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 Yaven, CT 06520-8285, USA

email: zucker-steven@cs.yale.edu

*This article is based on papers presented at the International Conference on Computer Vision, Kerkyra,

Preferred Address for Correspondence:

Kaleem Siddiqi

McGill University

School of Computer Science & Center for Intelligent Machines

3480 University Street

Montréal, QC H3A 2A7, Canada

phone: (514) 398 3371

fax: (514) 398 7348

Greece (ICCV'1999), and the European Conference on Computer Vision, Dublin, Ireland (ECCV'2000).

Keywords: Eikonal Equation, Hamiltonian Systems, 2D and 3D Skeletons, Shape Analysis.

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. Although these have received some consideration in the computer vision literature, they have yet to be widely used. In this paper we first introduce a new algorithm for simulating the eikonal equation, which offers a number of computational and conceptual advantages over the earlier methods when it comes to shock tracking. Next, we introduce a procedure for shock detection where the key idea is to measure the net outward flux of a vector field per unit volume, and to detect locations where a conservation of energy principle is violated. We specialize to the case of Blum's grassfire flow, and 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 [26]. We consider these in the context of the eikonal equation, which arises in geometrical optics and, recently, which has become of great interest for problems in computer vision [10, 23]. Indeed, it is the basis for continuous versions of mathematical morphology [9, 45, 62], as well as for Blum's grassfire transform [5], and dynamic theories of shape representation including [24, 59]. It has also been used for applications in image processing and analysis [48, 11], shape-from-shading [21, 44, 38, 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 Figures 3 and 4). To continue the evolution while preserving shocks, the technology of level set methods introduced by Osher and Sethian [40], 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 [40], 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 [39, 55].

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 [21, 44, 38], the numerical methods have yet to be widely used. The purpose of this paper is twofold. First, we introduce a straightforward algorithm for simulating the eikonal equation, which is based on Hamilton-Jacobi theory and offers a number of computational and conceptual advantages when it comes to shock tracking. The proposed algorithm is Lagrangian in nature, i.e., the front is explicitly represented as a sequence of marker particles. The motion of these particles is then governed by an underlying Hamiltonian system. Such 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, e.g., see [35]. Second, we introduce a procedure for shock detection, based on measuring the net outward flux per unit volume of the vector field underlying the Hamiltonian system. For the special case of Blum's grassfire flow [5] we combine the flux measurement 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 Oliensis and Dupuis' formulation of the shape-from-shading problem [38]. 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 begin by reviewing some relevant background on skeletons. We shall then focus on the eikonal equation and develop its Hamilton-Jacobi formulation.

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 [36, 57], registration [29] and segmentation [47]. They have also been used for graph-based object recognition in computer vision [37, 64, 51, 30, 56], for animating objects in graphics [60, 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, 27, 8, 34]. 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 [34, 4, 27].

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 [37], as well as extensions to 3D [52, 53]. 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) [19, 36]. Both types of methods preserve topology 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 [36].

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, 28, 20]. 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 promising recent approaches to computing 2D and 3D skeletons which combine aspects of thinning, Voronoi diagrams and distance functions [33, 63, 8, 61]. In spirit, our method is close to that of Malandain and Fernandez-Vidal who use a heuristic estimation of the singularities of a distance function to obtain an initial skeleton or medial surface, and then perform a topological reconstruction to ensure homotopy with the original shape [33]. It is also related to the approach of Pudney [43] who employs a distance-ordered thinning process, where points are removed in order of decreasing distance from the boundary while preserving end points as well as centers of maximal balls identified from a chamfer distance function. However, in contrast to both these approaches, our work is grounded in principles from physics, and the analysis leads to a clear notion of how to localize the skeleton based on a conservation of energy principle. We now turn to the eikonal equation and illustrate the connection to Blum's grassfire flow.

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

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

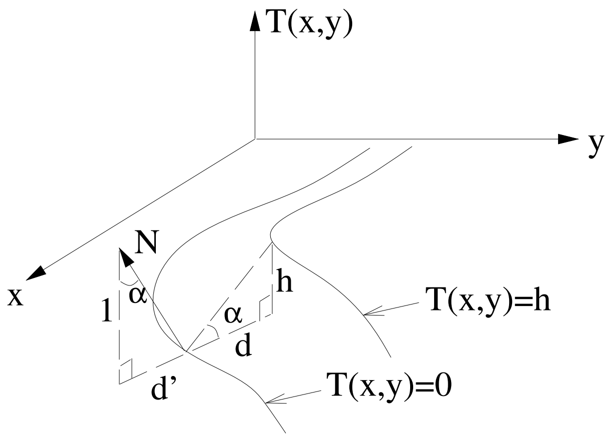

Figure 1: A geometric view of a monotonically advancing front (Eq. 1). $T(x, y)$ is a graph of the 'solution' surface, the level sets of which are the evolved curves.

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 systemically constructs T using only upwind values, and Rouy and Tourin's viscosity solutions approach [44]. However, neither 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, 59]. 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 [24], 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 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

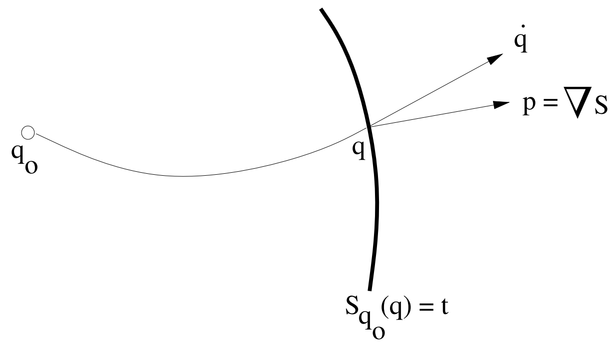

Figure 2: Direction of a ray $\dot{\mathbf{q}}$ and the direction of motion of the wave front $\mathbf{p}$. From [3].

used in classical mechanics, and rests on the use of a Legendre transformation (see [3] for the precise definition) which takes a system of n second-order differential equations to a (mathematically equivalent) system of $2n$ first-order differential equations. We believe that for a number of vision problems involving shock tracking and skeletonization, this represents a natural way of implementing the eikonal equation.

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

Following Arnold [3, pp. 248–258], we shall use Huygens' principle to show the connection between the eikonal equation and the Hamilton-Jacobi equation. For every point $\mathbf{q_0}$, define the function $S_{\mathbf{q}_0}(\mathbf{q})$ as the optical length of the path from $\mathbf{q_0}$ to $\mathbf{q}$ (see Figure 2). The wave front 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., $\mathbf{p} \cdot \dot{\mathbf{q}} = 1$. Note that these directions do not coincide in an anisotropic medium.

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 *action function* minimized, $S(\mathbf{q}, t)$, is defined as

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

along the extremal curve γ connecting the points $(\mathbf{q}_0, t_0)$ and $(\mathbf{q}, t)$. Here the Lagrangian

$$L = \frac{1}{F(x,y)} \, ||\partial\gamma/\partial t||$$

is 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.* Note that for an isotropic medium the extremals are straight lines, and that 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

$$\frac{\partial S}{\partial t} = -H\left(\frac{\partial S}{\partial \mathbf{q}}, \mathbf{q}\right), \tag{3}$$

where the Hamiltonian function $H(\mathbf{p}, \mathbf{q})$ is the Legendre transformation with respect to $\dot{\mathbf{q}}$ of the Lagrangian function $L(\mathbf{q}, \dot{\mathbf{q}})$. 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 $(\mathbf{p}, \mathbf{q})$ under the equivalent Hamiltonian system

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

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, 24].

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 $\mathbf{q} = (x, y)$, $\mathbf{p} = (S_x, S_y)$, 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 a Legendre transformation to the Lagrangian $L = ||\dot{\mathbf{q}}||$ is given by

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

The associated Hamiltonian system is:

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}} = (0,0), \qquad \dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}} = -(S_x, S_y). \tag{4}$$

$\tilde{C}$ 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, $\mathcal{C}(t)$.

4 Numerical Simulations

We now apply the above theory to formulate an efficient algorithm for simulating the eikonal equation, while tracking the shocks which form. Recall that since the approach is a Lagrangian one, marker particles will have to first be placed along the initial curve, which in our simulations is assumed to be a simple closed curve in the plane.[1] The evolution of marker particles is then governed by Eq. 4. With $\mathbf{q} = (x,y)$, $\mathbf{p} = (S_x, S_y) = \nabla S$, the system of equations

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

gives 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 $\mathcal{C}(t)$.

