other differential properties of the resulting surface. Note that the differential properties of the resulting surfaces can be controlled by the specific system of R-functions being used. Consequently, by adjusting the value of $\alpha$ and the "nearness" metric, one can 'grow' or 'shrink' these medial zones. The effect of changing $\alpha$ on the resulting medial zone is illustrated in Figures 8(b) and (c), as detailed in

[5]Recall that the R-function expression generates the surface in the $\mathbb{E}^{d+1}$ space.

[Eftekharian and Ilieş, 2010].

The implications to motion planning are important because these *medial zones* provide significantly increased flexibility of the resulting path compared to the case in which the paths are planned along the medial axis or skeleton of the same domain. Moreover, these medial zones can be computed with the *same* computational algorithms as those used to compute the skeletons described above. From a practical point of view, the paths planned based on the *medial zones* can be *shorter* and *smoother* than those planned based on skeletons, which is evident in the examples discussed in section 4.

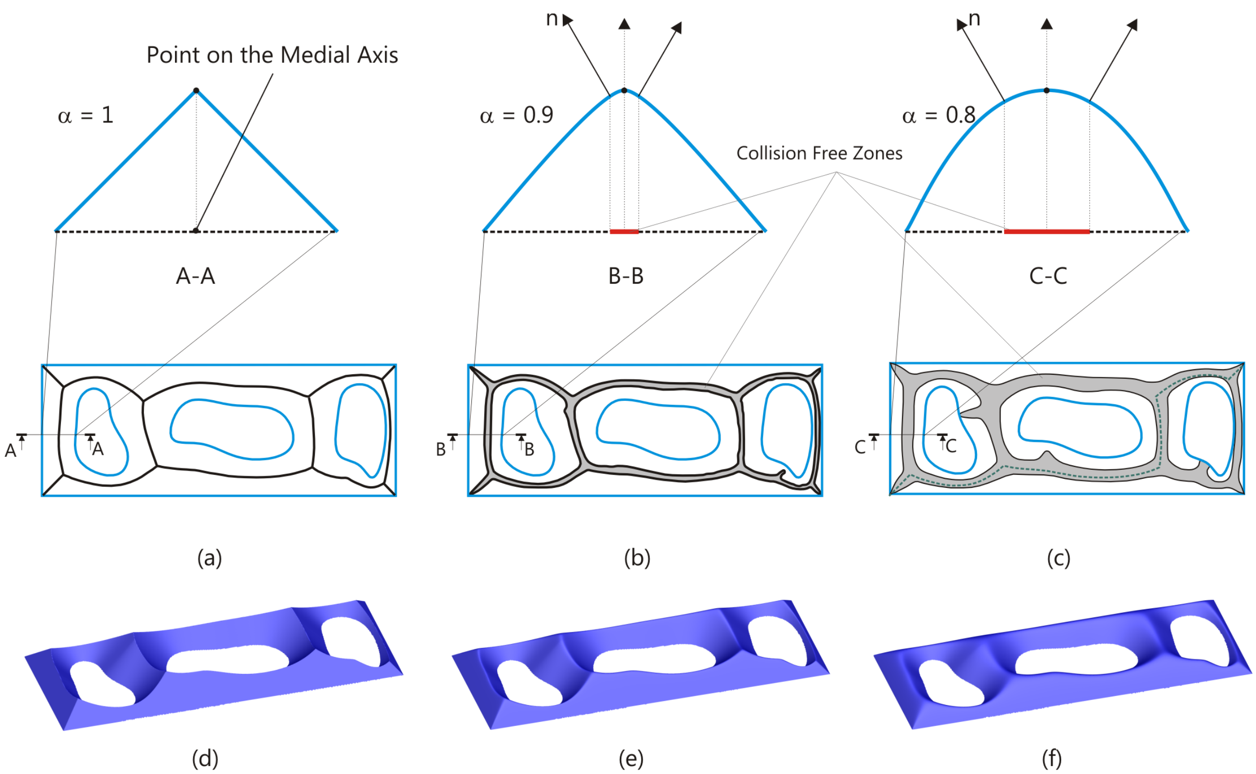


Figure 8: Medial zones controlled by prescribed $\alpha$ values.

## 4 Examples

We illustrate the capabilities of our approach with several examples that contain both linear and non-linear half-spaces in both 2D and 3D, as well as moving obstacles that merge with either other obstacles or the boundary of the domain $\partial\Omega$ to produce drastic topological changes of the environment. We implemented a commonly used finite-difference approximation of the Laplacian [Bovik, 2005] to extract the ridges and ravines of the weighted distance function.

Figure 9(a) shows the distance function of a domain and the corresponding shortest trajectory of a point between an initial and a final position. By taking a different level set of the distance function, we can plan within the same formulation the path of an object approximated by the smallest enclosing ball as shown in Figure 9(b).

