Computer-Aided Design 44 (2012) 413–423

Contents lists available at SciVerse ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Medial zones: Formulation and applications

Ata A. Eftekharian, Horea T. Ilieş *

Computational Design Laboratory, Department of Mechanical Engineering, University of Connecticut, Storrs, CT 06269-3139, United States

ARTICLE INFO

Article history:
Received 4 March 2011
Accepted 10 December 2011

Keywords:
Medial zones
Medial axis
Skeletons
Distance functions
R-functions

ABSTRACT

The popularity of medial axis in shape modeling and analysis comes from several of its well known fundamental properties. For example, medial axis captures the connectivity of the domain, has a lower dimension than the space itself, and is closely related to the distance function constructed over the same domain.

We propose the new concept of a *medial zone* of an n-dimensional semi-analytic domain Ω that subsumes the medial axis $\mathcal{MA}(\Omega)$ of the same domain as a special case, and can be thought of as a 'thick' skeleton having the same dimension as that of Ω. We show that by transforming the exact, non-differentiable, distance function of domain Ω into approximate but differentiable distance functions, and by controlling the geodesic distance to the crests of the approximate distance functions of domain Ω, one obtains families of medial zones of Ω that are homeomorphic to the domain and are supersets of $\mathcal{MA}(\Omega)$. We present a set of natural properties for the medial zones $\mathcal{MZ}(\Omega)$ of Ω and discuss practical approaches to compute *both* medial axes and medial zones for 3-dimensional semi-analytic sets with rigid or evolving boundaries. Due to the fact that the medial zones fuse some of the critical geometric and topological properties of both the domain itself and of its medial axis, re-formulating problems in terms of medial zones affords the 'best of both worlds' in applications such as geometric reasoning, robotic and autonomous navigation, and design automation.

1. Motivation

The *medial axis* introduced by Blum [1] for biological shape measurement and description has become an important tool in computational geometry and geometric modeling due to its compact representation of the essential topologic and geometric properties of a shape. Conceptually, the medial axis of a domain Ω can be described through the analogy with a fire simultaneously starting from all points of the boundary of Ω and moving with constant velocity in all directions. Then, $\mathcal{MA}(\Omega)$ is the locus of points where the moving front meets itself. For a planar figure, the medial axis is a collection of curves in the plane. The concept can be extended to k-dimensional geometric shapes in $\mathbb{R}^k$, case in which the $\mathcal{MA}$ 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 as illustrated in Fig. 1(a). More formally, if Ω is a closed, bounded, regular and semi-analytic subset of $\mathbb{R}^k$, then the medial axis $\mathcal{MA}(\Omega)$ is the set of points that have at least two closest points in $\partial\Omega$—the boundary of Ω. Thus, $\mathcal{MA}$ is the set of all centers of the closed maximal balls, and is a continuous version of the usual Voronoi diagram [2].

One can construct examples in which Ω is bounded and its boundary is C^∞-smooth yet the $\mathcal{MA}$ has infinitely many branches [3]. However, the finiteness of the medial axis has been proven in [4] for semi-analytic bounded open sets $\Omega \in \mathbb{R}^k$. In the same paper [4], it is proven that the medial axis of semi-analytic sets admits stratification, i.e., it can be decomposed into finite number of strata, each a connected i-manifold with boundary for $i < k$. This is particularly important since a wide variety of shapes encountered in science and engineering, including all solid models, can be described as semi-analytic sets [5]. Another important result described in [6] states that a shape and its medial axis are connected the same way regardless of the dimension of the space (i.e., they are homotopy equivalent), which is exploited in many applications, including those that focus on shape similarity.

Since its formulation, the medial axis has been used as an alternative solid modeling representation [7], as well as in many other applications such as shape matching and reconstruction [8–13], dimensional reduction in boundary value problems [14], representing and classifying 2D shapes [15–17], human vision [18], pattern analysis and shape recognition [19,20], and mesh generation [21–23]. Despite the fact that the mathematical properties of the

* Corresponding author. Tel.: +1 860 486 8813; fax: +1 860 486 5088.
E-mail addresses: eftekharian@engr.uconn.edu (A.A. Eftekharian), ilies@engr.uconn.edu (H.T. Ilieş).

0010-4485/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2011.12.003

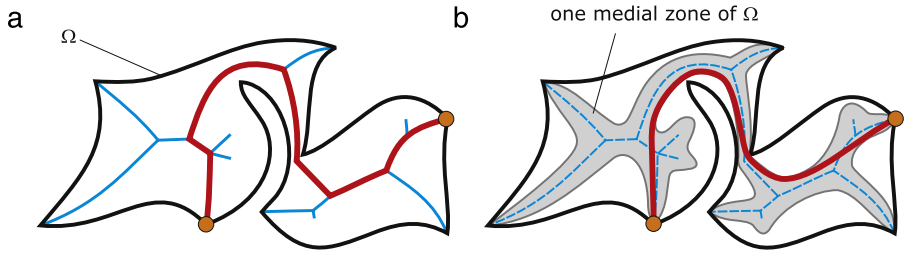

Fig. 1. (a) A planar domain Ω, its medial axis $\mathcal{MA}$ and the shortest path planned along $\mathcal{MA}$ between two configurations; (b) the same planar domain with a medial zone (shown as the shaded subset $\mathcal{MZ}$), and the corresponding shortest path that remains a subset of the medial zone $\mathcal{MZ}$ for the same two end configurations. The path in figure (b) is 13.34% shorter and has visibly more uniform curvature than the path shown in (a).

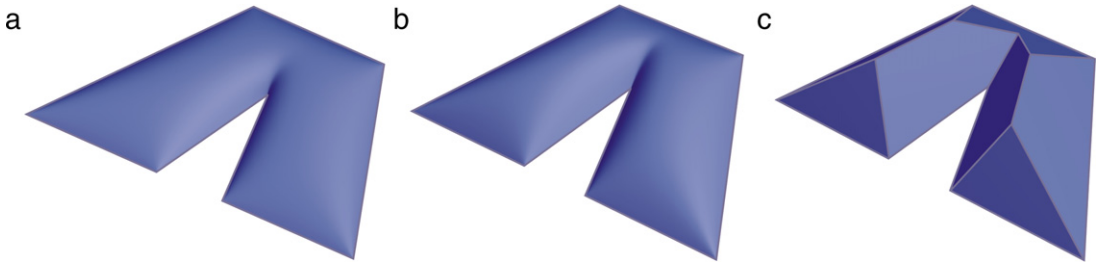

Fig. 2. Three implicit representations of a polygonal domain that correspond to (a) $\alpha = 0$, (b) $\alpha = 0.5$, and (c) $\alpha = 1$.

medial axis are fairly well understood, computing medial axis continues to remain a challenge, particularly for 3-dimensional non-polyhedral shapes.

One important application of medial axes is in the field of robot path planning for navigating along narrow passages where other practical algorithms including probabilistic roadmap (PRM) planners do not fare well. Since the medial axis contains, by definition, points that are equally spaced to the boundary of the domain, planning a path along the medial axis becomes a natural choice for domains with narrow passages. On the other hand, medial-axis based planners tend to produce non-optimal paths in the regions that are away from the narrow passages, because the medial axis continues to remain in the 'middle' of the domain as shown in Fig. 1(a).

We propose the new concept of a *medial zone* of an n-dimensional semi-analytic domain Ω that subsumes the medial axis $\mathcal{MA}(\Omega)$ of the same domain as a special case, and can be thought of as a 'thick' skeleton having the same dimension as that of Ω. We show that by transforming the exact, non-differentiable, distance function of domain Ω into approximate but differentiable distance functions, and by controlling the geodesic distance to the crests of the approximate distance function of domain Ω, one obtains families of medial zones of Ω that are homeomorphic to the domain and are supersets of $\mathcal{MA}(\Omega)$. We present a set of natural properties for the medial zones $\mathcal{MZ}(\Omega)$ of Ω and discuss practical approaches to compute *both* medial axes and medial zones for 3-dimensional semi-analytic sets with rigid or evolving boundaries.