In order to obtain accurate results, three numerical issues need to be addressed. First, in order to obtain a dense sequence of marker particles, a continuous representation of the initial shape's boundary ($T(x,y) = 0$, see Figure 1) is needed. Second, it is possible for marker particles to drift apart in rarefaction regions, i.e., concave portions of the curve may fan out. Hence, new marker particles must be interpolated when necessary. Third, whereas

[1] The method would also extend naturally to open curves, where an outward distance function would have to be defined.

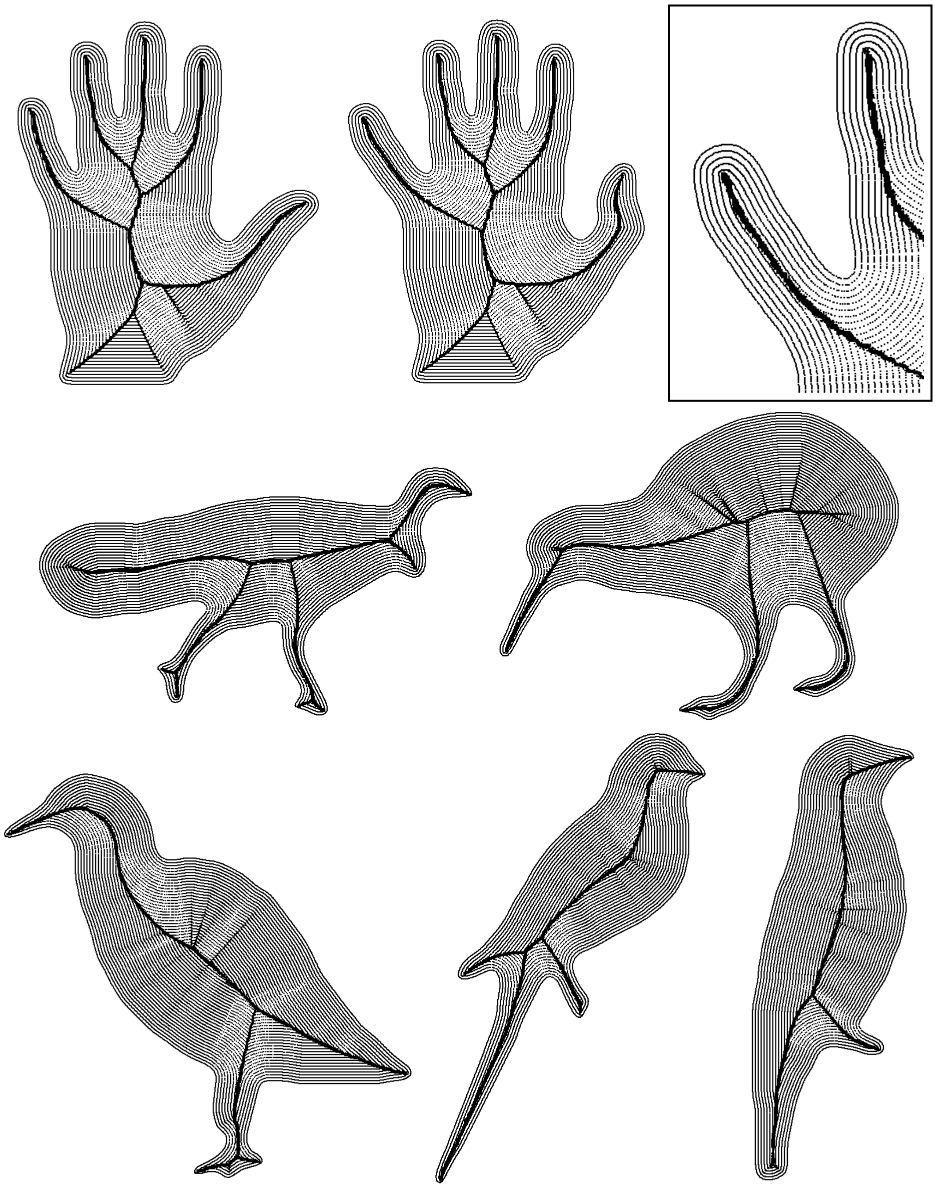

Figure 3: The evolution of marker particles under the Hamiltonian system. The initial particles are placed on the boundary, and iterations of the process are superimposed. These correspond to level sets of the solution surface $T(x,y)$ in Figure 1. Individual marker particles are more clearly visible in the zoom-in on the fingers of the hand (top right). See Section 4 for a discussion.

finite central differences are adequate for estimating the gradient of the Euclidean distance function in its smooth regime, such estimates will lead to errors near singularities, where S is not differentiable. Hence, we use ENO interpolants for estimating derivatives [39]; the key idea is to obtain information from the "smooth" side, in the vicinity of a singularity. The procedure is described in Algorithm 1.

Algorithm 1 The Hamilton-Jacobi Skeleton Flow Algorithm.

Take as the initial curve $T(x(s), y(s)) = 0$, the given boundary of an object, assumed to be a simple closed curve in the plane.

Create an ordered sequence of marker particles at positions Δs apart along the boundary.

Compute a Euclidean distance transform, where each grid point in the interior of the boundary is assigned its Euclidean distance to the closest marker particle.

For each grid point in the interior of the boundary compute and store the components of the vector field ∇S, using ENO interpolants.

Do for **step** from **0** to **TOTALSTEPS** {

  Do for **particle** from **0** to **NPARTICLES** {

- Update the particle's position based on
  ∇S at the closest grid point:
  $x(step + 1) = x(step) - \Delta t \times S_x$,
  $y(step + 1) = y(step) - \Delta t \times S_y$
- **if** (Distance(particle,next_particle) $> a\Delta s$){
  interpolate a new particle in between.

  }

 }

}

In our experiments, we have used a piecewise circular arc representation of the boundary, obtained using the contour tracer developed in [55], on the signed distance transform of the original binary shape. The distance transform is blurred very slightly to combat discretization. The birth of new marker particles (step 5) is also based on circular arc interpolation. Figures 3 and 4 depict the evolution of marker particles, with speed $F = 1$, for several different shapes. For all simulations, the spacing Δs of initial marker particles is 0.25 pixels, the spacing criterion for interpolating a new particle in the course of the evolution is $a\Delta s = 0.75$ pixels, and the resolution of the Euclidean distance transform S is the same as that of the original binary image. The timestep Δt is 0.5 pixels, and results for every second iteration are saved. The superposition of all the level curves gives the solution surface $T(x, y)$ in

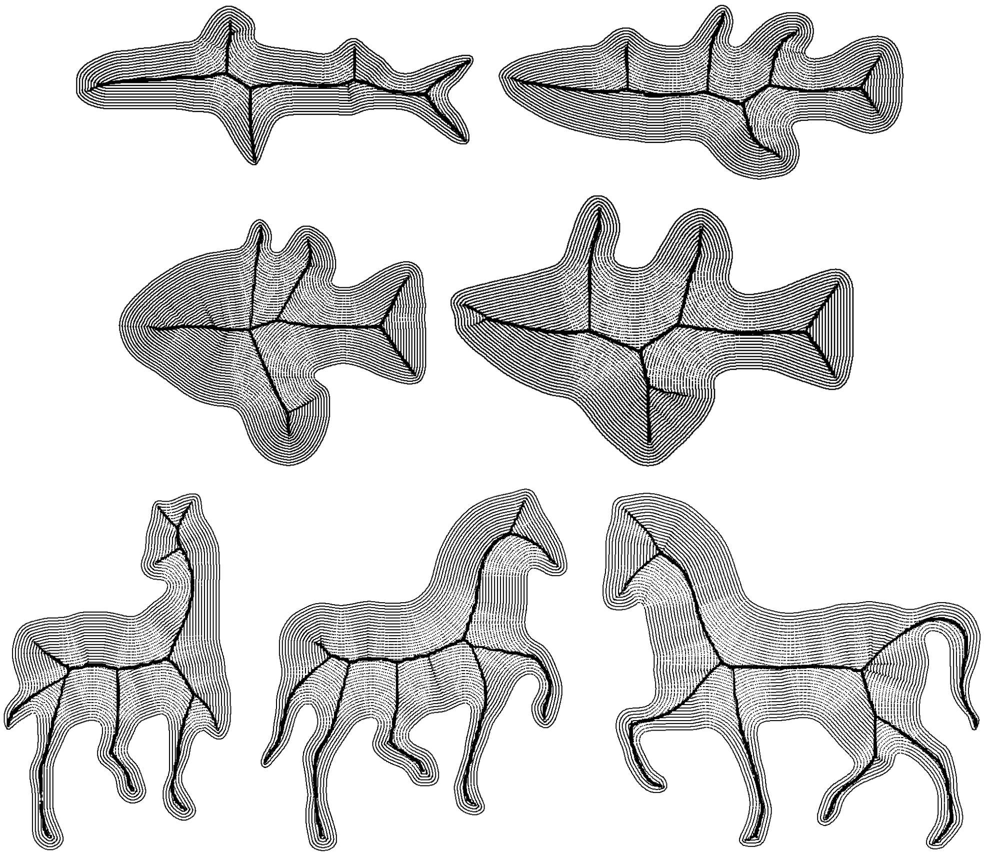

Figure 4: The evolution of marker particles under the Hamiltonian system. The initial particles are placed on the boundary, and iterations of the process are superimposed. Each iteration gives a level set of the solution surface $T(x, y)$ in Figure 1. See Section 4 for a discussion.

