Coarse-to-Fine Skeleton Extraction for High Resolution 3D Meshes

Luca Rossi[a,*], Andrea Torsello[a]

[a] *Università Ca' Foscari Venezia, Dipartimento di Scienze Ambientali, Informatica e Statistica, Via Torino 155, 30172 Venezia Mestre, Italy*

Abstract

This paper presents a novel algorithm for medial surfaces extraction that is based on the density-corrected Hamiltonian analysis of Torsello and Hancock [1]. In order to cope with the exponential growth of the number of voxels, we compute a first coarse discretization of the mesh which is iteratively refined until a desired resolution is achieved. The refinement criterion relies on the analysis of the momentum field, where only the voxels with a suitable value of the divergence are exploded to a lower level of the hierarchy. In order to compensate for the discretization errors incurred at the coarser levels, a dilation procedure is added at the end of each iteration. Finally we design a simple alignment procedure to correct the displacement of the extracted skeleton with respect to the true underlying medial surface. We evaluate the proposed approach with an extensive series of qualitative and quantitative experiments.

Keywords: Medial Surface, Hamilton-Jacobi Equations, 3D Shape Descriptor, Hierarchical Approach

*Corresponding author. Tel: +39 041 2348452, Fax: +39 041 2348419
Email addresses: lurossi@dsi.unive.it (Luca Rossi), torsello@dsi.unive.it (Andrea Torsello)

1. Introduction

The skeleton has proven to be a valuable and widely used shape descriptor for a number of tasks such as 2-D and 3-D shape recognition [2, 3], volumetric models deformation [4, 5], segmentation [6] and protein structure identification [7]. The interest in this descriptor stems from its being a concise representation of the original shape, which is topologically equivalent to it, and invariant to several shape deformations.

When working in two dimensions, the skeleton, or medial axis transform, is defined as the locus of the centers of the maximal inscribed circles bitangent to the shape boundary. Alternatively, it can be defined as the set of singularity points created by the inward evolution of the shape boundary with constant velocity according to the eikonal equation $\frac{\vec{B}(t)}{dt} = v\vec{N}(t)$, where $\vec{B}(t)$ is the equation of the boundary at time t, v is the constant velocity and $\vec{N}(t)$ is the normal to the boundary. Finally the skeleton can be seen as the set of ridge points of the distance map [8] [9], where the distance map is the function $D(x, y)$ that assigns to every point in the interior of a shape its distance to the closest point on the boundary.

1.1. 2D Skeleton Extraction

Over the years several methods have been proposed to compute the 2D skeleton of a shape, but all of them can be basically divided into four main categories.

The first class of methods are the thinning ones, which simulate Blum's grassfire transform by iteratively eroding layers from the shape [10] [11].

During the thinning procedure care must be given not to change the object topology and to ensure the correct geometrical position of the skeleton with respect to the original shape, since the result is clearly dependent on the order in which the erosion is performed. Unfortunately, while fast and simple to implement, these algorithms are quite sensitive to Euclidean transformations, so they typically fail to locating accurately the skeleton of the object.

The second class of methods exploits the fact that the skeleton coincides with the local extrema of the Euclidean distance transform [9] [12] [13]. This in turn relies on the computation of the Euclidean distance between each point in the interior of the object and the boundary of the shape, which can be done in linear time $O(n)$, where n is the number of pixels of the image [14]. These approaches then attempt to detect the ridges of the distance map either directly or by evolving a series of curves, such as snakes, under a potential energy field defined by the distance map. Although these methods fulfill the geometrical constraint, ensuring the topological correctness is not trivial.

A third class of methods is based on the Voronoi diagram of a subset of the boundary points [15]. The idea of these approaches is that, under appropriate smoothness conditions, the Voronoi diagram of a subset of the boundary points converges to the skeleton as the number of the sampled boundary points increases. These methods ensure topology preservation and invariance under Euclidean transformations, in addition to locate the skeleton with great accuracy, provided that the boundary of the shape is sampled densely enough. However, if the object being skeletonized is not a polygon, they obviously suffer from limitations due to the computational complexity of finding the Voronoi diagram of the shape (or alternatively the Delaunay

triangulation). Moreover, approximating a smooth shape with many straight line segments introduces a lot of spurious branches, which then need to be pruned with techniques typically based on heuristics.

