PDE-Based Medial Axis Extraction and Shape Manipulation of Arbitrary Meshes

Haixia Du[1], Terry Yoo[2], and Hong Qin[3]

[1]Sterne Kessler Goldstein & Fox, PLLC, USA
[2]National Library of Medicine, National Institutes of Health, USA
[3]Computer Science Department, Stony Brook University, USA

Abstract

Shape skeletonization (i.e., medial axis extraction) is powerful in many visual computing applications, such as pattern recognition, object segmentation, registration, and animation. In this paper, we expand the use of diffusion equations combined with distance field information to approximate medial axes of arbitrary 3D solids represented by polygonal meshes based on their differential properties. It offers an alternative but natural way for medial axis extraction for commonly used 3D polygonal models. By solving the PDE along time axis, our system can not only quickly extract diffusion-based medial axes of input meshes, but also allow users to visualize the extraction process at each time step. In addition, our model provides users a set of manipulation toolkits to sculpt extracted medial axes, then use diffusion-based techniques to recover corresponding deformed shapes according to the original input datasets. This skeleton-based shape manipulation offers a fast and easy way for animation and deformation of complicated mesh objects.

1 Introduction and Motivation

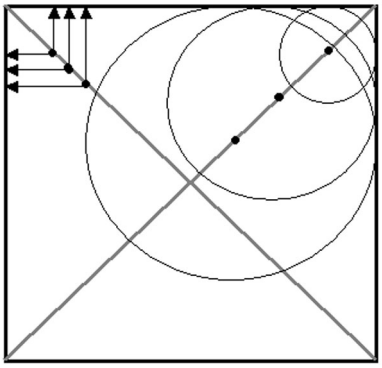
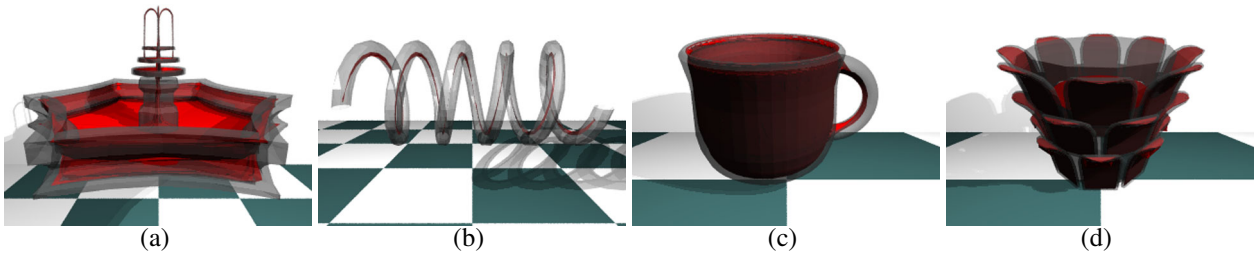


Figure 1: 2D illustration for medial axis.

Medial axis, also known as skeleton, offers much more simple and compact representations for arbitrary complex geometric and/or solid objects. Ever since it was first proposed and named by Blum [8][9], medial axis has started to gain more and more popularity in visual computing areas especially in recent years. It collectively provides useful shape information such as topology, orientation, and local properties in an intuitive and compact fashion. For instance, the medial axis of a 2D polygon can be directly associated with the concept of *grassfire transform*: By igniting boundary points of the polygon, the fire propagates inward from the boundary at a uniform speed, and where the fire front meets and extinguishes itself defines the medial axis in a natural and physically plausible way. More mathematically, the medial axis can be defined as the locus of all centers of circles inside the 2D polygon (or spheres inside the 3D object) that are tangent to the boundary in two or more places [10]. The points on the medial axis (or skeleton) of an object usually have more than one closest point on the boundary of the object. Fig. 1 shows an illustration of the medial axis for a 2D shape. In practice medial axis is also called medial surface and frequently



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

Figure 2: Medial axis extraction using our PDE technique. The medial axes are shown in red with transparent datasets surrounding them.

referred as the 3D skeleton (especially in bio-medical applications) for 3D models. Hence the extraction of medial axis is oftentimes called skeletonization.

There are several unique advantages of using medial axis or skeleton to model geometric objects. First, it provides localization of features such as anatomical landmarks (which are extremely valuable in bio-medical applications). Second, it separates thickness information (e.g., radius of medial axis or skeleton) from orientational and topological information, i.e., shape features can be subdivided into radial, orientational and location information in order to facilitate statistical analysis. Third, shape differences between objects can be quantified in a more intuitive and accurate way. Fourth, it is more expeditious to capture coarse-scale changes from the acquired models, making it more stable and robust to handle noisy datasets.

Skeletons can be used to repair topological errors on solid models in the form of small surface handles [31]. 3D line skeletons also contributes for modifying topologies of 3D models [20].

There are various developed algorithms using different techniques for medial axis extraction in both 2D and 3D. However, the stable numerical computation of medial axis remains a challenging problem.

On the other hand, PDE techniques use Partial Differential Equations (PDEs) to model a large variety of concepts in computer graphics and visual computing areas, such as visualization [38][40], and image processing [4][29], etc. In addition, PDE methods offer an alternative way to model both parametric and implicit geometric shapes [7][15][16]. In a nutshell, they define and govern geometric objects as solutions of a set of differential equations with boundary/initial conditions. In principle, PDE models can be controlled by physical laws, and the formulation of differential equations is well-conditioned and technically sound. Smooth objects that minimize certain energy functionals oftentimes are associated with differential equations, so optimization techniques can be unified with PDE models. Users can easily understand the underlying physical process associated with PDEs, therefore, it is possible to implement intuitive and natural control through the modification of physical parameters.

To take advantages of PDE techniques, [14] proposed a PDE-based technique to extract medial axes (or skeletons) for arbitrary 3D objects bounded by polygonal meshes. This method uses a diffusion-based equation with differential properties of the boundary surface to approximate a simplified medial axis of the object. The diffusion-based equation is solved numerically along the time axis, therefore users can obtain visual feedback during the medial axis extraction process. Users can define their own medial axis for an object by selecting desired boundary points of the object to be skeletal points on the medial axis. It provides users more degrees of freedom for shape skeletonization and further manipulation. To further improve the performance of the PDE-based medial axis extraction procedure, in this paper, we employ a distance field function based on Taylor polynomials to detect medial axis/surface point on the propagating surface. Fig. 2 demonstrates some examples of extracted medial axes from several objects.