Figure 1. It is important to note that in principle higher order interpolants can be used for the placement of marker particles, and the resolution of the exact distance transform is not limited by that of the original (binary) shape.

The results are comparable to those obtained using higher order ENO implementations (see Figure 5), although the algorithm is computationally more efficient (linear in the number of marker particles). Informal timing experiments indicate that the efficiency of the algorithm exceeds that of level set methods, except under the "fast marching" implementation, with which it compares favorably. However, when shock detection is included, the Hamiltonian approach has important conceptual and computational advantages. In particular, in contrast with level set approaches, the formation of shocks is made explicit (compare Figures 3 and 4 with Figure 5). We shall now show how to make the calculation of the locus of shocks precise

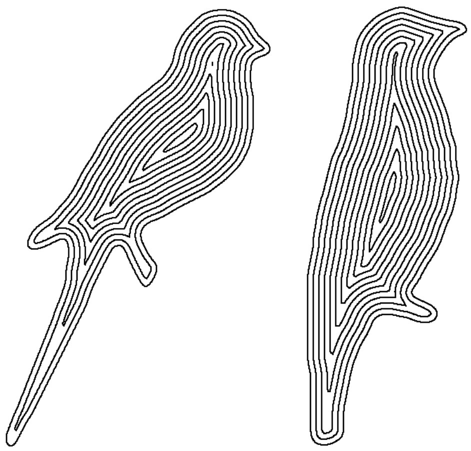

Figure 5: A level set implementation of the eikonal equation using a high order ENO scheme [48, 39]. The results are comparable to those in Figure 3, although the shock traces are absent.

by exploiting a conservation of energy principle.

5 Flux and Divergence

It is straightforward to show that a Hamiltonian system is conservative [41, p. 172]. More specifically:

Theorem 1 *The total energy* $H(\mathbf{p}, \mathbf{q})$ *of the Hamiltonian system (4) remains constant along trajectories of (4).*

Proof. The total derivative of $H(\mathbf{p}, \mathbf{q})$ along a trajectory $\mathbf{p}(t)$, $\mathbf{q}(t)$ of (4) is given by

$$\frac{dH}{dt} = \frac{\partial H}{\partial \mathbf{p}}.\dot{\mathbf{p}} + \frac{\partial H}{\partial \mathbf{q}}.\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}}.\frac{\partial H}{\partial \mathbf{q}} - \frac{\partial H}{\partial \mathbf{p}}.\frac{\partial H}{\partial \mathbf{q}} = 0.$$

Thus $H(\mathbf{p}, \mathbf{q})$ is constant along any trajectory of (4).

The Hamiltonian formulation applies under the assumption of a central field of extremals, see Section 3, such that trajectories of the Hamiltonian system do not intersect. Conversely, when shocks form due to the intersection of trajectories, the conservation of energy principle will be violated (energy will be absorbed). This loss of energy can be used to formulate a robust and very efficient algorithm for shock detection, based on an application of the divergence theorem.

The key is to measure the flux of the vector field $\dot{\mathbf{q}}$, which is analogous to the flow of an incompressible fluid such as water. Note that for an area within a closed contour, an excess of outward or inward flow through the contour indicates the presence of a *source*, or a *sink*, respectively. The latter case is the one we are interested in, and the net outward flux is related to the divergence of the vector field. More specifically, the divergence of a vector field at a point, $\mathrm{div}(\dot{\mathbf{q}})$, is defined as the net outward flux per unit area, as the area about the point shrinks to zero:

$$\mathrm{div}(\dot{\mathbf{q}}) \equiv \lim_{\Delta \mathrm{a} \to 0} \frac{\int_{\mathrm{S}} < \dot{\mathbf{q}}, \mathcal{N} >}{\Delta \mathrm{a}} \mathrm{dl} \tag{5}$$

Here Δa is the area, L is its bounding contour and $\mathcal{N}$ is the outward normal at each point on the contour. This definition can be shown to be equivalent to the more common definition of the divergence as the sum of the partial derivatives with respect to each of the vector field's component directions:

$$\mathrm{div}(\dot{\mathbf{q}}) = \frac{\partial \mathrm{q}_{\mathrm{x}_1}}{\partial \mathrm{x}_1} + \ldots + \frac{\partial \mathrm{q}_{\mathrm{x}_\mathrm{n}}}{\partial \mathrm{x}_\mathrm{n}} \tag{6}$$

However, Eq. 6 cannot be used at points where the vector field is singular, and hence is not differentiable. These are precisely the points we are interested in, and Eq. 5 offers significant advantages for shock detection. In particular, the numerator, which represents the net outward flux of the vector field through the contour which bounds the area, is an index computation on the vector field. As we shall see, this is numerically much more stable than the estimation of derivatives in the vicinity of singularities. Via the divergence theorem,

$$\int_a \mathrm{div}(\dot{\mathbf{q}}) \mathrm{da} \equiv \int_{\mathrm{L}} < \dot{\mathbf{q}}, \mathcal{N} > \mathrm{dl}. \tag{7}$$

Hence, the net outward flux through the contour which bounds a finite area is just the area integral of the divergence of the vector field within that area. Locations where the total outward flux is negative, and hence energy is lost, correspond to sink points or shocks.

It is important to note that 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. (4), and replace the area element with a volume element and the contour integral with a surface integral in Eq. (5) and Eq. (7).

Figure 6 illustrates the divergence-based computation on the silhouette of a panther shape. The gradient vector field of the Euclidean distance function is shown on the top, with the

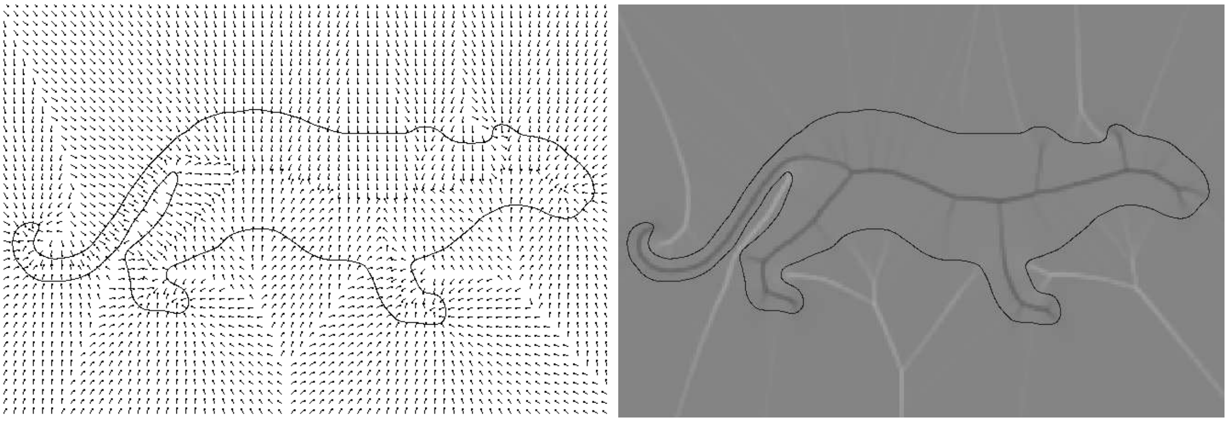

Figure 6: The gradient vector field of a signed distance function to the boundary of a panther shape (left), with the associated total outward flux (right). Whereas the smooth regime of the vector field gives zero flux (medium gray), its sinks correspond to the medial axis of the interior (dark gray) and its sources to the medial axis of the exterior (light gray).

total outward flux on the bottom. Observe that the smooth regime of the vector field gives zero flux (medium gray), its sinks coincide with the skeleton (medial axis) of the interior (black), and its sources with the skeleton of the exterior (light gray). Hence, a threshold on the divergence map yields a close approximation to the skeleton, as used in [54]. 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 7 (left). A low threshold yields a thin set, but it may be disconnected, Figure 7 (right). The solution, as we shall now show, is to introduce additional constraints to ensure that the topology of the shape is preserved. The essential idea is to incorporate a homotopy preserving thinning process, where the removal of points is guided by their divergence values. In the context of the Hamilton-Jacobi skeleton flow (Eq. 4), this leads to a robust and efficient algorithm for computing 2D and 3D skeletons.

