Distance Solutions for Medial Axis Transform

Hao Xia and Paul G. Tucker

Whittle Laboratory, Department of Engineering, University of Cambridge,
1 JJ Thomson Ave, Camridge, CB3 0DY, England
{hx222,pgt23}@cam.ac.uk

Abstract. A method towards robust and efficient medial axis transform (MAT) of arbitrary domains using distance solutions is presented. The distance field, d, is calculated by solving the hyperbolic-natured Eikonal (or Level Set) equation. The solution is obtained on Cartesian grids. Both the fast-marching method and fast-sweeping method are used to calculate d. Medial axis point clouds are then extracted based on the distance solution via a simple criteria: the Laplacian or the Hessian determinant of d. These point clouds in 2D-pixel and 3D-voxel space are further thinned to curves and surfaces through binary image thinning algorithms. This results in an overall hybrid approach. As an alternative to other methods, the current $d-$MAT procedure bypasses difficulties that are usually encountered by pure geometric methods (e.g. the Voronoi approach), especially in 3D, and provides better accuracy than pure thinning methods. It is also shown that the $d-$MAT approach provides the potential to sculpt/control the MAT form for specialized solution purposes. Various examples are given to demonstrate the current approach.

Keywords: Eikonal equation, wall distance, medial axis transform, thinning.

1 Introduction

In physics, the nearest normal wall distance d is still a key parameter in many turbulence modeling and simulation approaches [10, 28] and also in peripheral applications incorporating additional solution physics [17, 29]. Such examples include explosive front, multiphase flow, and electrostatic particle force modeling. Also, in grid generation the near-wall isovalues of d can be used to form the boundary layer mesh [25, 30], while the far-field d contours can be used as a rapid means of evaluating computational interfaces on unstructured overset meshes with relative movements. Not only being useful in traditional physics, distance field has also been important in computer vision, modeling and computational physics.

In the general fields of shape analysis and solid modeling, including automated meshing, obtaining the medial axis transform (MAT) for a given geometry (or shape/domain) is regarded as an essential step [14, 19, 22, 21, 23]. Efficient and robust techniques are therefore required. Medial axis has been

widely studied by researchers across the community. Several different types of methods exist, mainly including *thinning*, *Voronoi diagram*, *distance field*, and *hybrid methods*. A brief review on these methods can be found in [9]. Most widely discussed are the Voronoi diagram [6, 16, 3, 7] and distance field methods [1, 24, 27]. By nature, the Voronoi diagram is geometry-based and the distance field is often related to general differential equations. One of the advantages of differential equations is that their extension to 3D space is straightforward. On the other hand, although being quite intuitive, the Voronoi type methods often encounter increasing geometric and logical complexity and the approaches rigidity prevents MAT modification/customization.

In this paper, we propose a hybrid differential MAT approach based on the pixel/voxel distance field solution, namely $d-$MAT. Notice the advantage of such a hybrid method is to extract a well approximated medial point cloud on a properly calculated distance field, while pure thinning methods are fundamentally *discrete* and cannot ensure an accurate distance distribution. A similar approach has been suggested by Bouix [5]. However, the Laplacian or Hessian determinant criteria of the distance field proposed in this study seems simpler, more robust, and is independent on the thinning techniques. As shown later in this paper, the differential equation-based approach also provides a biased MAT, in other words the medial axis does not necessarily lie at the mid-point of the space, and it can be sculpted/controlled by the user.

To evaluate d, there are several methods. They can be broadly classified as: search procedures, integral approaches, and numerically solving differential equations. Crude search procedures often require $O(n_v n_s)$ operations where n_v and n_s correspond to the number of volume and surface node points [28]. This can be $O(n_v\sqrt{n_s})$ and $O(n_v \log n_s)$ operations, however, for complex geometries such specialized approaches are difficult to apply [28]. Differential equation-based methods have been discussed in detail in References [29, 28, 30]. Advantageously, they are naturally compatible with vector and parallel computer architectures. The focus here is on the solution of Eikonal differential equation within the framework of the integer Cartesian space using the fast-marching method (FMM) [25] or the fast-sweeping method (FSM) [31].

2 Solutions of Eikonal Equation

2.1 H-J/Eikonal Equation for d

To overcome the expense of calculating d, Sethian [25] considered viscosity solutions of the following general Hamilton-Jacobi equation, $\epsilon \to 0$,

$$\beta\frac{\partial\phi}{\partial t} + H(\nabla\phi, \mathbf{x}, \beta) = \epsilon\nabla^2\phi \tag{1}$$

with $H(\nabla\phi, \mathbf{x}, \beta) = F|\nabla\phi| - (1-\beta)$. A stationary Eikonal equation with hyperbolic nature is obtained when $\beta = 0$:

$$F|\nabla\phi| = 1 + \epsilon\nabla^2\phi \tag{2}$$

where the dependent variable ϕ describes first arrival times of propagating wave fronts from boundaries, and $F(\mathbf{x})$ is the local speed function of these fronts. The wall distance is then simply $d = F\,\phi$, if $F \equiv const.$. With $\epsilon \to 0$, Eq. (2) can be solved by numerical schemes with just enough dissipation to gain an entropy (physical sensible) solutions [25, 28]. As shown by Tucker [28, 30], the right hand side Laplacian is useful and often employed to control the front propagation velocity. However, this potential is not explored here.

With $\epsilon = 0$, Eq. (2) is also the multidimensional form of the boundary value formulation of the front tracking problem, $F = \frac{dx}{d\phi}$, where $F > 0$. Details of the procedures to solve Eq. (2) with $\epsilon \equiv 0$ using FMM and FSM are given in Appendix.

2.2 Domain and Initialization

For simplicity and efficiency, uniform Cartesian grids[1] are used in this d calculation context, namely $\Delta x = \Delta y = \Delta z = h$. The numbers of grid points in x, y and z direction are $I+1$, $J+1$ and $K+1$ respectively. The physical space is transformed and scaled into $x \in [0 : I]$, $y \in [0 : J]$ and $z \in [0 : K]$, so that no grid storage is needed and point locating is straightforward. To locate a given point (x_0, y_0, z_0) in the Cartesian grid, the floor function $\lfloor \cdot \rfloor$ is simply applied. For example, if $i = \lfloor x_0 \rfloor$, then $i \le x_0 \le i+1$, and so are y_0, z_0.

