Hamilton-Jacobi Skeletons

Kaleem Siddiqi[†] Sylvain Bouix[†] Allen Tannenbaum[§]
Steven W. Zucker[‡]

[†]McGill University
School of Computer Science & Center for Intelligent Machines
Montreal, QC H3A 2A7, Canada
email: {siddiqi,sbouix}@cim.mcgill.ca

[§]Georgia Institute of Technology
Department of Electrical & Computer Engineering
Department of Biomedical Engineering
Atlanta, GA 30082-0250, USA
email: tannenba@ece.gatech.edu

[‡]Yale University
Departments of Computer Science and Electrical Engineering
Center for Computational Vision & Control
New Haven, CT 06520-8285, USA
email: zucker-steven@cs.yale.edu

Technical Report No. SOCS-01.10, School of Computer Science,
McGill University, September 2001.

Abstract The eikonal equation and variants of it are of significant interest for problems in computer vision and image processing. It is the basis for continuous versions of mathematical morphology, stereo, shape-from-shading and for recent dynamic theories of shape. Its numerical simulation can be delicate, owing to the formation of singularities in the evolving front and is typically based on level set methods. However, there are more classical approaches rooted in Hamiltonian physics which have yet to be widely used by the computer vision community. In this paper we review the Hamiltonian formulation, which offers specific advantages when it comes to the detection of singularities or shocks. We specialize to the case of Blum's grassfire flow and measure the average outward flux of the vector field that underlies the Hamiltonian system. This measure has very different limiting behaviors depending upon whether the region over which it is computed shrinks to a singular point or a non-singular one. Hence, it is an effective way to distinguish between these two cases. We combine the flux measurement with a homotopy preserving thinning process applied in a discrete lattice. This leads to a robust and accurate algorithm for computing skeletons in 2D as well as 3D, which has low computational complexity. We illustrate the approach with several computational examples.

1 Introduction

Variational principles have emerged naturally from considerations of energy minimization in mechanics [27]. We consider these in the context of the eikonal equation, which arises in geometrical optics and has become of great interest for problems in computer vision [10, 22]. It is the basis for continuous versions of mathematical morphology [9, 45, 64], as well as for Blum's grassfire transform [5] and dynamic theories of shape representation [23, 61]. It has also been used for applications in image processing and analysis [48, 11], shape-from-shading [20, 44, 39, 25] and stereo [15].

As is well known, some care must be taken with the numerical simulation of this equation, since it is a hyperbolic partial differential equation for which a smooth initial front may develop singularities or *shocks* as it propagates. At such points, classical concepts such as the normal to a curve and its curvature are not defined. Nevertheless, it is precisely these points that are important for the above applications in computer vision since, e.g., it is they which denote the skeleton (see Figure 3). To continue the evolution while preserving shocks, the technology of level set methods introduced by Osher and Sethian [41] has proved to be extremely powerful. The approach relies on the notion of a weak solution, developed in viscosity theory [12] and the introduction of an appropriate entropy condition to select it. The representation of the evolving front as a level set of a hypersurface allows topological changes to be handled in a natural way and robust, efficient implementations have recently been developed [49].

As pointed out in [41], level set methods are Eulerian in nature because computations are restricted to grid points whose locations are fixed. For such methods, the question of computing the locus of shocks for dynamically changing systems remains of crucial importance, i.e., the methods are shock *preserving* but do not explicitly *detect* shocks. Shock detection methods which rely on interpolation of the underlying hypersurface are computationally very expensive. Numerical thresholds are introduced and high order accurate numerical schemes must be used [40, 56].

On the other hand, there are more classical methods rooted in Hamiltonian physics, which can also be used to study shock theory. Although such formulations have been applied to computer vision problems [20, 44, 39], the numerical methods have yet to be widely used. In this paper we review the Hamiltonian formalism for simulating the eikonal equation which offers a number of conceptual advantages when it comes to shock tracking. Hamiltonian systems are of course fundamental in classical physics and have a natural physical interpretation based on elementary Hamiltonian and Lagrangian mechanics. The existence of such simple differential equations is also relevant to considering whether these models have any possible biological implementations [36]. We specialize to the case of Blum's grassfire flow [5] and compute a measure of the average outward flux of the vector field underlying the Hamiltonian system. As the region over which this flux is computed shrinks to a point, the measure can be shown to have very different limiting behaviors depending upon whether or not that point is singular. Thus, it is a very effective way of distinguishing between medial and non-medial points. We combine the average outward flux measure with a homotopy preserving thinning process applied in a discrete lattice. This leads to a robust and efficient algorithm for computing skeletons in 2D as well as 3D which has low computational complexity. We illustrate the method with a number of examples of medial axes (2D) and medial surfaces (3D) of synthetic objects as well as complex anatomical structures obtained from medical images.

To the best of our knowledge, the closest work in computer vision is the formulation of Oliensis and Dupuis of the shape-from-shading problem [39]. Their method also uses Hamilton-Jacobi theory and has similar robust numerical properties. In particular, a density function for marker particles is propagated to obtain estimates of where particles accumulate. This strategy is used to distinguish sources from sinks in order to reconstruct shape from intensity images. We now review some relevant background on skeletons, followed by a brief overview of skeletonization approaches related to our method.

1.1 2D and 3D Skeletons

The 2D skeleton (medial axis) of a closed set $A \subset \mathcal{R}^2$ is the locus of centers of maximal open discs contained within the complement of the set [5]. An open disc is maximal if there exists no other open disc contained in the complement of A that properly contains the disc. The 3D skeleton (medial surface) of a closed set $A \subset \mathcal{R}^3$ is defined in an analogous fashion as the locus of centers of maximal open spheres contained in the complement of the set. Both types of skeletons have been widely used in bio-medicine for tasks involving object representation [37, 58], registration [30] and segmentation [47]. They have also been used for graph-based object recognition in computer vision [38, 66, 52, 31, 57], for animating objects in graphics [62, 42] and for manipulating them in computer-aided design. Despite their popularity, their numerical computation remains non-trivial. Most algorithms are not stable with respect to small perturbations of the boundary and heuristic measures for simplification are often introduced.

Interest in the skeleton as a representation for an object stems from a number of interesting properties: i) it is a *thin* set, i.e., it contains no interior points, ii) it is *homotopic* to the original shape, iii) it is invariant under Euclidean transformations of the object (rotations and translations) and iv) given the radius of the maximal inscribed circle or sphere associated which each skeletal point, the object can be reconstructed exactly. Hence, it provides a compact representation while preserving the object's genus and making certain useful properties explicit, such as its local width.

Approaches to computing skeletons can be broadly organized into three classes. First, methods based on *thinning* attempt to realize Blum's grassfire formulation [5] by peeling away layers from an object while retaining special points [2, 28, 8, 35]. It is possible to define erosion rules in a lattice such that the topology of the object is preserved. However, these methods are quite sensitive to Euclidean transformations of the data and typically fail to localize skeletal points accurately. As a consequence, only a coarse approximation to the object is usually reconstructed [35, 4, 28].

Second, it has been shown that under appropriate smoothness conditions the vertices of the Voronoi diagram of a set of boundary points converges to the exact skeleton as the sampling rate increases [46]. This property has been exploited to develop skeletonization algorithms in 2D [38], as well as extensions to 3D [53, 54]. The dual of the Voronoi diagram, the Delaunay triangulation (or tetrahedralization in 3D) has also been used extensively. Here the skeleton is defined as the locus of centers of circumscribed circles of each triangle (spheres of each tetrahedra in 3D) [18, 37]. Both types of methods ensure homotopy between objects and their skeletons and accurately localize skeletal points, provided that the boundary is sampled densely. Unfortunately, the techniques used to prune elements of the Voronoi graph which correspond to small perturbations of the boundary are typically based on heuristics. In practice, the results are not invariant under Euclidean transformations and the optimization step, particularly in 3D, can have a high computational complexity [37].

A third class of methods exploits the fact that the locus of skeletal points coincides with the singularities of a *Euclidean distance function* to the boundary. These approaches attempt to detect local maxima of the distance function, or the corresponding discontinuities in its derivatives [1, 29, 19]. The numerical detection of these singularities is itself a non-trivial problem; whereas it may be possible to localize them, ensuring homotopy with the original object is difficult. The are also some recent approaches to computing 2D and 3D skeletons which combine aspects of thinning, Voronoi diagrams and distance functions [34, 65, 8, 63].

1.2 Related Work

We now present a brief overview of selected approaches that are related to the method we develop in this paper. We refrain from an exhaustive review of the large body of work in computer vision and computational geometry on computing 2D and 3D skeletons, since this is beyond the scope of this paper.

Leymarie and Levine have simulated the grassfire by utilizing the magnitude of the gradient vector field of a signed distance function to attract a snake moving in from the object's boundary [29]. In this technique the contour has to be first segmented at curvature extrema, which is itself a challenging problem. Kimmel *et al.* have also proposed a method where the contour is first segmented at locations of positive curvature maxima [24]. Outward distance functions to each segment are then computed and the skeleton is obtained by interpolating the zero level set of the distance map differences and removing spurious points. Geiger *et al.* have introduced a variational approach to computing symmetric axis trees, where portions of a curve are matched against others, incorporating constraints including co-circularity and parallelism [32]. The approach leads to an abstraction that is related to the medial axis, but is comprised by a different locus of points. Dynamic programming is used to make the computation efficient. Tek and Kimia have proposed an approach for calculating symmetry maps, which is based on the combination of a wavefront propagation technique with an exact (analytic) distance function [63]. In this technique the representation must be pruned in order to distinguish salient branches from unwanted ones. Pudney has introduced a distance ordered homotopy preserving thinning procedure where points are removed in order of their distance from the boundary while anchoring end points and centers of maximal balls identified from a chamfer distance function [43]. Malandain and Fernandez-Vidal obtain two

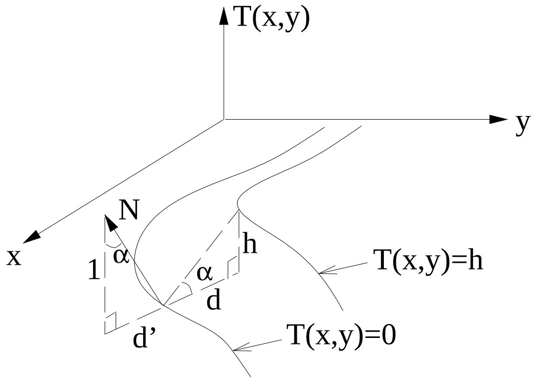

Figure 1: A geometric view of a monotonically advancing front (Eq. 1). $T(x,y)$ is a graph of the 'solution' surface, the level sets of which are the evolved curves.

sets based on thresholding a function of two heuristic measures, ϕ and d, to characterize the singularities of the Euclidean distance function [34]. The two sets are combined using a topological reconstruction process. Tari and Shah have proposed a characterization of the symmetries of n-dimensional shapes by looking at properties of the Hessian of a suitably defined scalar edge-strength functional [60]. Furst and Pizer have introduced a notion of an optimal parameter height ridge in arbitrary dimension by exploiting a sub-dimensional maximum property [16]. Kalitzin *et al.* have considered index computations on vector fields associated with scalar images in order to identify their singularities [21]. Vector fields rooted in magneto-statics have also been used for extracting symmetry and edge lines in greyscale images [13].

2 The Eikonal Equation

We begin by showing the connection between a monotonically advancing front and the well known eikonal equation. Consider the curve evolution equation

$$\frac{\partial \mathcal{C}}{\partial t} = F\mathcal{N}, \tag{1}$$

where $\mathcal{C}$ is the vector of curve coordinates, $\mathcal{N}$ is the unit inward normal and $F = F(x,y)$ is the speed of the front at each point in the plane, with $F \geq 0$ (the case $F \leq 0$ is also allowed). Let $T(x,y)$ be a graph of the solution surface, obtained by superimposing all the evolved curves in time (see Figure 1). In other words, $T(x,y)$ is the time at which the curve crosses a point (x, y) in the plane. Referring to the figure, the speed of the front is given by

$$F(x, y) = \frac{d}{h} = \frac{1}{\tan(\alpha)} = \frac{1}{d'} = \frac{1}{\|\nabla T\|}.$$

Hence, $T(x, y)$ satisfies the *eikonal equation*

$$\|\nabla T\| \, F = 1. \tag{2}$$

A number of algorithms have been recently developed to numerically solve this equation, including Sethian's fast marching method [49] which systematically constructs T using only upwind values, Rouy and Tourin's viscosity solutions approach [44] and Sussman *et al.*'s level set method for incompressible two-phase flows [59]. However, none of these methods address the issue of shock detection explicitly and more work has to be done to track shocks.