6 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

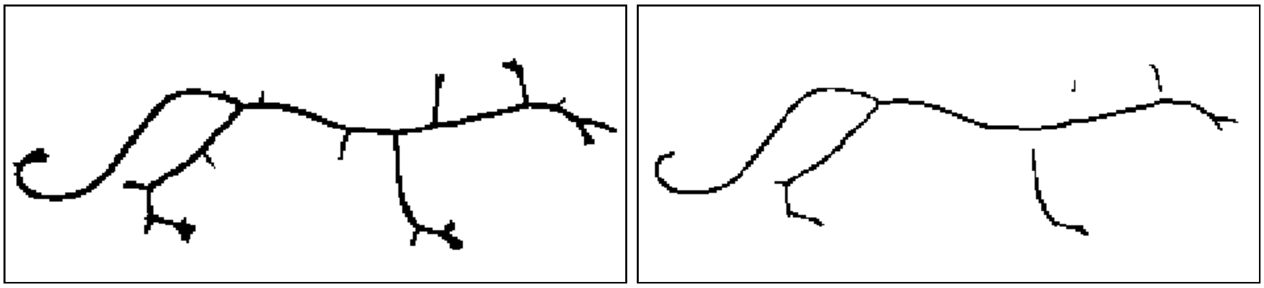

Figure 7: Thresholding the divergence map in Figure 6. A high threshold yields a connected set, but it is not thin, and unwanted branches are present (left). A low threshold yields a closer approximation to the desired medial axis, but the result is now disconnected (right).

in Figure 8 (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.

6.1 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, 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

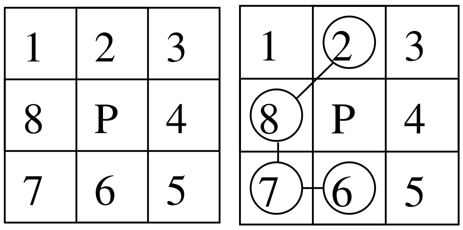

Figure 8: LEFT: A 3x3 neighborhood of a candidate point for removal P. RIGHT: An example neighborhood graph for which P is simple. Note that there is no edge between neighbors 6 and 8 (see text).

to 1. Note that this check only has to be performed locally, in the 3x3 neighborhood of P. Figure 8 (right) shows an example neighborhood graph for which P can be removed.

6.2 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 [32]: 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 2 *(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 ensuring 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, 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 divergence of the gradient vector field of the Euclidean distance function.

6.3 Divergence-Ordered Thinning

Recall from Section 5, that a conservation of energy principle is violated at skeletal points. The total outward flux of the gradient vector field of the Euclidean distance function is negative at such points, since they correspond to sinks.[2] The magnitude of the total outward flux is proportional to the amount of energy absorbed, and hence 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 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. To facilitate this task, we restrict our definition to a 6-connected neighborhood. In other words, an end point is either

[2] Conversely, skeletal points of the background correspond to sources, with positive total outward flux.

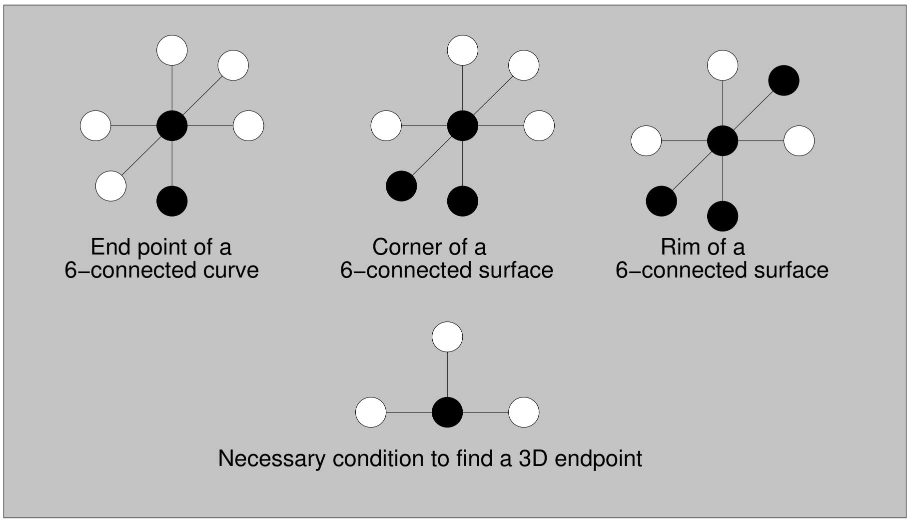

Figure 9: An end point is defined as the end of a 6-connected curve or the corner or rim of a 6-connected surface in 3D. For each configuration, there exists at least one plane in which the point has at least three background 6-neighbors.

the end of a 6-connected curve, or a corner or point on the rim of a 6-connected surface. It is straightforward to enumerate the possible 6-connected neighborhoods of the end point, and to show that they fall into one of three configurations (see Figure 9). Notice that for each configuration, there exists at least one plane in which the point has at least three background 6-neighbors, and that the end point must also be a simple point:

Proposition 3 *A 3D point P could be an end point of a 6-connected curve, or a corner or point on the rim of a 6-connected surface, if it is simple and there exists at least one plane in which the point has at least three background 6-neighbors.*

We note that a slightly different criterion has also been introduced recently for defining end points, where 9 different digital planes are examined in the 26-neighborhood of a candidate end point p [43].

6.4 The Algorithm and its Complexity

The essential idea behind the divergence-ordered thinning process is to remove simple points sequentially, ordered by divergence, 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 2.

We now analyze the complexity of the algorithm. The computation of the distance transform [6], the gradient vector field and the total 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 a $\mathcal{O}(klog(k))$ worst case complexity, where k is the number of points in 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 total outward flux as the sorting key 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 complexity of the algorithm is $\mathcal{O}(n) + \mathcal{O}(klog(k))$.

7 Examples

We first present examples of medial axes, computed for a range of shapes. The same divergence threshold was used in each example, to determine which end points to preserve. The implementation uses an exact distance function to a piecewise circular arc interpolation of the boundary, which allows for subpixel computations (details are presented in [14]). Figure 10 (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 10 (right), where the shape is reconstructed as the envelope of the maximal inscribed discs associated with each medial axis point. Figure 11 depicts subpixel

Algorithm 2 The Divergence-Ordered Thinning Algorithm.

Part I: Total Outward Flux

Compute the distance transform of the object D [6].

Compute the gradient vector field ∇D.

Compute the net outward flux of ∇D using Eq. 7

For each point P in the interior of the object

$Flux(P) = \sum_{i=1}^{n} < N_i, \nabla D(P_i) >$,

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 }

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.

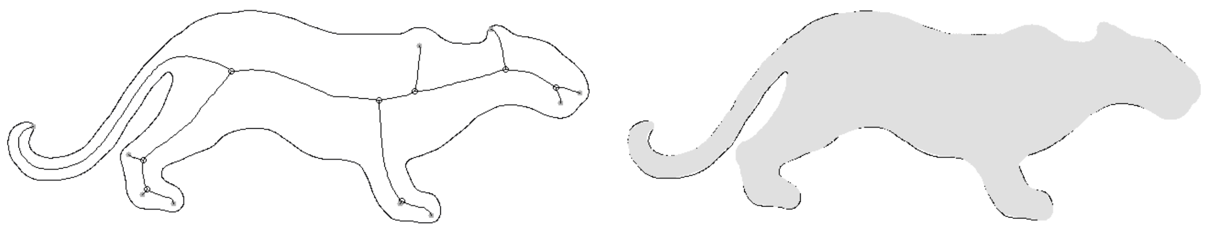

Figure 10: LEFT: A subpixel medial axis, with branch points shown as empty circles and end points as filled circles. Compare with the results in Figure 7. RIGHT: The reconstruction as the envelope of the maximal inscribed disks (grey) of the medial axis, overlayed on the original shape.

Next we illustrate the algorithm with both synthetic data and volumes segmented from MR and MRA 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 divergence 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 divergence.

Figures 12 and 13 compare our approach with the parallel thinning method introduced by Manzanera et al. [34]. The results reveal that both frameworks are robust, and yield structures that are homotopic to the underlying object. However, note that the latter method yields only a subset of the "true" medial surface for these data sets, and hence only a coarse approximation to the object is possible. In contrast, a near perfect reconstruction is possible from the divergence-based medial surfaces.