The formulation of medial zones builds on our earlier work on computing medial axes of planar non-rigid domains detailed in [24]. In practical terms, we demonstrate that the use of medial zones in path planning applications retains the properties of medial axes in the neighborhood of narrow passages, while resulting in globally shorter paths as illustrated in Fig. 1(b). Furthermore, we also point out that medial zones can provide a powerful new paradigm to automate the shape synthesis of mechanical artifacts.

2. Formulation

We construct distance functions to individual halfspaces bounding a semi-analytic domain Ω and combine them with R-functions into one single distance function over Ω. The R-functions provide the flexibility of constructing exact distance functions over Ω whose ridges and ravines correspond to classes of skeletons of Ω. For example, by projecting the ridges of the exact distance function one obtains the medial axis $\mathcal{MA}(\Omega)$ [24]. Furthermore, the R-functions can be used to construct approximate distance functions whose crests define the medial zones of Ω as we discuss in Section 2.3.

2.1. R-functions as logic operators on real-valued halfspaces

For any closed subset Ω of $\mathbb{R}^k$, one can construct a C^∞ function that vanishes on the boundary $\partial\Omega$ of Ω. 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 Ω of $\mathbb{R}^k$ by subdividing the boundary of Ω in primitive halfspaces f_i, followed by a combination of f_i into a single predicate using the standard Boolean logic operators AND, OR or NOT [5].

R-functions have been invented in the 60's by Rvachev [25,5], who called these functions "logically charged functions". They provide the means to construct a C^n function over a domain defined by primitive halfspaces. 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 Ω. 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 sufficiently complete systems of R-functions. One such system is known as the principal system of R-functions

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

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 α, we obtain different systems of R-functions. In particular, by setting $\alpha = 1$ in Eq. (1) we obtain the $R_1(\Delta)$ system of R-functions, while a value $\alpha = 0$ in Eq. (1) would result in the $R_0(\Delta)$ system (see Fig. 2).

For the $R_1(\Delta)$ system or R-functions, the expression under the square root in Eq. (1) becomes zero at points where the resulting function is non-differentiable. In other words, since the medial axis of a semi-analytic domain corresponds to the points where the distance function is non-differentiable, we look for points of the R-function where the expression under the square root vanishes [24].

Fig. 2 shows the difference between implicit functions constructed over a polygonal domain with the $R_\alpha(\Delta)$ system for three values of α. 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 Fig. 2(c) is made of piecewise planar patches that intersect along 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. Constructing Boolean expressions

Conceptually, the problem of constructing a Boolean expression for a domain bounded by halfspaces 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 [26,27]. 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 halfspaces need to be introduced—the so-called separating halfspaces. Determining a sufficient set of separating halfspaces 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 [28], solids bounded by linear or quadric surfaces [29], and more general 3D solids [30,31].

Note that the problem of B-rep to CSG conversion is well defined, but a CSG expression that uses only the halfspaces bounding the domain may not exist. This necessary and sufficient condition for the existence of a CSG expression is formally described by the 'Describability Theorem' in [30]. However, if a *canonical* CSG expression exists for a given set of halfspaces bounding a specific domain, then this canonical CSG expression is unique. Otherwise, additional halfspaces must be introduced into the CSG expression.

2.2.1. Two dimensional semi-analytic domains

The Boolean set representation of a polygonal domain can be computed based on the Convex Deficiency Tree [27], which treats each polygon as its convex hull *minus* a finite number of concavities. Note that the polygonal domain is a closed set, so the subtraction of concavities must necessarily be regularized (i.e., one must use regularized Boolean operations). Importantly, this construction algorithm for simple polygons can be extended to some other point sets, such as curved polygons [28], 3D polyhedra, and more general 3D solids. Finally, we perform a syntactic substitution to replace the union and intersection with the R-disjunction and R-conjunction given in Eq. (1), which results in an implicit function whose zero level set is the original domain.

By following this procedure, we obtain an implicit function that is the exact distance function for any convex polygon, and an approximate distance function for a concave planar domain. These approximate distance functions can be used to extract the straight skeleton for polygonal domains or the C-skeletons as we detail in [24]. We note that such a distance function is obtained from the principal system of R-functions given in Eq. (1) by setting the value of $\alpha = 1$, i.e., the $R_1(\Delta)$ system.

These approximate distance functions can be converted into an exact distance function by introducing additional halfspaces at the concave vertices of the domain [24]. Specifically, each concave vertex requires a conical halfspace with a half angle of $\pi/4$ and two separating/trimming halfspaces that are normal to the boundary curves that are incident at the concave vertex as illustrated in Fig. 3.

2.2.2. Three dimensional polyhedral domains

The same approach to constructing exact distance functions over planar domains can be extended to three-dimensional domains. While the extension to 3D piecewise-linear domains is intuitively explained in this section, constructing distance functions for more general 3D domains[1] requires trimming halfspaces for which an analytic solution may not exist. For planar shapes we added trimmed distance functions to all concave vertices. For a 3D polyhedral domain, we need to add not only the trimmed distance functions to all concave vertices as we did for planar domains, but also trimmed distance functions to all concave edges.

For practical reasons, since our domains can evolve and therefore the convexity of a given vertex or edge can change during the evolution, we add trimmed distance functions to *all* vertices and edges of the domain. We observe that only those functions that are added to the concave vertices and edges will affect the distance function, but this intentional redundancy eliminates the need to keep track of changes in convexity.

The unit normal of a plane given by the standard implicit equation $ax + by + cz + d = 0$ is

$$\mathbf{n} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$$

where $\mathbf{v} = [a, b, c]^t$ is a vector normal to the plane. Therefore, the shortest signed distance between a point $\mathbf{P} = [x, y, z]^t$ and the plane is given by

$$F(x, y, z) = \mathbf{n} \cdot (\mathbf{P} - \mathbf{P}_0)$$

where $\mathbf{P}_0 = [x_0, y_0, z_0]^t$ is any point of the plane, which can be rewritten as

$$F(x, y, z) = \frac{ax + by + cz + d}{\sqrt{a^2 + b^2 + c^2}}. \tag{2}$$

Similarly, the shortest distance between a point $\mathbf{P} = [x, y, z]^t$ and a 3D line L passing through a point $\mathbf{P}_0 = [x_0, y_0, z_0]^t$ having a direction given by the unit vector $\mathbf{v} = [v_x, v_y, v_x]^t$, i.e., $\mathbf{L}(t) = \mathbf{P}_0 + \lambda\mathbf{v}$, where $\lambda \in \mathbb{R}$, can be written as

$$E(x, y, z) = [(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - (v_x(x - x_0) + v_y(y - y_0) + v_z(z - z_0))^2]^{\frac{1}{2}}. \tag{3}$$

Furthermore,

$$V(x, y, z) = [(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2]^{\frac{1}{2}} \tag{4}$$

is the shortest distance between point (x, y, z) and a point (x_0, y_0, z_0).

Note that each face F_i induces a signed distance field f_i, where the positive distance corresponds to the interior of the domain. Similarly, each edge E_j and vertex V_k induce distance fields e_j and v_k respectively.

Consider the solid shown in Fig. 4, which has 11 faces, 24 edges and 16 vertices that generate an equal number of halfspaces

[1] Note that a 3D shape will generate a signed distance function in 4D.