The remaining of this paper is organized as follows. Section 2 reviews previous work of medial axis extraction and some related work of PDE techniques. Section 3 presents the formulation and numerical approximation of our PDE-based method for medial axis extraction with distance field information. Section 4 details the PDE-based medial axis extraction from objects with arbitrary polygonal boundary surfaces as well as shape sculpting and manipulation based on extracted skeletons. In Section 5, we discuss our method with possible improvements. Finally, Section 6 concludes the paper.

2 Related Work

2.1 Medial Axis Extraction

In the past several decades, medial axis extraction has been well studied and there are various techniques for detecting medial axes of 2D and 3D objects. Here we briefly review several typical approaches of computing medial axes or skeletons:

- **Thinning**

  To extract the medial axis of an object, one intuitive way is to peel off the object's boundary layer by layer. Such *thinning* process can be performed iteratively in the discrete domain. It will retain points on skeletons and maintain object's topology [2][23][26]. However, the thinning-based methods are fundamentally *discrete* processes and require fully segmented, compact, and connected objects. These techniques have difficulties to deal with partial data and are sensitive to Euclidean transformations of the data.

- **Distance functions**

  Because the skeletal or medial surface points usually coincides with the singularities of a *Euclidean distance function* to the boundary, distance functions can be employed for medial axis extraction. The approaches based on distance functions construct distance field transformation of an object and extract the medial axis based on the distance field [3][5][17][19][24]. However, usually it's difficult to ensure homotopy with original objects using techniques based on distance functions.

- **Voronoi skeletons**

  Because the vertices of the Voronoi diagram of a set of boundary points can converge to the skeleton as the sampling rate increases under appropriate smoothness conditions [32], Voronoi diagram and its dual Delaunay triangulation have been widely adopted for medial axis extraction [1][12][18][27][28] [33][35] [34]. These types of methods can preserve topology and accurately localize skeletal or medial surface points for densely sampled object. However, for algorithms based on Voronoi diagrams, it's more time consuming to build a 3D Voronoi diagram with increasing number of sample points, thus, direct computing method for Voronoi skeletons is less suitable for large datasets.

- **Level set method**

  Another class of methods casts the surface as the level set of a 4D embedded object and finds the weak solution of a PDE which models the wave propagation process whose singularities yield the medial axis. Kimmel *et al.* [21] introduced a level-set-based method for skeletonization using numerical approximation of distance maps of an object. Ma *et al.* [25] proposed a practical approach for extracting skeletons from general 3D models using radial basis functions (RBFs).

- **Direction testing**

  Bloomenthal and Lim [6] proposed an implicit method based on direction testing that defines the skeleton as the set of points at which the direction to the nearest point on the object undergoes a sudden transition. The geometric skeleton is derived from a static object using an implicit *direction* method. The object may be reconstructed from the modified skeleton using implicit distance and convolution techniques.

- **Hybrid techniques**

  In addition, many skeletonization techniques combine several aforementioned methods into a single framework for medial axis extraction. For instance, Siddiqi *et al.* [10] proposed a method combining the thinning process and the distance transformation and using a Hamilton-Jacobi equation to calculate the medial axis of volume data. This method provides accurate medial axis extractions and preserves homotopy of objects. However, it mainly focuses on volumetric datasets. Medial axis extraction for arbitrary polygonal meshes hasn't been considered. And sometimes the *real* medial axis for an irregular complex model may have noisy branches which are difficult to handle in the interest of shape manipulation.

2.2 Diffusion Equation and Applications

PDEs are at the heart of many computer analysis models or simulations of continuous physical systems, such as fluids, electromagnetic fields, the human body, and so on. Diffusion equation, wave equation, Laplacian equation, heat equation, as well as the equations of fluid dynamics, i.e., Navier-Stokes equations, are all popularly used PDEs [41] for modeling and simulation. Because most of the physics-based modeling techniques and many CAD/CAM applications are related to certain PDEs, PDE techniques are playing a more and more important role in computer graphics areas. In this paper, we mainly focus on diffusion equations.

A diffusion equation is defined as a PDE describing the variation in space and time of a physical quantity which is governed by diffusion. It provides a good mathematical model for the variation of temperature through conduction of heat and the propagation of electromagnetic waves in a highly conducting medium. The diffusion equation is a parabolic PDE whose characteristic form relates the first partial derivative of a field u with respect to time t to its second partial derivatives with respect to spatial coordinates $\mathbf{x}$:

$$\frac{\partial u}{\partial t} = D\nabla^2 u, \tag{1}$$

where $u = u(\mathbf{x},t), \mathbf{x} = (x_1, x_2, \cdots, x_n) \in \Omega \subset \mathbf{R}^n, t \geq 0$, and D is called the diffusion coefficient. The operator $\nabla^2 = \sum_i \frac{\partial^2}{\partial x_i^2}$ is called the *Laplacian*. When D is not constant, but depends on spatial coordinates and time: $D = D(\mathbf{x},t)$, this spatial variation leads to *anisotropic diffusion equation*:

$$\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u). \tag{2}$$

The solution of a diffusion equation is subject to both initial and boundary conditions. The numerical solution of diffusion equations usually makes use of the finite-difference method, which employs Forward Time Centered Space (FTCS) finite-difference approximation to the diffusion equation. Because $\mathbf{x}$ in above equations can be of arbitrary variables in any dimensions, diffusion equations can be applied for various applications in computer graphics fields. Using such equations, researchers have developed visually convincing models of fire, smoke, and other gaseous phenomena. The diffusion equations can also be used in scientific visualization of medical images. The applications of diffusion equations include depicting gaseous phenomena [36], surface fairing [11], texture synthesis using reaction-diffusion system [40][38], visualizing vector field [30][13], etc.

In this paper, we employ a diffusion-based equation to approximate skeletons of objects bounded by arbitrary meshes (or other boundary representations) and reconstruct the original shape.

3 PDE Formulation of Medial Axis Extraction for Arbitrary Meshes

To directly detect and extract skeletons of 3D solid objects bounded by arbitrary meshes, we employ a diffusion-based PDE combined with distance field information to allow any given 3D objects to propagate inward their boundaries and approximate simplified skeletons with user interactions, which can provide users instant feedback and interactive control during the extraction process. The distance information from skeletal points to the boundaries are recorded for reconstruction and deformation purposes. When manipulating the skeleton, the original model can be deformed accordingly. Other immediate applications include model simplification, skeleton-driven parameterization, and animation control of complex, articulated characters.

3.1 Diffusion-based Equation

The grassfire flow on a 3D surface $\mathbf{S}$ is governed by

$$\frac{\partial \mathbf{S}}{\partial t} = \mathbf{N}, \tag{3}$$

which allows the fire front propagating at unit speed along the inward surface normal $\mathbf{N}$.