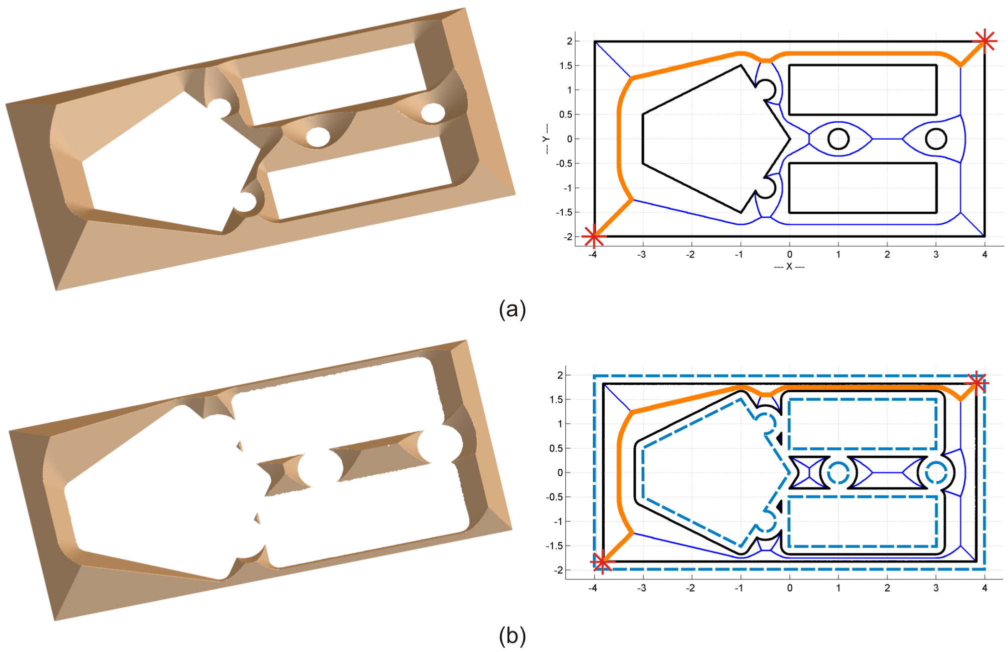


Figure 9: Trajectory planning of a point is shown in (a). By taking different level sets of the implicit function induced by the R-function expression, we can plan the path of an object approximated by the smallest enclosing ball (b).

---

The example in Figure 10 shows three instances during the evolution of an environment in which two obstacles are moving independently while merging and splitting with other obstacles. The planned path adapts to the changing environment as shown in the third row of images in the same figure.

The environment from Figure 10(b) is illustrated in Figure 11, but with a different path planning goal, namely to have 'sufficient' clearance to the boundary of $\Omega$. The corresponding weight $\lambda$ corresponding to the R-function expression defining the outer boundary of the domain is set to 0.2679 (equivalent to an angle of 15 degrees - see section 2.4). Note the difference between the

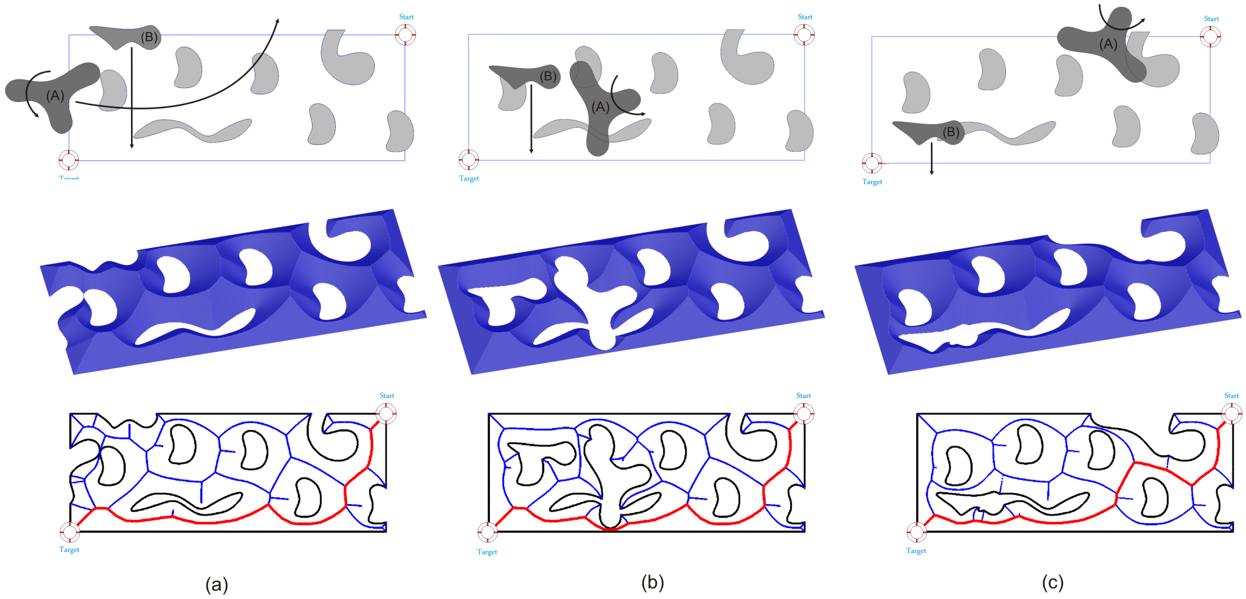


Figure 10: An environment with both static and dynamic obstacles moving independently. The planned path adapts to the changing environment.

resulting shortest path in Figure 11 and that of Figure 10(b).