Recall from Section 6 that the computation of the distance transform, its gradient vector field and the total outward flux are all $\mathcal{O}(n)$ operations. Hence, one may be tempted to simply threshold the divergence map below a certain (negative) value to obtain a medial surface extremely efficiently, as in [54]. Unfortunately, due to discretization, the result will not always be satisfactory. On the one hand, if the threshold is too high, slight perturbations of the boundary will be represented, and the resulting structure will not be a thin set. On the other hand, lowering the threshold can provide a thin set, but at the cost of altering the object's topology. This is illustrated in Figure 14 (second column) where the medial surfaces corresponding to the views in the first column are accurate and thin, but have

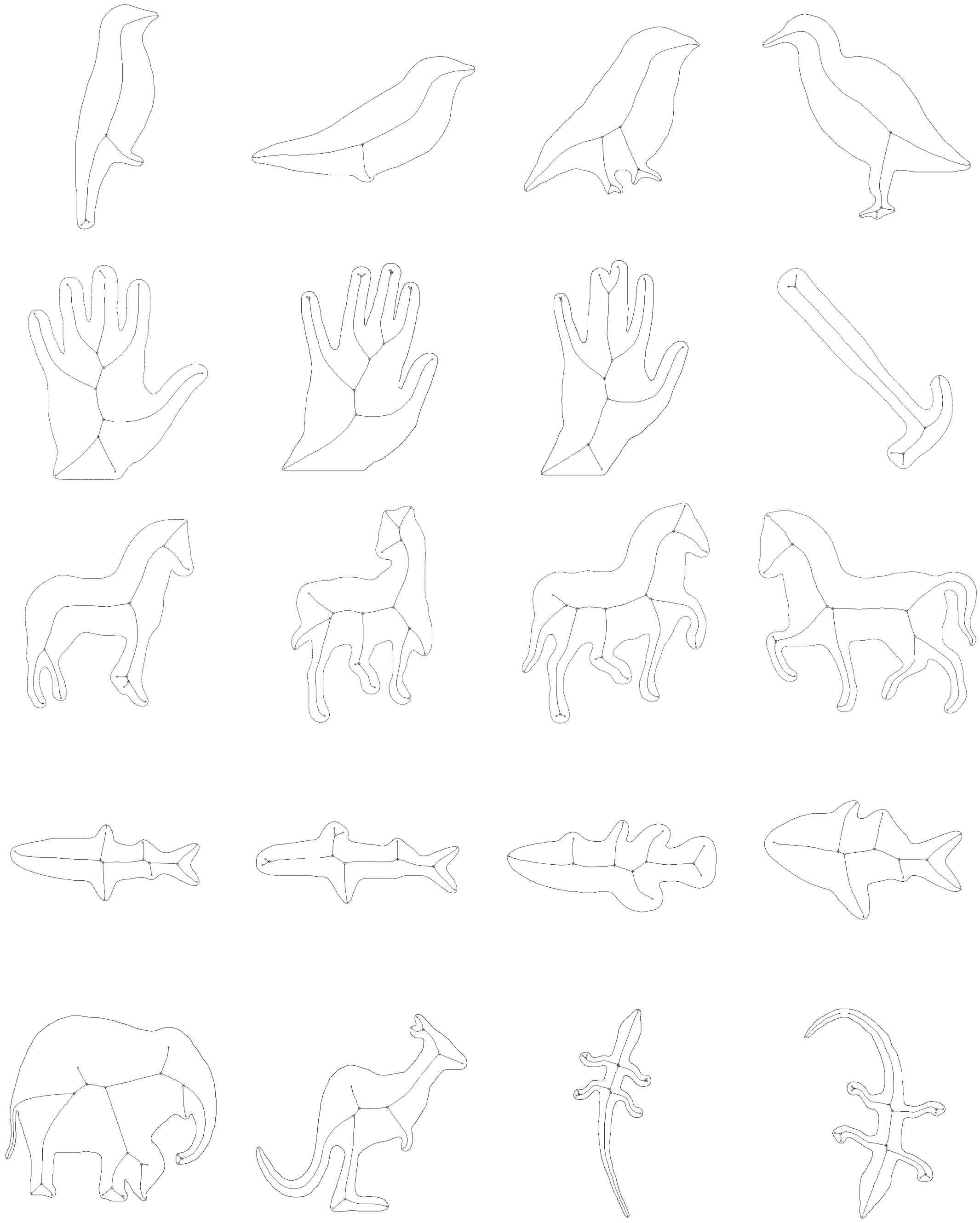

Figure 11: Subpixel medial axes for a range of shapes, obtained by divergence-ordered thinning. The detected end points and branch points are circled.

holes. This motivates the need for the topological constraints along with the characterization of endpoints discussed in Section 6. Observe that with the same threshold as before, the divergence-based thinning algorithm now yields a thin structure which preserves topology,

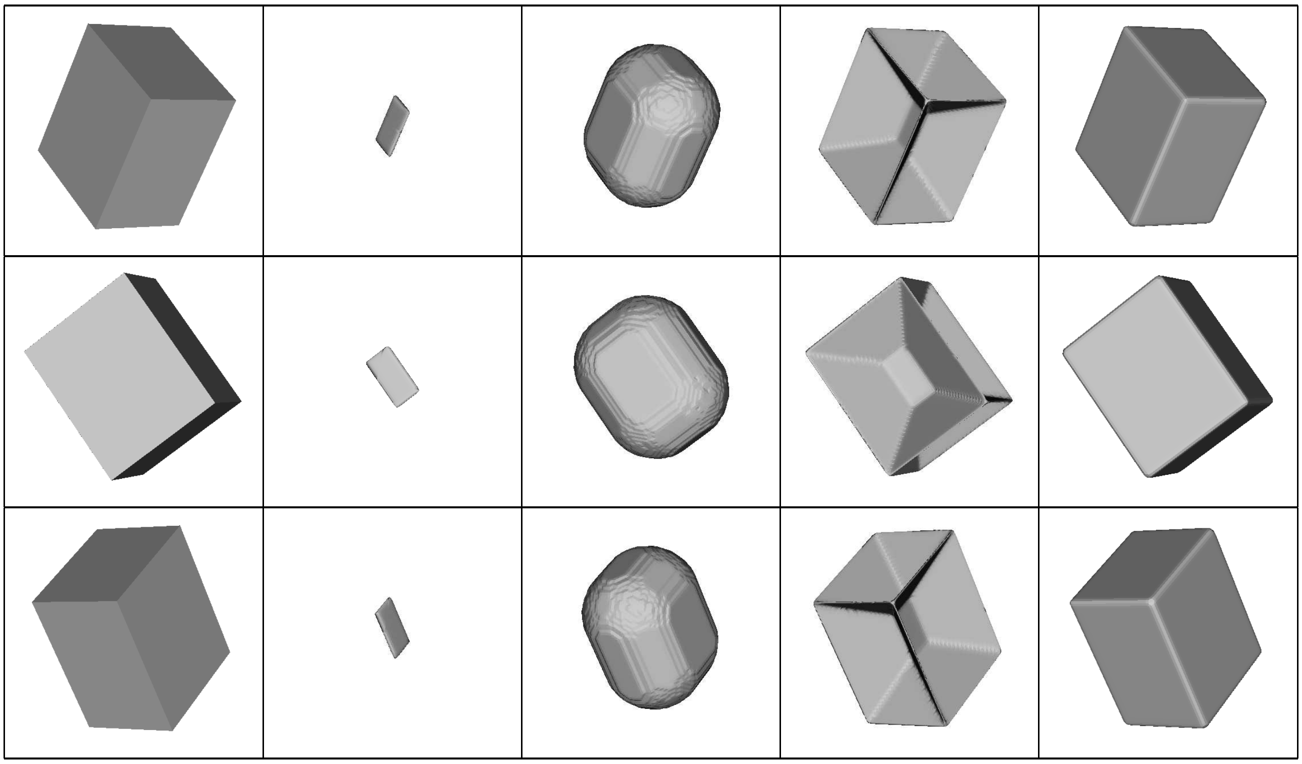

Figure 12: FIRST COLUMN: Three views of a cube. SECOND COLUMN: The corresponding medial surfaces computed using the algorithm of [34]. THIRD COLUMN: The object reconstructed from the medial surfaces in the previous column. FOURTH COLUMN: The corresponding divergence-based medial surfaces. FIFTH COLUMN: The object reconstructed from the medial surfaces in the previous column.

Figure 14 (third column). The ventricles reconstructed from the medial surfaces in the fourth column are shown in the fifth column.