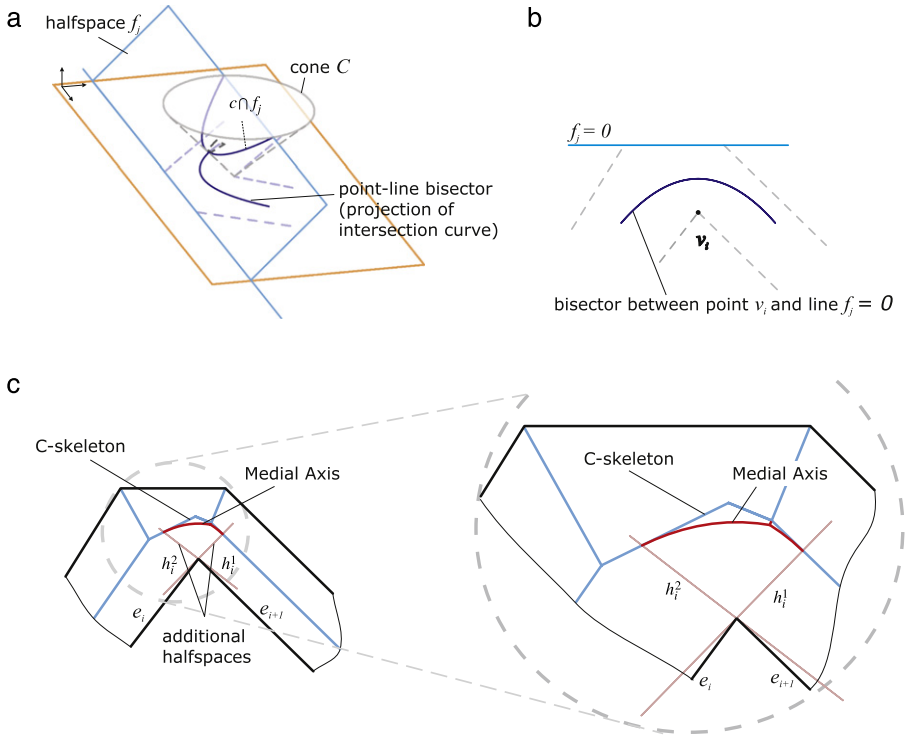

Fig. 3. Additional halfspaces are required to construct exact distance functions for concave polygons: (a) the conic section obtained by intersecting the halfspace f_j with the conical halfspace added at the convex vertex v_i; (b) the bisector between a point and a line is obtained by projecting this conic section onto the plane of the domain; (c) trimming the bisector by introducing additional halfspaces h_i^1 and h_i^2 normal to the two incident edges at v_i.

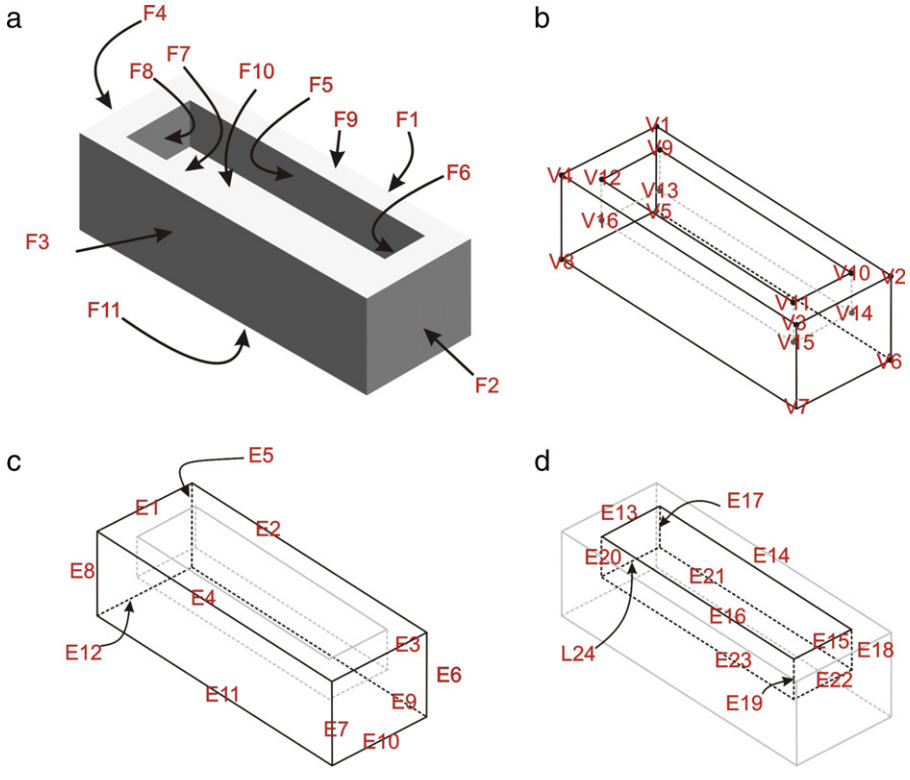

Fig. 4. A solid object with 11 faces, 24 edges and 16 vertices.

according to Eqs. (2)–(4). We can construct the approximate distance function (whose zero level set is the solid itself) by first constructing the equations for the parallelepiped and the halfspace defined by the boundary of the pocket followed by an intersection of the two according to:

$$\Omega_{\text{approx}} = (f_1 \cap f_2 \cap f_3 \cap f_4 \cap f_9 \cap f_{11}) \cap (f_5 \cup f_6 \cup f_7 \cup f_8 \cup f_{10}) \quad (5)$$

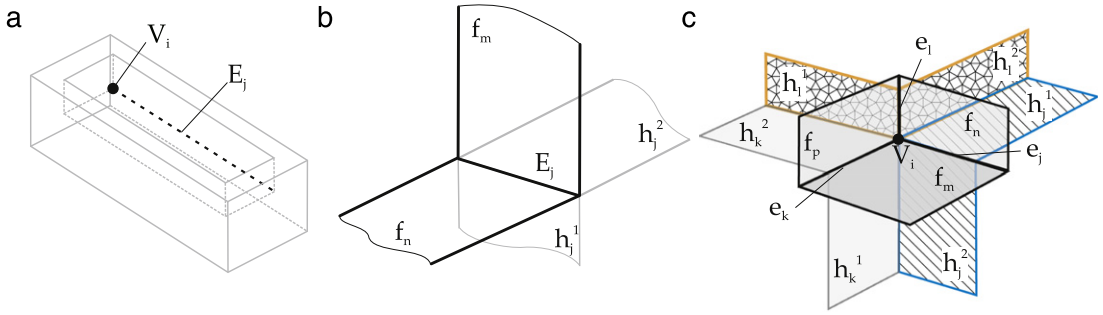

Fig. 5. The distance functions to each edge E_j of the pocket is trimmed by two (linear) halfspaces h_j^1 and h_j^2; the distance function to each vertex V_i of the pocket is trimmed by a number of pairs of (planar) halfspaces that equals the number of incident edges at that vertex.

where the sign of each signed distance field is adjusted so that its positive side is towards the interior of the solid.

By adding trimmed halfspaces corresponding to *each* concave edge and vertex of the pocket (i.e., edges E_{17}–E_{24} and vertices V_9–V_{16} to the function given in Eq. (5), we obtain the *exact* distance field for this polyhedral solid. Note that the zero level set of this field would correspond to the solid shown in Fig. 4.

Trimming halfspaces for 3D polyhedra

Constructing the trimming halfspaces is illustrated in Fig. 5. In [24] we have shown that, in the planar case, each concave vertex required two halfspaces that were normal to the two incident edges. We have seen that if one adds trimming halfspaces to all vertices of a planar figure, only those that belong to a concave vertex would contribute to the medial axis. This clearly increases the complexity of the distance function expression, but this designed redundancy eliminates the need of keeping track of changes in vertex convexity.

A vertex of a manifold 3D solid is obtained as the intersection of three or more incident edges, and each edge is shared by exactly two faces of the solid. Thus, we use two (linear) halfspaces h_j^1 and h_j^2 to trim the halfspace e_j corresponding to each concave (linear) edge E_j of the pocket as illustrated in Fig. 5(b):

$$\widetilde{e}_j = e_j \cap (h_j^1 \cap h_j^2) \tag{6}$$

where $\widetilde{e}$ denotes the trimmed halfspace e.