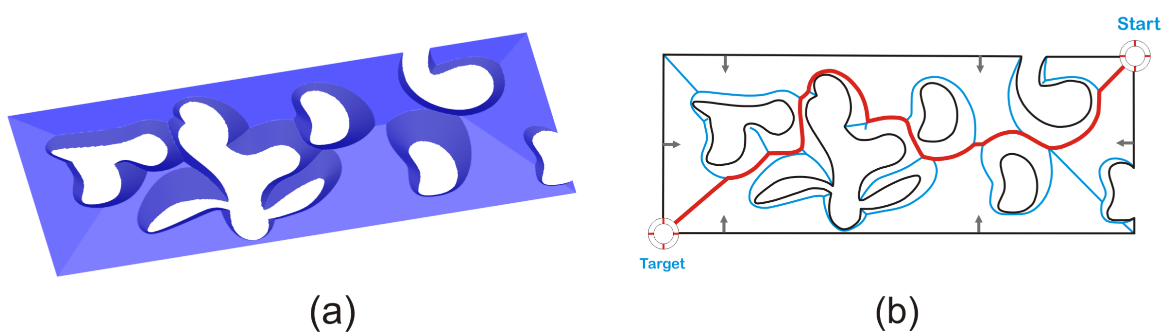


Figure 11: An environment with obstacles similar to the one in figure 10(b). The weight of the outer boundary is $\lambda = 0.2679$, which is equivalent to an angle of $15^o$.

Figure 12 shows the path planning for a point within a domain with curved boundaries (a) and a second path of an object enclosed by a minimal ball while moving in the same environment. Note that the path planned for the object is different than the path of the point even though the space is the same. This is due to the fact that planning the path of the object is transformed into the path planning of a point by taking a different level set of the R-function expression describing the original environment. In this example the radius of the minimal enclosing ball is $0.3m$.

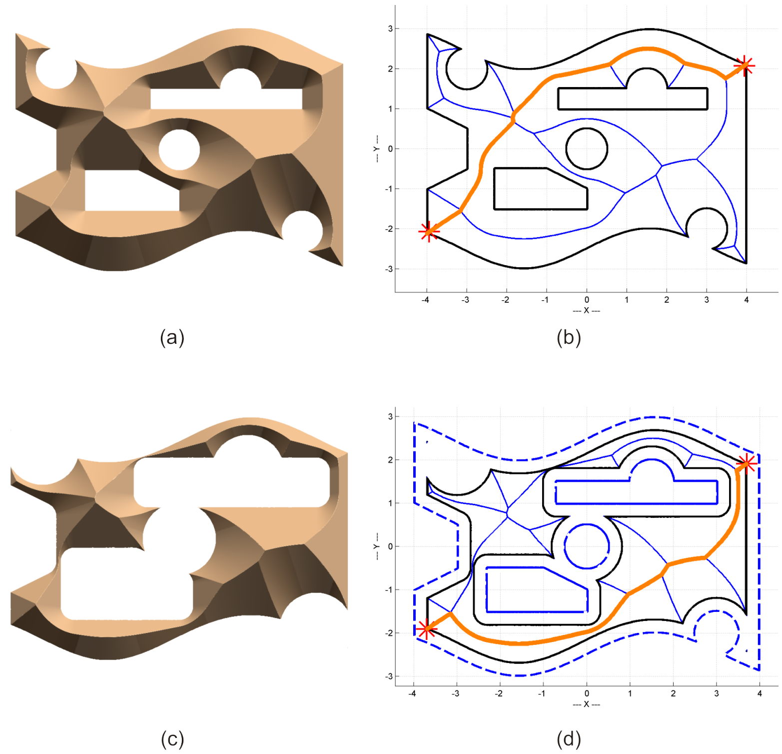


Figure 12: (a) an environment with curved boundaries and the corresponding trajectory of a point; (b) the problem of planning a path of a moving object is transformed into the path planning of a point by taking a different level set of the distance function corresponding to the original environment. Note the difference in the two planned paths.

---

A modified skeleton is shown in Figure 13 along with the computed path of a point within a domain with curved boundaries. The resulting roadmap computed based on the medial axis is shown in Figure 13(a). Furthermore, the R-function expression defining the crosshatched rectangular obstacle has been weighted by a factor $\lambda$ that corresponds to an angle of $85^o$ as discussed in section 2.4. This effectively brings the skeleton, and therefore the computed path, closer to that particular obstacle. Note that there are 5 obstacles in this example, some of which are merged together, and that only one of the 5 corresponding distance functions is weighted. Clearly, other distance functions could be weighted as well depending on the specific path planning task being investigated with important applications in assembly, robotic, or navigation applications (as shown schematically in Figure 1).