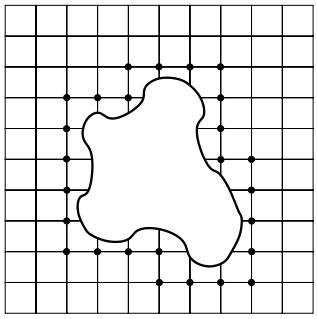
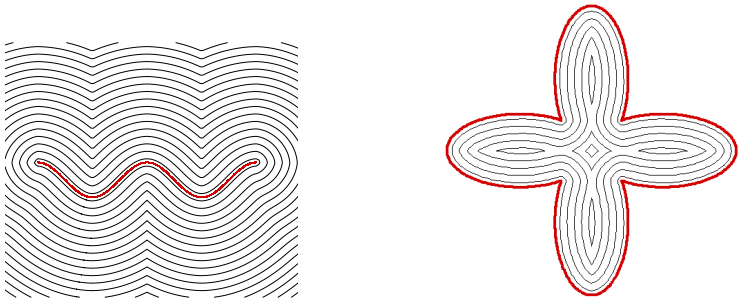


Fig. 1. A 2D schematic showing the wall boundary (or initial front) and the initialization of the solution ($F > 0$ for outward propagation).

Figure 1 shows the position of the wall boundary or initial front inside a Cartesian grid. This arbitrary curve can either be open or closed. To initialize, all immediate nearby grid points of the curve are marked as known (black

[1] Non-uniform Cartesian grids are particularly attractive when the geometric features have a broad range of scales.



(a) (b)

Fig. 2. Distance field contours for (a) a cosine wave, and (b) a curved cross domain

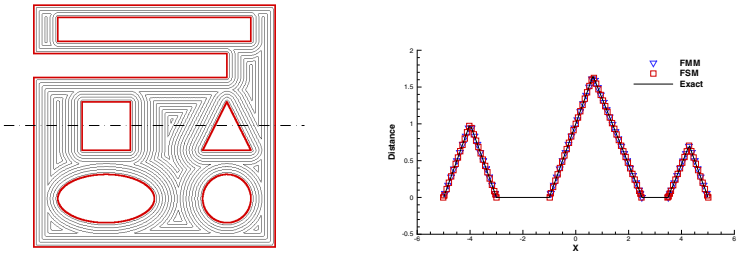
dots in Figure 1) and the value of d on these points are evaluated exactly. The remaining points are marked as 'unknown' and the value of d on them set as ∞. Then, the whole domain is solved numerically with FMM or FSM.

2.3 Distance Solution Examples

Figure 2 shows the distance contours of example 1 and 2 solved on a uniform Cartesian grid with $I = J = 200$. In Example 1, frame (a), the boundary (initial front) is a piece of cosine curve $x \in [0, 4\pi]$, and in example 2, frame (b), the domain is the closed area of a curved cross profile. Both examples show the entropy conformed fronts propagating from the initial position in 2D space. The analogous compression from the concave feature and the expansion from the convex and sharp feature are also clear.

Example 3 is a slightly more complex multiply connected 2D domain. Various shapes (including a rectangle, a square, a triangle, an ellipse and a circle) are subtracted from the domain leaving internal voids with disparate d scales. A Cartesian grid with $I = J = 400$ is used. Both FMM and FSM demonstrate good efficiency to obtain accurate solutions with FMM using 4.7 seconds and a single 2^2-sweep FSM using 2.3 seconds on a single 2.33GHz Intel Xeon CPU core. Figure 3a shows the distance contour. Collisions of fronts are seen in the central area. A center line cut (shown as the dash-dot line) is made through the domain. Figure 3b compares numerical solutions together with the exact d distribution on the center line. The FMM and FSM solutions are close to each other and agree with the analytical solution.

Example 4 is a 3D cubic box, where a variety of shape subtractions are made. FSM is applied to get the solution on a $I = J = K = 200$ uniform 3D Cartesian grid. Figure 4 plots the d contours at three cut planes. The predicted maximum d is within 0.05% of the exact solution.



(a) (b)

Fig. 3. Distance field contours for example 3

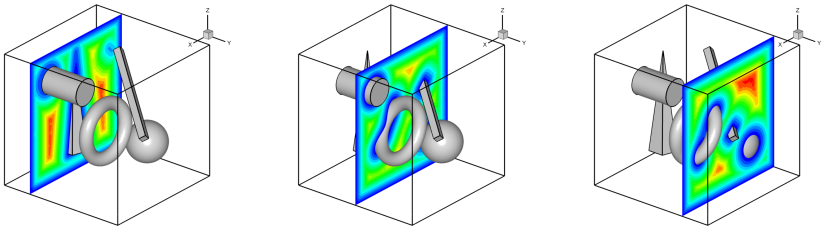


Fig. 4. Distance field obtained for the cubic box domain at various planes. The contour range evenly spreads from 0 (blue) to d_{max} (red).

3 Medial Axis Transform via Distance Field: $d-$MAT

The medial axis of a shape provides a compact representation of its features and their connectivity. As a result, researchers have discovered and are still exploring its use in many fields, such as topology recognition for grid generation [26, 23]. The medial axis is defined when the shape is embedded in an Euclidean space and is endowed with a distance function. Therefore, an expedient route is to efficiently obtain d (as discussed in previous sections) followed by the medial axis construction. (Notice that this approach is different from the pure geometric Voronoi-diagram based approach, in which the equal distance nature of the Voronoi-diagram is a key.) In 3D, a sphere is called *medial* if it meets S, the domain boundary, only tangentially in at least two points. The medial axis M is defined as the closure of the set of centers of all medial spheres. Figure 5 is an illustration in 2D. Only the medial axis inside the domain is considered here. Informally, the medial axis of a surface in 3D is the set of all points that have more than one closest point on the surface. They are often called the medial axis transform (MAT) for that 3D bounded domain.