A different approach, which is related to the solution surface $T(x, y)$ viewed as a graph, has been proposed by Shah *et al* [50, 61]. Here the key idea is to use an edge strength functional v in place of the surface $T(x, y)$, computed by a linear diffusion equation. The equation can be efficiently implemented and the framework extends to greyscale images as well as curves with triple point junctions. It provides an approximation to the reaction-diffusion space introduced in [23], but does not extend to the extreme cases, i.e., morphological erosion by a disc structuring element (reaction) or motion by curvature (diffusion). Hence, points of maximum (local) curvature of the evolved curves are interpreted as skeletal points. This regularized skeleton is typically not connected and its relation to the classical skeleton, obtained from the eikonal equation with $F = 1$, is as yet unclear.

In the next section, we shall consider an alternate framework for solving the eikonal equation, which is based on the canonical equations of Hamilton. The technique is widely used in classical mechanics and rests on the use of a Legendre transformation (see [3, 51]) which takes a system of n second-order differential equations to a mathematically equivalent system of $2n$ first-order differential equations. We believe that the method offers specific advantages over alternatives for a number of vision problems that involve shock tracking and skeletonization.

3 Hamilton's Canonical Equations and the Hamilton-Jacobi Skeleton Flow

We begin this section with an overview of the Hamiltonian formalism, taken from [3, 51]. Although this is standard material in classical mechanics, these techniques may be unfamiliar to the general computer vision audience. In a Lagrangian formulation the independent variables are the coordinates $\mathbf{q}$ of particles and their velocities $\dot{\mathbf{q}}$. For example, in the context of the Eq. (1) these would be the positions of points along the curve $\mathcal{C}$ and their associated velocities $F\mathcal{N}$. Each particle follows the path of least action in reaching a future location at a future time. In mathematical terms, the *action function* minimized, $S_{\mathbf{q}_0,t_0}$, is given by

$$S_{\mathbf{q}_0,t_0}(\mathbf{q},t) = \int_\gamma \mathcal{L}dt.$$

Here γ is an extremal curve connecting the points $(\mathbf{q}_0, t_0)$ and $(\mathbf{q}, t)$ and $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ is the Lagrangian. In other words, of all possible paths connecting $(\mathbf{q}_0, t_0)$ and $(\mathbf{q}, t)$, the trajectory γ followed by the particle is the one that minimizes the action function. The associated Euler-Lagrange equation is

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = 0 \tag{3}$$

where the momenta are derived quantities given by

$$\mathbf{p} = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}.$$

The key to the Hamiltonian formalism is to exchange the roles of $\dot{\mathbf{q}}$ and $\mathbf{p}$ by replacing the Lagrangian $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ with a Hamiltonian $\mathcal{H}(\mathbf{q}, \mathbf{p})$ such that the velocities now become the derived quantities

$$\dot{\mathbf{q}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}.$$

This can be done by applying the following *Legendre transformation*

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) = \mathbf{p} \cdot \dot{\mathbf{q}} - \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) \tag{4}$$

where the $\dot{\mathbf{q}}$'s are written as functions of $\mathbf{q}$'s and $\mathbf{p}$'s. It is a simple exercise to verify that the above expression for the velocities $\dot{\mathbf{q}}$ then holds. One can also take partial derivatives of the Hamiltonian with respect to the $\mathbf{q}$'s and verify that

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = -\frac{\partial \mathcal{L}}{\partial \mathbf{q}}.$$

The Lagrangian formalism	The Hamiltonian formalism
The state of the system is described by $(\mathbf{q}, \dot{\mathbf{q}})$.	The state of the system is described by $(\mathbf{q}, \mathbf{p})$.
The state may be represented by a point moving with a velocity in an n-dimensional configuration space.	The state may be represented by a point in a 2n-dimensional phase space.
The n coordinates evolve according to n second-order equations.	The 2n coordinates and momenta obey 2n first-order equations.
For a given $\mathcal{L}$ several trajectories may pass through a given point in the configuration space.	For a given $\mathcal{H}$ only one trajectory passes through a given point in the phase space.

Table 1: A comparison of the Lagrangian and Hamiltonian formalisms, taken from [51].

Using Eq. (3), $\frac{\partial \mathcal{L}}{\partial \mathbf{q}}$ can be replaced with $\dot{\mathbf{p}}$ to give *Hamilton's canonical equations*:

$$\dot{\mathbf{p}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}, \qquad \dot{\mathbf{q}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}. \tag{5}$$

Thus, in the Hamiltonian formalism one starts with the initial positions and momenta $(\mathbf{q}(0), \mathbf{p}(0))$ and integrates Eq. (5) to obtain the phase space $(\mathbf{q}(t), \mathbf{p}(t))$ of the system. A comparison of the Lagrangian and Hamiltonian formalisms is presented in Table 1.

Following Arnold [3, pp. 248–258], we now use Huygens' principle to show the connection between the eikonal equation and a Hamilton-Jacobi equation. For every point $\mathbf{q}_0$, define the function $S_{\mathbf{q}_0}(\mathbf{q})$ as the cost of the path from $\mathbf{q}_0$ to $\mathbf{q}$ (see Figure 2). As indicated earlier, the trajectory followed from $\mathbf{q}_0$ to $\mathbf{q}$ will be the path of least action. The wave front generated at time t is given by $\{\mathbf{q} : S_{\mathbf{q}_0}(\mathbf{q}) = t\}$. The vector $\mathbf{p} = \frac{\partial S}{\partial \mathbf{q}}$ is called the *vector of normal slowness of the front.* By Huygens' principle the direction of the ray $\dot{\mathbf{q}}$ is conjugate to the direction of motion of the front, i.e., $\mathbf{p} \cdot \dot{\mathbf{q}} = 1$. In an anisotropic medium the vectors $\mathbf{p}$ and $\dot{\mathbf{q}}$ have different directions in general.

Let us specialize to the case of a monotonically advancing front in an inhomogeneous but isotropic medium (Eq. 1). Here the speed $F(x, y)$ depends only on position (not on direction), and the directions of $\mathbf{p}$ and $\dot{\mathbf{q}}$ coincide. The Lagrangian associated with the action function

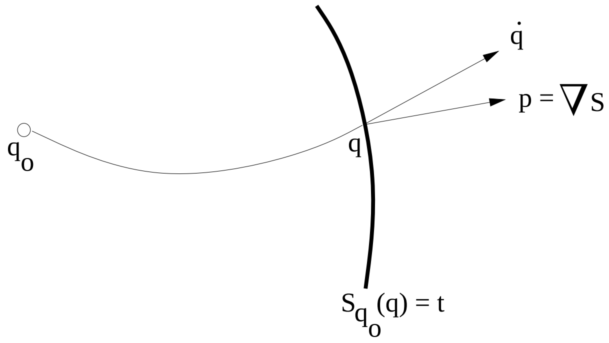

Figure 2: Direction of a ray $\dot{\mathbf{q}}$ and the direction of motion of the wave front $\mathbf{p}$. From [3].

minimized (Eq. (3)) is given by

$$\mathcal{L} = \frac{1}{F(x,y)} \; \|\partial\gamma/\partial t\| = \frac{1}{F(x,y)} \; \|\dot{\mathbf{q}}\| \; .$$

This can be interpreted as a conformal (infinitesimal) length element and we have assumed that the extremals emanating from the point $(\mathbf{q}_0, t_0)$ do not intersect elsewhere, i.e., they form a *central field of extremals.* For an isotropic medium the extremals turn out to be straight lines and for the special case $F(x,y) = 1$ the action function becomes Euclidean length.

It can be shown that the vector of normal slowness, $\mathbf{p} = \frac{\partial S}{\partial \mathbf{q}}$, is not arbitrary but satisfies the Hamilton-Jacobi equation

$$\frac{\partial S}{\partial t} = -\mathcal{H}\left(\mathbf{q}, \frac{\partial S}{\partial \mathbf{q}}\right), \tag{6}$$

where the Hamiltonian function $\mathcal{H}(\mathbf{q}, \mathbf{p})$ is the Legendre transformation with respect to $\dot{\mathbf{q}}$ of the Lagrangian discussed earlier [3]. Rather than solve the nonlinear Hamilton-Jacobi equation for the action function S (which will give the solution surface $T(x,y)$ to Eq. (2)), it is much more convenient to look at the evolution of the phase space $(\mathbf{q}(t), \mathbf{p}(t))$ under the equivalent Hamiltonian system given by Eq. (5). This offers a number of advantages, the most significant being that the equations become linear and hence trivial to simulate numerically. We shall now derive this system of equations for the special case of a front advancing with speed $F(x,y) = 1$. This case is of particular interest for shape analysis because the locus of shocks which form coincides with the trace of the Blum skeleton [5, 9, 23].

For the case of a front moving with constant speed, recall that the action function being minimized is Euclidean length and hence S can be viewed as a Euclidean distance function from the initial curve $\mathcal{C}_0$. Furthermore, the magnitude of its gradient, $||\nabla S||$, is identical to 1 in its smooth regime, which is precisely where the assumption of a central field of extremals is valid.

With $\mathbf{q} = (x, y)$, $\mathbf{p} = (S_x, S_y)$, $||\mathbf{p}||= 1$, we associate to the evolving plane curve $\mathcal{C} \subset \mathbf{R}^2$ the surface $\tilde{C} \subset \mathbf{R}^4$ given by

$$\tilde{C} := \{(x, y, S_x, S_y) : (x, y) \in C,\ S_x^2 + S_y^2 = 1,\ \mathbf{p} \cdot \dot{\mathbf{q}} = 1\}.$$

The Hamiltonian function obtained by applying the Legendre transformation (Eq. 4) to the Lagrangian $\mathcal{L} =||\dot{\mathbf{q}}||$ is given by

$$\mathcal{H} = \mathbf{p} \cdot \dot{\mathbf{q}} - \mathcal{L} = 1 - (S_x^2 + S_y^2)^{\frac{1}{2}}.$$

The associated Hamiltonian system is

$$\dot{\mathbf{p}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = (0, 0), \qquad \dot{\mathbf{q}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} = -(S_x, S_y). \tag{7}$$

$\tilde{C}$ can be evolved under this system of equations, with $\tilde{C}(t) \subset \mathbf{R}^4$ denoting the resulting (contact) surface. The projection of $\tilde{C}(t)$ onto $\mathbf{R}^2$ will then give the parallel evolution of $\mathcal{C}$ at time t, $\mathcal{C}(t)$.

We illustrate this flow by representing the initial curve with a sequence of marker particles and then evolving them according to Eq. 7. Further details are presented in [55]. With $\mathbf{q} = (x, y)$, $\mathbf{p} = (S_x, S_y) = \nabla S$, the system of equations becomes

$$\{\dot{S}_x = 0,\ \dot{S}_y = 0; \quad \dot{x} = -S_x,\ \dot{y} = -S_y\},$$

a gradient dynamical system. The second equation indicates that the trajectory of the marker particles will be governed by the vector field obtained from the gradient of the Euclidean distance function S and the first indicates that this vector field does not change with time and can be computed once at the beginning of the simulation. Projecting this 4D system onto the (x, y) plane for each instance of time t will give the evolved curve $\mathcal{C}(t)$. The superposition of all the level curves gives the solution surface $T(x, y)$ in Figure 1. Figure 3 depicts the evolution of marker particles, with speed $F = 1$, for several different shapes. Whereas a variety of methods can be used to simulate the eikonal equation, including level set techniques and their fast marching versions [49], the Hamiltonian formalism offers the advantage that the formation of shocks can be made explicit. As shown in the following section the key idea is to exploit a measure of the average outward flux of the vector field $\dot{\mathbf{q}}$.