The simplest way to simulate (3) for medial axis extraction of a polygonal mesh is to let the sample points on the boundary surface travel along the surface normal inward (i.e., shrinking the boundary) at each time step, and where the points meet with each other forms the skeleton. However, the time step for this simulation process needs to be very small to guarantee a close approximation of medial axis. Therefore, it's difficult to achieve satisfactory results using direct simulation of (3). Furthermore, the complexity of the medial axis structure of an object depends on its geometric shape. For a complex object with various detailed features, its real medial axis will be very complicated with noisy branches. Such structures are not suitable for shape manipulation operations. In addition, because our goal is to extract medial axes of objects bounded by discrete arbitrary polygonal meshes, we can only approximate the surface normal at discrete sample points on the boundary surfaces where the regular parametrization is not applicable, and the mesh qualities will directly affect the results for direct simulation of (3).

On the other hand, diffusion equations are frequently used for denoising in image processing. They can also provide smooth results for geometric surface fairing [11]. Because of their smoothing properties, we apply the diffusion process for medial axis extraction from polygonal meshes, which will provide simplified approximations and remove noisy branches on medial axes for easy storage and manipulation. Since our main purpose of medial axis extraction is to offer users a compact geometric representation for shape manipulation and deformation, such approximation can provide satisfactory results.

We formulate the diffusion-based equation for the simplified medial axis approximation as:

$$\frac{\partial \mathbf{S}}{\partial t} = \mathbf{D}(\mathbf{N}, \kappa) \nabla^2 \mathbf{S}, \tag{4}$$

where $\mathbf{S} = \mathbf{S}(\mathbf{p},t)$ is the propagating boundary surface of an object, $\mathbf{p} = (x,y,z)$ is the coordinate vector, $t \geq 0$ is the time variable, $\nabla^2 \mathbf{S}$ is the Laplacian of the surface, and $\mathbf{D}$ is the diffusion coefficient function related to the surface normal $\mathbf{N}$ and curvature κ. The normal $\mathbf{N}$ provides directions for boundary propagation during the medial axis extraction process. The curvature κ is used as a threshold to detect skeletal points on the medial axis. We consider the curvature as the threshold for skeletal point detection is because the Laplacian will smooth the boundary surface and eliminate sharp features during the propagation. By

using curvature of the boundary surface as a threshold, the propagation process can detect sharp features of an object and preserve such properties on its simplified medial axis.

(4) is formulated to guide the boundary surface propagation. It provides the direction of the propagating boundary surface while smoothing out unnecessary noises at each time step. By solving (4), the object's surface will moving inward from the original boundary guided by its normal, and the Laplacian will smooth the surface to avoid noisy branches during the propagating process. The curvature acts as a threshold to preserve feature points of the object on the approximated medial axis. Therefore, after all the points on the propagating surfaces collide with others which means they reside on a thin set, we can obtain a compact structure without interior points inside. We consider it as an approximation for the *real* medial axis because it's a thin set inside the object and preserve features of the original dataset. Since our major goal is to use a compact and simple representation for shape manipulation, such an approximation is enough to provide satisfactory results for this purpose. Alternatively, because the skeletal or medial surface points usually coincides with the singularities of a *Euclidean distance function* to the boundary, we also use distance field information to determine if the point on the propagating surfaces reside on the medial axis. We employ a distance function based on Taylor polynomials to calculate the value of the points in the distance field, which could indicate if the points belong to the medial axis.

Note that, the shape reconstruction from skeletons is a reverse process of medial axis extraction by applying the normal outward to original boundaries. The diffusion equation is suitable for continuous geometric objects including surfaces and solids. Although in this paper, we mainly use numerical techniques to solve it on discrete polygonal boundary surfaces, it can be readily applied to other type of solid representations for medial axis extraction.

3.2 Numerical Simulations

Diffusion equations can be easily solved through numerical techniques. One of the most popular numerical methods to solve a diffusion equation is the finite-difference method. It discretizes the equation by applying finite-difference approximations of partial derivatives in the equation. The finite-difference technique is straightforward for regular parametric objects. However, regular parametrization for an arbitrary mesh surface is a challenging problem, because an arbitrary mesh usually has arbitrary connectivities among surface points, therefore it is difficult to discretize the surface into regular, uniform grids. As a result, finite-difference approximations of partial derivatives [15] cannot be applied in such situation. We have to seek for alternative techniques to approximate the partial derivatives to obtain a discretized equation.

3.2.1 Umbrella Operator

There is a type of difference operators called *umbrella operator* which is commonly used in surface fairing to approximate the Laplacian operator for 2D meshes [37][22]. A simple umbrella operator assumes the mesh has underlying regular parametrization where every edge length equals with each other and every angle between neighbor vertices in the parametrization domain is the same. Then the parametrization of (u_i, v_i) can be represented by

$$(u_i, v_i) = (cos\frac{2\pi i}{n}, sin\frac{2\pi i}{n}),$$

where n is the number of direct neighbors (points in the 1-neighborhood) of the point at (u_i, v_i).

The Laplacian operator can be approximated by the discretized umbrella operator:

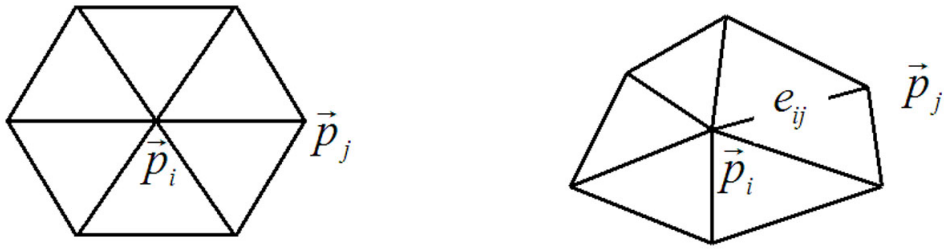
$$\nabla^2 \mathbf{p}_i = \frac{1}{n} \sum_{j \in N_1(i)} \mathbf{p}_j - \mathbf{p}_i,$$

where $\mathbf{p}_i$ is a surface point, $\mathbf{p}_j$ is a point in the 1-neighborhood ($N_1(i)$) of $\mathbf{p}_i$.