Furthermore, the distance function to each vertex V_i of the pocket will be trimmed by a number of pairs of halfspaces that is equal to the number of incident edges at that vertex. Since all vertices of the pocket have 3 incident edges, we need to add 3 pairs of trimming halfspaces to each vertex. For example, if the edges that are incident at vertex V_i are E_j, E_k and E_l as shown in Fig. 5(c), then the trimmed halfspace added at vertex V_i is given by:

$$\widetilde{v}_i = v_i \cap [(h_j^1 \cap h_j^2) \cap (h_k^1 \cap h_k^2) \cap (h_l^1 \cap h_l^2)]. \tag{7}$$

Thus, the equation describing the distance function for the pocket is

$$\Omega_{\text{pocket}} = (f_5 \cup f_6 \cup f_7 \cup f_8 \cup f_{10}) \cup \widetilde{e} \cup \widetilde{v} \tag{8}$$

where $\widetilde{e}$ and $\widetilde{v}$ are the trimmed halfspaces corresponding to the edges and vertices of the pocket:

$$\begin{aligned} \widetilde{v} &= \bigcup_{i\in[9,16]} \widetilde{v}_i \\ \widetilde{e} &= \bigcup_{j\in[17,22]} \widetilde{e}_j. \end{aligned} \tag{9}$$

By subtracting Ω_{pocket} from $\Omega_{\text{parallelepiped}}$ from Eqs. (5) and (8), we obtain the exact distance function for the polyhedral solid object of Fig. 4:

$$\Omega_{\text{exact}} = \Omega_{\text{parallelepiped}} \cap \Omega_{\text{pocket}}. \tag{10}$$

2.2.3. General three dimensional semi-analytic domains

So far, we have used signed distance fields together with trimming halfspaces for the edges and vertices of the domain to obtain the exact distance field over 3D polyhedral domains. One of the advantages was that other types of skeletons, such as the straight skeleton and the C-skeleton [24] could be easily computed from the same formulation. On the other hand, the 'straight skeleton' is only defined for sets bounded by piecewise linear boundaries. Furthermore, implicitizing more general surfaces that bound non-polyhedral domains is almost always very demanding from a computational point of view even when it can be carried out, and analytic solutions for the trimming halfspaces may not exist. This is why we adopt a different procedure for constructing the distance function over the domain for general three dimensional semi-analytic sets.

The problem of constructing distance functions to known arbitrary curves and surfaces has been raised in a number of applications such as geometric modeling, and computer vision. For example, squared distance functions to curves and surfaces are investigated in [32], where second order approximations for the distance functions to curves and surfaces are being constructed. At the same time, this problem can be reformulated as a numerical search for the foot point of a curve or surface, but the search can be slow, does not always converge and may not be robust [33]. Given these difficulties, an alternative strategy can be employed: rather than searching for the point(s) on a boundary surface or edge that is(are) closest to a given point of the space, we can

(1) enumerate (a dense set of) points $\mathbf{P}_c$ on the boundary surface/curve;
(2) construct the vector/plane normal to the surface/curve at a given point $\mathbf{P}_c$ on the surface/curve;
(3) enumerate (a dense set of) points $\mathbf{P}'_c$ along the normal vector $\mathbf{n_P}$ or in the normal plane;
(4) set the *known* distance from $\mathbf{P}'_c$ to $\mathbf{P}_c$ as the value of the distance function that we are looking for, that is

$$d(\mathbf{P}'_c, \mathbf{P}_c).$$

Observe that this procedure: (1) is general, in the sense that it remains valid for any semi-analytic halfspace; (2) it *enumerates* values of the distance function *without requiring a search*; (3) it avoids the problems with the foot-point computations mentioned above; and (4) it is conceptually equivalent to a line that sweeps a surface as point $\mathbf{P}_c$ travels along the surface/curve in the directions suggested by the parametrization (i.e. u & v for surfaces and u for curves). The proposed construction should be followed by a postprocessing step that would eliminate the self-intersections of the sweeping line, for example through a visibility test. The resulting point cloud can be interpolated to obtain a continuous hyper-surface approximating the boundary of the distance function.

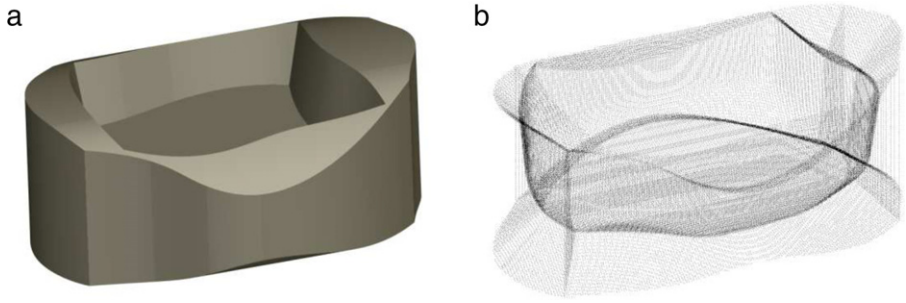

Fig. 6. An arbitrary domain (a) and points on its medial axis (b); the solid shown in (a) is also the zero level set of the resulting squared distance function.

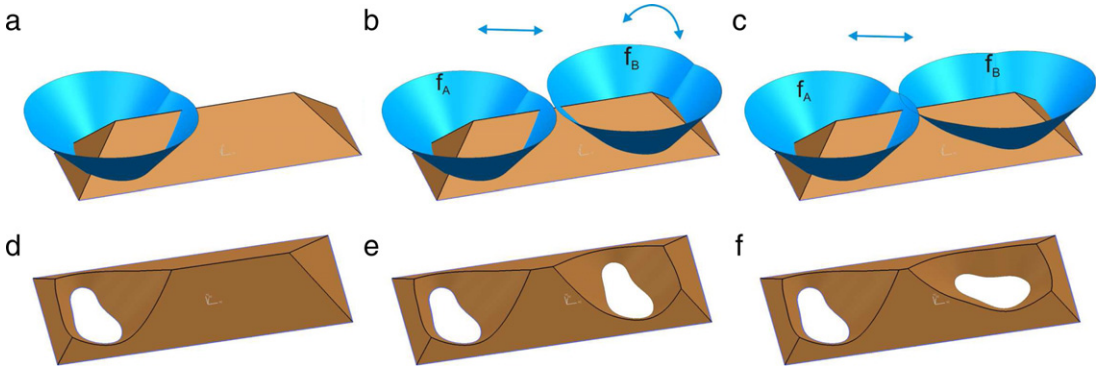

Fig. 7. Holes/voids can be added to (or removed from) the environment. Distance functions f_A and f_B to individual holes are illustrated in (a–c). The resulting distance functions of the domain are shown in (d–f).

Alternatively, the minimum distance from a point to a face, edge or vertex of a solid can be found through the API of most geometric kernels. By doing so, or by using other available algorithms, such as those outlined above or the newer breed of GPU accelerated algorithms, one can compute the *unsigned* distance field for every closed[2] face and edge bounding the domain. This is followed by a combination of all these (positive) distance functions with the *R*-conjunction that corresponds to the intersection operator. The fact that we employ unsigned distance functions eliminates the additional trimming operation described above.

One important consequence of using unsigned distance fields is that the B-rep to CSG conversion becomes independent of the order in which we combine the geometric entities bounding the 3D shape, because the Boolean expression will contain one single commutative operator, i.e., *R*-conjunction or intersection. The disadvantage is that the sign of the resulting function can no longer be used as a Point Membership Classification (PMC) test for the solid because all points will have a positive distance value. Hence, computing medial sets of the solid from this resulting distance function would need to employ a PMC test against the original representation of the solid.

Fig. 6 shows an arbitrary domain bounded by NURBS surfaces, which can be thought of as a deformed version of the polyhedron from Fig. 4. As mentioned above, we can construct the distance function for this domain from the individual unsigned distance functions to each closed face and edge, as illustrated in Fig. 6(b).

2.2.4. Adding/removing holes in the domain

One of the main advantages of our approach to compute skeletons based on *R*-functions is that holes/voids can be easily added to Ω by simply adding Jordan curves in 2D or Jordan surfaces[3] in 3D corresponding to the boundary of each hole/void. The *R*-function expression corresponding to the hole is subtracted from the *R*-function defining the outer boundary of Ω according to the *R*-subtraction[4]:

$$R_\alpha(\Delta):\ \frac{1}{1+\alpha}\left(f_1 - f_2 - \sqrt{f_1^2 + f_2^2 + 2f_1f_2}\right) \tag{11}$$

where f_1 is the function describing the outer domain and f_2 represents the hole/void. Fig. 7 shows how holes/voids can be added to the main space. Fig. 7(a–c) illustrate this process in which two holes are sequentially added to the environment, then one of them is rotated. The corresponding distance functions and the associated topological changes are illustrated in Fig. 7(d–f).

The Boolean formulation discussed above also ensures that holes can be easily relocated and re-sized to perform, for example, parametric or topology optimization [35,36]. These modifications exploit the connection between the resulting implicit functions and level sets so that the boundary of the domain defining the implicit function is the zero level set of the function itself (see also [24]). Furthermore, moving boundaries of Ω can be handled as easily. It is important to note that each halfspace affects the medial axis only locally, i.e., a subset of all branches of the medial axis, and this correspondence is explicit in our formulation (see also [24]). This allows us, in principle, to localize the computations as long as the Boolean expression defining the domain does not change.