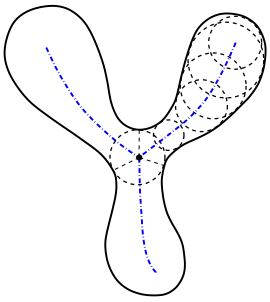


Fig. 5. Schematic of the medial axis for a 2D domain defined by a closed curve

3.1 Feature Detection Criteria

The key step for the MAT construction here is to detect the medial axis feature in a given d field. Notice that the unique property of a medial axis point is that it has equal distance to multiple boundaries. Hence, in space, the medial axis represents the 'local maxima' or non-smoothness of the distance function $d(\mathbf{x})$. For example, in one dimension the medial axis is a single point as shown in Figure 6.

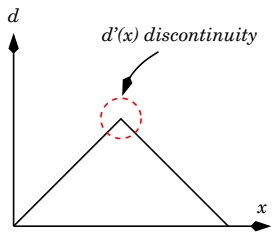
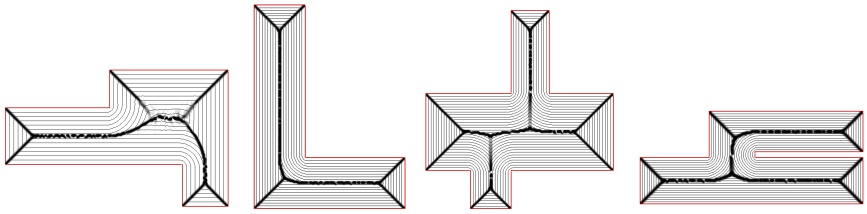


Fig. 6. An example of 1D distance field and its medial axis (vertex)

In multi-dimensions, one of the general formulations to represent this discontinuity in gradient or 'slope jump' is the Laplacian, $\nabla^2 d$. In smooth regions, $\nabla^2 d \equiv 0$. In the vicinity of the local maxima of d, $\nabla^2 d \to -\infty$. Hence, with the numerical approximation of d, the medial axis area can be identified by specifying a criteria such as $\nabla^2 d < -\epsilon$, where positive small number ϵ is a user specified threshold.

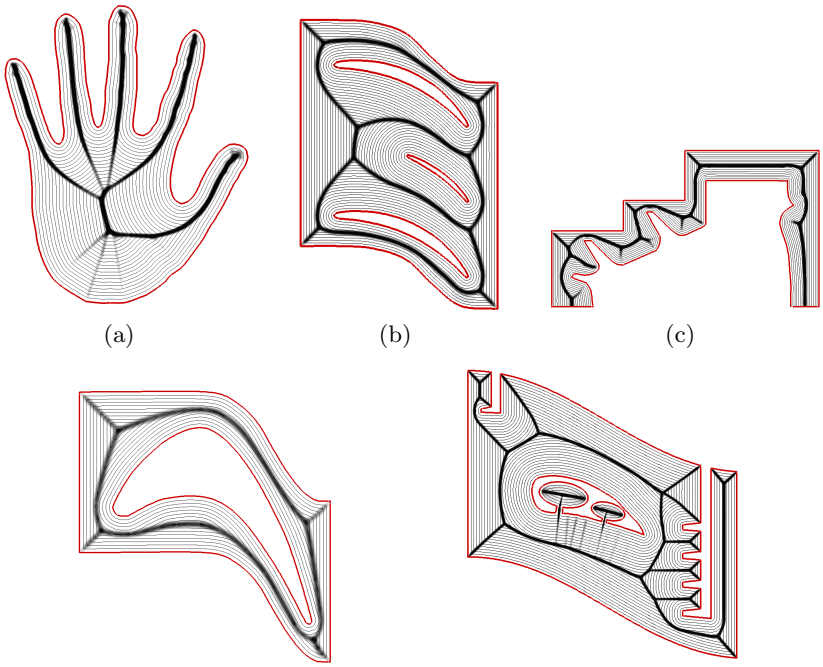
Another way to detect the medial axis feature is through the Hessian matrix determinant. The Hessian matrix, $H_{ij}(d) = \frac{\partial^2 d}{\partial x_i \partial x_j}$, is a square matrix of second-order partial derivatives of a function. It is the true 'second order' derivative in multi-dimensional space. The determinant of the Hessian matrix has been widely used in computer vision [15] (e.g. the 'blob detection') and shock wave detection in computational fluid dynamics [2, 20].



(a) (b) (c) (d)

Fig. 7. Distance field and marked medial axes for simple domains: (a, b) Laplacian based medial axes and (c, d) Hessian based.

Figure 7 gives geometries where d distance contours and medial axes (in thicker line) are shown. For the medial axis identification, Figures 7(a) and (b) use the Laplacian criteria while 7(c) and 7(d) apply the Hessian matrix criteria. The results are very similar. We note that generally the Laplacian is simpler and cheaper to calculate. Figure 8 gives Laplacian based medial



(a) (b) (c)

(d) (e)

Fig. 8. Distance field and marked medial axes for complex domains

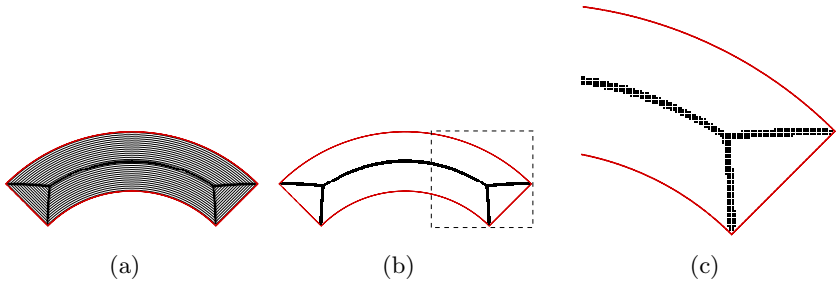


Fig. 9. Point cloud represented medial axis for a bended duct: (a) d contours and marked MA, (b) Underlying point cloud of the MA, (c) Zoom-in.

axes for more complex geometries. These include a hand profle, compressor passage with two blades and one splitter, a turbine blade shroud, a turbine blade in a passage, and a test section with an airfoil and other devices.