Next, we illustrate the robustness of the approach on a (partial) data set of blood vessels obtained from an MRA image of the brain, in Figure 15. 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 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, 63]. There is a straightforward modification of our framework which allows this, provided that certain special points on the medial surface have been identified. The essential idea is to preserve such special points, but

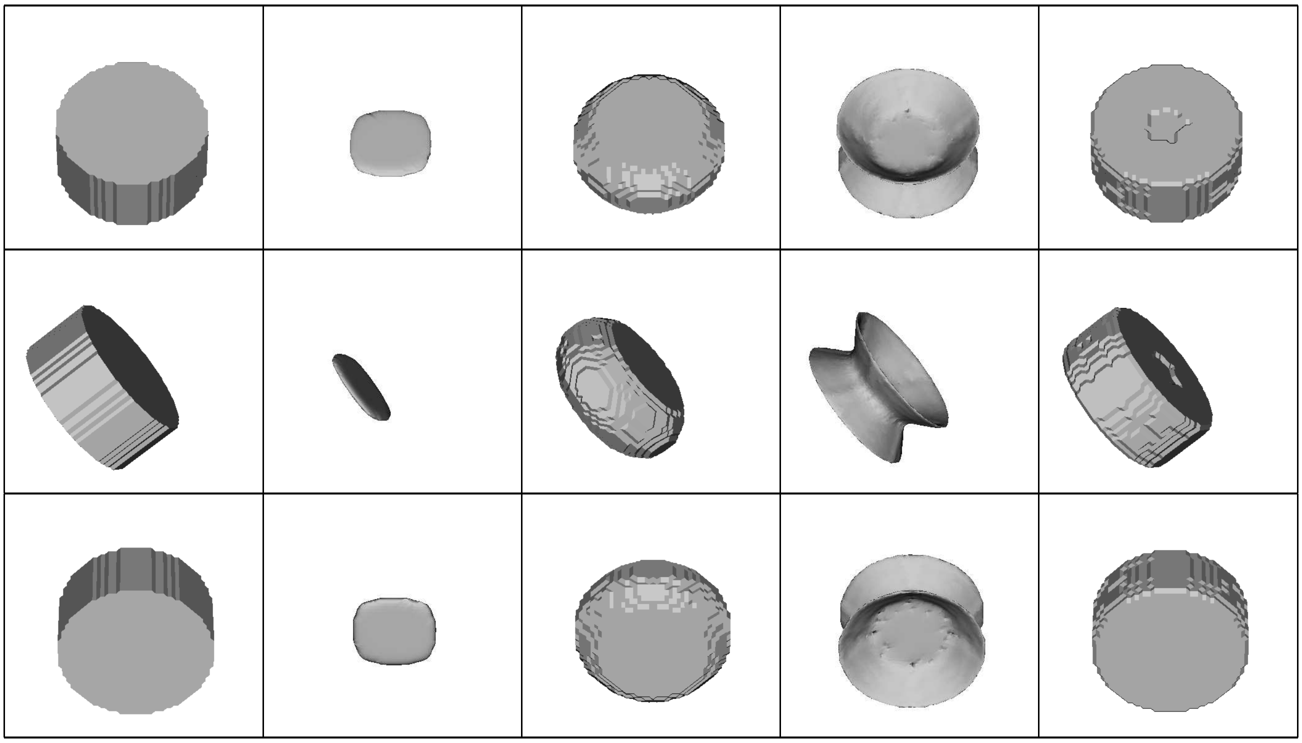

Figure 13: FIRST COLUMN: Three views of a cylinder. SECOND COLUMN: The corresponding medial surfaces computed using the algorithm of [34]. THIRD COLUMN: The object reconstructed from the medial surfaces in the previous column. FOURTH COLUMN: The corresponding divergence-based medial surfaces. FIFTH COLUMN: The object reconstructed from the medial surfaces in the previous column.

to remove all other simple points in a sequence ordered by their divergence values. This is illustrated for a portion of the vessel data in Figure 16, where the endpoints of 3 branches were selected as special points. Observe that the result is now composed of three 1 voxel wide 26-connected 3D digital curves.

As a final example, Figure 17 illustrates the medial surface of the sulcii of a brain, where we have shown an X, Y and Z slice through the volume. Observe that 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 1. As predicted by the complexity analysis in Section 6.4, 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 on k, the number of points contained within the volume.

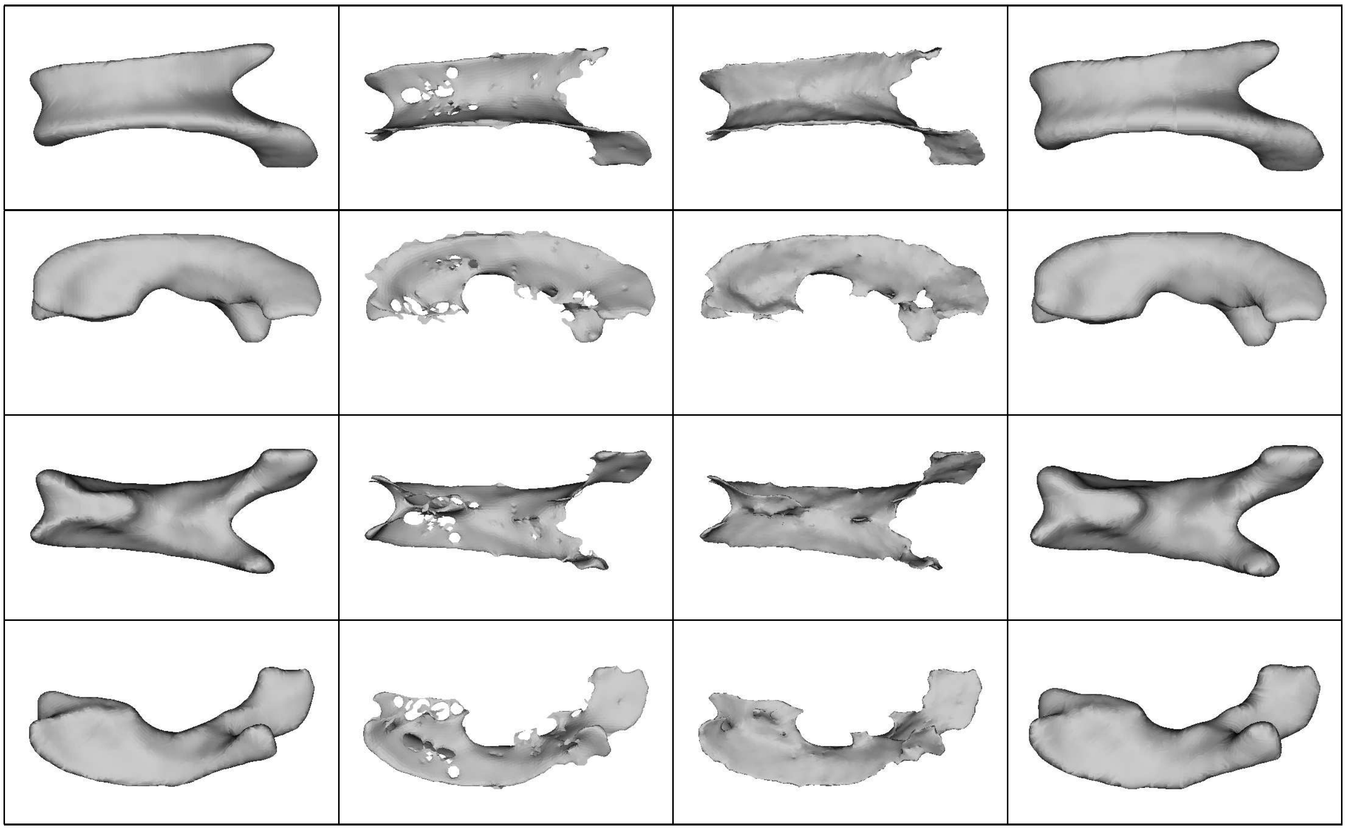

Figure 14: FIRST COLUMN: Four views of the ventricles of a brain, segmented from volumetric MR data using an active surface. SECOND COLUMN: The corresponding medial surfaces obtained by thresholding the divergence map. THIRD COLUMN: The divergence-based medial surfaces obtained using the same threshold, but with the incorporation of homotopy preserving thinning. FOURTH COLUMN: The ventricles reconstructed from the divergence-based medial surfaces in the previous column.