2.3. Medial zones

We have been using so far the principal system of *R*-functions from Eq. (1) with a value of $\alpha = 1$. This in turn generates

[2] A closed face (or edge) contains its boundaries.

[3] These are the surfaces satisfying the Jordan–Brouwer separation theorem—see for example [34].

[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.

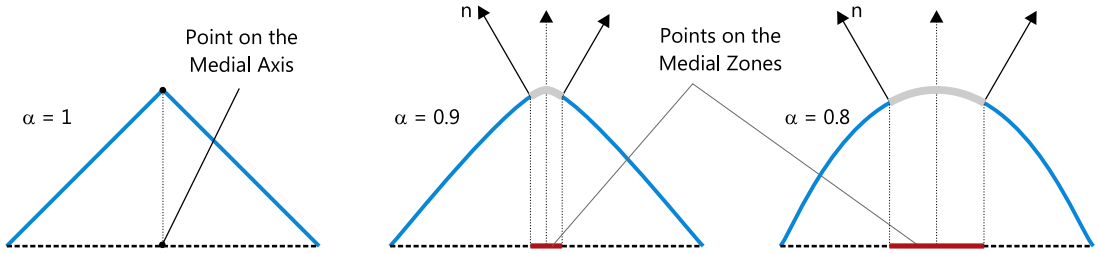

Fig. 8. A 'slice' of a distance function with a plane perpendicular to the medial axis. By adjusting the values of α and of the geodesic 'proximity' metric one can control the 'thickness' of the medial zone.

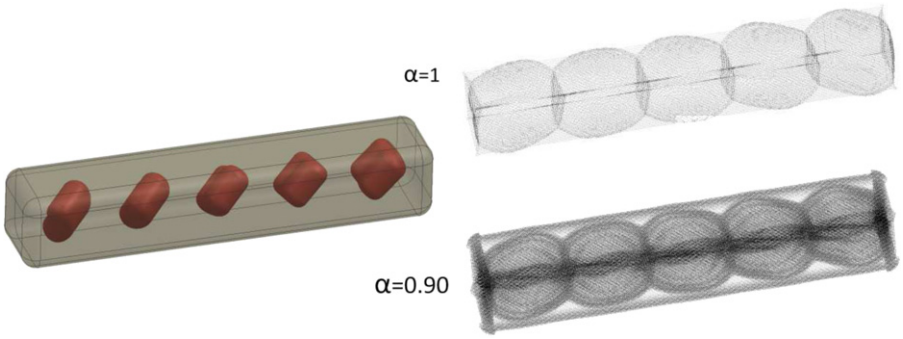

Fig. 9. An arbitrary solid with five internal voids, its medial skeleton $\alpha = 1$ and the medial zone for $\alpha = 0.9$.

piecewise smooth distance functions and the skeletons are obtained as the non-differentiable points of the distance functions as discussed above. However, one important feature of all systems of R-functions is that they can generate continuous and smooth functions over the domain that are differentiable almost everywhere (see [30,5] for a detailed discussion). By simply choosing a value of $0 \leq \alpha < 1$ for the R_α system, one eliminates the singular points of the exact distance function that correspond to interior points of the domain. The differential properties of the resulting function, which is no longer an exact distance function, play a key role in developing meshfree solution procedures for boundary value problems [5].

Fig. 8 shows a cross-section of the distance function of a planar domain with a plane perpendicular to the medial axis, and the effect of decreasing the value of α on the shape of the resulting function. Note that as α decreases, the maximum curvature of the resulting function decreases. Furthermore, by controlling the geodesic distance to the crests of the approximate distance function of the domain, one obtains families of medial zones for the same domain Ω.

It is apparent that there are multiple ways to construct medial zones. We postulate that a medial zone of a closed, bounded and semi-analytic domain Ω must satisfy the following requirements:

I: *Relationship to the domain*: $\mathcal{MZ}(\Omega) \subseteq \Omega$;
II: *Relationship to the medial axis of the domain*: $\mathcal{MZ}(\Omega) \supseteq \mathcal{MA}(\Omega)$;
III: *Dimension*: the medial zone $\mathcal{MZ}(\Omega)$ of Ω should be dimensionally homogeneous and have the same dimension as Ω itself, except when $\mathcal{MZ}(\Omega) = \mathcal{MA}(\Omega)$;
IV: *Homeomorphism*: Ω and its medial zones $\mathcal{MZ}(\Omega)$ should be homeomorphic, except when $\mathcal{MZ}(\Omega) = \mathcal{MA}(\Omega)$.

Note that α is a 'global' parameter, while the geodesic 'proximity' metric is, in principle, a local parameter defined at a point. One practical and efficient approach to control this geodesic distance is to use the smooth Laplace operator Δ, or a suitable discrete version of it [37,38] depending on the geometric representation of the halfspaces bounding the domain. In this paper we used a prescribed threshold of the discrete Laplacian as a uniform geodesic 'proximity' metric for all points of the domain and used this threshold in conjunction with α to generate the medial zones. However, we note that the Laplacian condition may result in topological 'anomalies' that may violate one or more of the above requirements. More robust geodesic proximity metrics, possibly in terms of other differential operators or spectral methods [39], are currently under investigation.

Informally, the 'thickness' of the medial zone increases as α goes to 0 and as the geodesic distance to the crests of the approximate distance function increases—see Fig. 8 and the examples in Section 3. The influence of modifying α on the resulting medial zone is illustrated in Fig. 9.

We note that both medial axis $\mathcal{MA}(\Omega)$ and the family of medial zones $\mathcal{MZ}(\Omega)$ can be extended outside domain Ω.

3. Examples and applications

We now demonstrate the capabilities and effectiveness of the proposed approach on several examples involving polyhedra as well as non-polyhedra. In all these examples we have used a parallelized version of the code that relies on Message Passing Interface (MPI) libraries. More specifically, we have used MPICH2 implementation on a PC workstation with an Intel Xeon 3 GHz computer (dual quad cores) with 4 GB RAM.

3.1. Examples

The first example shown in Fig. 10(a) is a polyhedron undergoing drastic topological changes due to uniform growth of the three concave regions labeled A, B and C in Fig. 10(b). The medial axis of the final instance of the evolving polyhedron is illustrated in Fig. 10(c). As expected, the CPU time linearly decreases with each additional core of the processor used by our parallel algorithm.

The medial axis of the gear-like shape shown in Fig. 11(a) is computed and displayed in Fig. 11(c). We modify the gear-like shape by subtracting a helical shape from the original solid, which, in turn, transforms the connected solid model into a disconnected

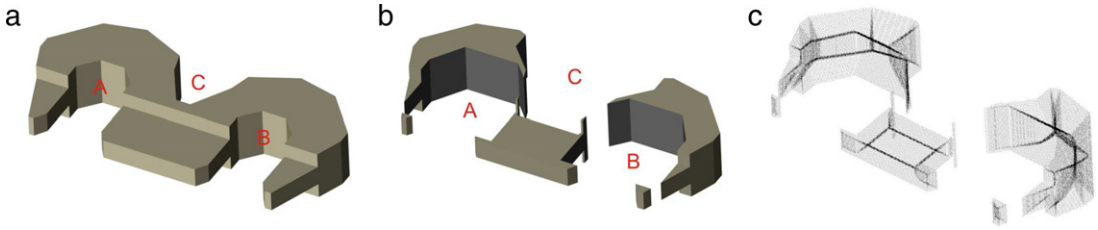

Fig. 10. An evolving, initially connected, polyhedron whose concave regions grow uniformly, and the medial axis of the final (disconnected) instance of the polyhedron.