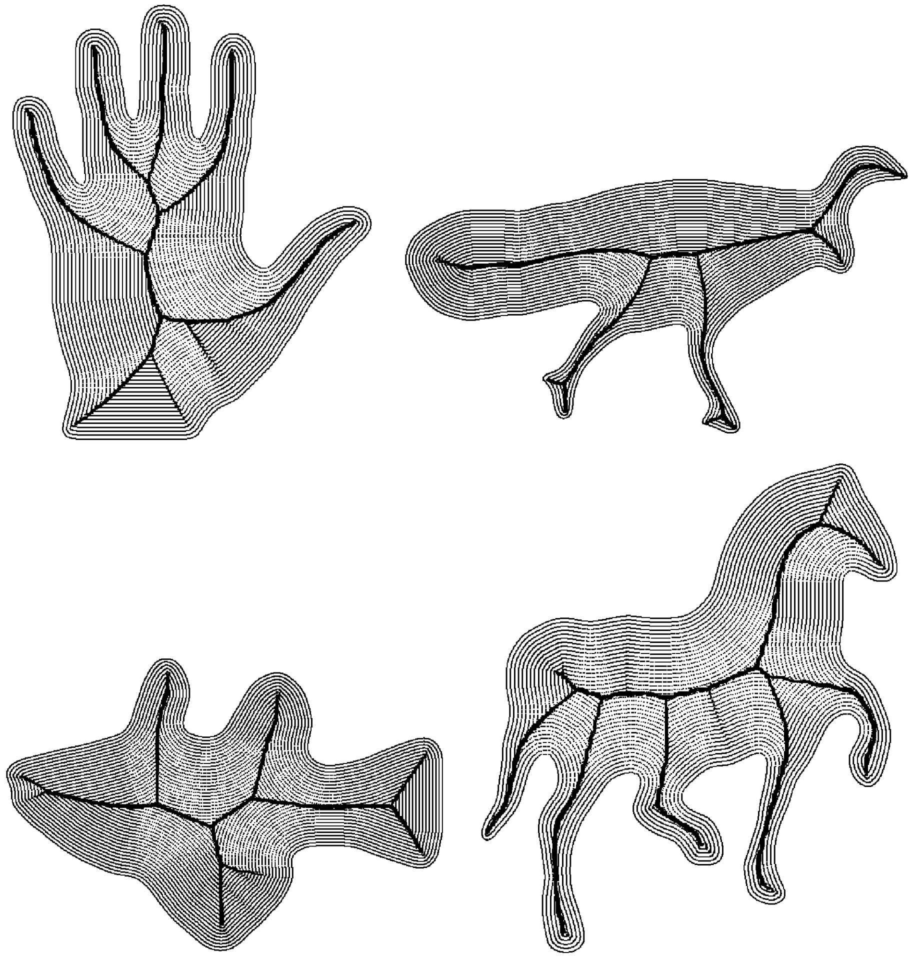

Figure 3: The evolution of marker particles under the Hamiltonian system. The initial particles are placed on the boundary and iterations of the process are superimposed. These correspond to level sets of the solution surface $T(x,y)$ in Figure 1.

4 Flux and Divergence

We approach the discrimination of medial points, which coincide with the shocks of the grassfire flow, from non-medial ones, by computing the average outward flux of the vector field $\dot{\mathbf{q}}$ about a point. The average outward flux is defined as the outward flux through the boundary of a region containing the point, normalized by the length of the boundary

$$\frac{\int_{\delta R} < \dot{\mathbf{q}}, \mathcal{N} > ds}{\text{length}(\delta R)} \tag{8}$$

Here ds is an element of the bounding contour δR of the region R and $\mathcal{N}$ is the outward normal at each point of the contour. Via the divergence theorem

$$\int_R \mathrm{div}(\dot{\mathbf{q}})\mathrm{da} \equiv \int_{\delta \mathrm{R}} < \dot{\mathbf{q}}, \mathcal{N} > \mathrm{ds}, \tag{9}$$

where da is an area element. Thus the outward flux is related to the divergence in the following way

$$\mathrm{div}(\dot{\mathbf{q}}) \equiv \lim_{\Delta \mathrm{a} \to 0} \frac{\int_{\delta \mathrm{R}} < \dot{\mathbf{q}}, \mathcal{N} > \mathrm{ds}}{\Delta \mathrm{a}}. \tag{10}$$

It is well known that the outward flux, or equivalently the integral of the divergence of $\dot{\mathbf{q}}$ measures the degree to which the flow generated by $\dot{\mathbf{q}}$ is area preserving for the region over which it is computed. To elaborate, the outward flux (and hence also the average outward flux) is negative if the area enclosed by the region δR is shrinking under the action of the Hamiltonian flow, positive if it is growing and zero otherwise. This quantity is clearly strongly dependent on the shape of the region R. However, it can be shown that in the limit as the region δR shrinks to a non-medial point, the average outward flux approaches zero.

When considering a region δR that contains a medial point, unfortunately the standard form of the divergence theorem does not apply since the vector field $\dot{\mathbf{q}}$ becomes singular. Instead, the limiting behavior of the average outward flux as the region δR shrinks to a medial point can be considered. Furthermore, it can be shown that this quantity approaches a strictly negative number proportional to $< \dot{\mathbf{q}}, \mathcal{N}' >$, where $\mathcal{N}'$ is now a one-sided normal to the medial axis or surface.[1] The constant of proportionality depends upon whether the axis point is a regular point or a branch point. Thus, in the limit as δR shrinks to a point the average outward flux calculation is an effective way of detecting the singularities of the vector field $\dot{\mathbf{q}}$. Non-medial points give values that are close to zero and medial points corresponding to strong singularities give large negative values. Whereas thus far we have focussed on the case of a (2D) closed curve, the very same analysis applies to a closed (3D) surface evolving according to an eikonal equation. One simply has to replace the initial closed curve $\mathcal{C}$ with the closed surface $\mathcal{S}$ in Eq. (1), add a third coordinate z to the phase space in Eq. (7) and

[1] The proof of this fact was conveyed to us by Jim Damon in personal communications. He pointed us to an alternate form of the divergence theorem that could be applied in regions intersecting the medial axis or surface, which he used to work out the limiting behavior of the average outward flux. We are grateful to him for this analysis.

replace the area element with a volume element and the contour integral with a surface integral in Eqs. (8), (9) and (10).

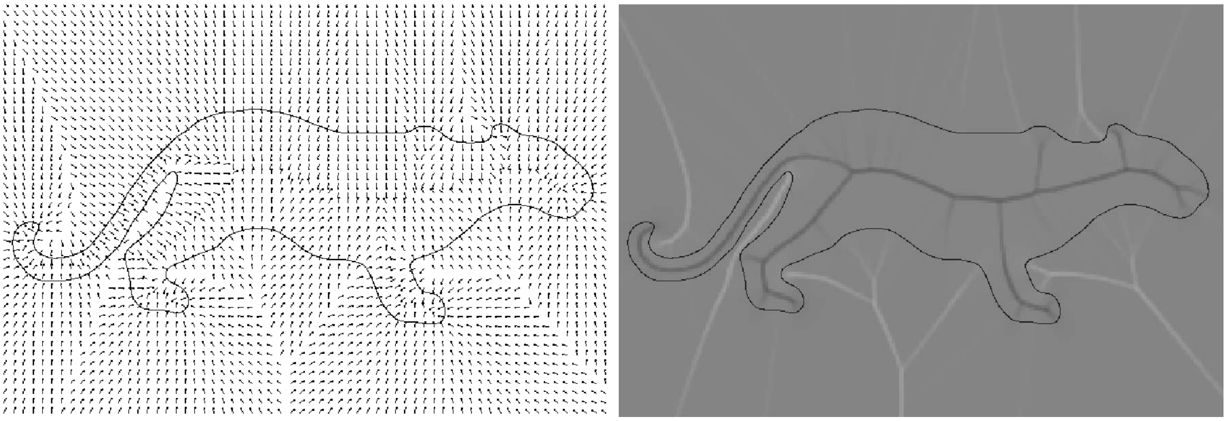

Figure 4: The gradient vector field of a signed distance function to the boundary of a panther shape (left), with the associated average outward flux (right). Whereas the smooth regime of the vector field gives zero flux (medium grey), strong singularities give either large negative values (dark grey) in the interior or large positive values (light grey) in the exterior.

Figure 4 illustrates the average outward flux computation on the silhouette of a panther shape, where values close to zero are shown in medium grey. All computations are carried out on a rectangular lattice, although the bounding curve is shown in interpolated form. Strictly speaking, the average outward flux is desired only in the limit as the region shrinks to a point. However, the average outward flux over a very small neighborhood (a circle in 2D or a sphere in 3D) provides a sufficient approximation to the limiting values. Strong singularities correspond either to high magnitude negative (dark grey) or positive numbers (light grey), depending upon whether the vector field is collapsing at or emanating from a particular point. A threshold on the average outward flux yields a close approximation to the skeleton, as used in [55]. However, in general it is impossible to guarantee that the result obtained by simple thresholding is homotopic to the original shape. A high threshold may yield a connected set, but it is not thin and unwanted branches may be present, Figure 5 (left). A low threshold yields a thin set, but it may be disconnected, Figure 5 (right). The solution, as we shall now show, is to introduce additional constraints to ensure that the resulting skeleton is homotopic to the shape. The essential idea is to incorporate a homotopy preserving thinning process, where the removal of points is guided by the average outward flux values. In the context of the Hamilton-Jacobi skeleton flow (Eq. 7), this leads to a robust and efficient algorithm for computing 2D and 3D skeletons.

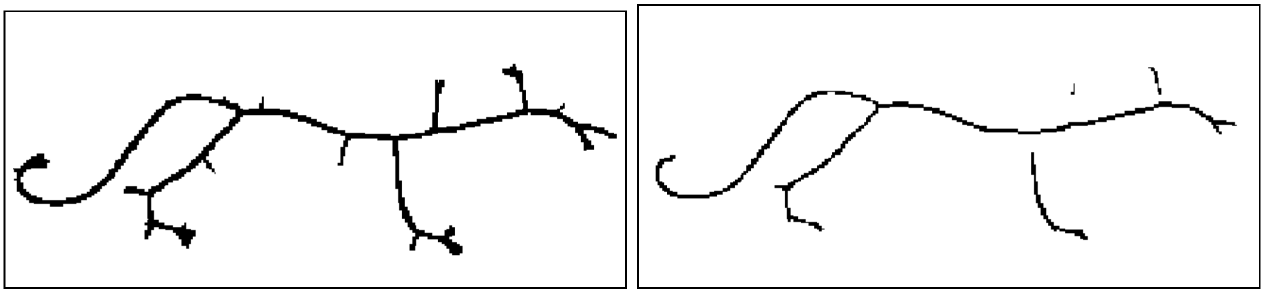

Figure 5: Thresholding the average outward flux map in Figure 4. A high threshold yields a connected set, but it is not thin and unwanted branches are present (left). A low threshold yields a closer approximation to the desired medial axis, but the result is now disconnected (right).

5 Homotopy Preserving Skeletons

Our goal is to combine the divergence computation with a digital thinning process, such that as many points as possible are removed without altering the object's topology. In digital topology a point is *simple* if its removal does not change the topology of the object. In 2D we shall consider rectangular lattices, where a point is a unit square with 8 neighbors, as shown in Figure 6 (left). Hence, a 2D digital point is simple if its removal does not disconnect the object or create a hole. In 3D we shall consider cubic lattices, where a point is a unit cube with 6 faces, 12 edges and 8 vertices. Hence, a 3D digital point is simple if its removal does not disconnect the object, create a hole, or create a cavity [26].

5.1 2D Simple Points

Consider the 3x3 neighborhood of a 2D digital point P contained within an object and select those neighbors which are also contained within the object. Construct a neighborhood graph by placing edges between all pairs of neighbors (not including P) that are 4-adjacent or 8-adjacent to one another. If any of the 3-tuples $\{2,3,4\}$, $\{4,5,6\}$, $\{6,7,8\}$, or $\{8,1,2\}$, are nodes of the graph, remove the corresponding diagonal edges $\{2,4\}$, $\{4,6\}$, $\{6,8\}$, or $\{8,2\}$, respectively. This ensures that there are no degenerate cycles in the neighborhood graph (cycles of length 3). Now, observe that if the removal of P disconnects the object,

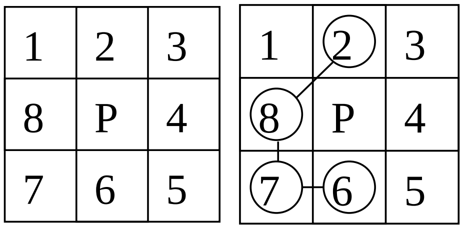

Figure 6: LEFT: A 3x3 neighborhood of a candidate point for removal P. RIGHT: An example neighborhood graph for which P is simple. There is no edge between neighbors 6 and 8 (see text).

or introduces a hole, the neighborhood graph will not be connected, or will have a cycle, respectively. Conversely, a connected graph that has no cycles is a tree. Hence, we have a criterion to decide whether or not P is simple:

Proposition 1 *A 2D digital point P is* simple *if and only if its 3x3 neighborhood graph, with cycles of length 3 removed, is a tree.*

A straightforward way of determining whether or not a graph is a tree is to check that its Euler characteristic $|V| - |E|$ (the number of vertices minus the number of edges) is identical to 1. This check only has to be performed locally, in the 3x3 neighborhood of P. Figure 6 (right) shows an example neighborhood graph for which P can be removed.