The fourth, and final, class of methods is based on the analysis of the differential structure of the boundary. In [16], the boundary is segmented at points of maximal curvature and the authors show that the skeleton is a subset of the Voronoi diagram of these segments. Despite its accuracy, the main drawback of this approach is the need to estimate the boundary curvature by fitting a curve to it, which is a computationally demanding and quite delicate task. A somehow similar approach is that of Leymarie and Levine [13], which is based on the concept of active contours introduced in [17]. Kass, Witkin and Terzopoulos cast the problem of boundary location into a curve evolution framework, where the curve is evolved in a potential energy field under certain smoothness constraints. By using the distance map as the energy function, Leymarie and Levine are able to estimate the shape skeleton by simulating the grassfire transform and identifying the points where the wavefront collapses as the skeletal points. Unfortunately, as in [16] this requires an initial segmentation of the boundary at curvature extrema, which is itself a challenging problem.

Another important method that belongs to this class stems from the Hamiltonian analysis of the boundary flow dynamics [18]. Siddiqi et al. state that the singular points where the system ceases to be Hamiltonian (i.e., an energy conservation principle is violated) are responsible for the formation of skeletal points. Unfortunately, their analysis fails to take into account the effects of the boundary curvature, a problem which they only partially

solve in [19]. Subsequently, however, Torsello and Hancock [1] show how to completely overcome the problem by performing a Hamilton-Jacobi analysis of the flow under conditions where the flow density varies due to curvature.

1.2. 3D Skeletons

Although there exist a considerable number of algorithms for the extraction of skeletons from 2D shapes which yield reasonably good results, the problem of medial surface extraction is still an open one. This is because the addition of a third dimension makes the task of medial surfaces extraction particularly challenging. At the same time, the wide availability of cheap 3D scanning devices demands for a robust representation which provides a simple venue to perform shape analysis and representation under deformation and articulation. For this reason, the design of efficient algorithms for 3D skeleton extraction is of key importance.

Luckily, while in 2D the skeleton extraction needs to be preceded by a segmentation of the image, in 3D it is common to model objects as distinct meshes, and thus the skeletonization can be much more practical. However, when a third dimension is added the task of medial surfaces extraction turns out to be much more challenging than in 2D. The reason is threefold. First, we observe an exponential growth of the number of voxels, which may render the computation impracticable, especially if a high resolution is needed. Further, while in 2D it is common to work with raster images, and thus there is no need to discretize the shape, volumetric objects are usually represented as triangle meshes, that may eventually need to be voxelized before any further computation is done. Clearly, the result of this discretization depends on the resolution chosen. Moreover, the topology itself may change as the resolu-

tion changes. Finally, tasks that are almost trivial in two dimensions, such as ensuring the topological correctness of the skeleton, i.e., the equivalence between the object and its skeleton, require particular attention when a third dimension is added.

According to the analysis that we need to perform on the shape, in the literature there are two competing 3D generalizations of the skeleton: the curve (or line) skeleton [20, 21] and the medial surfaces. The curve skeleton provides a minimal yet efficient representation for shape analysis and recognition. The medial surfaces, on the other hand, carry enough information to accurately reconstruct the original shape from the skeleton. In fact, while the line skeleton is a lossy simplification of the shape, the medial surface is topologically equivalent to the original shape, i.e., it is possible to map its segments, considered as two oriented surfaces, to the original mesh through a homotety. In some degenerate cases, moreover, the curve skeleton turns out to be ill-defined. Consider for example the shape of a cup, which clearly cannot be abstracted in terms of a medial axis. For these reasons, in this paper we decide to concentrate on the extraction of medial surfaces from triangulated meshes.

Recently, Arcelli et al. [22] proposed a distance-driven algorithm for medial surfaces extraction. Although the algorithm proves to be effective and it is shown to preserve the topology of the original shape, it works only on voxelized objects, and as a consequence cannot cope with high resolution inputs. The work of Siddiqi et al. [3] bears some similarities with the present paper, as it generalizes to three dimensions the Hamilton-Jacobi skeleton. However, it suffers from the same limitations of its two-dimensional counter-

part, since it doesn't take into account the effects of boundary curvature. A more robust algorithm is that of Reniers et al. [23], where both the curve and the surface skeletons are located by means of an advection-based importance measure. Unfortunately this measure turns out to be well defined only for genus 0 shapes, and both [3] and [23] share again the problem of requiring a complete voxelization of the space, which makes the use of these algorithms limited to low resolution objects.