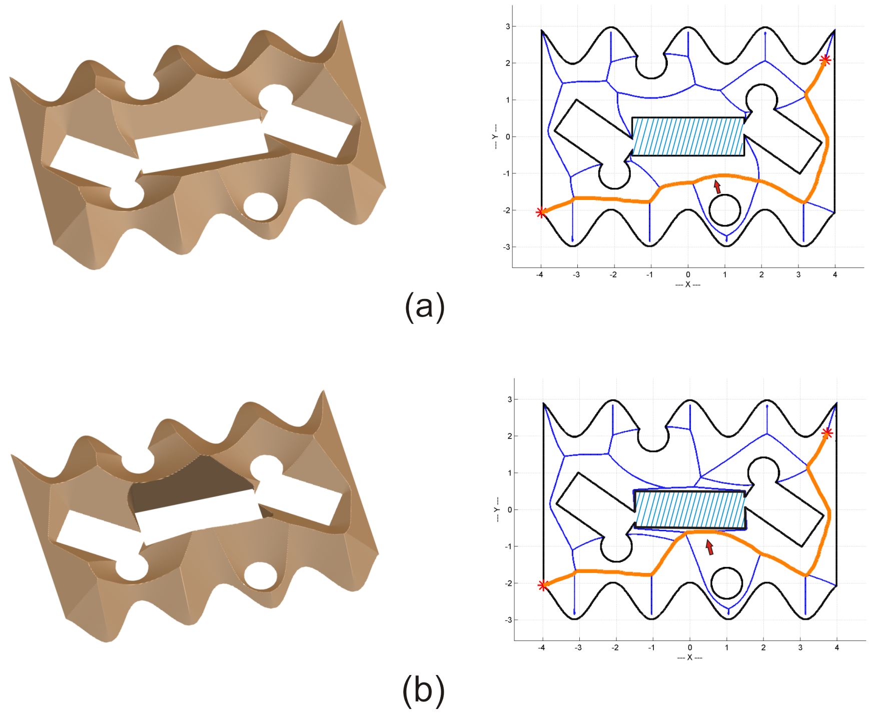


Figure 13: Shortest path of a point within a domain with curved boundaries and 5 obstacles that was computed based on: (a) medial axis; (b) a skeleton obtained by weighing the distance function defining the rectangular (crosshatched) obstacle.

---

Figure 14 shows a domain containing multiple obstacles with free-form planar geometry. The three different cases illustrated there correspond to three values of $\alpha$, namely $\alpha = 1$, $\alpha = 0.95$, and $\alpha = 0.8$, which result in the illustrated distance functions. For the case when $\alpha = 1$ illustrated Figure (a) the resulting skeleton is the medial axis, and the resulting shortest path is shown in orange (lighter color). By decreasing the value of $\alpha$ we increase the smoothness of the distance function, which in turn increases the 'thickness' of the medial zones, which is illustrated in Figures 14(b) and (c). Note that as $\alpha$ increases the length of the roadmap decreases, while its smoothness increases.

The R-function expression describing the domain of Figure 10(b) is illustrated in Figure 15(a) with a value of $\alpha = 0.90$, which results in the medial zone shown in Figure 15(b). Observe that this new distance function effectively results in a smoother shortest path (Figure 15(b)) that takes

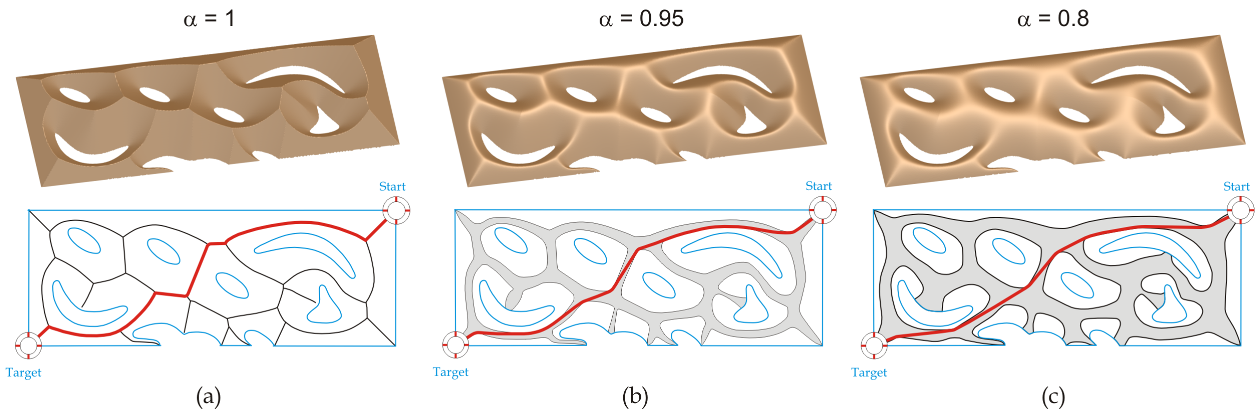


Figure 14: Shortest paths of a domain containing multiple obstacles with free-form planar geometry. The paths have been computed based on medial zones for three values of $\alpha$.

a different route than the shortest path shown in Figure 10(b) for the same domain.

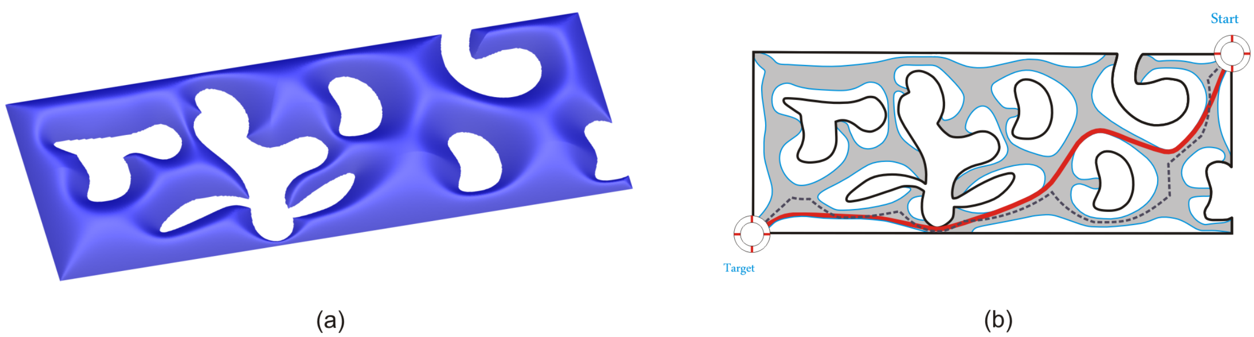


Figure 15: Medial zone and shortest path for the domain of Figure 10(b) for a value of $\alpha = 0.9$. The path computed in 10(b) is shown with dotted line for comparison purposes.