3.2 Thinning and Representation

It is very often still not enough to just have the medial axis as a finite thickness curve or surface. The ultimate aim of MAT is to reduce the marked area to thin curves/surfaces and possibly to represent them in a parametric form. This requires a 'thinning' operation. As shown, for example, in Figure 9 once the medial axis area is marked, by say application of $\nabla^2 d < -\epsilon$ one only obtains a point cloud of the underlying mesh nodes. However, to build up the topology, data reduction and connectivity of the point cloud are needed. Because the complex medial axis area does not represent a simple bounding line or surface, surface reconstruction methods or point cloud simplification methods cannot be applied directly, particularly due to branching points of the medial axis. Hence, here, shape recognition methods are considered. A typical procedure is to first convert the point cloud into a binary image and then thin the 'width' of the band (or shell) in the image, and finally transfer back to medial axis skeleton of the domain.

Notice that the term 'thinning' is a morphological operation that is used to remove selected foreground pixels/voxels from 2D/3D binary images, somewhat like erosion or opening. The result of thinning can be regarded as the minimum set of the topology preserving representation of the original shape. It can be used for many applications, and is often useful for skeletonization. Thinning is normally only applied to binary images, and produces another binary image as output. In the current case, because the uniform Cartesian grids are used, they are conformed with binary (black & white) images. The marked medial axis points can be regarded as 'black' pixels and the rest are 'white'. Therefore, the thinning methodologies for binary images can be applied here to thin medial axis point clouds.

Thinning in $\mathbb{Z}^2$

Thinning methodologies for 2D binary images have been extensively studied since 1980s (see [13] for a review). In this study, the algorithm described in [11] is adopted. Consider a pixel at p and its eight neighbors, see Figure 10. The following notations are used. The pixels $x_1, x_2, \ldots, x_8$ are the 8-neighbors of p and are collectively denoted by $N(p)$. They are said to be 8-adjacent to p. The pixels, x_1, x_3, x_5, x_7 are the 4-neighbors of and 4-adjacent to p. The value of each pixel can either be 0 or 1 meaning 'white' and 'black'.

x_4	x_3	x_2
x_5	p	x_1
x_6	x_7	x_8

Fig. 10. Pixel adjacencies of $N_8(p)$ in $\mathbb{Z}^2$, where the value at each pixel can either be 0 or 1

The thinning algorithm is generally described as: p is deleted (i.e. value changed from 1 to 0), if only if all the following conditions are satisfied:

C1: $X_H(p) = 1$,
C2: $2 \leq \min[n_1(p),\ n_2(p)] \leq 3$,
C3: $(x_2 \vee x_3 \vee x_8) \wedge x_1 = 0$ for odd iterations; $(x_6 \vee x_7 \vee x_4) \wedge x_5 = 0$ for even iterations.

where $\vee$, $\wedge$ are logical 'OR', 'AND' operators, and

$$X_H(p) = \sum_{i=1}^{4} b_i \quad \text{and} \quad b_i = \begin{cases} 1, & \text{if } (x_{2i-1} = 0) \text{ and } (x_{2i-1} \text{ or } x_{2i+1} = 1) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k} \quad \text{and} \quad n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1} \tag{4}$$

A note to this algorithm is that it preserves the connectivity of the pixels, or to say: no two remaining pixels after thinning are 'disconnected' if they were neighbors before the thinning. Therefore, if p and x_1 are the remaining pixels in Figure 10, the local connectivity is straightforward, because x_1 is either directly or diagonally adjacent to p. A simple $N_8(p)$ neighborhood check restores the full connectivity. p is identified as a medial axis branching point if $\sum N_8(p) > 2$; for p, where $\sum N_8(p) = 1$, p is an end point of a medial axis; and finally for any p with $\sum N_8(p) = 2$, p is a middle point in a medial axis curve.

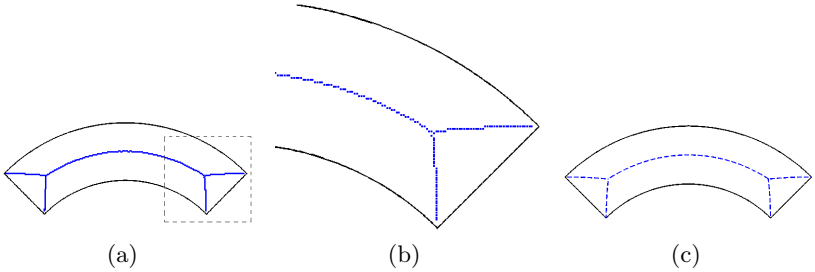


Fig. 11. The bended duct after applying thinning: (a) Single-pixel-wide point cloud, (b) Zoom-in, (c) Rebuilt connectivity and splined MA curves.

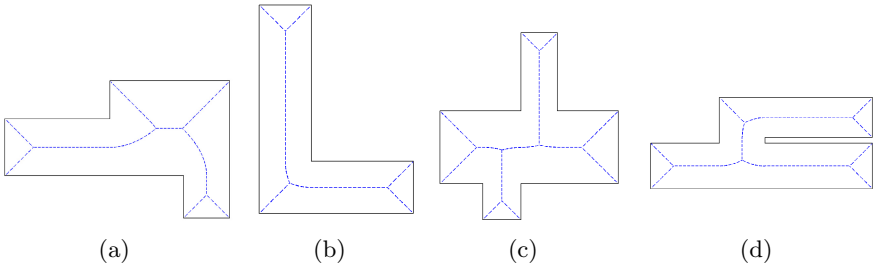


Fig. 12. Fina medial axes for simple domains previously shown in Figure 7

The typical procedure of applying this algorithm can be described with the aid of Figures 9 & 11. First, the binary image is obtained from the marked medial axis point cloud by one of the feature detecting criteria on the d solution, as in Figure 9b & c. Secondly, the binary image is thinned using the above pixel thinning algorithm, shown in Figure 11a & b. Thirdly, the pixels are transferred from $\mathbb{Z}^2$ back to $\mathbb{R}^2$. Finally, and optionally, the MAT points can be splined in smooth curves, as in frame (c). Figures 12 & 13 show the results of pixel thinning applied to the cases previously shown in Figures 7 & 8.