However, the assumption of regular parametrization is only suitable for ideal situations. In most occasions for arbitrary meshes, such type of parametrization cannot give a satisfactory result. The umbrella operator can be further improved by adding weights based on the connectivity of the mesh which allows vertices drifting in the parametric space and leads to non-uniform mesh parametrization. One way is to allow edge lengths between points not to be constant. The discretized Laplacian operator can be approximated by

$$\nabla^2 \mathbf{p}_i = \frac{2}{E} \sum_{j \in N_1(i)} \frac{\mathbf{p}_j - \mathbf{p}_i}{e_{i,j}},$$

where $E = \sum_{j \in N_1(i)} e_{i,j}$ and $e_{i,j}$ is the edge length between $\mathbf{p}_i$ and $\mathbf{p}_j$. Fig. 3 shows an illustration of umbrella operators. Note that, the angles between edges in the 1-neighborhood of a point on the mesh can also be considered as weights to improve the umbrella operator.



(a) (b)

Figure 3: Umbrella operators. (a) Regular umbrella operator. (b) Improved umbrella operator with edge lengths as weights.

To simplify the process and provide a fast algorithm for medial axis extraction, we employ the finite-difference discretization associated with umbrella operators for iterative computations of the evolving surface and its Laplacian operator. The diffusion-based equation (4) can be discretized as follows:

$$\frac{\mathbf{p}_i^{n+1} - \mathbf{p}_i^n}{\Delta t} = \mathbf{D}(\mathbf{N}_i, \kappa_i)(\frac{1}{n} \sum_{j \in N_1(i)} \mathbf{p}_j^n - \mathbf{p}_i^n). \tag{5}$$

3.2.2 Surface Normal Approximation

To calculate surface normal at sample points of an arbitrary mesh object, we also resort to numerical approximation techniques. The simplest way is first calculating the normals of surface patches around the target point, then averaging the surface patch normals to approximate the normal at the point. This only provides a rough approximation of surface normal at the sample point. There is another way to approximate the normal at a point proposed in [42] to provide more satisfying normal approximations. The normal at a surface point $\mathbf{p}_i$ can be computed using approximated tangent vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ along the surface at $\mathbf{p}_i$ which can be computed as:

$$\mathbf{t}_1 = \sum_{j=0}^{n-1} cos\frac{2\pi j}{n}\mathbf{p}_j, \mathbf{t}_2 = \sum_{j=0}^{n-1} sin\frac{2\pi j}{n}\mathbf{p}_j,$$

where n is the valence of the point on the mesh, and $\mathbf{p}_j$'s are in the 1-neighborhood of $\mathbf{p}_i$.

Therefore, the sampled surface normal $\mathbf{N}_i$ at $\mathbf{p}_i$ can be computed as

$$\mathbf{N}_i = \mathbf{t}_1 \times \mathbf{t}_2. \tag{6}$$

3.2.3 Gaussian Curvature of Arbitrary Meshes

Since the diffusion process is also influenced by the surface curvature, we need to evaluate curvature values at the boundary surface. In this paper, we consider the contribution of Gaussian curvature for medial axis extraction. The curvature is used as a threshold to define skeletal points on the medial axis to preserve shape features, therefore other types of curvature instead of Gaussian curvature can also be employed for this purpose. We use Gaussian curvature because it's very easy to calculate using the approximation scheme for polygonal meshes. We use a local approximation scheme to compute Gaussian curvature of sample points on the boundary surface based on Gauss-Bonnet theorem. The Gaussian curvature at a surface point is related to angles and faces connected to the point on the surface [39]. The Gaussian curvature κ can be approximated as:

$$\kappa = \frac{a}{A},$$

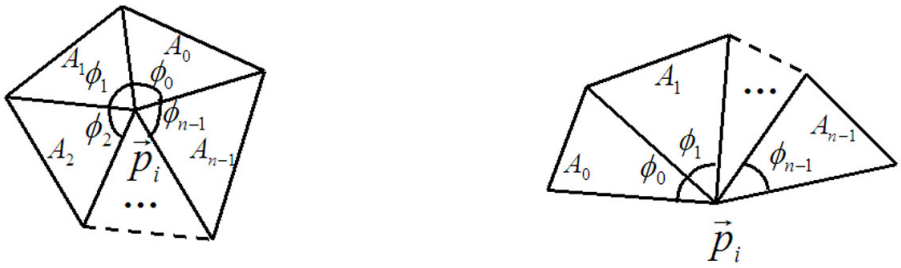
where a is the angular defect at the point which is defined as (2π - sum of the interior angles of faces meeting at the point) and A is the area associated to the point that is equal to $\frac{1}{3}$ of the sum of areas of triangles meeting at the point. Therefore, the Gaussian curvature κ_i at point $\mathbf{p}_i$ can be computed as follows:

$$\kappa_i = \frac{2\pi - \sum_{j=0}^{n-1} \phi_j}{\frac{1}{3}\sum_{j=0}^{n-1} A_j}, \tag{7}$$

where ϕ_j is the angle of the jth face connected to $\mathbf{p}_i$ and A_j is the corresponding triangle's area. (7) is for inner points on a mesh and suitable for any points in a closed surface. As for open surfaces, the approximation for Gaussian curvature of a boundary point can be evaluated using the following scheme:

$$\kappa_i = \frac{\pi - \sum_{j=0}^{n-1} \phi_j}{\frac{1}{3}\sum_{j=0}^{n-1} A_j}, \tag{8}$$

An illustration of Gaussian curvature approximation for a mesh point is shown in Fig. 4.



(a) (b)

Figure 4: The evaluation of Gaussian curvature for a mesh point. (a) Gaussian curvature for an internal vertex. (b) Gaussian curvature for a boundary vertex.

3.2.4 Distance Function Based On Taylor Polynomials

According to the grassfire simulation, the surface points on a polygonal mesh travel along the surface normal inward (i.e., shrinking the boundary) and where the points meet with each other forms the medial axis. Therefore, the points on the medial axis have equal distance to at least two surface points on the original mesh. During the numerical simulation of (4), we can use distance field information of surface points on the propagating surface to determine if the points reach the medial axis.

A distance field based on a given 3D object can be calculated by an implicit interpolation function

$$f(\mathbf{p}) = \sum_{i=1}^{n} w_i(\mathbf{p})\phi(\mathbf{p} - \mathbf{c}_i),$$

where $f(\mathbf{c}_i) = d_i$ for any point $\mathbf{c}_i$ in the dataset. w_i are the weight functions and ϕ is the interpolation basis function.