Finally, Figure 16 shows several 3D examples and the corresponding shortest paths. The 3D domains shown in Figures 16(a) and (c) contain one intermediate site that must be part of the path that is being planned. Note that a path along the medial axis that would satisfy these constraints does not exist. These examples contain 2 paths that do not satisfy the intermediate site constraint, which are planned inside 2 medial zones corresponding to values of $\alpha = 1$ and $\alpha = 0.9$ respectively. The same examples show a third path, which, in each case, is the shortest path connecting the beginning and end configurations, passing through the intermediate site, while remaining inside a medial zone corresponding to a value of $\alpha = 0.9$. To achieve this, the distance function corresponding to the proximate obstacle ($A$) was modified in both examples as described in section 2.4 and obstacle $A$ ‘attracts’ the resulting path. The example shown in Figures 16(d) illustrates another path planning scenario in an environment that contains many narrow passages and freeform obstacles.

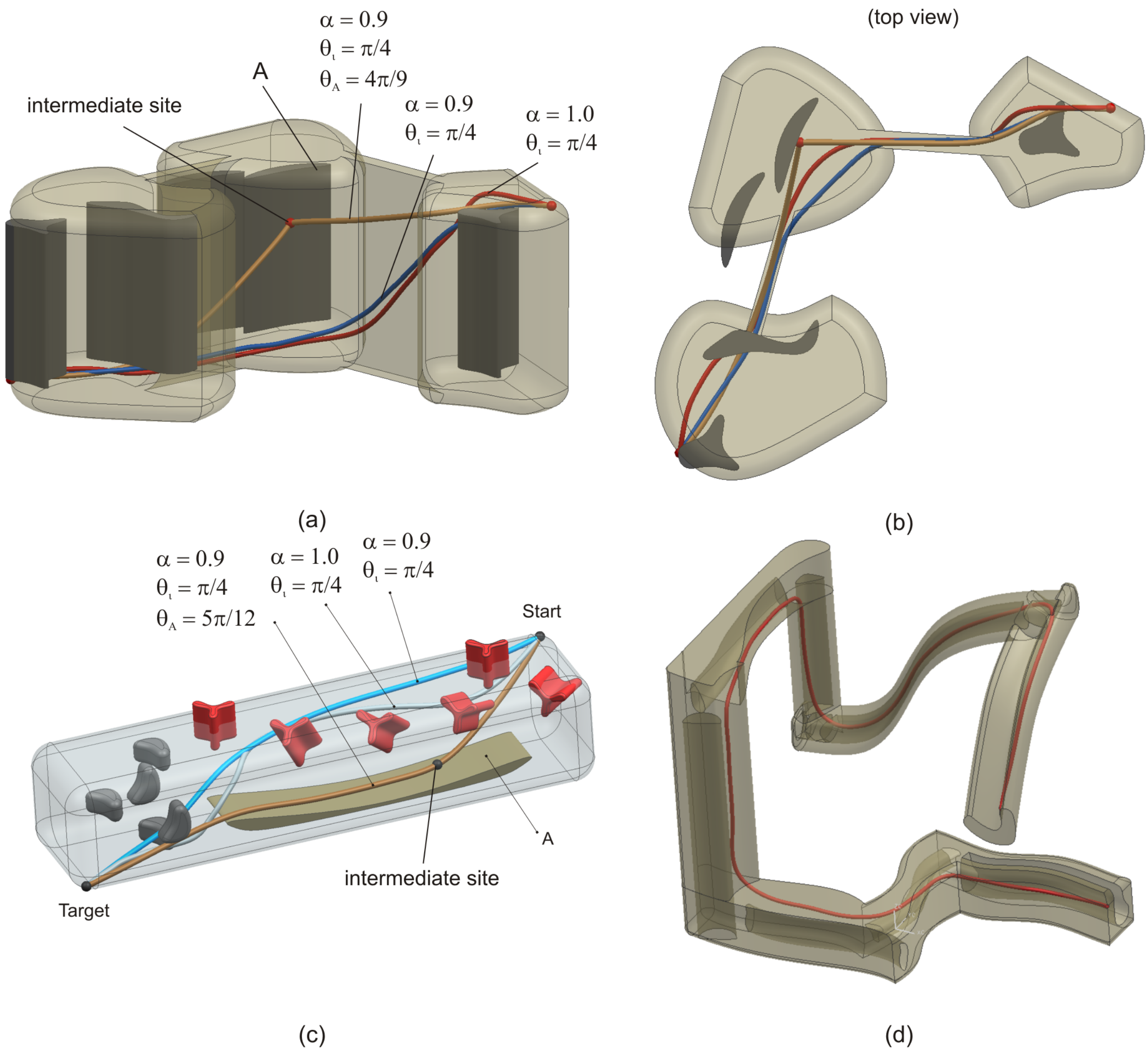


Figure 16: Collision-free paths for several 3D domains computed based on medial zones. The intermediate sites in (a) and (c) are shown as a points – the three paths correspond to values of $\alpha = 1$ and $\alpha = 0.9$; (b) shows the top view of the environment from (a); (d) illustrates an environment with narrow passages.