In order to cope with increased spatial and time complexity, Bai et al. [24] and Quadros et al. [25] propose to use adaptive octrees, which allow some parts to be discretized more densely while the rest is analyzed at a coarser scale. However, both these approaches work on a precomputed octree, where the grid refinement criterion is based on simple heuristics. In [24] the authors propose to increase the grid resolution on those voxels that are roughly at the center of the shape, where the medial surface is more likely to be located. Anyway, they clearly state that the design of an optimal grid adaptation criterion for skeleton computation is beyond the scope of their paper, and a more efficient heuristic should be used instead. In [25] the octree nodes are generated according to the vertices and centroids of the facets of an input CAD model, therefore the density of the nodes is higher in the presence of small features or regions of high curvature. The resulting skeleton, however, is disconnected, and it is composed of sets of nodes at different levels of resolution.

Finally, Yoshizawa et al. [5] and Hisada et al. [26] propose a generalization of the Voronoi-based approach to three dimensions. These approaches work directly on the original mesh by approximating the medial surface with

a skeletal mesh which has the same number of vertices and connectivity as the original mesh. More precisely, the QuickHull algorithm [27] is used to extract the Voronoi diagram of the mesh vertices, then for each mesh vertex v they define a skeletal point p at a distance d along v's normal, where the displacement d is computed as the distance from v to the arithmetic mean of the Voronoi vertices of the Voronoi region containing v. The connectivity between skeletal vertices is then defined according to the connectivity between the corresponding mesh vertices. These approaches are fast and do not require an initial voxelization, but extract only an approximation of the skeleton and are extremely sensitive to small perturbations of the boundary.

Recently we [28] proposed a hierarchical skeletonization algorithm where the refinement criterion is based on the density-corrected Hamiltonian analysis [1]. In order to deal with the discretization errors incurred at the coarser levels, we proposed to dilate the skeleton at each step of the hierarchical refinement. Although this procedure clearly increases the quality of the extracted skeleton, some discretization artifacts remain unsolved. In particular, due to the discrete nature of the voxelization procedure, the center of the final skeletal voxels tend to be displaced with respect to the true underlying medial surface. This in turn will affect the quality of the extracted skeletal mesh.

1.3. Our Contribution

Our purpose in this paper is to extend the work by Rossi and Torsello [28]. We propose a novel algorithm for medial surfaces extraction that is based on a generalization to three dimensions of the density-corrected analysis of Torsello and Hancock [1], while taking an adaptive octree-based approach for

the discretization of the initial mesh in a manner that is similar to that proposed by Bai et al. [24] and Quadros et al. [25]. Contrary to these approaches, we decide not to precompute the whole octree in advance, but instead we keep the original mesh, that is used for distance computations, and we iteratively decide whether to refine a voxel or not based on the local value of the divergence of the momentum field, i.e., the confidence we have in that point being skeletal. Finally we design a simple alignment procedure to correct the displacement of the extracted skeleton with respect to the true underlying medial surface. We evaluate the proposed approach with an extensive series of qualitative and quantitative experiments, comparing our method against other approaches in the literature under varying mesh conditions.

2. Preliminaries

In this section we review the two-dimensional continuous formulation of the Hamilton-Jacobi skeleton [18] and its density corrected counterpart [1], where the latter will form the basis for our medial surface extraction algorithm.

2.1. Hamilton-Jacobi Skeleton

Let the distance map D be a function that assigns to each point in the interior of the shape its distance to the closest point on the object boundary $\vec{B}$, and let $\vec{F} = \nabla D$ be the corresponding velocity field, where $\nabla = (\partial/\partial x, \partial/\partial y)^T$ is the gradient operator. We define the outward flux of $\vec{F}$ through the boundary ∂A of an arbitrary area A as $\phi_A(\vec{F}) = \int_{\partial A} \vec{F} \cdot \vec{n}\, dl$, where $\vec{n}$ denotes the normal to ∂A and dl is the length differential on ∂A.