Thinning in $\mathbb{Z}^3$

It is much more complicated to thin a binary image with $p \in \mathbb{Z}^3$. The recent study of Palágyi [18] shows that a robust (both topology-preserving and maximum thinning) surface-thinning algorithm is possible. It deals with the full 26-adjacent voxels to p, see Figure 14. This is an analogous extension from the 2D thinning algorithms but different in the underlying details. We adopt this method in this study to thin the equivalent $\mathbb{Z}^3$ binary image representing the marked medial axis point cloud in 3D.

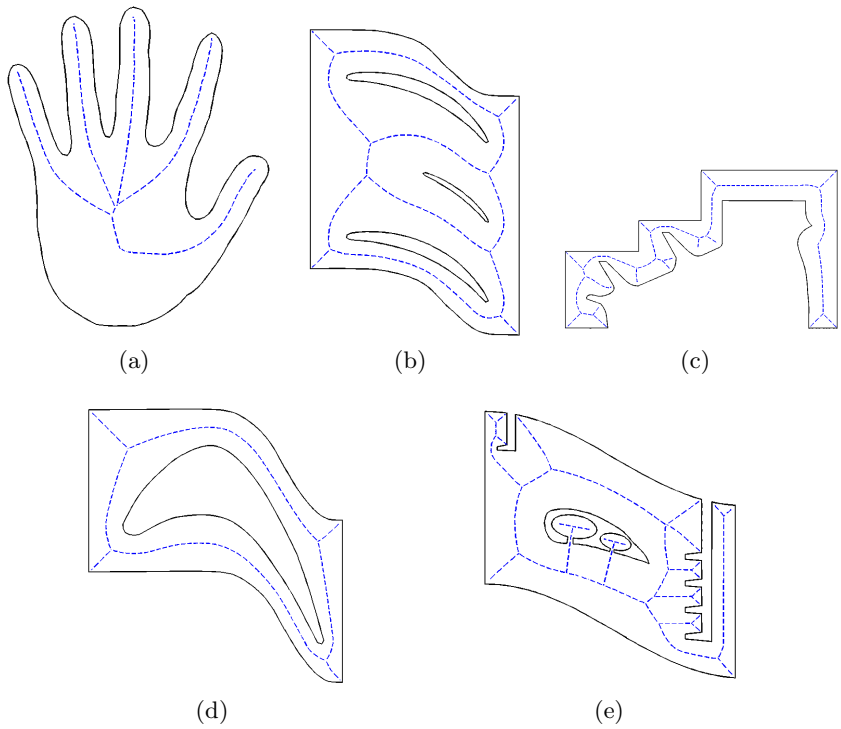


Fig. 13. Final medial axes for more complex domains previously shown in Figure 8

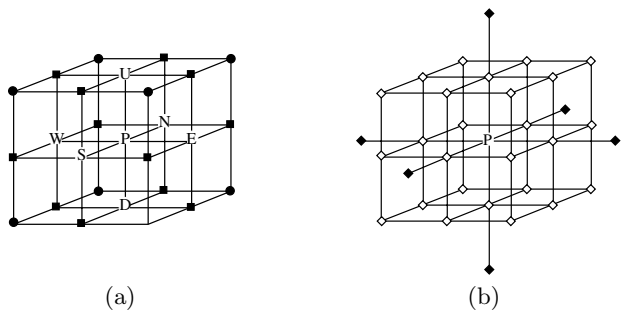
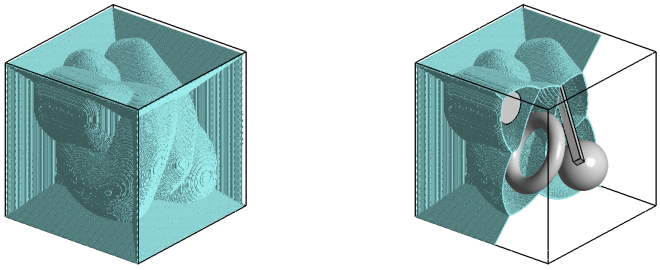


Fig. 14. Voxel adjacencies of $N_{26}(p)$ in $\mathbb{Z}^3$

Consider a voxel p. Its directly connected 6-adjacent neighbors are denoted as **U**, **N**, **E**, **S**, **W** and **D**, shown in Figure 14a, also denoted as $N_6(p)$. The set $N_{18}(p)$ contains the set $N_6(p)$ and the 12 points marked "black square". The set $N_{26}(p)$ contains the set $N_{18}(p)$ and the 8 points marked "black circle". Whether p is deletable depends on $N_{26}(p)$ marked "diamond" and six

additional points marked "black diamond", see Figure 14b. An important step is to construct structuring elements to delete the extra voxels. Due to the extended neighbor dependency, this procedure, although becoming more complex than its 2D counterpart, still seems manageable. A complete set of 114 structuring elements are suggested in [18]. Details of this algorithm can be found in the original paper. A note to the application of this thinning algorithm is that in our study the extracted medial point cloud by the Laplacian criteria is already quite thin, usually a few voxels thick, say $N_{th} \sim 4$ voxels. The iterations involved in the above thinning is at most $N_{th}/2 \sim 2$, and this number is independent on the size of the image, i.e. the total voxels in space.



(a) (b)

Fig. 15. Full and partial view of the 3D medial axis surface through voxel thinning

For example, the 3D medial surfaces, shown in Figure 15, are obtained using this approach. The FSM d solution of the $200 \times 200 \times 200$ Cartesian grid in $\mathbb{Z}^3$ is calculated first. The voxels with local maxima of d are marked using the Laplacian criteria, and finally thinned into the shown complex medial surface inside the domain, between objects.