# 5 Conclusion

This paper formulates a family of geometric skeletons that contains the medial axis as a special case, and explores their use in robotic path planning applications in highly dynamic and topologically evolving environments. We propose a strategy for constructing these skeletons that relies on constructive representations of shapes with R-functions that operate on real-valued half-spaces as logic operations. The flexibility provided by the underlying Boolean nature of the proposed framework makes this approach well suited to problems involving substantial geometric and topologic changes of the environment. Importantly, our approach:

- is applicable for general closed, bounded, regular and semi-analytic domains;
- results in a family of skeletons that contains the medial axis as a special case. The computed skeleton can be modified so that the computed path is attracted or repulsed by prescribed obstacles within the domain;
- can compute the medial zones which, intuitively, represent thick skeletons of the domain. One important advantage of these zones in the context of motion planning applications is that they generate shorter and smoother paths than medial-axis path planning strategies for the *same* environment;
- can handle problems with evolving environments or in which the environment is not fully known a priori because obstacles can be introduced/removed at any time during the planning stage and within the same problem formulation;
- intrinsically supports local and parallel skeleton computations for domains with rigid or evolving boundaries due to the explicit mapping between the branches of the medial axis and the half-spaces bounding the environment;
- can handle trajectory planning of points as well as path planning for 2D/3D objects within the same formulation due to the close relationship between R-function expressions and level sets. The simplest approach to extend the trajectory planning to solid objects is to approximate the moving object by its smallest enclosing ball, and take different level sets of the distance function defining the environment. This effectively offsets the boundary of the environment

so that the outer boundaries of the environment shrink while, at the same time, the obstacles grow by the same radius of the disk enclosing the object.

- supports multiple representations of the input geometry;

We argue in this paper that the degrees of freedom that we used to control the subset of the domain in which we are looking for the shortest path afford significant flexibility in computing paths that satisfies specific geometric constraints that are common in geometric reasoning and motion planning applications. In addition, the attractive and repulsive factors $\lambda_i$ provide a simple and elegant mechanism to deform the skeleton so that the resulting path gets closer to or farther away from specific sites or obstacles in the environment.

We conjectured in section 2.4 that the family of skeletons defined by weights $\lambda$ is homeomorphic to the medial axis, which suggests that the skeletons in the family are homotopic to the domain. This property coupled with the fact that the family of skeletons is really defined by pseudo distance functions could prove to be crucial in numerous potential applications such as unmanned vehicles navigation systems, robotics, autonomous assembly systems, and shape representation and recognition.

## Acknowledgments

This work was supported in part by the National Science Foundation grants CMMI-0555937, CAREER award CMMI-0644769, CMMI-0856401, and CNS-0927105.

## References

[Amenta et al., 2001] Amenta, N., Choi, S., and Kolluri, R. K. (2001). The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19:127–153.

[Attali et al., 2008] Attali, D., Boissonnat, J.-D., and Edelsbrunner, H. (2008). Stability and computation of medial axes — a state-of-the-art report. In Möller, T., Hamann, B., and Russel, R., editors, *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Mathematics and Visualization. Springer Verlag.

[Blum, 1967] Blum, H. (1967). A transformation for extracting new descriptions of shape. *In Models for the Perception of Speech and Visual Form*, pages 362–380.

[Blum and Nagel, 1978] Blum, H. and Nagel, R. (1978). Shape description using weighted symmetric axis features. *Pattern Recognition*, 10.

[Bookstein, 1979] Bookstein, F. (1979). The line-skeleton. *Computer Graphics and Image Processing*, 11(2):123–137.

[Bovik, 2005] Bovik, A. C. (2005). *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Academic Press, Inc., Orlando, FL, USA.

[Brandt and Algazi, 1992] Brandt, J. W. and Algazi, V. R. (1992). Continuous skeleton computation by Voronoi diagram. *CVGIP: Image Underst.*, 55(3):329–338.

[Buchele and Crawford, 2003] Buchele, S. F. and Crawford, R. H. (2003). Three-dimensional halfspace constructive solid geometry tree construction from implicit boundary representations. In *Proceedings of the* $8^{th}$ *ACM Symposium on Solid Modeling and Applications*, pages 135–144. ACM Press.

[Canny, 1993] Canny, J. (1993). Computing roadmaps of general semi-algebraic sets. *The Computer Journal*, 36(5):504–514.

[Cao and Liu, 2008] Cao, L. and Liu, J. (2008). Computation of medial axis and offset curves of curved boundaries in planar domain. *Comput. Aided Des.*, 40(4):465–475.

[Chazal and Soufflet, 2004] Chazal, F. and Soufflet, R. (2004). Stability and finiteness properties of medial axis and skeleton. *Journal of Dynamical and Control Systems*, 10(2):149–170.

[Chen et al., 2007] Chen, J., Shapiro, V., Suresh, K., and Tsukanov, I. (2007). Shape optimization with topological changes and parametric control. *International Journal for Numerical Methods in Engineering*, 71:313–346.

[Choi et al., 1997] Choi, H. I., Choi, S. W., and Moon, H. P. (1997). Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 1(181):57–88.

[Choset, 2000] Choset, H. (2000). Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, 9:247–253.

[Culver et al., 2004] Culver, T., Keyser, J., and Manocha, D. (2004). Exact computation of the medial axis of a polyhedron. *Comput. Aided Geom. Des.*, 21(1):65–98.

[Damon, 2005] Damon, J. (2005). Determining the geometry of boundaries of objects from medial data. *Int. J. Comput. Vision*, 63:4564.

[Dasgupta et al., 2008] Dasgupta, S., Papadimitriou, C. H., and Vazirani, U. (2008). *Algorithms*. McGraw-Hill.

[de León S. and Sossa A., 1998] de León S., J. D. and Sossa A., J. (1998). Automatic path planning for a mobile robot among obstacles of arbitrary shape. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(3):467–472.