Under the assumption that the vector field $\vec{F}$ is conservative everywhere except on the skeleton, the skeletal points can be identified by looking for those points where the system ceases to be conservative. Since the net flux of $\vec{F}$ through the boundary of the shape is positive, by virtue of the divergence theorem the interior of the shapes contains a set of sink points, i.e., the skeletal points. Hence, in their original formulation, Siddiqi et al. propose to label as skeletal those points in which the divergence of $\vec{F}$ is non-zero [18]. However, under a compressing front, the divergence can be negative also at non-skeletal locations. More precisely, the density of the compressing front changes during its inward evolution in a way which is proportional to the boundary curvature, and as a result the velocity field is no longer conservative. Initially, Siddiqi et. al tried to overcome this problem with the introduction of the concept of normalized flux. They show that by normalizing the flux of the velocity field by the perimeter of a circular integration area, as the radius of the circle approaches zero so does the value of the divergence, if the point is not skeletal. Due to the discrete structure of the lattice, however, the integration radius has a lower bound of one pixel. Since the divergence of the velocity field in $\vec{p}$ depends on the local boundary curvature, assuming an integration radius of one pixel, the value of the normalized flux at $\vec{p}$ will be $N\phi_A(\vec{F})(p) = -\frac{1}{2}k(\vec{p})$, where $k(\vec{p})$ is the curvature of the evolving boundary at $\vec{p}$ and $N\phi_A(\vec{F})$ denotes the normalized flux of $\vec{F}$. The problem is that near the endpoints of the skeleton the value of the curvature will tend to infinity, thus the discrete normalized flux diverges in their proximity.

2.2. Density-Corrected Analysis

Based on the observation that when the front is curved the average linear density is not constant over time, Torsello and Hancock [1] propose to change the problem into a mass conservation one. More precisely, they state that, rather than the velocity field, it is the momentum field $\vec{M} = \rho\vec{F}$ that is conservative, where ρ is a scalar field that assigns to each point along the inward-evolving boundary front its linear density. As a result, the divergence of the momentum field is zero at any non-skeletal point, i.e., $\nabla \cdot (\rho\vec{F}) = 0$, and thus also $\phi_A(\rho\vec{F}) = 0$ for any region A not containing a skeletal point.

The density of the inward-evolving boundary can then be determined by applying the rule of product differentiation to the conservation equation and setting $\sigma = \log(\rho)$, thus yielding

$$\nabla\sigma \cdot \vec{F} = -\nabla \cdot \vec{F}\,. \tag{1}$$

Finally, this can be further reduced to the system of ordinary differential equations along the path of boundary points

$$\begin{cases} \frac{\partial}{\partial t}\sigma(s(t)) = -\nabla \cdot \vec{F}(s(t)) \\ \frac{\partial}{\partial t}s(t) = \vec{F}(s(t)) \end{cases} \tag{2}$$

where $s(t)$ is the trajectory of a boundary point under the eikonal equation.

3. Hierarchical Skeletonization

Our algorithm works as follows. We are given a triangulated mesh, a starting resolution res_{min} and a desired resolution res_{max}. Initially we compute a complete voxelization of the shape at resolution res_{min}. Given this

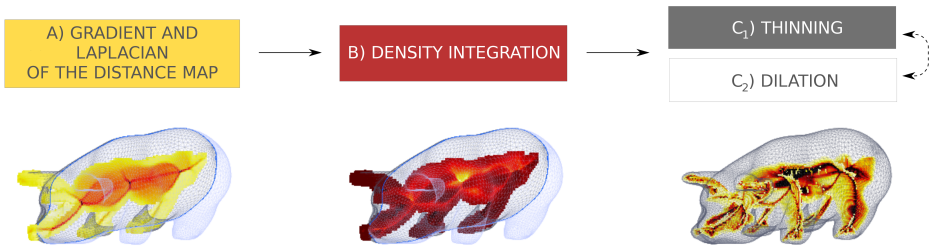


Figure 1: Steps to refine the skeleton: a) computation of the gradient and Laplacian of the distance map; b) integration of the log-density in the voxels with a full neighborhood; c) alternating thinning and dilation step to detect skeletal voxels at the current level of the octree.

initial coarse discretization, we compute the distance transform D, its gradient $\vec{F} = \nabla D$ and the divergence $\nabla \cdot \vec{F}$, then we integrate the density $\sigma = \log(\rho)$ and finally we compute the divergence of the momentum field $\nabla \cdot (\rho \vec{F})$. With this information to hand, we are able to extract a first approximation of the medial surface. Assuming that a very low starting resolution res_{min} is given as input, we now wish to further refine the extracted skeleton up to a res_{max} resolution.

To this end, we iteratively increase the resolution by subdividing the leaves of the octree with a large value of $\nabla \cdot (\rho \vec{F})$, i.e., those voxels that are most likely to contain skeletal points. The Hamiltonian analysis is then carried over the newly created octree level and the refinement process is iterated until the required resolution res_{max} and octree level $\log_8(res_{max})$ is reached.