3.3 Solution Superposition

One useful feature of the current $d-$MAT approach is that it allows solution superposition. For instance, in Figure 16, frames (a), (b) and (c) are three independent solutions. The first is simply a square domain. The other two are solid points at two different locations placed in an infinite domain. After applying the minimum superposition:

$$d = \min(d_a, d_b, d_c) \tag{5}$$

where d_a, d_b and d_c are the independent solutions, the solution shown in frame (d) is obtained. The solution is the same as if it was solved with the three boundary conditions imposed simultaneously. This decomposition is particularly useful because different speed functions can be applied to different solutions before the superposition. For example, Figure 17 represents a different pattern of the distance field after the superposition of

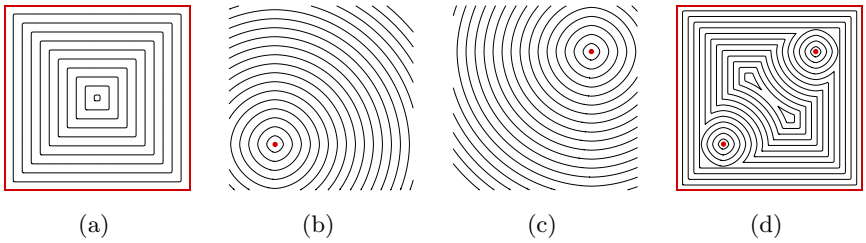


Fig. 16. Eikonal d solution superposition for a square domain and two points in infinite domains: (a) Square domain solution; (b, c) Point source solutions; and (d) Superposition of solutions

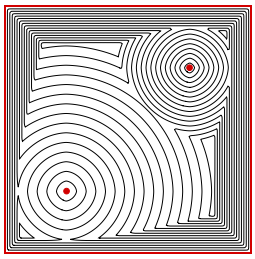


Fig. 17. Solution superposition for a square box and two points distance field solution with $F_a : F_b : F_c = 1 : 4 : 2$

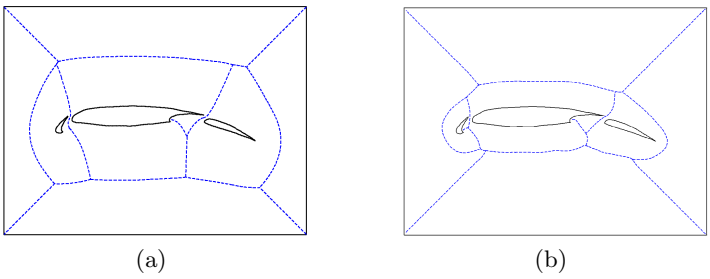


Fig. 18. Standard and biased medial axes for a multi-element airfoil domain

three solutions with different speeds, in which case the speed functions are $F_a : F_b : F_c = 1 : 4 : 2$.

With this useful feature, one can build biased medial axes by setting different speed functions to superimposed solutions. Figure 18(a) shows the standard medial axes of the flow domain of a multi-element airfoil, and frame (b) demonstrates the biased medial axes where front speed from the far field

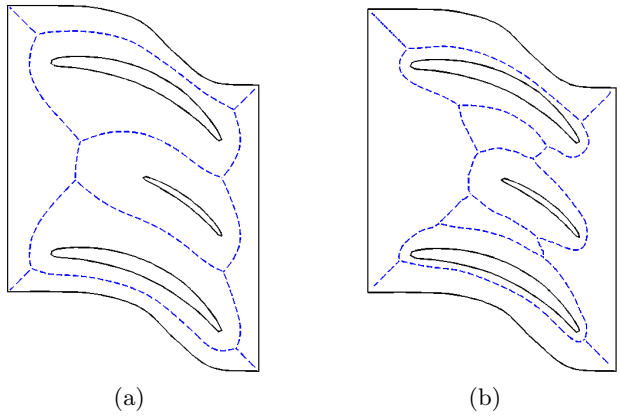


Fig. 19. Standard and biased medial axes for a compressor passage with two blades and one splitter

boundaries is three times faster than the one from the airfoil. Sometimes to create more sensible domain decomposition, it is desirable to have such a biased medial axis. For instance, in Figure 19, the biased medial axes may provide more sensible information for domain division. We also note that to further sculpt the medial axis (see Reference [30]) other d dependent functions can be added to the differential Eikonal type equation. This sculpting can provide a $d-$MAT for high quality hexehedral meshing. This is left as future work.

4 Conclusion

Fast marching and sweeping methods of the Eikonal equation have been applied to obtain distances, d, in arbitrary domains. A continuous distance field based method for creating the medial axis transform $d-$MAT has been proposed and demonstrated with various examples. This $d-$MAT approach combines fast Eikonal equation solutions with the pixel/voxel thinning. The link between the distance solution and the medial axes are established upon the use of simple criteria of $\nabla^2 d$ and $|H_{ij}(d)|$. It has been found to be a robust alternative to the classical pure geometric or pure thinning methods. Solution superposition and boundary dependent front speeds have been found useful to generate the biased medial axis, providing more general information than the standard medial axis. Because the differential equation does not pose any particular challenge when extended from 2D to 3D, it is a promising approach for medial axes evaluation in geometrically complex domains. The differential equation basis allows for flexible $d-$MAT customization through the addition of source terms that are functions of d itself. Future work will be focused on the representation of the 3D thinned medial surfaces which

will involve building the connectivity through local voxel neighborhood, resampling the voxels and splining the surfaces.

Acknowledgments. This work is funded by UK Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/E057233/1. We would like to thank Professor Kálmán Palágyi from University of Szeged for sharing the implementation of the 3D thinning algorithm. Helpful discussions with Dr Neil Dodgson from the Computer Laboratory at Cambridge University are also appreciated.