[Dey et al., 2003] Dey, T. K., Woo, H., and Zhao, W. (2003). Approximate medial axis for CAD models. In *SM '03: Proceedings of the eighth ACM Symposium on Solid Modeling and Applications*, pages 280–285, New York, NY, USA. ACM.

[Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271.

[Dobkin et al., 1988] Dobkin, D., Guibas, L., Hershberger, J., and Snoeyink, J. (1988). An efficient algorithm for finding the CSG representation of a simple polygon. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer Graphics and Interactive Techniques*, pages 31–40, New York, NY, USA. ACM.

[Du and Qin, 2004] Du, H. and Qin, H. (2004). Medial axis extraction and shape manipulation of solid objects using parabolic PDEs. In *SM '04: Proceedings of the ninth ACM Symposium on Solid Modeling and Applications*, pages 25–35, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Eftekharian and Ilieş, 2009] Eftekharian, A. and Ilieş, H. (2009). Distance functions and skeletal representations of rigid and non-rigid planar shapes. *Computer-Aided Design*, 41(12):865 – 876.

[Eftekharian and Ilieş, 2010] Eftekharian, A. and Ilieş, H. (2010). Medial zones: Formulation, properties and applications. Technical report, University of Connecticut.

[Foskey et al., 2001] Foskey, M., Garber, M., Lin, M. C., and Manocha, D. (2001). A Voronoi-based hybrid motion planner. In *In Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 55–60.

[Fraichard, 1993] Fraichard, T. (1993). Dynamic trajectory planning with dynamic constraints: A 'state-time space' approach. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems '93 IROS '93*, volume 2, pages 1393–1400.

[Garber and Lin, 2002] Garber, M. and Lin, M. C. (2002). Constraint-based motion planning for virtual prototyping. In *SMA '02: Proceedings of the seventh ACM symposium on Solid Modeling and Aapplications*, pages 257–264, New York, NY, USA. ACM.

[Geraerts and Overmars, 2007] Geraerts, R. and Overmars, M. H. (August 2007). Creating high-quality paths for motion planning. *The International Journal of Robotics Research*, 26, no. 8:845–863.

[Goh, 2008] Goh, W.-B. (2008). Strategies for shape matching using skeletons. *Comput. Vis. Image Underst*, 110:326–345.

[Gomes and Faugeras, 2000] Gomes, J. and Faugeras, O. (2000). Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation*, 11(2):209–223.

[Hoff III et al., 2000] Hoff III, K., Culver, T., Keyser, J., Lin, M., and Manocha, D. (2000). Interactive motion planning using hardware-accelerated computation of generalized Voronoi diagrams. In *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA'00*, volume 3.

[Ishikawa, 1991] Ishikawa, S. (1991). A method of indoor mobile robot navigation by using fuzzy control. In *Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems, IROS '91*, volume 2, pages 1013–1018.

[Jiang et al., 1999] Jiang, K., Seneviratne, L. D., and Earles, S. W. E. (1999). A shortest path based path planning algorithm for nonholonomic mobile robots. *J. Intell. Robotics Syst.*, 24(4):347–366.

[Kallman and Mataric, 2004] Kallman, M. and Mataric, M. (2004). Motion planning using dynamic roadmaps. *ICRA '04: 2004 IEEE International Conference on Robotics and Automation, 2004.*, 5:4399–4404.

[Kavraki et al., 1996] Kavraki, L. E., Vestka, P., Latombe, J.-C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12:566–580.

[Kimia and Tamrakar, 2002] Kimia, B. B. and Tamrakar, A. (2002). The role of propagation and medial geometry in human vision. In *BMCV '02: Proceedings of the Second International Workshop on Biologically Motivated Computer Vision*, pages 219–229, London, UK. Springer-Verlag.

[Kimmel et al., 1995] Kimmel, R., Shaked, D., Kiryati, N., and Bruckstein, A. (1995). Skeletonization via distance maps and level sets. *Computer Vision and Image Understanding*, 62(3):382–391.

[Latombe, 1991] Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA.

[LaValle, 2006] LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, Cambridge, U.K. Also available at http://planning.cs.uiuc.edu/.

[Lee and Choset, 2005] Lee, J. Y. and Choset, H. (May 2005). Sensor-based planning for a rod-shaped robot in three dimensions. *The International Journal of Robotics Research*, 24:343–383.

[Lieutier, 2003] Lieutier, A. (2003). Any open bounded subset of rn has the same homotopy type than its medial axis. In *SM '03: Proceedings of the eighth ACM Symposium on Solid Modeling and Applications*, pages 65–75, New York, NY, USA. ACM.

[Lingelbach, 2004] Lingelbach, F. (2004). Path planning using probabilistic cell decomposition. In *Proceedings IEEE International Conference on Robotics & Automation*.

[Liu et al., 1998] Liu, T.-L., Geiger, D., and Kohn, R. V. (1998). Representation and self-similarity of shapes. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 1129, Washington, DC, USA. IEEE Computer Society.

[Luo et al., 2007] Luo, Z., Tong, L., Wang, M. Y., and Wang, S. (2007). Shape and topology optimization of compliant mechanisms using a parameterization level set method. *J. Comput. Phys.*, 227:680–705.

[Masehian and Katebi, 2007] Masehian, E. and Katebi, Y. (winter 2007). Robot motion planning in dynamic environments with moving obstacles and target. *International Journal of Mechanical Systems Science and Engineering*.

[Neus M., 2005] Neus M., M. S. (2005). Motion planning using the modified visibility graph. In *IEEE IEEE International Conference on Systems man and Cybernetics*, volume 4, pages 651–655.