Among various implicit interpolation functions, we use Taylor polynomials as interpolation basis functions that make use the differential properties of the 3D object to provide higher-order continuity for the resulting distance field. The Taylor polynomials are often used to approximate values in its neighborhood based on the value and differential properties at a given point. For example, the Taylor approximation using the first order derivative is:

$$f_{1st}(\mathbf{c} + \Delta\mathbf{p}) \approx f(\mathbf{c}) + \Delta\mathbf{p} \cdot \nabla f(\mathbf{c}), \tag{9}$$

where $\nabla f = (\partial f/\partial x, \partial f/\partial y, \partial f/\partial z)^T$ and in most cases, we assume it's equivalent to the normal vector $\mathbf{N}_i$ at a given point $\mathbf{c}_i$. And the Taylor approximation using the first and second order derivatives is:

$$f_{2nd}(\mathbf{c} + \Delta\mathbf{p}) \approx f(\mathbf{c}) + \Delta\mathbf{p} \cdot \nabla f(\mathbf{c}) + \frac{1}{2}\Delta\mathbf{p} \cdot \mathbf{H}(\mathbf{c}) \cdot \Delta\mathbf{p}, \tag{10}$$

where $\mathbf{H}(\mathbf{c}) = \begin{pmatrix} \partial^2 f/\partial x^2 & \partial^2 f/\partial x \partial y & \partial^2 f/\partial x \partial z \\ \partial^2 f/\partial x \partial y & \partial^2 f/\partial y^2 & \partial^2 f/\partial y \partial z \\ \partial^2 f/\partial x \partial z & \partial^2 f/\partial y \partial z & \partial^2 f/\partial z^2 \end{pmatrix}$.

The weight functions w_i for the interpolation are defined as:

$$w_i = \frac{1}{|\mathbf{p} - \mathbf{c}_i|} \Big/ \sum_{j=1}^{n} \frac{1}{|\mathbf{p} - \mathbf{c}_j|}. \tag{11}$$

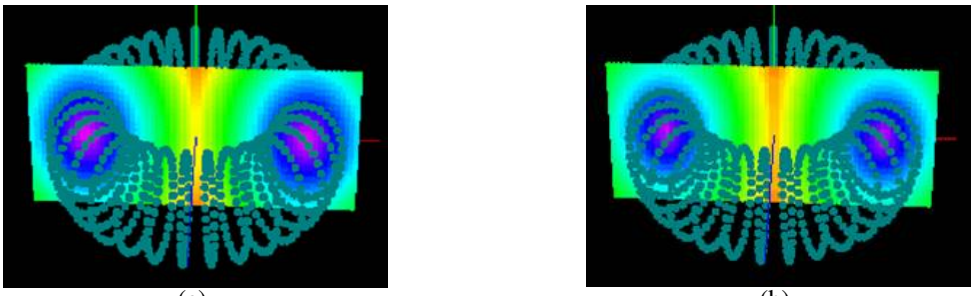
The weight functions have the property $\sum_{i=1}^{n} w_i = 1$.

Therefore, the interpolation function using the first order Taylor polynomial $f(\mathbf{p}) = \sum_{i=1}^{n} w_i f_{1st}(\mathbf{c}_i + (\mathbf{p} - \mathbf{c}_i)$ can be rewritten as:

$$f(\mathbf{p}) = \sum_{i=1}^{n} \frac{d_i + (\mathbf{p} - \mathbf{c}_i) \cdot \mathbf{N}_i}{|\mathbf{p} - \mathbf{c}_i|} / \sum_{j=1}^{n} \frac{1}{|\mathbf{p} - \mathbf{c}_i|}. \tag{12}$$

The function using the second order Taylor polynomial can be formulated similarly.

Fig. 5 shows examples of calculating distance fields of a mesh torus based on the first and second order Taylor polynomials.



(a) (b)

Figure 5: Distance fields calculated based on Taylor polynomials. (a) The distance field of a torus calculated based on the first order Taylor polynomial. (b) The distance field of a torus calculated based on the second order Taylor polynomial.

Using Taylor polynomials for calculate the distance field of a polygonal mesh has several advantages. First, the distance value can be calculated directly based on data points on the 3D object and their derivative information. Second, weight functions are naturally defined according to data points on the 3D object so that no equations need to be solved. Third, the distance of a given point can be approximated by a simplified interpolation function, which only use the closest surface point for the calculation. With such simplification, we can quickly detect the points on the medial axis.

With numerical discretizations and approximations for Laplacian operator, surface normal, and curvature, (5) can be easily solved by iterative method along time axis. The diffusion equation will evolve along time axis according to the surface curvature and normal. The medial axis resides on the locations where different parts of the propagating surface meet, which can be determined by a collision detection algorithm or the distance values calculated by the distance function based on Taylor polynomials. Because of the discrete property of the numerical technique, users can freeze certain points on the mesh to let them stay at their current positions during the process to obtain different skeletons. Furthermore, we also allow users to select a region to extract the medial axis inside the region to obtain any localized results.

For shape reconstruction, because we save the distance information between the skeleton and the original boundary surface, the object can be recovered along the normal outward without any difficulty. In addition, after the skeleton manipulation, the corresponding deformed shape can be reconstructed through diffusion propagation to follow changes of the skeleton.

4 PDE-based Skeletonization and Shape Manipulation

4.1 Diffusion-based Medial Axis Extraction: An Algorithmic Outline

We use the finite-difference technique to approximate the solution for the time-dependent diffusion-based equation numerically to provide users progressive results for medial axis approximation and shape reconstruction. Our techniques can be applied for solid objects with polygonal boundary surfaces and is also suitable for other boundary representations.

Starting with the original mesh, our system extracts the medial axis according to the differential properties of the boundary mesh, and allow the mesh to shrink to its medial axis. Our algorithm consists of following operations:

- Initialization: at the initialization stage, the system approximates the surface normal for the boundary surface using (6) and other differential properties such as curvature by (7) and Laplacian using umbrella operators.