8 Conclusions and Related Work

This paper makes two main contributions. First, a natural algorithm for simulating the eikonal equation has been introduced. The method is rooted in Hamiltonian physics and offers a number of computational advantages when it comes to shock tracking. Second, based on the violation of a conservation of energy principle at singular points of the Hamiltonian system, we have provided a computational framework for shock detection. For the special case of Blum's grassfire flow this leads to a straightforward algorithm for computing 2D and 3D skeletons, which is robust and accurate, computationally efficient, invariant to Euclidean transformations and homotopy preserving. The essential idea is to combine a

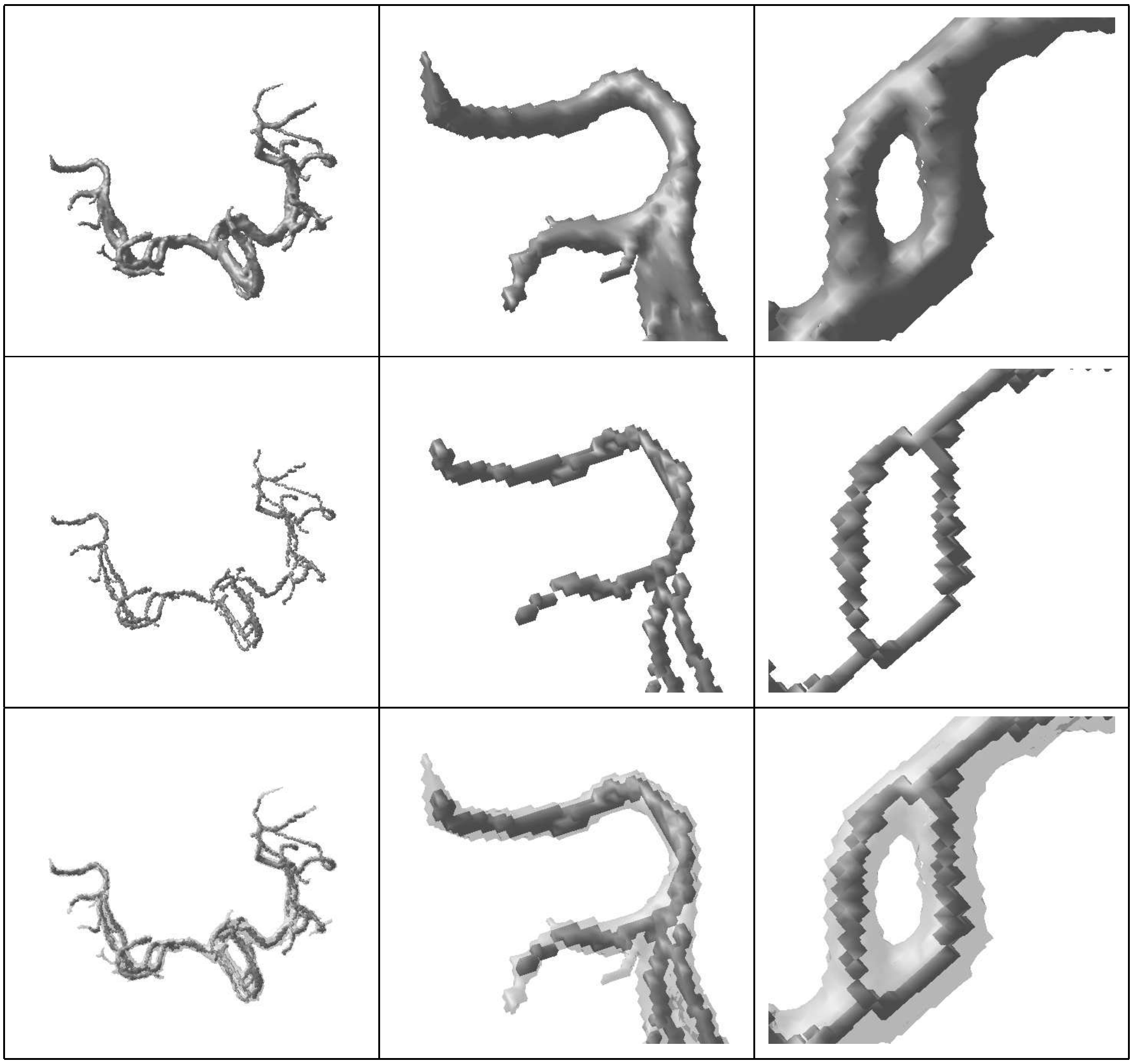

Figure 15: TOP ROW: Blood vessels segmented from volumetric MRA data, with magnified parts shown in the middle and right columns. MIDDLE ROW: The corresponding divergence-based medial surfaces. THIRD ROW: The divergence-based medial surfaces (solid) are shown within the vessel surfaces (transparent).

divergence computation on the gradient vector field of the Euclidean distance function to the object's boundary with a thinning process that preserves topology. We have illustrated the advantages of the approach on synthetic and real binary 2D and 3D shapes of varying complexity.

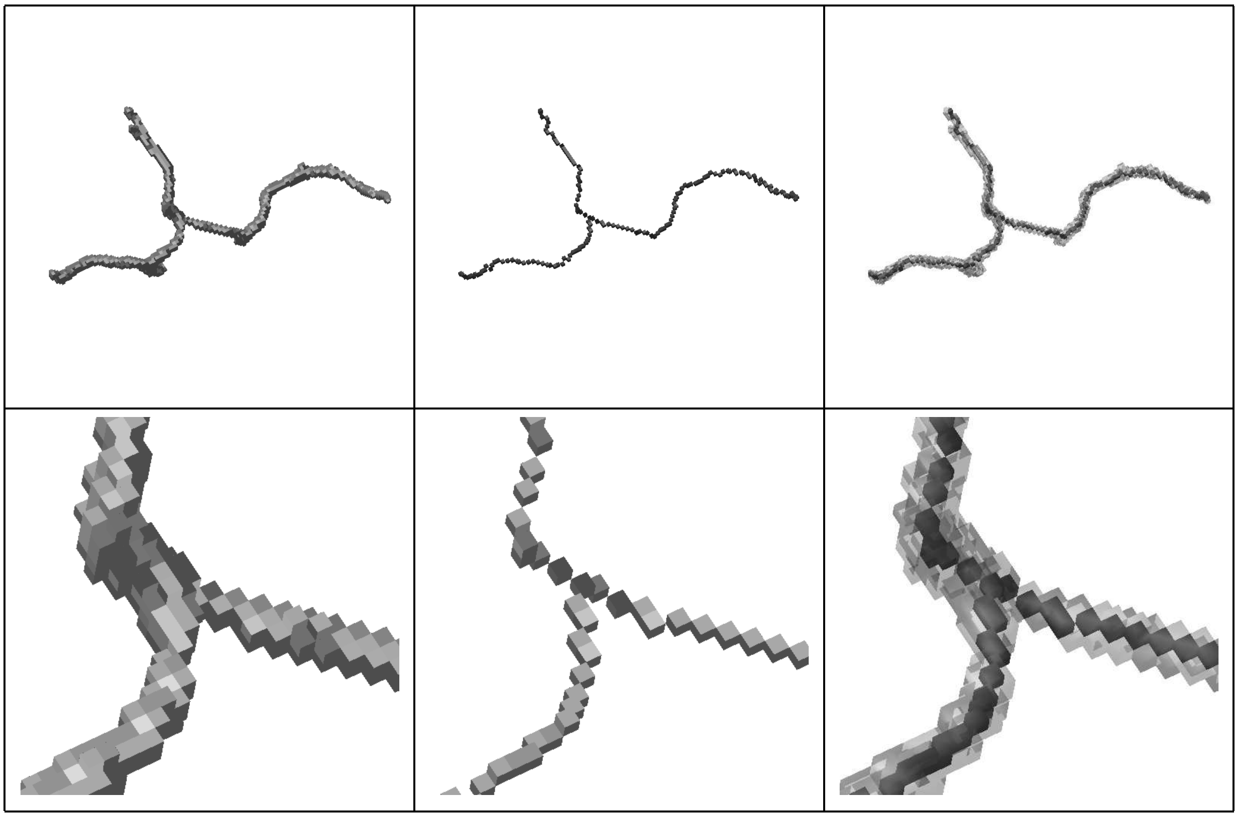

Figure 16: LEFT COLUMN: Blood vessels segmented from volumetric MRA data, with a magnified portion shown in the second row. MIDDLE COLUMN: The divergence-based 3D curves. RIGHT COLUMN: The divergence-based 3D curves are shown embedded within the vessel data.