[Pizer et al., 2003] Pizer, S. M., Siddiqi, K., Székely, G., Damon, J. N., and Zucker, S. W. (2003). Multiscale medial loci and their properties. *Int. J. Comput. Vision*, 55(2-3):155–179.

[Quadros et al., 2004] Quadros, W., Shimada, K., and Owen, S. (2004). Skeleton-based computational method for the generation of a 3D finite element mesh sizing function. *Engineering with Computers*, 20(3):249–264.

[Ramamurthy and Farouki, 1999] Ramamurthy, R. and Farouki, R. T. (1999). Voronoi diagram and medial axis algorithm for planar domains with curved boundaries — II: detailed algorithm description. *J. Comput. Appl. Math.*, 102(2):253–277.

[Requicha, 1980] Requicha, A. (1980). Representations for rigid solids: Theory, methods and systems. *Computing Surveys*, 12(4):437–463.

[Rimon and Koditschek, 1992] Rimon, E. and Koditschek, D. (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518.

[Sacks et al., 1999] Sacks, E., Pisula, C., and Joskowicz, L. (1999). Visualizing 3d configuration spaces for mechanical design. *IEEE Computer Graphics and Applications*, 19(5):50–53.

[Sampl, 2000] Sampl, P. (2000). Semi-structured mesh generation based on medial axis. *IMR*, pages 21–32.

[Sampl, 2001] Sampl, P. (2001). Medial axis construction in three dimensions and its application to mesh generation. *Engineering with Computers*, 17:234–248.

[Sebastian et al., 2004] Sebastian, T., Klein, P., and Kimia, B. (2004). Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571.

[Shah, 2005] Shah, J. (2005). Gray skeletons and segmentation of shapes. *Computer Vision and Image Understanding*, 99:96–109.

[Shaham et al., 2004] Shaham, A., Shamir, A., and Cohen-Or, D. (2004). Medial axis based solid representation. In *9th ACM Symposium on Solid Modeling and Applications*.

[Shapiro, 1991] Shapiro, V. (1991). Theory of R-functions and applications: A primer. Technical Report TR91-1219, Computer Science Department, Cornell University, Ithaca, NY.

[Shapiro, 2001] Shapiro, V. (2001). A convex deficiency tree algorithm for curved polygons. *Int. J. Computational Geometry and Applications*, 11(2):215–238.

[Shapiro, 2007] Shapiro, V. (2007). Semi-analytic geometry with R-functions. *Acta Numerica*, 18:239–303.

[Shapiro and Vossler, 1991] Shapiro, V. and Vossler, D. L. (1991). Construction and optimization of CSG representations. *Computer-Aided Deign*, 23(1):4–20.

[Shapiro and Vossler, 1993] Shapiro, V. and Vossler, D. L. (1993). Separation for boundary to CSG conversion. *ACM Transactions on Graphics*, 12(1):3555.

[Sherbrooke et al., 1995] Sherbrooke, E., Patrikalakis, N., and Brisson, E. (1995). Computation of the medial axis transform of 3-D polyhedra. In *Proceedings of the third ACM symposium on Solid modeling and applications*, pages 187–200. ACM New York, NY, USA.

[Siddiqi et al., 2002] Siddiqi, K., Bouix, S., Tannenbaum, A., and Zucker, S. W. (2002). Hamilton-Jacobi skeletons. *Int. J. Comput. Vision*, 48(3):215–231.

[Siddiqi et al., 1999] Siddiqi, K., Shokoufandeh, A., Dickinson, S. J., and Zucker, S. W. (1999). Shock graphs and shape matching. *Int. J. Comput. Vision*, 35(1):13–32.

[Snavely et al., 2008] Snavely, N., Garg, R., Seitz, S., and Szeliski, R. (2008). Finding paths through the world's photos. *ACM Transactions on Graphics (TOG)*, 27(3):15.

[Suresh, 2003] Suresh, K. (2003). Automating the CAD/CAE dimensional reduction process. In *SM '03: Proceedings of the eighth ACM symposium on Solid Modeling and Applications*, pages 76–85, New York, NY, USA. ACM.

[Takaoka, 2005] Takaoka, T. (2005). An $O(n^3 \log\log n / \log n)$ time algorithm for the all-pairs shortest path problem. *Inf. Process. Lett.*, 96(5):155–161.

[Tor and Middleditch, 1984] Tor, S. B. and Middleditch, A. E. (1984). Convex decomposition of simple polygons. *ACM Trans. Graph.*, 3(4):244–265.

[van den Berg and Overmars, 2005] van den Berg, J. and Overmars, M. (2005). Roadmap-based motion planning in dynamic environments. *IEEE Transactions on Robotics*, 21(5):885–897.

[van Eede et al., 2006] van Eede, M., Macrini, D., Telea, A., Sminchisescu, C., and Dickinson, S. (2006). Canonical skeletons for shape matching. In *Proc. 18th International Conference on Pattern Recognition ICPR 2006*, volume 2, pages 64–69.

[Wilmarth et al., 1999] Wilmarth, S. A., Amato, N. M., and Stiller, P. F. (1999). Motion planning for a rigid body using random networks on the medial axis of the free space. In *SCG '99: Proceedings of the Fifteenth Annual Symposium on Computational Geometry*, pages 173–180, New York, NY, USA. ACM.

[Xidias et al., 2007] Xidias, E. K., Azariadis, P. N., and Aspragathos, N. A. (2007). Two-dimensional motion-planning for non-holonomic robots using bump-surface concept. *Computing*, 79:109–118.