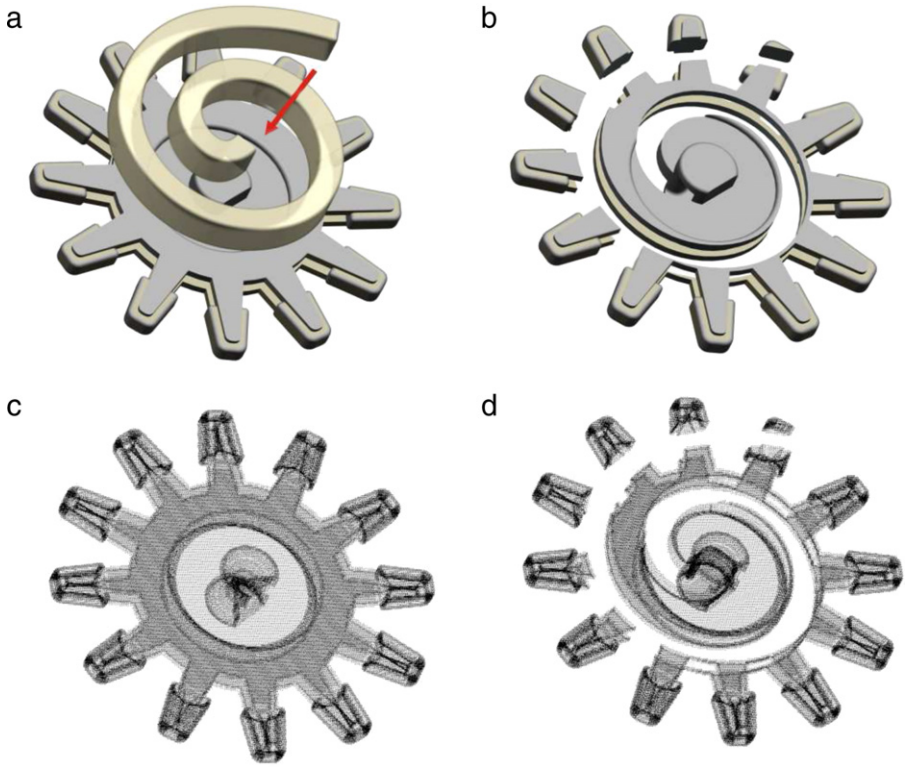

Fig. 11. A gear solid (a) that interacts with a helical solid, which significantly changes the solid topology (b).

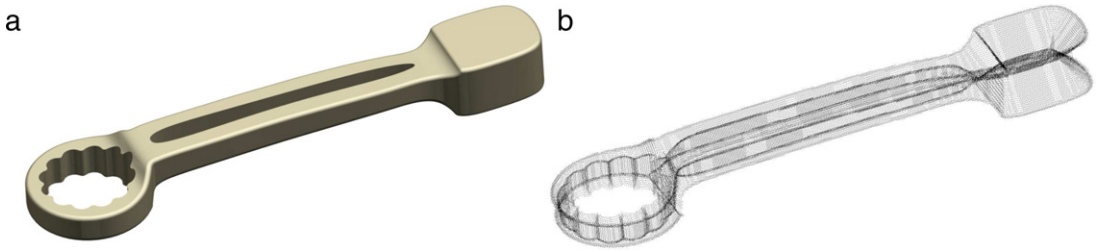

Fig. 12. A wrench like solid and its medial axis.

one as illustrated in Fig. 11(b). The same formulation used to compute the medial axis point of the original gear-like object was used to compute the medial axis of the disconnected set as shown in Fig. 11(d).

Fig. 12(a) shows the model of a wrench that consists of 55 faces, 138 edges and 84 vertices. The medial axis of the wrench is illustrated in 12(b).

The final example that we present here is the 2D domain and its 3D extrusion bounded by both linear and free-form boundaries, and illustrated in Fig. 13(a) and (e). The medial axis of the two shapes are presented in Fig. 13(b) and (f), and the medial zones corresponding to two different values of $\alpha = 0.95$ and $\alpha = 0.9$ are given in Fig. 13(c) and (g) and (d) and (h) respectively.

3.2. Two engineering applications

The concept of a medial zone is as fundamental as that of the medial axis and captures some of the intrinsic properties of a shape. The fact that the medial zones possess some of the critical properties of both the domain itself and of its medial axis, suggests that re-formulating problems in terms of medial zones affords the 'best of both worlds' in practically any application that makes use of geometric reasoning algorithms, including manufacturing and assembly, robotic and autonomous navigation, and design automation. Furthermore, the generality and flexibility of the computational framework is well suited to most general 2- and 3-dimensional problems that can involve deformable shapes and evolving domains.

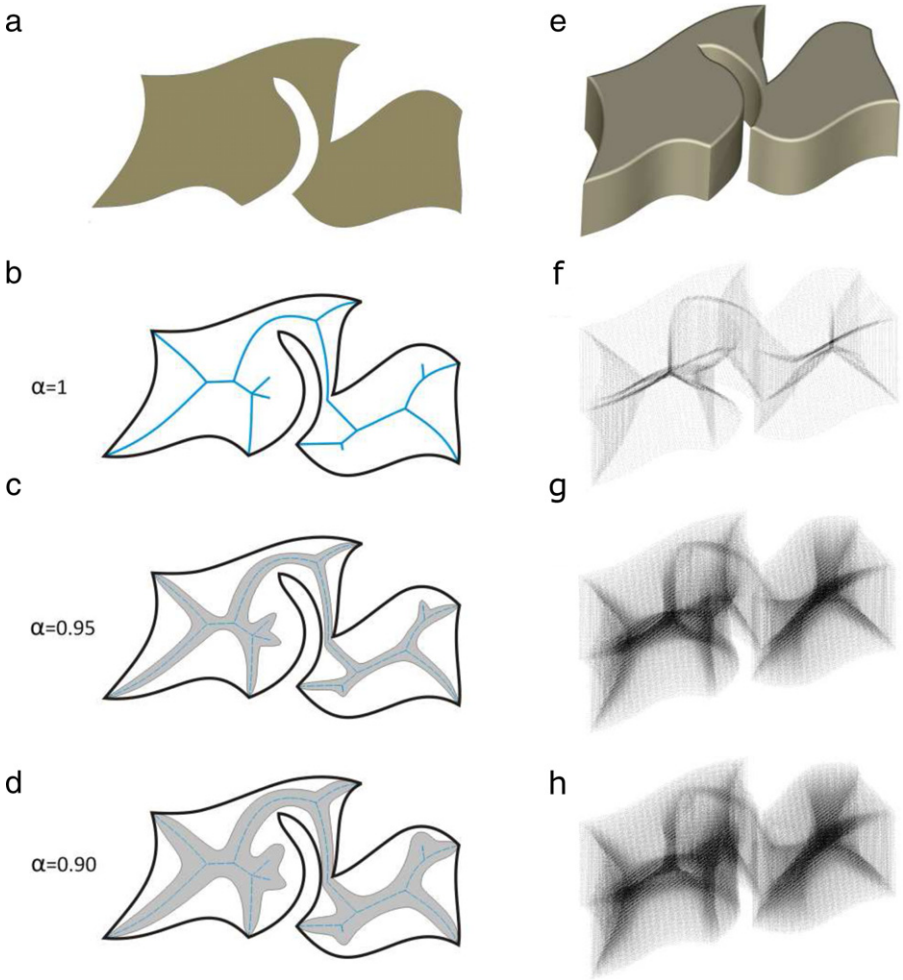

Fig. 13. A 2D domain (a) 3D solid object (e), 2D medial axis (b), 3D medial surface (f), 2D medial zone (c, d), 3D medial zone (g, h).

Path planning

Specifically, in the case of path planning applications, the medial zones define a subset of the domain in which one searches for paths, which is larger than the medial axis set. The resulting path is not only guaranteed to avoid the boundary of the domain and obstacles, in a similar manner with the paths planned along the medial axes, but can be shorter and have a more uniform curvature than any path defined merely along the medial axis.

Specifically, Fig. 14(a) shows the domain of Fig. 13, its medial axis and an initial and final configuration of the path to be computed. Note that these extreme configurations do not have to be on the medial axis or even on the boundary of the domain. Fig. 14(a–c) illustrate the computed shortest paths based on (a) medial axis; (b) medial zone with $\alpha = 0.95$ and (c) a medial zone with $\alpha = 0.9$. It is apparent that as the 'thickness' of the medial zone increases, the length of the path decreases and its curvature becomes more uniform. In fact, the path shown in Fig. 14(c) is 13.34% shorter and with a visibly more uniform curvature than the path of Fig. 14(a). For the 3D domain of Fig. 13(e), the 3D paths planned along the medial axis and inside two medial zones are illustrated[5] in Fig. 14(d). Moreover, the properties of the medial zones guarantee that the resulting path remains inside the medial zone and therefore is guaranteed to be collision-free.