A Finite Difference Solution Procedures

A.1 Fast-Marching Method

The gradient $\nabla\phi$ in Eq. (2) is discretized by Godunov's upwind difference scheme. This correctly chooses the physically vanishing weak viscosity solution. The following form is suggested by Osher [17] and Sethian [25],

$$\begin{pmatrix} \max(D^{-x}_{ijk}\phi - D^{+x}_{ijk}\phi,\ 0)^2 + \\ \max(D^{-y}_{ijk}\phi - D^{+y}_{ijk}\phi,\ 0)^2 + \\ \max(D^{-z}_{ijk}\phi - D^{+z}_{ijk}\phi,\ 0)^2 \end{pmatrix} = \frac{1}{F^2_{ijk}} \tag{6}$$

where D^-_{ijk} and D^+_{ijk} are the first-order backward and forward finite difference operators given by:

$$\begin{aligned} D^{-x}_{ijk}\phi &= \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\Delta x}, & D^{+x}_{ijk}\phi &= \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x} \\ D^{-y}_{ijk}\phi &= \frac{\phi_{i,j,k} - \phi_{i,j-1,k}}{\Delta y}, & D^{+y}_{ijk}\phi &= \frac{\phi_{i,j+1,k} - \phi_{i,j,k}}{\Delta y} \\ D^{-z}_{ijk}\phi &= \frac{\phi_{i,j,k} - \phi_{i,j,k-1}}{\Delta z}, & D^{+z}_{ijk}\phi &= \frac{\phi_{i,j,k+1} - \phi_{i,j,k}}{\Delta z} \end{aligned} \tag{7}$$

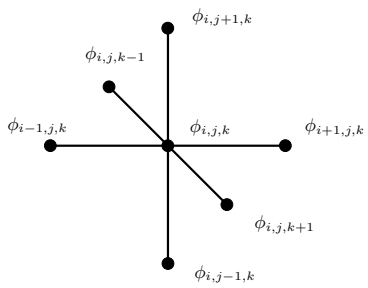


Fig. 20. Finite difference stencil at (i, j, k)

The stencil of grid points involved is shown in Figure 20. Although schemes with higher order accuracy are available, Refs. [25, 28] suggest the first-order scheme is sufficient for d calculation. Through substitution of the above equation Eq. (6) can be rewritten as:

$$\max\left(\frac{\phi-\phi_1}{\Delta x},0\right)^2+\max\left(\frac{\phi-\phi_2}{\Delta y},0\right)^2+\max\left(\frac{\phi-\phi_3}{\Delta z},0\right)^2=\frac{1}{F_{ijk}^2} \quad (8)$$

where

$$\begin{aligned}\phi &= \phi_{i,j,k}\\ \phi_1 &= \min(\phi_{i-1,j,k},\ \phi_{i+1,j,k})\\ \phi_2 &= \min(\phi_{i,j-1,k},\ \phi_{i,j+1,k})\\ \phi_3 &= \min(\phi_{i,j,k-1},\ \phi_{i,j,k+1})\end{aligned} \quad (9)$$

In a way similar to the Dijkstra algorithm [8], the idea of the Fast-Marching Method (FMM) is to introduce an order to the selection of grid points. This order is based on a causality criteria, where the arrival time ϕ at any point depends only on the neighbors that have smaller values. Therefore, the FMM relies on propagating the information in one direction from smaller values of ϕ to larger ones. This is also convenient for Eq. (8), since it can be further reduced to a standard quadratic equation,

$$\left(\frac{\phi-\phi_1}{\Delta x}\right)^2+\left(\frac{\phi-\phi_2}{\Delta y}\right)^2+\left(\frac{\phi-\phi_3}{\Delta z}\right)^2=\frac{1}{F_{ijk}^2} \quad (10)$$

The general algorithm can be described as follows. The front constructs a narrow band of trial points, distinct from the accepted (known) points and the far (unknown) points. Among the current trial points, the point, e.g. denoted as A, with smallest ϕ is moved from 'trial' status to 'known'. Each neighbor of A is added to trial points if it was 'unknown' before, and the quadratic equation (10) is solved for each neighboring point of A. Therefore, a new narrow band of trial points is formed. A recursive procedure is performed until there are no more 'unknown' points. The key to this algorithm is to find the smallest ϕ among the trial points. To store the trial points A special min-heap data structure [4] is suggested in [25], in which the worst case of finding the smallest ϕ has the complexity of $O(\log N)$. This is significantly improved compared with a crude search, which is of $O(N)$.

A.2 Fast-Sweeping Method

To avoid implementing any complex data structure essentially needed in the marching methods, the Fast-Sweeping Method (FSM) [31] is also considered. For discretization, FSM shares the same finite difference scheme with FMM, i.e. the first-order Godunov's scheme Eq. (6). However, the FSM does not have the FMM's convenience of reducing Eq. (8) to a standard quadratic

equation. A solution procedure of Eq. (8) is suggested in [31] by seeking the solution of the following general form,

$$\max(x - a_1,\ 0)^2 + \max(x - a_2,\ 0)^2 + \cdots + \max(x - a_n,\ 0)^2 = b^2 \tag{11}$$

where n is the number of dimensions. The above equation is re-organized so that $a_1 \leq a_2 \leq \cdots \leq a_n$. When $n = 2$, the unique solution of the above equation is:

$$\bar{x} = \begin{cases} a_1 + b\,, & \text{if } \ a_1 + b \leq a_2, \\ \dfrac{a_1 + a_2 + \sqrt{2b^2 - (a_1 - a_2)^2}}{2}\,, & \text{otherwise.} \end{cases} \tag{12}$$

and for $n \geq 3$ a recursive procedure, detailed in [31], is required to find $\bar{x}$, in which an integer p, $1 \leq p \leq n$, is sought such that $a_p < \bar{x} \leq a_{p+1}$ is the unique solution of

$$(x - a_1)^2 + (x - a_2)^2 + \cdots + (x - a_p)^2 = b^2 \tag{13}$$

FSM involves Gauss-Seidel iterations with alternating sweep orderings. At each grid point $x_{i,j,k}$ whose value is not initialized as known (or fixed), the solution of Eq. (8), denoted by $\bar{\phi}_{i,j,k}$, is computed from the most recent values of its neighbors $\phi_{i\pm1,j,k}$, $\phi_{i,j\pm1,k}$ and $\phi_{i,j,k\pm1}$, and then the value at $x_{i,j,k}$ is updated by:

$$\phi^{new}_{i,j,k} = \min(\phi^{old}_{i,j,k},\ \bar{\phi}_{i,j,k}) \tag{14}$$

The whole domain is repeatedly swept with 2^n alternating orderings. For example when $n = 3$, these orderings are

$$\{i = 0 : I \text{ or } I : 0\} \times \{j = 0 : J \text{ or } J : 0\} \times \{k = 0 : K \text{ or } K : 0\}$$

Notice that often just one 2^n sweep cannot guarantee a converged solution for an arbitrary speed function $F(\mathbf{x})$, especially those with complex characteristics [31, 12]. However, in practice, for constant F or those with simple characteristics one 2^n sweep is sufficient for an accurate solution.