5.2 3D Simple Points

In 3D a digital point can have three types of neighbors: two points are *6-neighbors* if they share a face; two points are *18-neighbors* if they share a face or an edge; and two points are *26-neighbors* if they share a face, an edge or a vertex. This induces three *n-connectivities*, where $n \in \{6, 18, 26\}$, as well as three *n-neighborhoods* for x $(N_n(x))$. An n-neighborhood without its central point is defined as $N_n^* = N_n(x) \backslash \{x\}$. An object A is *n-adjacent* to an object B, if there exist two points $x \in A$ and $y \in B$ such that x is an n-neighbor of y. A *n-path* from x_1 to x_k is a sequence of points $x_1, x_2, ..., x_k$, such that for all x_i, $1 < i \leq k$, x_{i-1} is n-adjacent to x_i. An object represented by a set of points O is *n-connected*, if for every pair of points $(x_i, x_j) \in O \times O$, there is a n-path from x_i to x_j.

Based on these definitions, Malandain *et al.* provide a topological classification of a point x in a cubic lattice by computing two numbers [33]: i) C^*: the number of 26-connected components 26-adjacent to x in $O \cap N_{26}^*$ and ii) $\bar{C}$: the number of 6-connected components

6-adjacent to x in $\bar{O} \cap N_{18}$. An important result with respect to our goal of thinning is the following:

Theorem 1 *(Malandain* et al. *1993) P is* simple *if $C^*(P) = 1$ and $\bar{C}(P) = 1$.*

We can now determine whether or not the removal of a point will alter the topology of a digital object. When preserving homotopy is the only concern, simple points can be removed sequentially until no more simple points are left. The resulting set will be thin and homotopic to the object. However, without a further criterion the relationship to the skeleton will be uncertain since the locus of surviving points depends entirely on the order in which the simple points are removed. In the current context, we have derived a natural criterion for ordering the thinning, based on the average outward flux of the gradient vector field of the Euclidean distance function.

5.3 Flux-Ordered Thinning

Recall from Section 4, that the average outward flux of the gradient vector field of the Euclidean distance function can be used to distinguish non-medial points from medial ones. This quantity tends to zero for the former, but approaches a negative number below a constant times $< \dot{\mathbf{q}}, \mathcal{N}' >$ for the latter, where $\mathcal{N}'$ is the one-sided normal to the medial axis or surface. Hence, the average outward flux provides a natural measure of the "strength" of a skeletal point for numerical computations. The essential idea is to order the thinning such that the weakest points are removed first and to stop the process when all surviving points are not simple, or have a total average outward flux below some chosen (negative) value, or both. This will accurately localize the skeleton and also ensure homotopy with the original object. Unfortunately the result is not guaranteed to be a thin set, i.e., one without an interior.

One way of satisfying this last constraint is to define an appropriate notion of an end point. Such a point would correspond to the end point of a curve (in 2D or 3D), or a point on the rim of a surface, in 3D. The thinning process would proceed as before, but the threshold criterion for removal would be applied *only to end points.* Hence, all surviving points which were not end points would not be simple and the result would be a thin set.

In 2D, an end point will be viewed as any point that could be the end of a 4-connected or

8-connected digital curve. It is straightforward to see that such a point may be characterized as follows:

Proposition 2 *A 2D point P could be an end point of a 1 pixel thick digital curve if, in a 3x3 neighborhood, it has a single neighbor, or it has two neighbors, both of which are 4-adjacent to one another.*

In 3D, the characterization of an end point is more difficult. An end point is either the end of a 26-connected curve, or a corner or point on the rim of a 26-connected surface. In $\mathcal{R}^3$, if there exists a plane that passes through a point p such that the intersection of the plane with the object includes an open curve which ends at p, then p is an end point of a 3D curve, or is on the rim or corner of a 3D surface. This criterion can be discretized easily to 26-connected digital objects by examining 9 digital planes in the 26-neighborhood of p as in [43].

5.4 The Algorithm and its Complexity

The essential idea behind the flux-ordered thinning process is to remove simple points sequentially, ordered by their average outward flux, until a threshold is reached. Subsequently, simple points are removed if they are not end points. The procedure converges when all remaining points are either not simple or are end points. The thinning process can be made very efficient by observing that a point which does not have at least one background point as an immediate neighbor cannot be removed, since this would create a hole or a cavity. Therefore, the only potentially removable points are on the border of the object. Once a border point is removed, only its neighbors may become removable. This suggests the implementation of the thinning process using a heap. A full description of the procedure is given in Algorithm 1.

We now analyze the complexity of the algorithm. The computation of the distance transform [6], the gradient vector field and the average outward flux are all $\mathcal{O}(n)$ operations. Here n is the total number of points in the array. The implementation of the thinning is more subtle. We claim an $\mathcal{O}(klog(k))$ worst case complexity, where k is the number of points inside the object. The explanation is as follows. At first, store only the points that are on the outer layer of the object in a heap, using the average outward flux as the sorting key

Algorithm 1 The Flux-Ordered Thinning Algorithm.

```
Part I: Average Outward Flux
Compute the distance transform of the object $D$ [6].
Compute the gradient vector field $\nabla D$.
Compute the average outward flux of $\nabla D$ using Eq. 9
    For each point $P$ in the interior of the object
        $Flux(P) = \sum_{i=1}^{n} < N_i, \nabla D(P_i) > /n$,
        where $P_i$ is an n-neighbor ($n = 8$ in 2D, $n = 26$ in 3D) of $P$ and
        $N_i$ is the outward normal at $P_i$ of the unit (disc in 2D, sphere in 3D)
        centered at $P$.
Part II: Homotopy Preserving Thinning
For each point $P$ on the boundary of the object
if ($P$ is simple)
    insert($P$, Heap) with $Flux(P)$
    as the sorting key for insertion
While (Heap.size $> 0$)
$P =$ HeapExtractMax(Heap)
    if ($P$ is simple)
        if ($P$ is not an end point) or ($Flux(P) > Thresh$)
            Remove $P$
            for all neighbors $Q$ of $P$
                if ($Q$ is simple)
                    insert($Q$, Heap)
        else mark $P$ as a skeletal (end) point
        end { if }
    end { if }
end { while }
```

for insertion. The extraction of the maximum from the heap will provide the best candidate for removal. If this point is removable, then delete it from the object and add its *simple* (potentially removable) neighbors to the heap. A point can only be inserted a constant number of times (at most 26 times for a 3D, 26-neighborhood and at most 8 times for a 2D, 8-neighborhood) and insertion in a heap, as well as the extraction of the minimum, are both $\mathcal{O}(log(l))$ operations, where l is the number of elements in the heap. There cannot be more than k elements in the heap, because we only have a total of k points within the object. The worst case complexity for thinning is therefore $\mathcal{O}(klog(k))$. Hence, the worst case complexity of the algorithm is $\mathcal{O}(n) + \mathcal{O}(klog(k))$. We should point out that this is a very loose upper bound. The heap only contains points from the object's surface and therefore in practice the complexity is almost linear in the number of digital points n.

6 Examples

6.1 Medial Axes

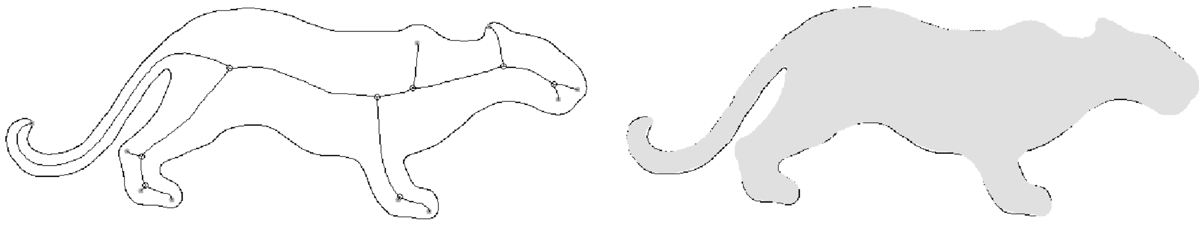

Figure 7: Left: A subpixel medial axis, with branch points shown as empty circles and end points as filled circles. Compare with the results in Figure 5. Right: The reconstruction as the envelope of the maximal inscribed disks (grey) of the medial axis, overlaid on the original shape.

We first present examples of medial axes, computed for a range of 2D binary shapes. The same outward flux threshold was used in each example to determine which end points to preserve. The input is a 2D binary array where the foreground and background are identified by distinct values. The implementation then uses an exact (signed) distance function to a piecewise circular arc interpolation of the boundary, which allows for subpixel computations (details are presented in [14]). Figure 7 (left) shows the subpixel medial axis for the panther silhouette with branch points shown as empty circles and end points as closed circles. The accuracy of the representation is illustrated in Figure 7 (right), where the shape is reconstructed as the envelope of the maximal inscribed discs associated with each medial axis point. Figure 8 depicts subpixel medial axes for a number of other shapes. The results demonstrate the robustness of the framework under Euclidean transformations, as well as changes in scale.

6.2 Medial Surfaces

Next we illustrate the algorithm with both synthetic data and volumetric structures segmented from medical images. For these we used the D-Euclidean distance function [6] which provides a good approximation to the true distance function. Once again, the only free parameter is the choice of the outward flux threshold below which the removal of end points is blocked. For these examples, the value was selected so that approximately 25% of the points within the volume had a lower average outward flux.

6.2.1 Accuracy, Stability and Robustness

Accuracy: We test the method by using synthetic objects for which the expected structure of the medial surface is known. Figures 9 and 10 illustrate the computation of the flux-based medial surfaces for a cube and a cylinder, respectively. In both cases the computations lead to the structures one would expect when considering the loci of centres of maximal inscribed spheres. Also, the reconstruction from the medial surface and its associated distance function is accurate.

Stability: Next we test the sensitivity of the method to boundary perturbations. Figure 11 shows the same cube as earlier, but with points randomly removed (top row) or added (bottom row), up to a depth of four voxels. The resulting medial surface is no longer as smooth as before but no spurious sheets have been added. This illustrates the stability of the method in the presence of moderate boundary noise. In fact, the boundary protrusions or indentations have to be significant in order for spurious branches or sheets to appear, as illustrated in Figure 12.