- Skeletonization: during the skeletonization process, at each time step, the system first computes the evolving surface based on (5). Then a determination step is performed on the resulting surface to decide if a surface point is on the medial axis. Two algorithms are employed for the determination step. (a) Collision detection. If a surface point collides with any other point, edge, or face, it is considered as residing on the skeleton. In such case, it is marked as a skeletal point with its position fixed and the distance information from original surface point to this skeletal point is recorded. After

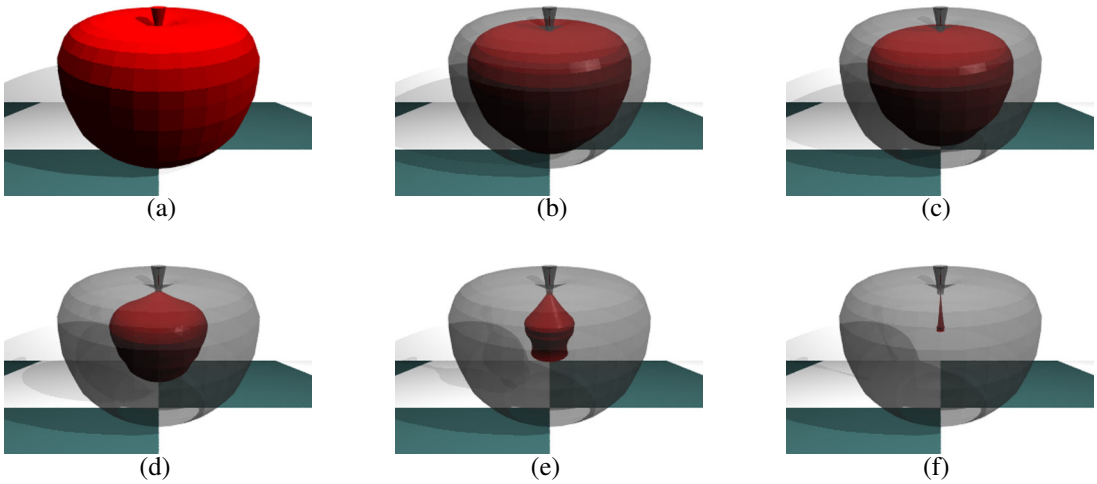


Figure 6: An example of PDE-based medial axis extraction for arbitrary meshes. (a) Original dataset; (b), (c), (d) and (e) are shrinking objects during medial axis extraction at different time step by performing (5); (f) is the final skeleton.

all the points are checked for collision detection, surface optimization is applied to delete redundant points and faces with too small areas. (b) Distance Function. Alternatively, users can choose to use the distance function based on Taylor polynomials to approximate the distance value of a propagating surface point. According to the definition of medial axis, a medial axis point will have equal distance to at least two surface points on the 3D object. During the propagation process, a point on the propagating surface will reach the medial axis if its distance value can be calculated based on a point other than its corresponding point on the original surface. By calculating the distance values of propagating surface points at each time step, we can determine which points reside on the medial axis. This process is repeated until all points on the propagating surface are marked as skeletal points.

- User Interaction: in addition, during the process, users can interactively select any points on the propagating surface to be skeletal points, thus they can define the user-controlled skeleton based on their own criteria. Users are also allowed to define local regions for local medial axis extraction.

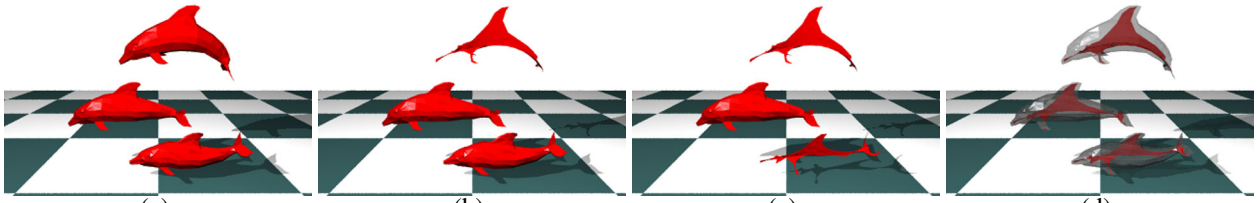
The medial axis extraction using our technique is a progressive process along time, which offers users visual feedback during the extraction. Fig. 6 shows an example of progressively extracting medial axis for an object.

After this skeletonization process, we can obtain a simplified skeleton approximating the medial axis of the object associated with distance information between the skeleton and the original boundary surface. With such information, we can manipulate the object by sculpting its skeleton with ease.

4.2 Local Region Skeletonization and User Interaction

To explore local features, our system allows users to extract medial axis from a selected part of an object. This can be done by selecting a region in the 3D working space and the system will only extract skeleton for part of the object residing in the region. By allowing medial axis to be extracted locally, it will reduce the time complexity for shape skeletonization of complex models and enable the mechanism for the direct user control. Refer to Fig. 7 for an example.

Because we provide a simplified approximation of medial axis for an object, the result may not satisfy users expectation sometimes. For example, there are certain points on the object that users want to be on the medial axis, but the system doesn't mark them as skeletal points during medial axis extraction process. Therefore, we allow users to select desired points on the boundary surface to be skeletal points for any user defined skeleton during the extraction process, which can provide more degrees of freedom for later skeleton-based shape manipulation (Fig. 8). Since manipulations on different shape of skeletons can result in different shape deformations, these user interactions provide more flexibility/freedom and control for skeleton-based shape sculpting. Furthermore, because the diffusion-based equation is solved on polygonal meshes, the number of points on the meshes will extremely affect the performance of the medial axis extraction process. It's time consuming to extract medial axes for complex models. Therefore, local medial axis extraction in selected regions will be useful for such cases. It's also possible to integrate parallel techniques with our method for shape skeletonization of large datasets.



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

Figure 7: An example of PDE-based medial axis extraction for selected parts from arbitrary boundary meshes. (a) Original dataset; (b) and (c) are two examples of extracting skeletons for part of the objects; and (d) is the skeleton for the entire dataset.

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

Figure 8: Examples of PDE-based medial axis extraction with user-defined skeletal points. (a), (b), and (c) are different skeletons obtained after fixing different surface points as skeletal points. (d) is a different skeleton for Fig. 6 (a) by fixing a point at the bottom of the dataset.

4.3 Skeleton-based Shape Sculpting

One of the advantages of medial axes is that they provide much more compact and natural representations for objects. Therefore, shape deformation/manipulation and other processes based on medial axes alleviate the burden of tedious and less insightful operations for deforming and animating complex objects, as well as other shape queries and interrogations. In this paper, we provide users various sculpting tools to manipulate medial axes, then propagate the deformation to original datasets according to the distance information. However, the deformed result may not be satisfactory if we just simply reconstruct the objects from their medial axes according to the distances from medial axes to original datasets. Therefore, we employ the diffusion-based equation with normal pointing outward to the original boundaries to reconstruct the modified datasets. Fig. 9 and Fig. 10 have two examples of shape manipulation based on skeletons and recovered using diffusion propagation. Fig. 11 shows a deformation sequence of an object through skeleton manipulations. It may be noted that, from this point of view, our technique also serves as an aid for shape parameterization and can be potentially improved for a powerful shape analysis tool (beyond shape sculpting and synthesis).