References

1. Ahuja, N., Chuang, J.H.: Shape Representation Using a Generalized Potential Field Model. IEEE Trans. Pattern Anal. Mach. Intell. 19(2), 169–176 (1997)
2. Ait-Ali-Yahia, D., Habashi, W.G., Tam, A., Vallet, M., Fortin, M.: A directionally adaptive methodoloty using an edge-based error estimate on quadrilateral grids. International Journal for Numerical Methods in Fluids 23(7), 673–690 (1998)
3. Ang, P.Y., Armstrong, C.G.: Adaptive Curvature-Sensitive Meshing of the Medial Axis. In: Proceedings of the 10th International Meshing Roundtable, pp. 155–165 (2001)

4. Atkinson, M.D., Sack, J.R., Santoro, B., Strothotte, T.: Min-max heaps and generalized priority queues. Commun. ACM 29(10), 996–1000 (1986)
5. Bouix, S., Siddiqi, K.: Divergence-Based Medial Surfaces. In: Proceedings of the Sixth European Conference on Computer Vision, pp. 603–618 (2000)
6. Chou, J.J.: Voronoi diagrams for planar shapes. IEEE Computer Graphics and Applications 15(20), 52–59 (1995)
7. Dey, T.K., Zhao, W.: Approximate medial axis as a voronoi subcomplex. In: Proceedings of the seventh ACM symposium on Solid modeling and applications, pp. 356–366 (2002)
8. Dijkstra, E.: A Note on Two Problems in Connection with Graphs. Numerische Mathematik 1, 269–271 (1959)
9. Du, H., Qin, H.: Medial axis extraction and shape manipulation of solid objects using parabolic PDEs. In: Proceedings of the ninth ACM symposium on Solid modeling and applications, pp. 25–35 (2004)
10. Fares, E., Schröder, W.: A differential equation for approximate wall distance. International Journal for Numerical Methods in Fluids 39, 743–762 (2002)
11. Guo, Z., Hall, R.W.: Parallel thinning with two-subiteration algorithms. Commun. ACM 32(3), 359–373 (1989)
12. Jeong, W.-K., Whitaker, R.T.: A Fast Iterative Method for Eikonal Equations. SIAM J. Sci. Comput. 30(5), 2512–2534 (2008)
13. Lam, L., Lee, S.-W., de Suen, C.Y.: Thinning Methodologies-A Comprehensive Survey. IEEE Trans. Pattern Anal. Mach. Intell. 14(9), 869–885 (1992)
14. Li, T.S., Mckeag, R.M., Armstrong, C.G.: Hexahedral meshing using midpoint subdivision and integer programming. Comput. Methods Appl. Mech. Engrg. 124, 171–193 (1995)
15. Mikolajczyk, K., Schmid, C.: Scale & Affine Invariant Interest Point Detectors. Int. J. Comput. Vision 60(1), 63–86 (2004)
16. Näf, M., Székely, G., Kikinis, R., Shenton, M.E., Kübler, O.: 3D Voronoi skeletons and their usage for the characterization and recognition of 3D organ shape. Comput. Vis. Image Underst. 66(2), 147–161 (1997)
17. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi f ormulations. Journal of Computational Physics 79, 12–49 (1988)
18. Palágyi, K.: A 3D fully parallel surface-thinning algorithm. Theor. Comput. Sci. 406(1-2), 119–135 (2008)
19. Price, M.A., Armstrong, C.G.: Hexahedral mesh generation by medial surface subdivision: part II. solids with flat and concave edges. International Journal for Numerical Methods in Engineering 40, 111–136 (1997)
20. Qin, N., Liu, X.: Flow feature aligned grid adaptation. International Journal for Numerical Methods in Engineering 67(6), 62–72 (2006)
21. Quadros, W.R., Owen, S.J., Brewer, M., Shimada, K.: Finite Element Mesh Sizing for Surfaces Using Skeleton. In: Proceedings of the 13th International Meshing Roundtable, pp. 389–400 (2004)
22. Quadros, W.R., Ramaswami, K., Prinz, F.B., Gurumoorthy, B.: LayTracks: A New Approach To Automated Quadrilateral Mesh Generation using MAT. In: Proceedings of the 9th International Meshing Roundtable, pp. 239–250 (2000)
23. Rigby, D.L.: TopMaker: A Technique for Automatic Multi-Block Topology Generation Using the Medial Axis, NASA Report: NASA/CR-2004-213044 (2004)
24. Rumpf, M., Telea, A.: A continuous skeletonization method based on level sets. In: Proceedings of the symposium on Data Visualisation, p. 151 (2002)

25. Sethian, J.: Level Set Methods and Fast Marching Methods: Evolving Interfaces. In: Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, Cambridge (1999)
26. Sheehy, D.J., Armstrong, C.G., Robinson, D.J.: Shape Description By Medial Surface Construction. IEEE Transactions on Visualization and Computer Graphics 2(1), 62–72 (1996)
27. Sud, A., Otaduy, M.A., Manocha, D.: DiFi: Fast 3D Distance Field Computation Using Graphics Hardware. In: Proceedings of Eurographics 2004, pp. 557–566 (2004)
28. Tucker, P.G.: Differential equation-based wall distance computation for DES and RANS. J. Comput. Phys. 190(1), 229–248 (2003)
29. Tucker, P.G., Rumsey, C.L., Bartels, R.E., Biedron, R.T.: Transport Equation Based Wall Distance Computations Aimed at Flows with Time-Dependent Geometry, NASA Report, NASA-TM-2003-212680 (2003)
30. Tucker, P.G., Rumsey, C.L., Spalart, P.R., Bartels, R.E., Biedron, R.T.: Computation of Wall Distances Based on Differential Equations. AIAA Journal 43(3) (2005)
31. Zhao, H.: A Fast Sweeping Method for Eikonal Equations. Mathematics of Computation 74(250), 603–627 (2004)