Shape synthesis

Another critical application of medial zones is automatic design synthesis of parts that satisfy prescribed 'functions'. Since this functional information almost never defines the geometry completely, a combinatorial model of a part described in terms of known interacting 'functional boundaries' [40] may be more appropriate for product development activities. This is consistent with the principle of energy exchange of a system through the so-called *energy ports* (see [41,42]), such as attachment and contact/mating features, or subsets where we prescribe boundary conditions. More precisely, medial zones can provide in this context an *explicit* representation of the correspondence between mechanical/functional characteristics and spatial properties.

To illustrate the concept, consider the relatively simple 2D scenario shown in Fig. 15(a), and assume three attachment constraints and known boundary conditions over known subsets of the domain. Assume that the resulting part must provide clearance for a neighboring part, which is prescribed here as a hole H. The goal is to obtain a shape that will remain in the elastic regime of deformation during the energy transfer through the prescribed energy ports. Observe that by computing the medial zone of the convex hull defined by the constraints, we obtain an initial shape that satisfies the topological requirements, i.e., connects the energy ports and satisfies the clearance requirement mentioned above. A more general case of a 3D problem is illustrated in Fig. 15(d–f).

[5] We observe that the medial axis contains sharp features (vertices and edges), which should be reflected in the path planned along the medial axis, i.e., for $\alpha = 1$. However, in order to improve the clarity of the illustration, we offset all three paths and show them in Fig. 14 as generalized cylinders. In turn, this visual artifact is hiding the sharp features that would otherwise be seen in path planned along the medial axis.

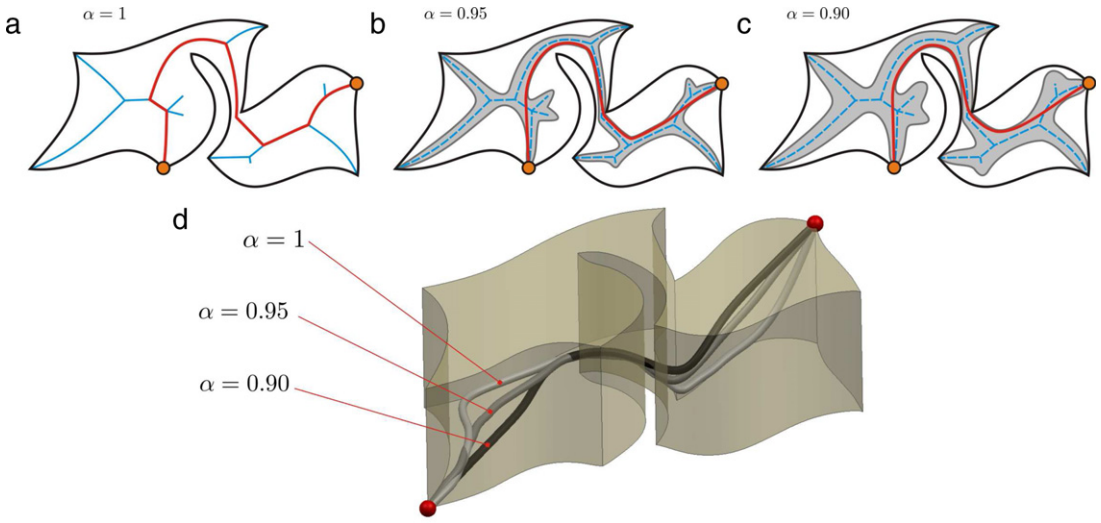

Fig. 14. The 2D domain presented in Fig. 13 with the planned motion between two prescribed configurations for three different values of α.

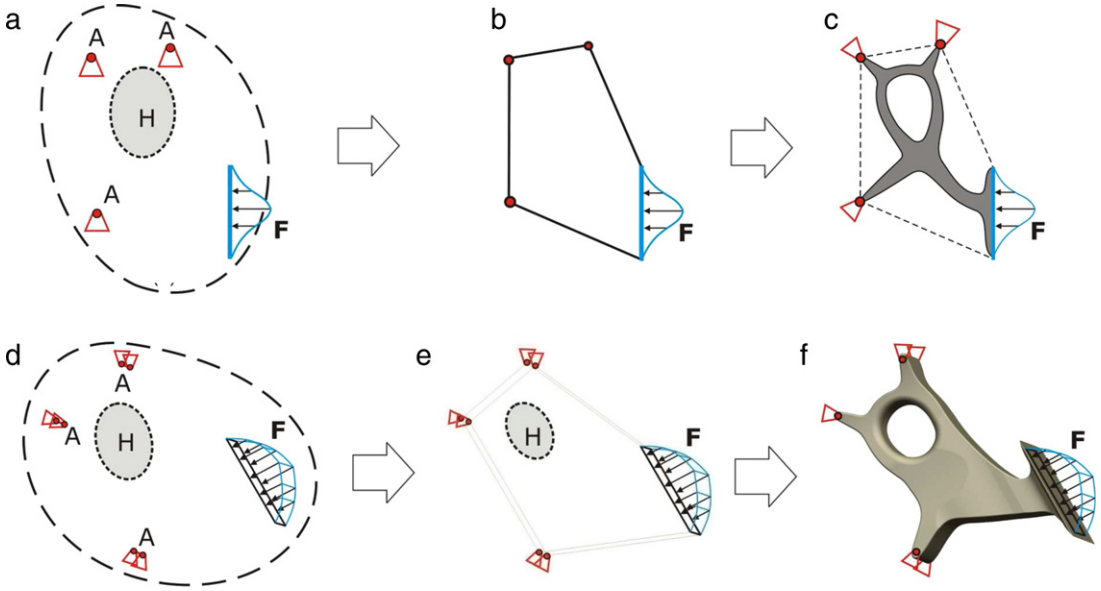

Fig. 15. Automating shape synthesis: the medial zone provides a topologically correct initial guess whose size can be adjusted by controlling the values of α and $\gamma_{\max}$. Note that α provides global control, while $\gamma_{\max}$ is defined at a point and hence it provides, in principle, local control.

Note that this *initial* and *topologically valid* shape can be coupled with solvers of the accompanying boundary value problem to refine the shape until some termination criterion is met [43]. For example, the termination criterion can be formulated in terms of an objective function that minimizes a combination of the maximum compliance with volume/area constraints.

4. Conclusions

The popularity of the medial axis in shape modeling and analysis comes from several of its well known fundamental properties. For example, medial axes capture the connectivity of the domain, have a lower dimension than the space itself, and are closely related to the distance function constructed over the same domain.

We propose the new concept of *medial zones* of an n-dimensional semi-analytic domain Ω that subsumes the medial axis $\mathcal{MA}(\Omega)$ of the same domain as a special case, and can be thought of as a 'thick' skeleton having the same dimension as that of Ω. The medial zones combine some of the intrinsic geometric and topological properties of the domain, such as dimensionality, genus, and connectedness, with the appealing property of the medial axis that keeps the medial axis points in the 'middle' of the domain in a very precise sense [3].

The concept of a medial zone appears to be as fundamental to solid shapes as that of the medial axis, and captures all of the same intrinsic properties of a shape, but has the same dimension as that of the domain. The fact that the medial zones fuse some of the critical geometric and topological properties of both the domain itself and of its medial axis, suggests that re-formulating problems in terms of medial zones can afford the 'best of both worlds' in those applications that rely on geometric reasoning algorithms, including manufacturing and assembly, robotic and autonomous navigation, and design automation.

It is important to note that the proposed framework, which uses constructive representations of shapes with R-functions that operate on real-valued halfspaces as logic operations, can be used to compute both medial axes and medial zones of complex 3D shapes defined as general semi-analytic (or semi-algebraic) rigid or deformable/evolving domains. Our formulation is general,

in the sense that it can handle solid shapes having arbitrarily complex geometry and topology, and supports multiple geometric representations as long as one can compute distance functions to individual subsets of the boundary. At the same time, our formulation intrinsically supports local and parallel computations for domains with rigid or evolving boundaries due to the explicit mapping between the halfspaces bounding the environment and the points of the medial axis/zones.