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

Figure 9: An example of skeleton-based shape sculpting. (a) Original dataset; (b) is the skeleton; (c) is the sculpted skeleton; (d) is the corresponding deformed dataset recovered from (c).

4.4 Curvature Manipulation

In this paper, we employ Gaussian curvature of the polygonal boundary surface in the diffusion equation for shape skeletonization. It works as the threshold for medial axis extraction to decide which surface points will be skeletal points on the medial

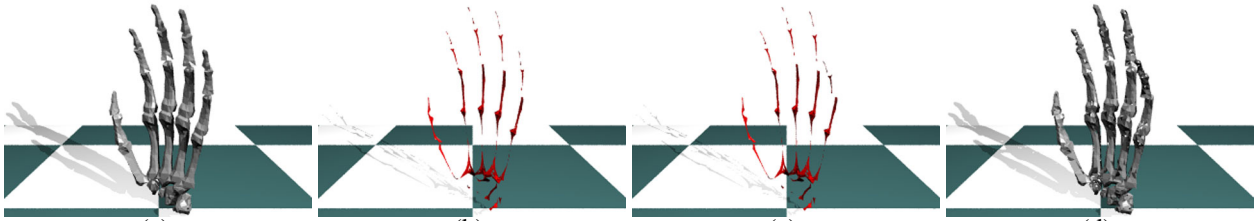


Figure 10: Another example of skeleton-based shape sculpting.

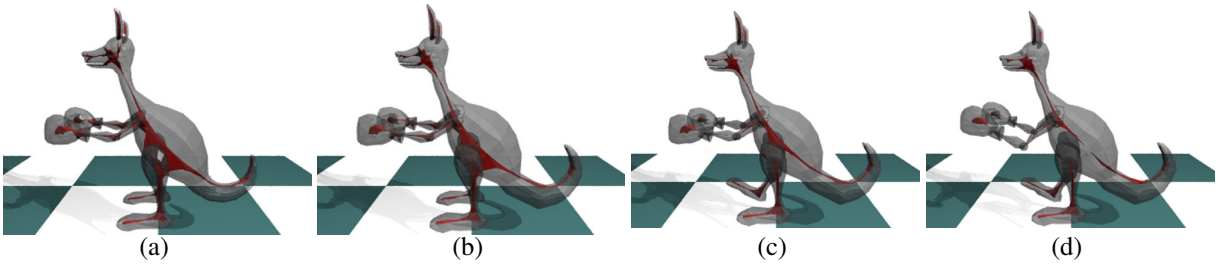


Figure 11: A sequence of deformed shapes through skeleton-based shape sculpting.

axis. Thus, different values of the threshold for Gaussian curvature on the boundary polygonal mesh will result in different shapes of skeletons. By allowing users to define the threshold themselves, they can obtain the medial axis for an object according to their own criteria. Fig. 12 shows examples for several medial axes extracted from an object with different Gaussian curvature thresholds.

5 Discussion and Future Work

In this paper, we propose a diffusion-based medial axis extraction method which combines the grassfire flow simulation and diffusion propagation to approximate skeletons for solid objects whose boundary surfaces are polygonal meshes or other types of B-reps. The diffusion-based formulation naturally unifies the thinning process along surface normals with surface smoothing for propagating boundaries. The system is implemented using Visual C++ and runs on Windows systems. The examples shown in this paper are provided by 3D CAFE and rendered using POV-RAY.

Our method offers smooth approximations of medial axes in a visually progressive way. For complex objects bounded by polygonal meshes, the *real* medial axes may have numerous noisy branches to preserve objects' features. Such structures are difficult to manipulate for shape sculpting. In contrast, our technique provides simplified approximations for medial axes, which are smooth thin sets residing inside objects without noisy branches. The approximated results are smoothed because of the Laplacian operator, which eliminates noisy branches of the real medial axis, so that the resulting medial axis is relatively simple and easy to manipulate.

Because our medial axis extraction algorithm is applied directly to objects bounded by arbitrary polygonal meshes, the resolution of meshes and the point distribution on meshes will affect the quality of extracted skeletons. For instance, when the two end points of a long edge on the propagating surface stop on the skeleton, all the points on the edge will be assumed to

Figure 12: Examples of skeleton extraction with different value of curvature thresholds.

be skeletal points, although there may be still spaces between them and real skeletal points. Therefore, a mesh optimization process may be considered to extract more accurate results.

In addition, the approximating techniques to calculate the differential properties of the boundary surface sometimes are not accurate enough for extremely irregular meshes. On the other hand, there are techniques available to provide regular parametrization for irregular polygonal meshes. The differential calculation will be much easier under such parametrization. Thus, mesh parametrization techniques might be applied to our model for better results.

Other future work may include using other distance functions in the future and comparing the results with different distance measures. We will also consider other skeleton-based shape sculpting techniques to further enhance our system.

6 Conclusion

In this paper we present a PDE-based technique using diffusion-based propagation for medial axis extraction of geometric objects bounded by arbitrary polygonal meshes. By numerically solving the time-dependent diffusion-based equation using finite-difference approximations, we simulate the skeletonization process progressively along the time axis to offer users instant feedback of the medial axis extraction. The diffusion-based equation is formulated to unify the grassfire flow simulation and diffusion propagation based on differential properties of the boundary surfaces such as curvature. The evolving surface is propagating from the boundary surface inwards according to the PDE combined with distance field information based on Taylor polynomials and approximates a simplified and smoothed medial axis of the object associated with distant information between the skeleton and the original model. With such information, the original model can be easily reconstructed. In addition, shape sculpting based on skeleton manipulations can be conducted without any difficulty. Because the diffusion equation can essentially smooth out noises of an irregular dataset, our medial axis extraction process is much less noise-sensitive and able to provide smoother skeletons for irregular datasets. Our method offers user control of curvature threshold and selecting desired skeletal points on the propagating surface for the skeleton extraction process that can allow users to define different medial axis extraction criteria and thus obtain satisfactory skeleton representations. Furthermore, our system also can extract localized skeletons for selected parts of the objects, which is useful for medial axis extraction of complex models. To illustrate properties of the extracted medial axis, we also provide interactive manipulation toolkits to deform the medial axis, and use diffusion propagation to recover the corresponding deformed shape. Our PDE-based approach unifies several modeling tasks such as shape smoothing (denoising), simplification, editing, and deformation together within a single framework. We hope that it can be further improved to become a more powerful and convenient tool for shape modeling, synthesis, and analysis of complex real-world objects.