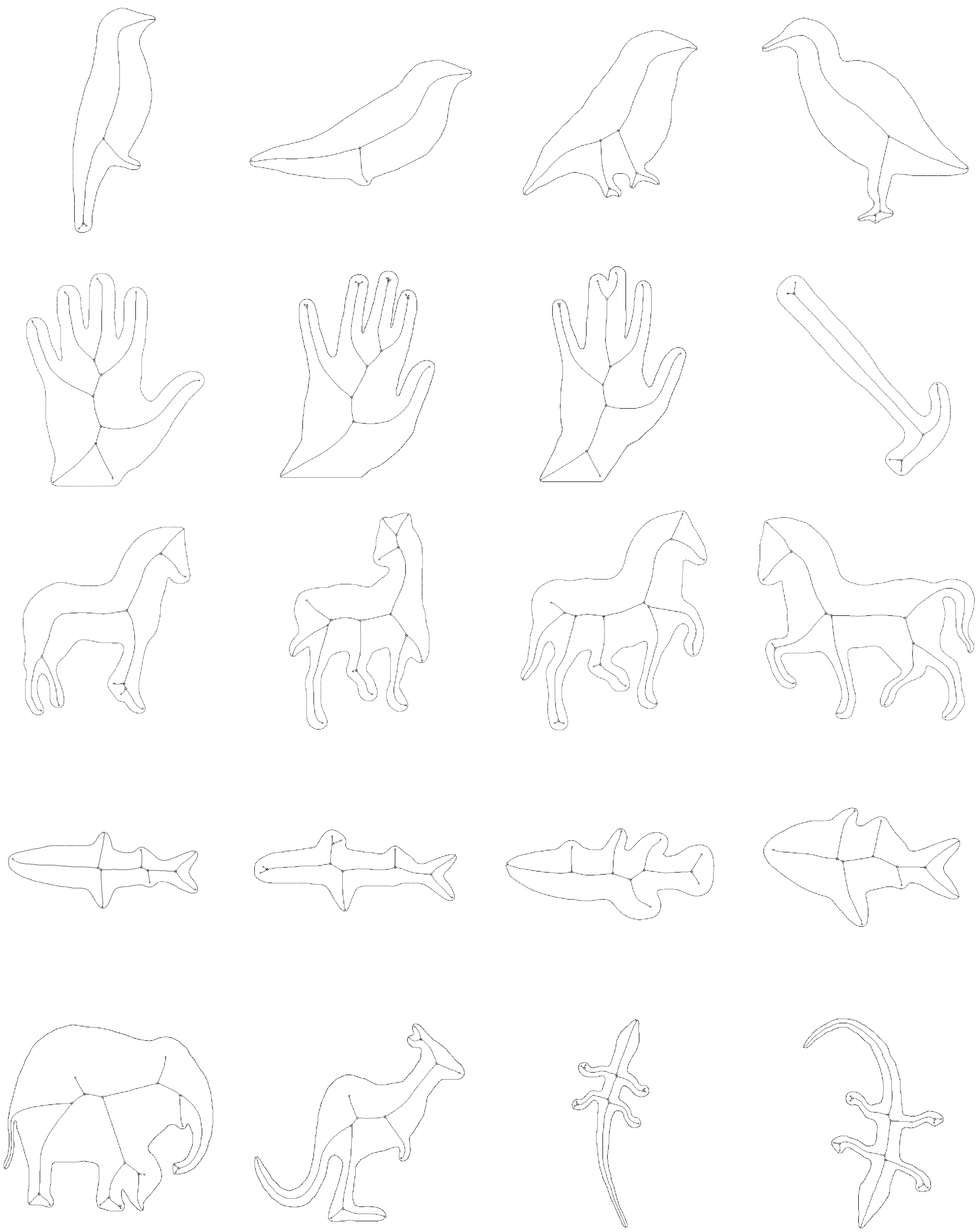

Figure 8: Subpixel medial axes for a range of shapes, obtained by flux-ordered thinning. The detected end points and branch points are circled.

Robustness: Third, we test the robustness of the method under rotation. We rotate the cube by 30 degrees around the z axis and compute its medial surface. Figure 13 compares this result with the medial surface of the original cube. The two outputs are clearly almost

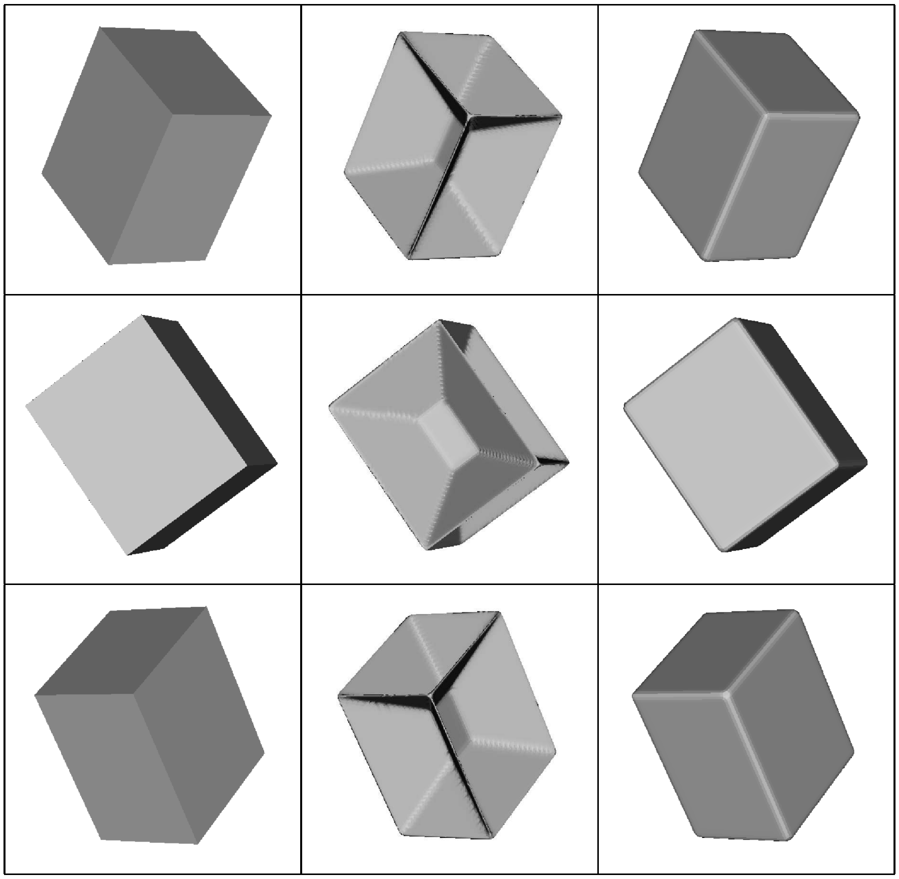

Figure 9: First Column: Three views of a cube. Second Column: The corresponding flux-based medial surfaces. Third Column: The object reconstructed from the medial surfaces in the previous column.

identical.

6.2.2 Labeling the Medial Surface

The medial surface can be labeled using the classification of Malandain *et al.* [33]. Specifically, the numbers C^* and $\bar{C}$, described in Section 5, can be used to classify curve points, surface points, border points and junction points. However, junction points can be misclassified as surface points when certain special configurations of voxels occur and these cases have to be dealt with using a new definition for simple surfaces [33].

Let x be a surface point ($\bar{C} = 2$ and $C^* = 1$). Let B_x and C_x be the two connected components of $\bar{O} \cap N_{18}$ 6-adjacent to x. Two surface points x and y are in an equivalence

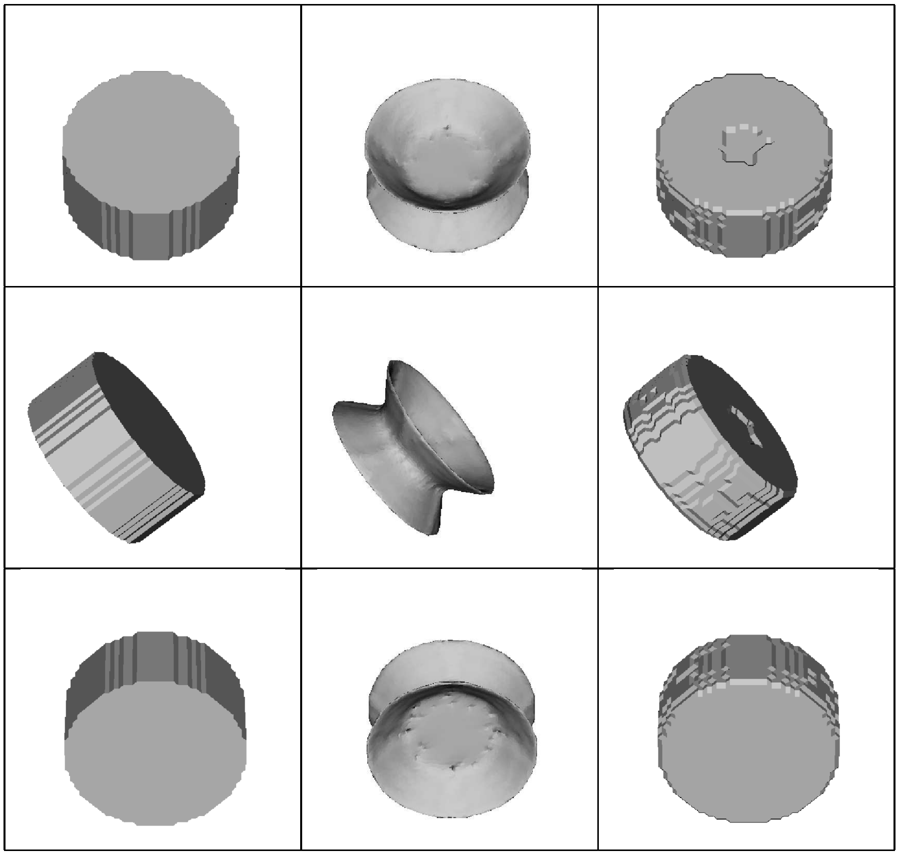

Figure 10: FIRST COLUMN: Three views of a cylinder. SECOND COLUMN: The corresponding flux-based medial surfaces. THIRD COLUMN: The object reconstructed from the medial surfaces in the previous column.

relation if there exists a 26-path $x_0, x_1, ..., x_i, ..., x_n$ with $x_0 = x$ and $x_n = y$ such that for $i \in [0, ..., n-1]$, $(B_{x_i} \cap B_{x_{i+1}} \neq \emptyset$ and $C_{x_i} \cap C_{x_{i+1}} \neq \emptyset)$ or $(B_{x_i} \cap C_{x_{i+1}} \neq \emptyset$ and $C_{x_i} \cap B_{x_{i+1}} \neq \emptyset)$. A *simple surface* is then defined as any equivalence class of this equivalence relation.

We use this definition in our framework to find all the misclassified junctions. If the 26-neighborhood of a previously classified surface point x is not a *simple surface*, then x is a junction point. Figures 13 and 14 illustrate the labeling of the medial surface of a cube and a cylinder. The medial surface of the cylinder is correctly labeled as having two *simple* sheets connected by a 3D digital curve through two junction points.

The same definition can be used to extract the individual simple surfaces comprising the medial surface of an object. The idea is to find an unmarked surface point on the medial surface and use it as a "source" to build its associated simple surface using a depth first

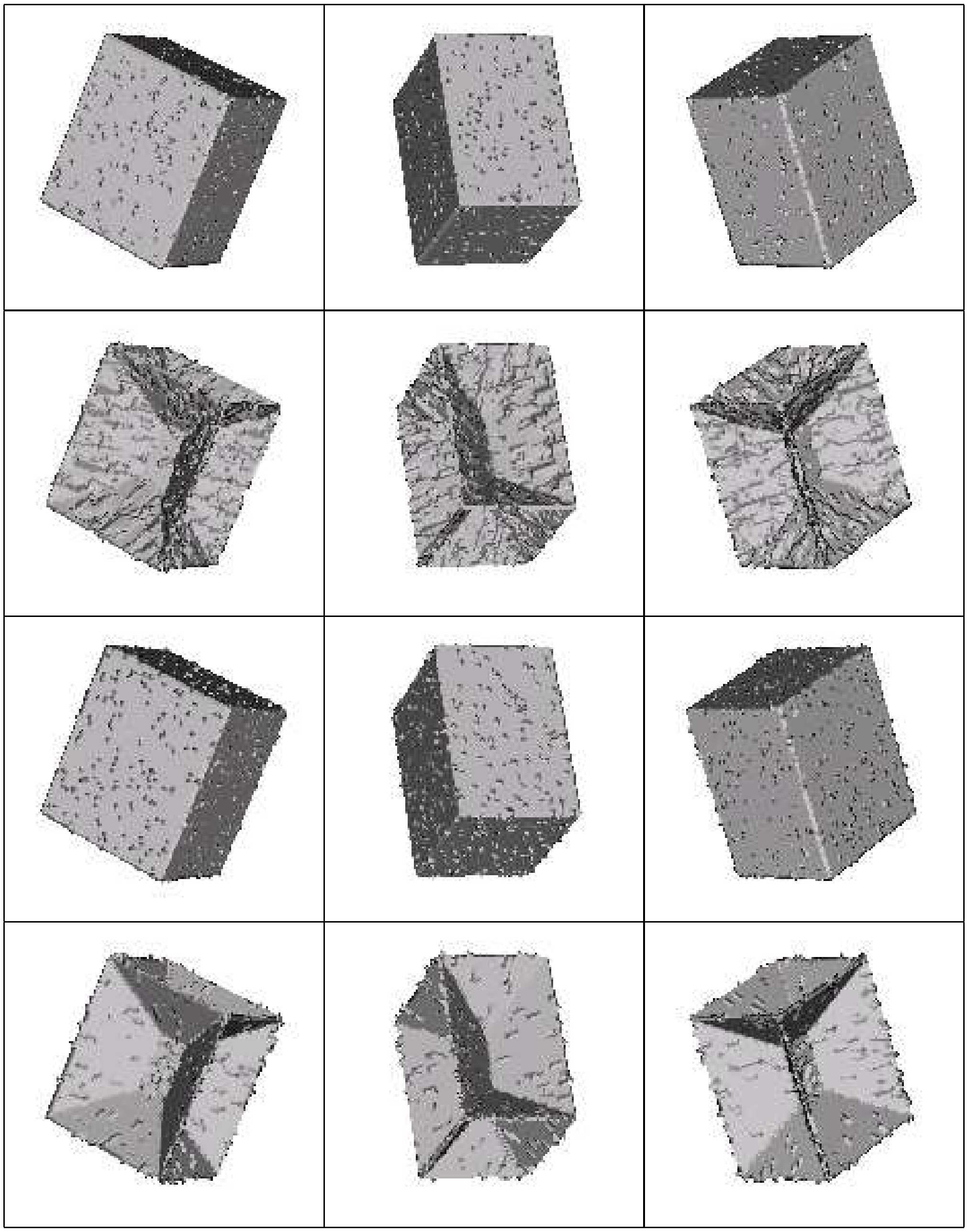

Figure 11: First Row: Three views of the cube in Figure 9, but with up to 4 voxels in depth *removed* randomly from the surface. Second Row: The resulting medial surface. Third Row Three views of a cube in Figure 9, but with up to 4 voxels in depth *added* randomly to the surface. Fourth Row: The resulting medial surface.

search strategy. The next simple surface is built from the next unmarked surface point and so on, until all surface points are marked.