The geometry of the medial zones can be controlled both globally, through changes in the parameter α as well as locally through changes in the geodesic proximity to the crests of the approximate distance fields that can be, at least theoretically, defined at a point as discussed in Section 2.3. While performing local changes can prove to be a powerful mechanism for manipulating the geometry of the medial zones to support specific applications, such as shape synthesis in engineering design, it can also introduce discontinuities in the boundary of the medial zones and can alter their topological properties. This problem is the subject of current investigations.

Acknowledgments

This work was supported in part by the National Science Foundation grants CMMI-0555937, CAREER award CMMI-0644769, CMMI-0927105, and CNS-0923158. We would like to thank the reviewers for their constructive comments.

References

[1] Blum H. A transformation for extracting new descriptions of shape. In: Models for the perception of speech and visual form. 1967. p. 362–80.
[2] Choi HI, Choi SW, Moon HP. Mathematical theory of medial axis transform. Pacific Journal of Mathematics 1997;1(181):57–88.
[3] Attali D, Boissonnat J-D, Edelsbrunner H. Stability and computation of medial axes—a state-of-the-art report. In: Möller T, Hamann B, Russel R, editors. Mathematical foundations of scientific visualization, computer graphics, and massive data exploration. Mathematics and visualization, Springer Verlag; 2008. p. 109–25.
[4] Chazal F, Soufflet R. Stability and finiteness properties of medial axis and skeleton. Journal of Dynamical and Control Systems 2004;10(2):149–70.
[5] Shapiro V. Semi-analytic geometry with *R*-functions. Acta Numerica 2007; 18(3):239–303.
[6] Lieutier A. Any open bounded subset of $\mathbb{R}^n$ has the same homotopy type than its medial axis. In: SM'03: proceedings of the eighth ACM symposium on solid modeling and applications. New York (NY, USA): ACM; 2003. p. 65–75.
[7] Shaham A, Shamir A, Cohen-Or D. Medial axis based solid representation. In: 9th ACM symposium on solid modeling and applications. 2004. p. 37–44.
[8] van Eede M, Macrini D, Telea A, Sminchisescu C, Dickinson S. Canonical skeletons for shape matching. In: Proc. 18th international conference on pattern recognition ICPR 2006. vol. 2. 2006. p. 64–9.
[9] Liu T-L, Geiger D, Kohn RV. Representation and self-similarity of shapes. In: ICCV'98: proceedings of the sixth international conference on computer vision. Washington (DC, USA): IEEE Computer Society; 1998. p. 1129.
[10] Goh W-B. Strategies for shape matching using skeletons. Computer Vision and Image Understanding 2008;110:326–45.
[11] Sebastian T, Klein P, Kimia B. Recognition of shapes by editing their shock graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence 2004; 26(5):550–71.
[12] Siddiqi K, Shokoufandeh A, Dickinson SJ, Zucker SW. Shock graphs and shape matching. International Journal of Computer Vision 1999;35(1):13–32.
[13] Damon J. Determining the geometry of boundaries of objects from medial data. International Journal of Computer Vision 2005;63:45–64.
[14] Suresh K. Automating the CAD/CAE dimensional reduction process. In: SM'03: proceedings of the eighth ACM symposium on solid modeling and applications. New York (NY, USA): ACM; 2003. p. 76–85.
[15] Sherbrooke EC, Patrikalakis NM, Brisson E. Computation of the medial axis transform of 3-D polyhedra. In: Third ACM symposium on solid modeling and applications. 1995. p. 187–200.
[16] Pizer SM, Siddiqi K, Székely G, Damon JN, Zucker SW. Multiscale medial loci and their properties. International Journal of Computer Vision 2003;55(2–3): 155–79.
[17] Shah J. Gray skeletons and segmentation of shapes. Computer Vision and Image Understanding 2005;99:96–109.
[18] Kimia BB, Tamrakar A. The role of propagation and medial geometry in human vision. In: BMCV'02: proceedings of the second international workshop on biologically motivated computer vision. London (UK): Springer-Verlag; 2002. p. 219–29.
[19] Bookstein FL. The line skeleton. Computer Graphics and Image Processing 1979;11:123–37.
[20] Blum H, Nagel R. Shape description using weighted symmetric axis features. Pattern Recognition 1978;10.
[21] Sampl P. Semi-structured mesh generation based on medial axis. IMR. 2000. p. 21–32.
[22] Sampl P. Medial axis construction in three dimensions and its application to mesh generation. Engineering with Computers 2001;17:234–48.
[23] Quadros William, Shimada Kenji, Owen Steven. Skeleton-based computational method for the generation of a 3D finite element mesh sizing function. Engineering with Computers 2004;20(3):249–64.
[24] Eftekharian A, Ilieş H. Distance functions and skeletal representations of rigid and non-rigid planar shapes. Computer-Aided Design 2009;41(12):865–76.
[25] Rvachev V. Theory of *R*-functions and some applications. Kiev: Naukova Dumka; 1982.
[26] Tor SB, Middleditch AE. Convex decomposition of simple polygons. ACM Transactions on Graphics 1984;3(4):244–65.
[27] Dobkin D, Guibas L, Hershberger J, Snoeyink J. 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. New York (NY, USA): ACM; 1988. p. 31–40.
[28] Shapiro V. A convex deficiency tree algorithm for curved polygons. International Journal of Computational Geometry and Applications 2001;11(2): 215–38.
[29] Shapiro V, Vossler D. Separation for boundary to CSG conversion. ACM Transactions on Graphics 1993;12(1):35–55.
[30] Shapiro V, Vossler DL. Construction and optimization of CSG representations. Computer-Aided Design 1991;23(1):4–20.
[31] Buchele SF, Crawford RH. Three-dimensional halfspace constructive solid geometry tree construction from implicit boundary representations. In: SM'03: proceedings of the eighth ACM symposium on solid modeling and applications. New York (NY, USA): ACM; 2003. p. 135–44.
[32] Pottmann H, Hofer M. Geometry of the squared distance function to curves and surfaces. In: Visualization and mathematics III. Springer; 2003. p. 223–44.
[33] Aigner M, Jüttler B. Robust computation of foot points on implicitly defined curves. In: Mathematical methods for curves and surfaces: Troms. 2005. p. 1–10.
[34] Bing R. The geometric topology of 3-manifolds. Amer. Mathematical Society; 1983.
[35] Luo Z, Tong L, Wang MY, Wang S. Shape and topology optimization of compliant mechanisms using a parameterization level set method. Journal of Computational Physics 2007;227(1):680–705.
[36] Chen J, Shapiro V, Suresh K, Tsukanov I. Shape optimization with topological changes and parametric control. International Journal for Numerical Methods in Engineering 2007;71(3):313–46.
[37] Wardetzky M, Mathur S, Kälberer F, Grinspun E. Discrete Laplace operators: no free lunch. In: ACM SIGGRAPH ASIA 2008 courses. ACM; 2008. p. 19.
[38] Bovik AC. Handbook of image and video processing. Communications, networking and multimedia, Orlando (FL, USA): Academic Press, Inc.; 2005.
[39] Hildebrandt K, Schulz C, von Tycowicz C, Polthier K. Eigenmodes of surface energies for shape analysis. In: Mourrain B, Schaefer S, Xu G, editors. Advances in geometric modeling and processing. Lecture notes in computer science, vol. 6130. Springer; 2010. p. 296–314.
[40] Ilieş H, Shapiro V. An approach to systematic part design. In: Proceedings of the 5th IFIP WG5.2 workshop on geometric modeling in CAD. 1996. p. 383–92.
[41] Shapiro V, Voelcker H. On the role of geometry in mechanical design. Research in Engineering Design 1989;1(1):69–73.
[42] Filippo J, Delgado M, Brie C, Paynter H. A survey of bond graphs: theory, applications and programs. Journal of the Franklin Institute 1991;328(5–6): 565–606.
[43] Eftekharian AA, Ilieş HT. Shape and topology optimization with medial zones. In: IDETC, 37th design automation conference, ASME. 2011.