Acknowledgement

This research was supported in part by the NSF ITR grant IIS-0082035, the NSF grant IIS-0097646, Alfred P. Sloan Fellowship, Honda Initiation Award, and an appointment of Haixia Du to the NLM Research Participation Program sponsored by the National Library of Medicine and administered by the Oak Ridge Institute for Science and Education.

References

[1] N. Amenta, S. Choi, and R. K. Kolluri. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266. ACM Press, 2001.

[2] C. Arcelli and G. S. di Baja. A width-independent fast thinning algorithm. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 7(4):463–474, 1985.

[3] C. Arcelli and G. S. di Baja. Ridge points in euclidean distance maps. *Pattern Recognition Letters*, 13(4):237–243, 1992.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH 2000*, pages 417–424, New Orleans, LA USA, 2000.

[5] I. Bitter, A. Kaufman, and M. Sato. Penalized-distance volumetric skeleton algorithm. *IEEE Transaction on Visualization and Computer Graphics*, 7(3):195–206, 2001.

[6] J. Bloomenthal and C. Lim. Skeletal methods of shape manipulation. In *Shape Modeling International 1999, Aizu-Wakamatsu, Japan*, pages 44–47, 1999.

[7] M. I. G. Bloor and M. J. Wilson. Using partial differential equations to generate free-form surfaces. *Computer Aided Design*, 22(4):202–212, 1990.

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

[9] H. Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38:205–287, 1973.

[10] S. Bouix and K. Siddiqi. Divergence-based medial surfaces. In *European Conference on Computer Vision (ECCV) 2000*, Dublin, Ireland, 2000.

[11] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH 99*, pages 317–323, Los Angeles, CA USA, 1999.

[12] T. K. Dey and W. Zhao. Approximate medial axis as a voronoi subcomplex. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 356–366. ACM Press, 2002.

[13] U. Diewald, T. Preußer, and M. Rumpf. Anisotropic diffusion in vector field visualization on euclidean domains and surfaces. *IEEE Transaction on Visualization and Computer Graphics*, 6(2):139–149, 2000.

[14] H. Du and H. Qin. Medial axis extraction and shape manipulation of solid objects using parabolic PDEs. In *Proceedings of the ninth ACM Symposium on Solid Modeling and Applications*, 2004.

[15] H. Du and H. Qin. Dynamic PDE surface design using geometric and physical constraints. *Graphical Models*, 67(1):43–71, 2005.

[16] H. Du and H. Qin. Free-form geometric modeling by integrating parametric and implicit PDEs. *IEEE Transaction on Visualization and Computer Graphics*, 13(3):549–561, 2007.

[17] M. Foskey, M. C. Lin, and D. Manocha. Efficient computation of a simplified medial axis. In *Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 96–107. ACM Press, 2003.

[18] J. Goldak, X. Yu, A. Knight, and L. Dong. Constructing discrete medial axis of 3-D objects. *Int. J. Computational Geometry and Its Applications*, 1(3):327–339, 1991.

[19] J. Gomez and O. Faugeras. Reconciling distance functions and level sets. *Technical Report TR3666, INRIA*, 1999.

[20] T. Ju, Q. Zhou, and S. Hu. Editing the topology of 3d models by sketching. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 42, New York, NY, USA, 2007. ACM.

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

[22] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH 1998*, pages 105–114, USA, 1998.

[23] T. Lee and R. Kashyap. Building skeleton models via 3d medial surface/axis thinning algorithm. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994.

[24] F. Leymarie and M. Levine. Simulating the grassfire transform using an active contour model. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(1):56–75, 1992.

[25] W.-C. Ma, F.-C. Wu, and M. Ouhyoung. Skeleton extraction of 3d objects with radial basis functions. In *Shape Modeling International 2003, Seoul, Korea*, pages 207–215, 2003.

[26] A. Manzanera, T. Bernard, F. Preteux, and B. Longuet. Medial faces from a concise 3d thinning algorithm. In *ICCV'99*, pages 337–343, Kerkyra, Greece, 1999.

[27] M. Näf, O. Kübler, R. Kikinis, M. Shenton, and G. Szekely. Characterization and recognition of 3d organ shape in medical image analysis using skeletonization. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, 1996.

[28] R. Ogniewicz. *Discrete Voronoi Skeletons*. Hartung-Gorre, 1993.

[29] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (TOG)*, 22(3):313–318, 2003.

[30] T. Preußer and M. Rumpf. Anisotropic nonlinear diffusion in flow visualization. In *Visualization 99*, pages 325–332, 1999.

[31] T. J. Q. Zhou and S. Hu. Topology repair of solid models using skeletons. *IEEE Transaction on Visualization and Computer Graphics*, 13(4):675–685, 2007.

[32] M. Schmitt. Some examples of algorithms analysis in computational geometry by means of mathematical morphology techniques. *Lecture Notes in Computer Science, Geometry and Robotics, Springer-Verlag*, 391:225–246, 1989.

[33] D. Sheehy, C. Armstrong, and D. Robinson. Shape description by medial surface construction. *IEEE Transaction on Visualization and Computer Graphics*, 2(1):62–72, 1996.

[34] E. Sherbrooke, N. Patrikalakis, and E. Brisson. An algorithm for the medial axis transform of 3d polyhedral solids. *IEEE Transaction on Visualization and Computer Graphics*, 2(1):44–61, 1996.

[35] E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. Computation of the medial axis transform of 3-d polyhedra. In *Proceedings of the third ACM symposium on Solid modeling and applications*, pages 187–200. ACM Press, 1995.

[36] J. Stam and E. Fiume. Depicting fire and other gaseous phenomena using diffusion professes. In *SIGGRAPH 95*, pages 129–136, Los Angeles, CA USA, 1995.

[37] G. Taubin. A signal processing approach to fair surface design. In *SIGGRAPH 1995*, pages 351–358, Los Angeles, CA USA, 1995.

[38] G. Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *Computer Graphics*, 25(4):289–298, 1991.

[39] P. Veron and J. Leon. Static polyhedron simplification using error measurements. *Computer-Aided Design*, 29(4):287–298, 1997.

[40] A. Witkin and M. Kass. Reaction-diffusion textures. In *SIGGRAPH 91*, pages 299–308, Las Vegas, NV USA, 1991.

[41] E. Zauderer. *Partial Differential Equations of Applied Mathematics, Second Edition*. A Wiley-Interscience Publication, 1988.

[42] D. Zorin and P. Shröder. *Subdivision for Modeling and Animation*. SIGGRAPH 2000 Course Notes 36, 2000.