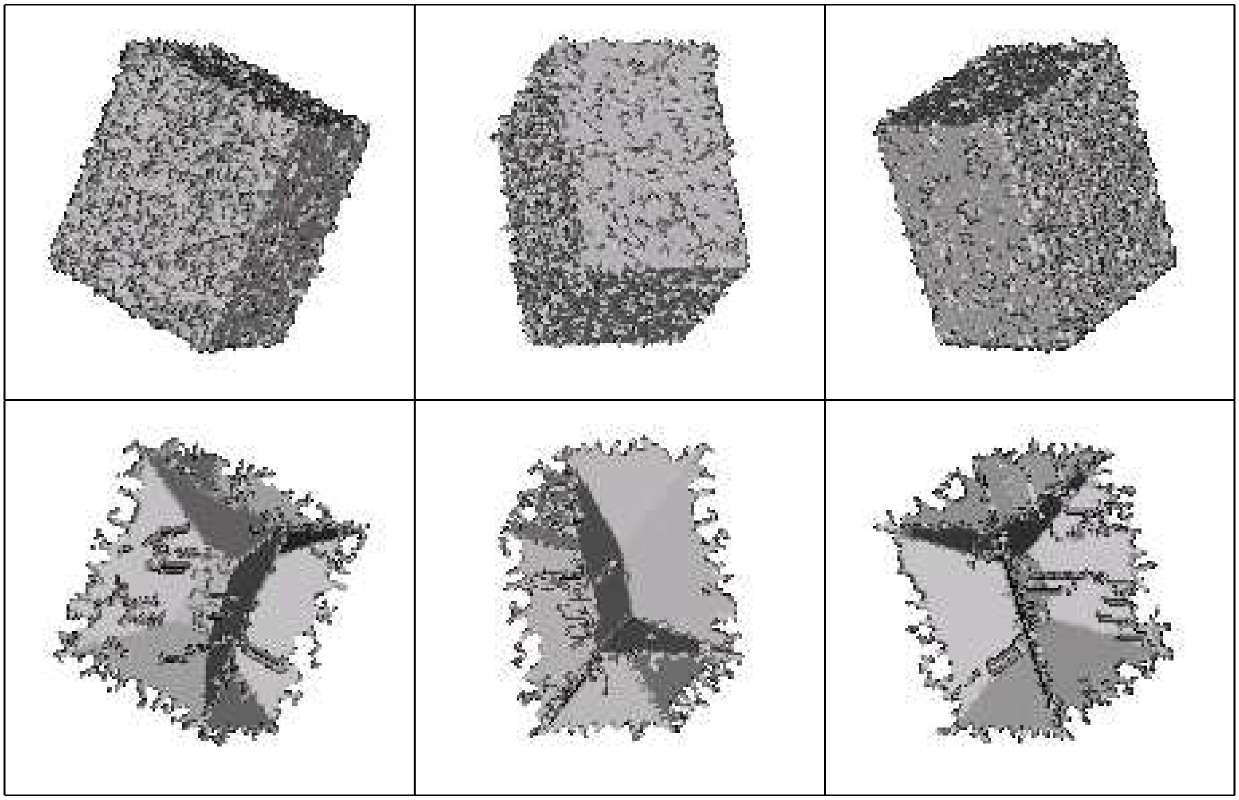

Figure 12: Top Row: Three views of the cube in Figure 9 but with up to 30 voxels in depth randomly removed or added to the surface. Bottom Row The resulting medial surface shows spurious branches and sheets, but has a smooth main structure.

6.2.3 Experiments on MR and MRA Data

We now illustrate the method on volumetric data segmented from medical images. Figure 15 illustrates the results on brain ventricles obtained from a magnetic resonance (MR) image. The medial surface consists of two main sheets which reflect the “butterfly-like” structure of the original object. The figure demonstrates that thresholding the flux (second column) results in erroneous topologies, whereas the full algorithm (third column) computes medial surfaces which are both thin and topologically correct. The ventricles reconstructed from the medial surface in the third column are shown in the fourth column.

Next, we illustrate the approach on a (partial) data set of blood vessels obtained from a magnetic resonance angiography (MRA) image of the brain, in Figure 16. The blood vessels have complex topology with loops (due to pathologies) and are already quite thin in several places. The bottom row illustrates the accuracy of the method, where the medial surfaces are

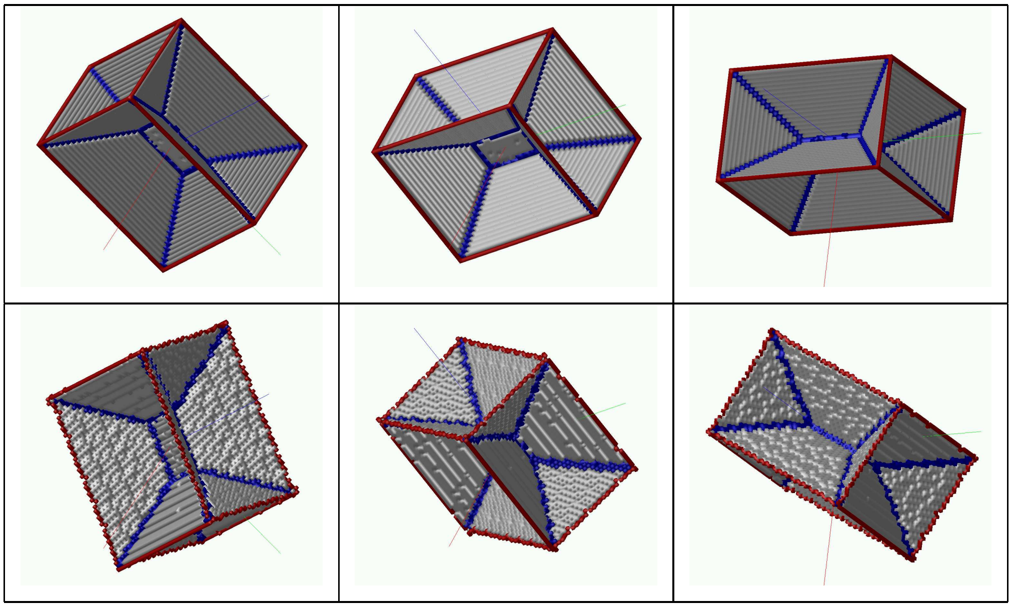

Figure 13: Top Row: The medial surface of the cube in Figure 9. Bottom Row: The cube is rotated by 30 degrees around the z axis and its medial surface is recomputed. The viewing directions are the same for the top and bottom rows. The two medial surfaces have also been automatically segmented into surface points (grey), junction points (blue) and border points (red) using the classification of [33].

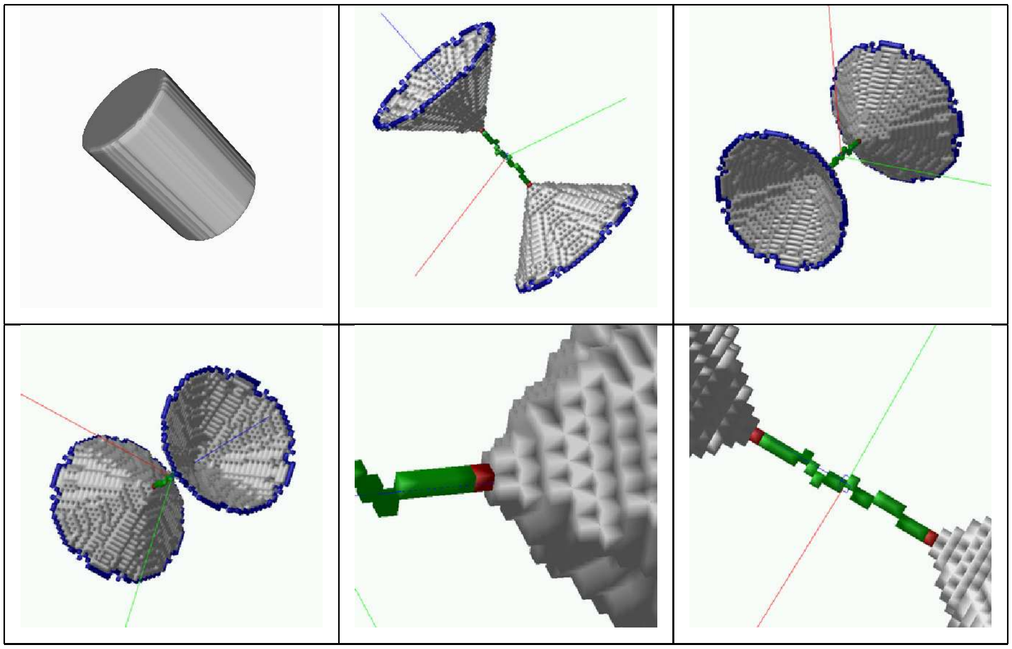

Figure 14: The medial surface of a cylinder is labeled into border points (blue), surface points (grey), curve points (green) and junction points (red)

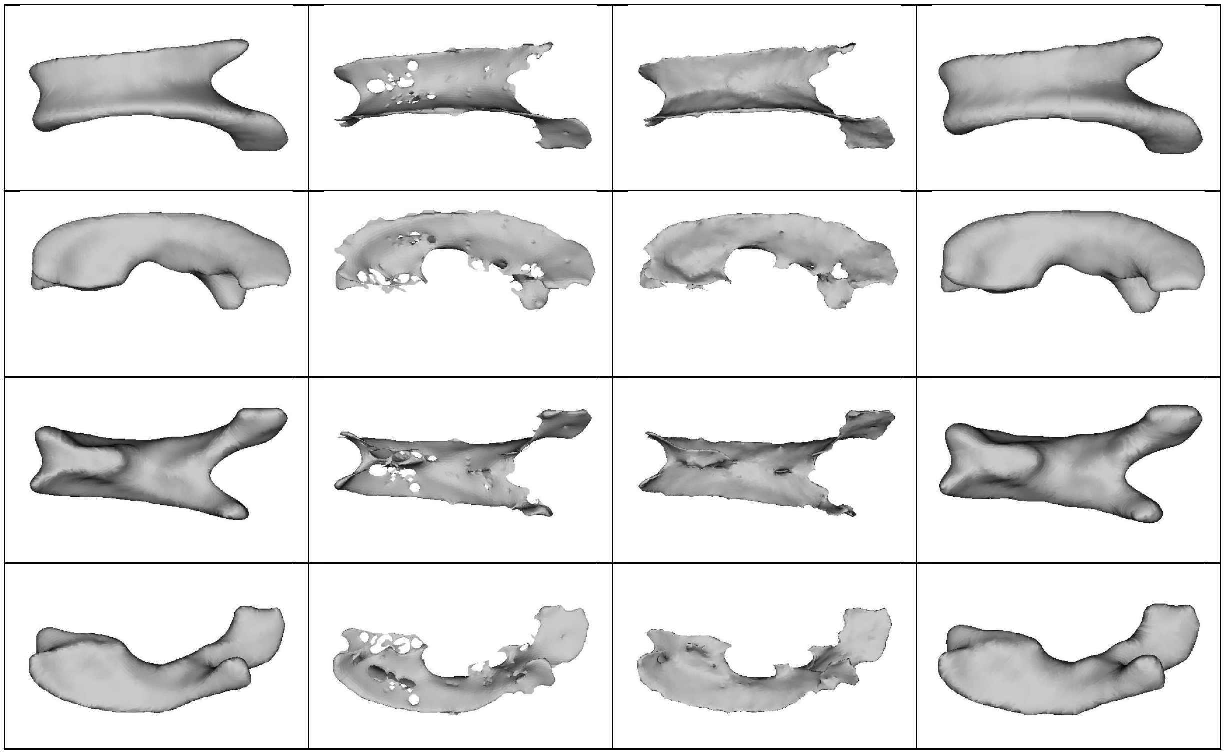

Figure 15: First Column: Four views of the ventricles of a brain, segmented from volumetric MR data using an active surface. Second Column: The corresponding medial surfaces obtained by thresholding the flux map. Third Column: The flux-based medial surfaces obtained using the same threshold, but with the incorporation of homotopy preserving thinning. Fourth Column: The ventricles reconstructed from the flux-based medial surfaces in the previous column.

shown embedded within the original data set. Generically these structures are thin sheets which approach 3D curves when the blood vessels become perfectly cylindrical. In a number of medical applications where the objects are tubular structures, an explicit reduction of the medial surface to a set of 3D curves is of interest [17, 7, 8, 65]. There is a straightforward modification of our framework which allows this. The essential idea is to modify the end point criteria such that only end points of 3D curves are preserved. Rim and corner points of surfaces are now considered to be removable points during the thinning process, resulting in a medial surface consisting only of 3D curves. This is illustrated for a portion of the vessel data in Figure 17, which gives three 1 voxel wide 26-connected 3D digital curves.