We should note that in related work, Leymarie and Levine utilized the magnitude of the gradient vector field to design a potential function to attract a snake moving in from the object's boundary [28]. This leads to a very intuitive simulation of the grassfire, however the interpretation of its singularities as a medial axis is not immediate. For example, the contour must first be segmented at curvature extrema, which is itself a challenging numerical problem. 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 [31]. Dynamic programming is used to make the computation efficient. Zhu has posed the computation of the medial axis as a statistical inference problem, leading to an approximation of the skeleton [65]. Tek and Kimia have proposed an approach for calculating symmetry maps, which is based on the combination of a wavefront propagation technique with the use of an exact (analytic) distance function [61]. The method has been

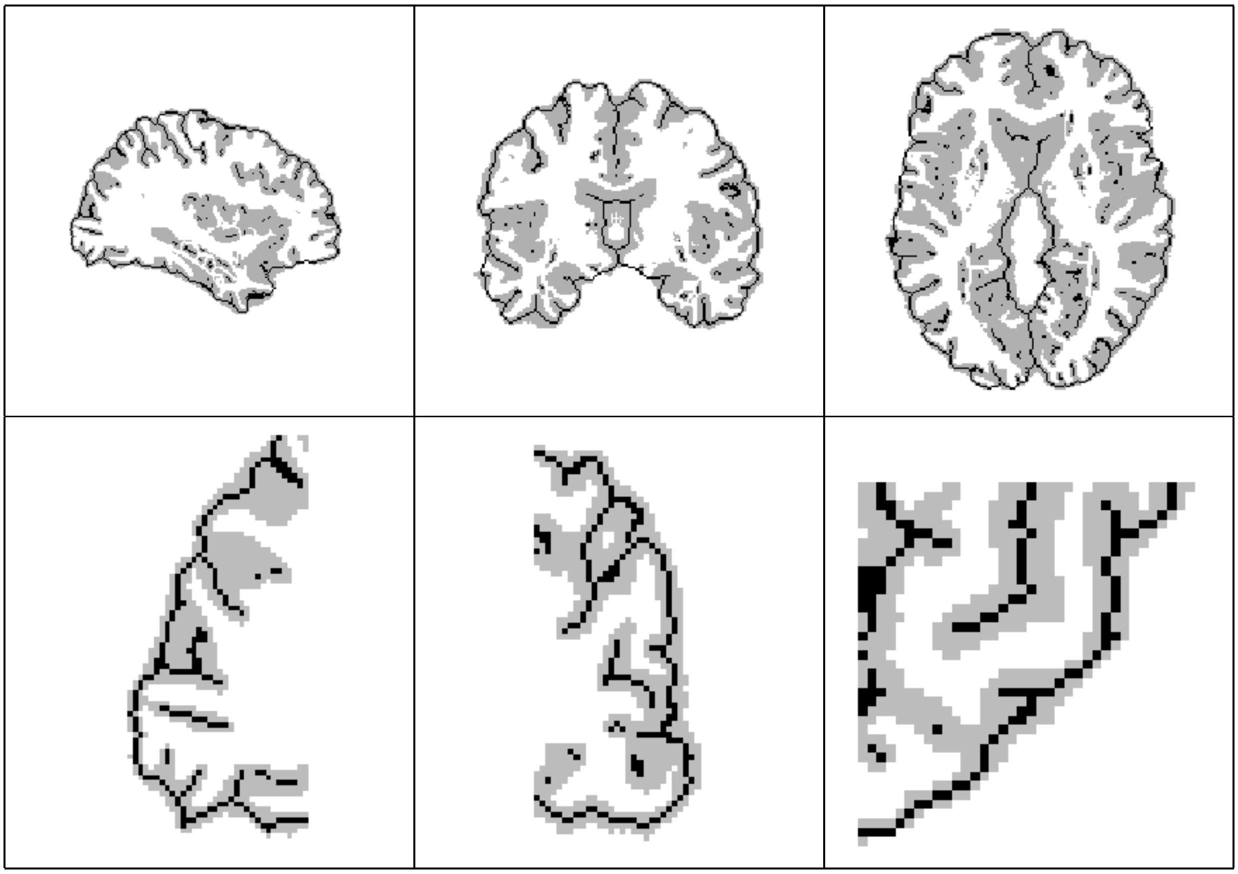

Figure 17: Top Row: Medial surfaces of the sulcii of a brain, segmented from an MR image. The three columns represent X, Y and Z slices through the volume. The cross section through the medial surface in each slice is shown in black, and the object is shown in grey. Bottom Row: A zoom-in on a selected region of the corresponding slice in the top row, to show detail.

demonstrated on both closed and open curves, and is efficient and accurate. However, the representation must be pruned in order to distinguish salient branches from unwanted ones. Pudney has introduced a distance ordered homotopic thinning procedure, where the idea is to remove points in order of their distance from the boundary while preserving end points and centers of maximal balls identified from a chamfer distance function [43]. Malandain and Fernandez-Vidal obtain two sets based on thresholding a function of two measures, ϕ and d, to characterize the singularities of the Euclidean distance function [33]. The two sets are combined using a topological reconstruction process. Whereas empirical results have been good the choice of appropriate thresholds are context dependent and ϕ and d, as well as their combination, are based on heuristics.

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
Sulcii	192x169x99	798221	38.71s	10.54s	240.85s	290.1s

Table 1: 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 6.4 .

There have also been a number of recent related developments in the curve and surface evolution and scale-space communities. Among these, Tari and Shah characterize the symmetries of n-dimensional shapes by looking at properties of the Hessian of a suitably defined scalar edge-strength functional [58]. Whereas the framework is mathematically elegant, robust and efficient numerical implementations remain to be developed. Giblin and Sapiro have introduced two definitions for affine invariant medial axes, along with a notion of affine skew symmetry [18]. Kalitzin et al. have considered index computations on vector fields associated with scalar images in order to identify their singularities [22]. Vector fields rooted in magneto-statics have been used for extracting symmetry and edge lines in greyscale images [13]. Finally, Furst and Pizer have introduced a notion of an optimal parameter height ridge in arbitrary dimension by exploiting a sub-dimensional maximum property. Their approach gives practical advantages when applied to ridge finding [16].

In contrast, our method is rooted in a physics-based analysis of the gradient vector field of the Euclidean distance function, which shows that a conservation of energy principle is violated at skeletal points and provides a single consistent framework within which to compute skeletons in 2D and 3D. We use the divergence theorem to compute the total outward flux of the vector field, and to locate points where energy is absorbed. It 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 that act as sources, and have positive total outward flux. Furthermore, both medial surfaces can be located with sub-voxel accuracy by using the local gradient vector field to shift the final set of digital points. In related work we have demonstrated this idea for 2D shapes, where a similar framework was used to compute sub-pixel 2D skeletons and skeletal graphs [14].

In future work we plan to incorporate the topological classification of [32] to parse the medial surface and obtain a more abstract representation of it, e.g., as a graph. We shall also explore the possibility of finding necessary as well as sufficient conditions for defining endpoints in a cubic lattice.

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.

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 PAMI*, 7(4):463–474, July 1985.

[3] V. 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. *J. Theor. Biol.*, 38:205–287, 1973.

[6] G. Borgefors. Distance transformations in arbitrary dimensions. *CVGIP*, 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 *SSPR'98*, 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 *CVPR'2000*, Hilton Head, South Carolina, June 2000.

[15] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *Fifth 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 *MICCAI'98*, 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] P. J. Giblin and G. Sapiro. Affine invariant medial axis and skew symmetry. In *ICCV'98*, Bombay, India, January 1998.

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

[20] J. Gomez and O. Faugeras. Level sets and distance functions. In *ECCV'2000*, Dublin, Ireland, June 2000.

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

[22] 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.

[23] 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.

[24] 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.

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

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

[28] F. Leymarie and M. D. Levine. Simulating the grassfire transform using an active contour model. *IEEE PAMI*, 14(1):56–75, jan 1992.

[29] A. Liu, E. Bullitt, and S. M. Pizer. 3d/2d registration via skeletal near projective invariance in tubular objects. In *MICCAI'98*, pages 952–963, 1998.

[30] T.-L. Liu and D. Geiger. Approximate tree matching and shape similarity. In *ICCV'99*, pages 456–461, Kerkyra, Greece, September 1999.

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

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

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

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

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

[36] 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.

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

[38] 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.

[39] 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.

[40] 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.

[41] L. Perko. *Differential Equations and Dynamical Systems.* Springer-Verlag, 1986.

[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. J. Numer. Analy.*, 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 *MICCAI'98*, 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. *Proc. Natl. Acad. Sci. 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] 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.

[52] 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.

[53] 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.

[54] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. The hamilton-jacobi skeleton. In *ICCV'99*, pages 828–834, Kerkyra, Greece, September 1999.

[55] 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.

[56] 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.

[57] 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 *IPMI'99*, pages 84–97, 1999.

[58] S. Tari and J. Shah. Local symmetries of shapes in arbitrary dimension. In *ICCV'98*, Bombay, India, January 1998.

[59] 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.

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

[61] H. Tek and B. B. Kimia. Symmetry maps of free-form curve segments via wave propagation. In *ICCV'99*, pages 362–369, Kerkyra, Greece, September 1999.

[62] 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.

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

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

[65] S. C. Zhu. Stochastic computation of the medial axis in markov random fields. In *CVPR'98*, pages 72–79, Santa Barbara, CA, June 1998.