Finally, Figure 18 illustrates the 3D medial surface of the sulci of a brain. Rather than show the entire surface, which is difficult to visualize, we have shown an X, Y and Z slice

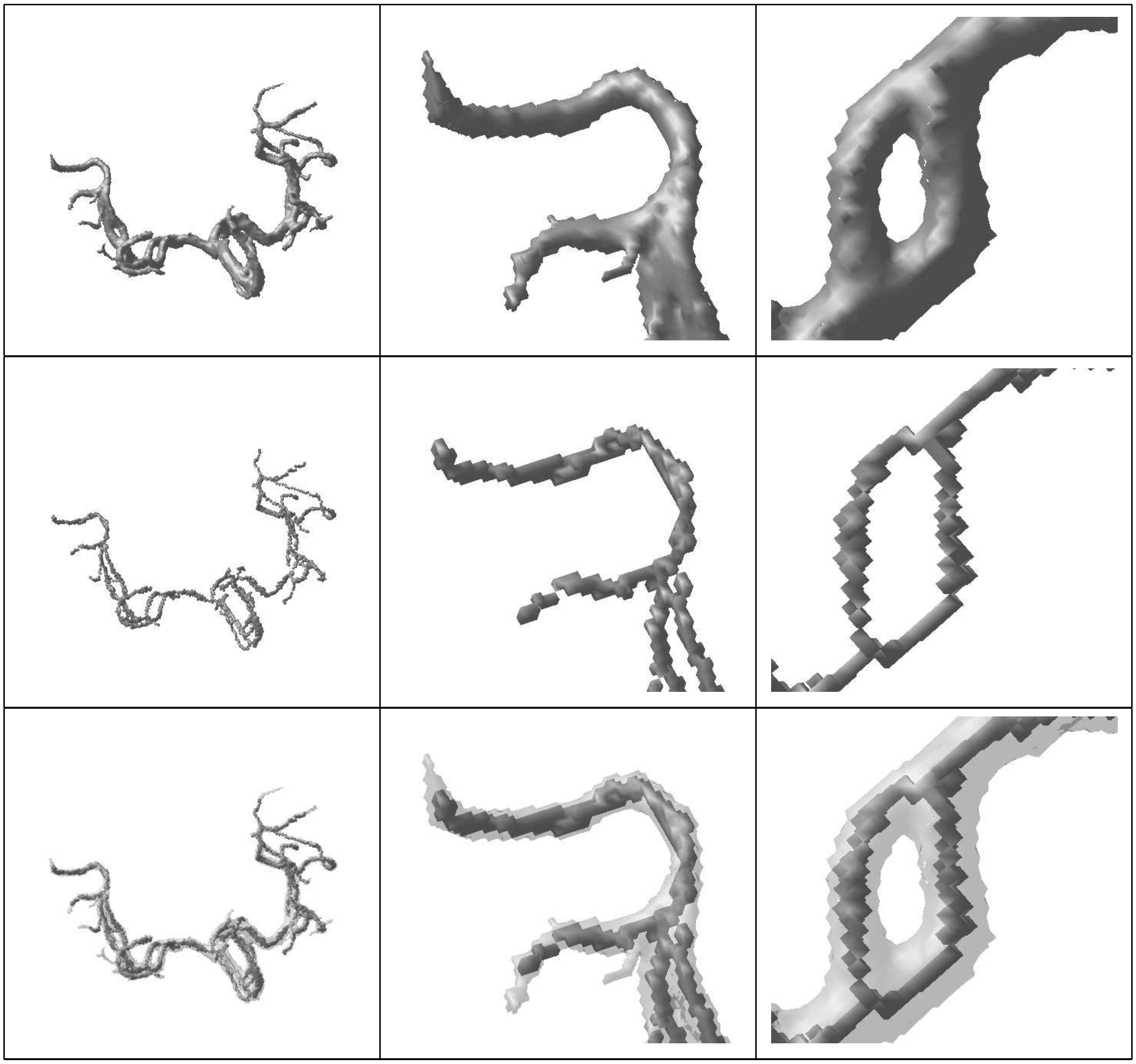

Figure 16: Top Row: Blood vessels segmented from volumetric MRA data with magnified portions shown in the middle and right columns. Middle Row: The corresponding flux-based medial surfaces. Third Row: The flux-based medial surfaces (solid) are shown within the vessel surfaces (transparent).

through the volume in grey, with the intersection of that slice and the medial surface shown in black. The medial surface is well localized and captures the complex topology of the object's shape. The computation times for the 3D examples running on a dual processor 550 MHz Pentium III machine are shown in Table 2. As predicted by the complexity analysis in Section 5.4, the computations of the distance transform and the divergence map are linear in the size n of the 3D array, while the thinning procedure has an $\mathcal{O}(klog(k))$ dependence

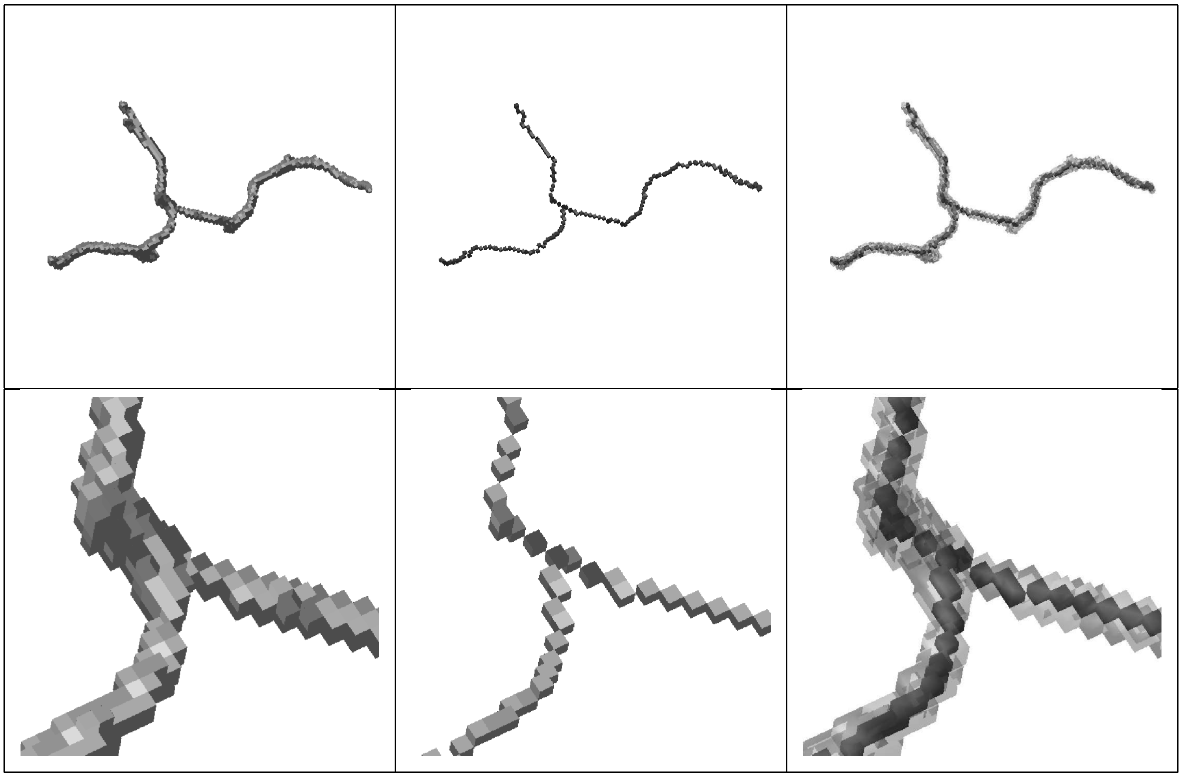

Figure 17: Left Column: Blood vessels segmented from volumetric MRA data, with a magnified portion shown in the second row. Middle Column: The flux-based 3D curves. Right Column: The flux-based 3D curves are shown embedded within the vessel data.

on k, the number of points contained within the volume.

7 Conclusions

In this paper we have applied a Hamiltonian formalism to the eikonal equation, which offers conceptual advantages when it comes to shock detection. The calculation shows that when applied to Blum's grassfire flow, the gradient vector field $\dot{\mathbf{q}}$ of the signed Euclidean distance function to the object's boundary drives the motion of points on the bounding curve (2D) or surface (3D). A measure of the average outward flux of this vector field can be used to distinguish medial points from non-medial ones. In the limit as the region about which this flux is computed shrinks to a point, the measure tends to zero for non-medial points but to a negative number below a constant factor times $\dot{\mathbf{q}} \cdot \mathcal{N}'$ for medial points, where $\mathcal{N}'$ is the one-sided normal to the medial axis or surface. We have combined this measure

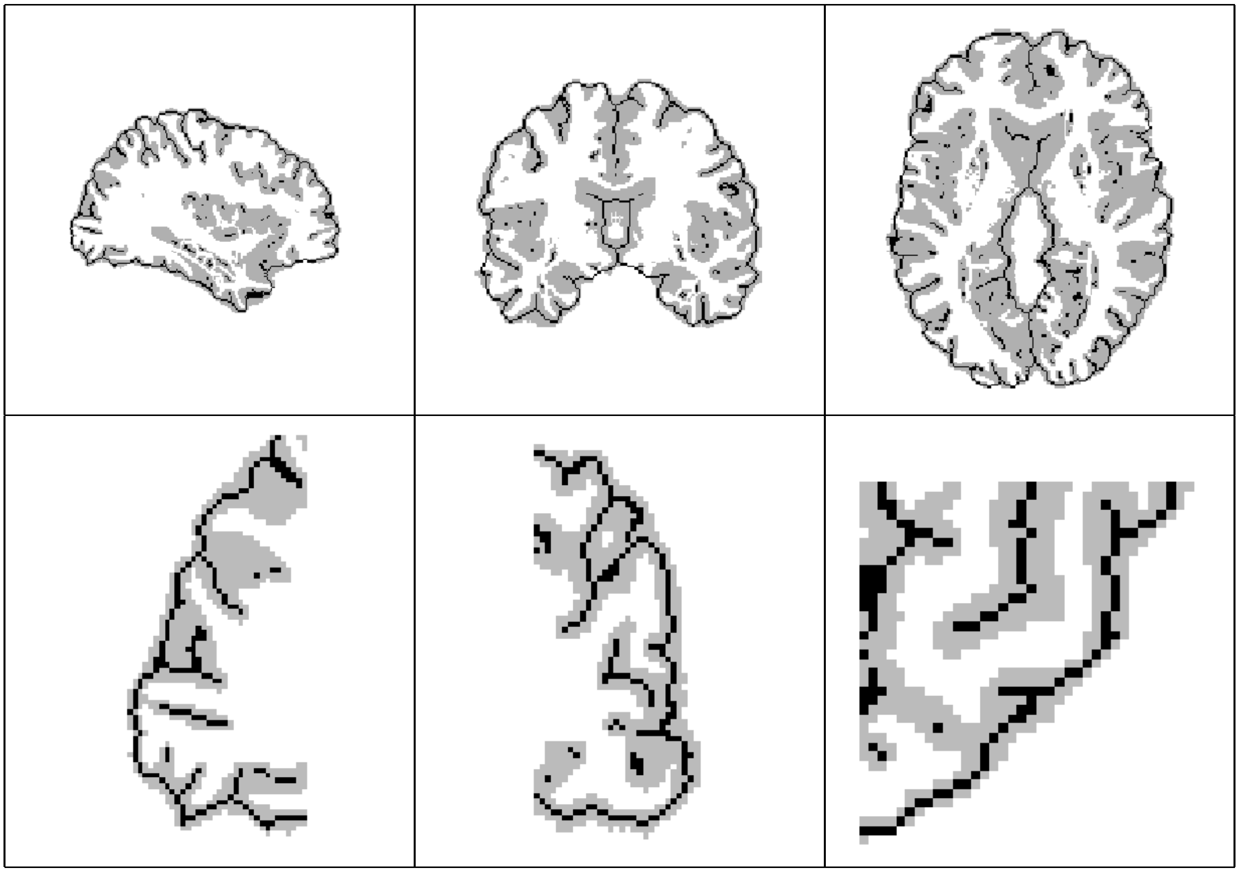

Figure 18: Top Row: Medial surfaces of the sulci of a brain, segmented from an MR image. The three columns represent X, Y and Z slices through the volume, shown in grey. The cross section through the 3D medial surface in each slice is shown in black. Bottom Row: A zoom-in on a selected region of the corresponding slice in the top row, to show detail.

with a homotopy preserving thinning process on a discrete lattice to develop an algorithm which is computationally efficient and yields skeletons that are homotopic to the original objects and thin in 2D as well as in 3D. Whereas in theory the average outward flux is desired only in the limit as the region shrinks to a point, our experiments show that the average outward flux computed over a very small neighborhood (a circle in 2D or a sphere in 3D) provides a sufficient approximation to the limiting values. This, being an integral formulation, is robust to boundary perturbations and digital rotations, as demonstrated by our numerical experiments. We have also illustrated that digital segmentations of medial axes and surfaces such as those proposed in [33], can be readily incorporated. Thus, the approach has a number of advantages over alternative methods in the literature. Finally, it

Data Set	Array Size (n)	Number of Points (k)	DT	DIV	THIN	TOTAL
Cube	128x128x128	120000	26.21s	6.18s	33.50s	65.89s
Cylinder	128x128x128	26260	26.40s	6.18s	8.15s	40.73s
Ventricles	192X169x99	30909	39.60s	9.44s	10.17s	59.21s
Vessels	63x151x164	14377	14.07s	3.31s	3.74s	21.12s
Sulci	192x169x99	798221	38.71s	10.54s	240.85s	290.1s

Table 2: The computation times for the 3D examples, running on a dual processor 550 MHz Pentium III machine. The times taken in seconds to compute the distance transform (DT), the divergence map (DIV) and to carry out the thinning (THIN), are each shown separately. The results are consistent with the complexity analysis in Section 5.4 .

should be clear that whereas we have focused on the interior of an object, the skeleton of the background can be similarly obtained by locating points with high positive average outward flux.

Acknowledgements This work was supported by the Natural Sciences and Engineering Research Council of Canada, FCAR Québec, the Canadian Foundation for Innovation, the National Science Foundation, the Air Force Office of Scientific Research, the Army Research Office and by a Multi University Research Initiative grant. We are grateful to Pavel Dimitrov and Carlos Phillips for help with the 2D simulations. Louis Collins, Georges Le Goualher, Belinda Lee and Terry Peters kindly supplied the medical data. Finally, we are particularly grateful to Jim Damon of the Department of Mathematics at the University of North Carolina, Chapel Hill with whom we have had many useful discussions. He pointed us to an alternate form of the divergence theorem that could be applied in regions containing medial points. He used this form to show that for medial points the average outward flux measure tends to a negative number in the limit as the region is shrunk to the point under consideration. This calculation lends our algorithm a very strong theoretical motivation.

References

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

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

[3] V. I. Arnold. *Mathematical Methods of Classical Mechanics, Second Edition.* Springer-Verlag, 1989.

[4] G. Bertrand. A parallel thinning algorithm for medial surfaces. *Pattern Recognition Letters*, 16:979–986, 1995.

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

[6] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision Graphics and Image Processing*, 27:321–345, 1984.

[7] G. Borgefors, I. Nystrom, and G. S. D. Baja. Skeletonizing volume objects part ii: From surface to curve skeleton. In *International Workshop on Syntatical and Structural Pattern Recognition*, pages 220–229, 1998.

[8] G. Borgefors, I. Nystrom, and G. S. D. Baja. Computing skeletons in three dimensions. *Pattern Recognition*, 32:1225–1236, 1999.

[9] R. Brockett and P. Maragos. Evolution equations for continuous-scale morphology. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, San Francisco, CA, March 1992.

[10] A. R. Bruss. The eikonal equation: Some results applicable to computer vision. In B. K. P. Horn and M. J. Brooks, editors, *Shape From Shading*, pages 69–87, Cambridge, MA, 1989. MIT Press.

[11] V. Caselles, J.-M. Morel, G. Sapiro, and A. Tannenbaum, editors. *IEEE Transactions on Image Processing, Special Issue on PDEs and and Geometry-Driven Diffusion in Image Processing and Analysis*, 1998.

[12] M. G. Crandall, H. Ishii, and P.-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27(1):1–67, 1992.

[13] A. D. J. Cross and E. R. Hancock. Scale-space vector fields for feature analysis. In *Conference on Computer Vision and Pattern Recognition*, pages 738–743, June 1997.

[14] P. Dimitrov, C. Phillips, and K. Siddiqi. Robust and efficient skeletal graphs. In *Conference on Computer Vision and Pattern Recognition*, pages 417–423, Hilton Head, South Carolina, June 2000.

[15] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *European Conference on Computer Vision*, volume 1, pages 379–393, 1998.

[16] J. D. Furst and S. M. Pizer. Marching optimal parameter ridges: An algorithm to extract shape loci in 3d images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 780–787, 1998.

[17] Y. Ge, D. R. Stelts, X. Zha, D. Vining, and J. Wang. Computing the central path of the colon from ct images. In *SPIE International Symposium on Medical Imaging*, volume 3338(1), pages 702–713, San Francisco, 1998.

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

[19] J. Gomez and O. Faugeras. Level sets and distance functions. In *European Conference on Computer Vision*, volume 1, pages 588–602, Dublin, Ireland, June 2000.

[20] B. K. P. Horn and M. J. Brooks, editors. *Shape From Shading.* MIT Press, Cambridge, MA, 1989.

[21] S. N. Kalitzin, B. M. ter Haar Romeny, A. H. Salden, P. F. M. Nacken, and M. A. Viergever. Topological numbers and singularities in scalar images. *Journal of Mathematical Imaging and Vision*, 9(3):253–269, 1998.

[22] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. On optimal control methods in computer vision and image processing. In B. ter Haar Romeny, editor, *Geometry-Driven Diffusion In Computer Vision*, pages 307–338. Kluwer, 1994.

[23] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. Shape, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1995.

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

[25] R. Kimmel, K. Siddiqi, B. B. Kimia, and A. Bruckstein. Shape from shading: Level set propagation and viscosity solutions. *International Journal of Computer Vision*, 16(2):107–133, 1995.

[26] T. Y. Kong and A. Rosenfeld. Digital topology: Introduction and survey. *Computer Vision Graphics and Image Processing*, 48(3):357–393, December 1989.

[27] C. Lanczos. *The Variational Principles of Mechanics.* Dover, 1986.

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

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

[30] A. Liu, E. Bullitt, and S. M. Pizer. 3d/2d registration via skeletal near projective invariance in tubular objects. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 952–963, 1998.

[31] T.-L. Liu and D. Geiger. Approximate tree matching and shape similarity. In *International Conference on Computer Vision*, pages 456–461, Kerkyra, Greece, September 1999.

[32] T.-L. Liu, D. Geiger, and R. V. Kohn. Representation and self-similarity of shapes. In *International Conference on Computer Vision*, 1998.

[33] G. Malandain, G. Bertrand, and N. Ayache. Topological segmentation of discrete surfaces. *International Journal of Computer Vision*, 10(2):183–197, 1993.

[34] G. Malandain and S. Fernandez-Vidal. Euclidean skeletons. *Image and Vision Computing*, 16:317–327, 1998.

[35] A. Manzanera, T. M. Bernard, F. Preteux, and B. Longuet. Medial faces from a concise 3d thinning algorithm. In *International Conference on Computer Vision*, pages 337–343, Kerkyra, Greece, September 1999.

[36] D. Miller and S. W. Zucker. Computing with self-excitatory cliques. *Neural Computation*, 11(1):21–66, 1999.

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

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

[39] J. Oliensis and P. Dupuis. An optimal control formulation and related numerical methods for a problem in shape construction. *Annals of Applied Probability*, 4(2):287–346, 1994.

[40] S. Osher and C.-W. Shu. High-order essentially non-oscillatory schemes for Hamilton-Jacobi equations. *SIAM Journal of Numerical Analysis*, 28:907–922, 1991.

[41] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[42] S. M. Pizer, A. Thall, and D. T. Chen. M-reps: A new object representation for graphics. *ACM Transactions on Graphics*, submitted, 1999.

[43] C. Pudney. Distance-ordered homotopic thinning: A skeletonization algorithm for 3d digital images. *Computer Vision and Image Understanding*, 72(3):404–413, 1998.

[44] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal of Numerical Analysis*, 29(3):867–884, June 1992.

[45] G. Sapiro, B. B. Kimia, R. Kimmel, D. Shaked, and A. Bruckstein. Implementing continuous-scale morphology. *Pattern Recognition*, 26(9), 1992.

[46] M. Schmitt. Some examples of algorithms analysis in computational geometry by means of mathematical morphology techniques. In J. Boissonnat and J. Laumond, editors, *Lecture Notes in Computer Science, Geometry and Robotics*, volume 391, pages 225–246. Springer-Verlag, 1989.

[47] T. B. Sebastian, H. Tek, J. J. Crisco, S. W. Wolfe, and B. B. Kimia. Segmentation of carpal bones from 3d ct images using skeletally coupled deformable models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1184–1194, 1998.

[48] J. Sethian. *Level Set Methods: evolving interfaces in geometry, fluid mechanics, computer vision, and materials science.* Cambridge University Press, Cambridge, 1996.

[49] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences, USA*, 93:1591–1595, February 1996.

[50] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Conference on Computer Vision and Pattern Recognition*, pages 136–142, June 1996.

[51] R. Shankar. *Principles of Quantum Mechanics.* Plenum Press, 1994.

[52] D. Sharvit, J. Chan, H. Tek, and B. B. Kimia. Symmetry-based indexing of image databases. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, June 1998.

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

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

[55] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. The hamilton-jacobi skeleton. In *International Conference on Computer Vision*, pages 828–834, Kerkyra, Greece, September 1999.

[56] K. Siddiqi, B. B. Kimia, and C. Shu. Geometric shock-capturing eno schemes for subpixel interpolation, computation and curve evolution. *Graphical Models and Image Processing*, 59(5):278–301, September 1997.

[57] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.

[58] G. D. Stetten and S. M. Pizer. Automated identification and measurement of objects via populations of medial primitives, with application to real time 3d echocardiography. In *International Conference on Information Processing in Medical Imaging*, pages 84–97, 1999.

[59] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–154, 1994.

[60] S. Tari and J. Shah. Local symmetries of shapes in arbitrary dimension. In *International Conference on Computer Vision*, Bombay, India, January 1998.

[61] Z. S. G. Tari, J. Shah, and H. Pien. Extraction of shape skeletons from grayscale images. *Computer Vision and Image Understanding*, 66:133–146, May 1997.

[62] M. Teichmann and S. Teller. Assisted articulation of closed polygonal models. In *9th Eurographics Workshop on Animation and Simulation*, 1998.

[63] H. Tek and B. B. Kimia. Symmetry maps of free-form curve segments via wave propagation. In *International Conference on Computer Vision*, pages 362–369, Kerkyra, Greece, September 1999.

[64] R. van den Boomgaard and A. Smeulders. The morphological structure of images: The differential equations of morphological scale-space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1101–1113, November 1994.

[65] Y. Zhou, A. Kaufman, and A. W. Toga. 3d skeleton and centerline generation based on an approximate minimum distance field. *International Journal of the Visual Computer*, 14(7):303–314, 1998.

[66] S. Zhu and A. L. Yuille. Forms: a flexible object recognition and modeling system. *International Journal of Computer Vision*, 20(3):187–212, 1996.