In order to carry over the Hamiltonian analysis at a lower octree level the following steps must be undertaken (see Fig. 1):

1. **Velocity field computation**. For each voxel $\vec{v}$ at the current resolu-

tion level we compute its distance to the shape boundary. Given the distance map, we first compute its gradient in $\vec{v}$ by fitting a hyperplane in a least squares sense on the voxel neighbors, then we determine its Laplacian by computing the flux of $\vec{F}$ through the surface of the convex-hull bounded by the neighbours of $\vec{v}$, divided by its volume.

2. **Integration of the front-density**. For each voxel at the current resolution level we compute the density of the evolving front by evaluating Eq. 2. We integrate the density starting from the current level boundary inward, under the assumption that the initial boundary has a complete 26-neighborhood where the value of the density is inherited from the parent voxels.
3. **Thinning and dilation**. With the divergence information to hand, we iteratively remove the current level boundary voxels in distance order when the value of the divergence is under a certain threshold. In order to guarantee the preservation of the object topology, we remove a voxel only if it is simple, i.e., if its removal does not alter the object topology by disconnecting the shape or introducing a hole [34]. Once the thinning procedure is completed, we dilate the skeleton to partially compensate for discretization errors incurred at the coarser levels. We alternate the thinning-dilation process until no voxels can be added to the thinned skeleton. Finally a last dilation is performed to guarantee that the exploded points have a complete neighborhood around each skeletal point.

With this high-level overview in mind, we will now present all the computational ingredients needed by the proposed approach.

3.1. Distance Computation

The distance transform computation is certainly one of the most expensive operations that we need to perform. We decide not to compute the distance map with respect to a discretized boundary, instead we keep the original mesh and we make distance queries with respect to it. In particular, the input mesh is saved on an Axis Aligned Bounding Box (AABB) tree [29], a common data structure that is used to make distance queries faster. A voxel is assigned either to the interior or exterior of the shape by casting a ray from the center of the voxel to a random direction and computing the number of intersections with the mesh. If the number of intersections is odd, the point is classified as interior, otherwise it is classified as exterior. We acknowledge that better algorithms for computing the signed distance transform have been proposed in the literature (e.g., [30]), but we also want to stress that the distance map issue is completely incidental to the main problem of skeletonization, which is the one we are addressing in this paper.

3.2. Gradient and Laplacian Computation

Once the distance map is to hand, its gradient and divergence can be determined. Note, however, that while in the beginning all the leaves of the octree are at the same level and thus the gradient and the Laplacian can be approximated using the finite difference method, as the skeleton is refined there will be several voxels at different levels of resolution. For this reason we need to resort to a different approximation method that is able to cope with a non-uniform grid setting.

Note that in the remainder of the paper we will operate on different neighborhoods of a voxel, according to the type of operation that we intend

to perform. This includes the $6-$, $18-$ and $26-$ neighborhoods, where $n-$ refers to the adjacency relation between the voxels. Recall that two voxels are 6-adjacent if they share a face, 18-adjacent if they share a face or an edge and 26-adjacent if they share a face, an edge or a vertex. In particular, we will always assume that a 26-neighborhood is used, with the exception of a few cases. As explained later in the text, when computing the laplacian of the distance map we only use local information and thus we restrict ourselves to a 6-neighborhood. On the other hand, during the integration of the density, we will use the subset of the 26-neighbors that have already been visited by the inward-evolving boundary. Finally, when ensuring the topology preservation, we will refer to the work of Malandain et al. [34], where the $6-$, $18-$ and $26-$ neighborhoods are used to characterize the voxels.

Following [31], we compute the gradient by performing a 4D linear regression over all the neighbors of $\vec{x}$. More formally, given a set of points $\{(x_i, y_i, z_i, d_i)\}_{i=1}^{m}$, where $(x_i, y_i, z_i)^T$ is a neighbor of $\vec{x}$ and d_i its distance to the boundary, we look for the coefficients A, B, C, D so that the hyperplane $d = Ax + By + Cz + D$ best fits the samples in a weighted least squares sense. Minimizing

$$E(A, B, C, D) = \sum_i w_i (Ax_i + By_i + Cz_i + D - d_i)^2 \,. \tag{3}$$

the gradient is then $\vec{F}(\vec{x}) = \frac{(A,B,C)^T}{||(A,B,C)^T||}$, where as a weight w_i we used the inverse of the distance of the point $(x_i, y_i, z_i)^T$.

Note that this approach has a problem whenever the skeleton crosses the convex hull of the neighborhood, as we integrate across a singularity resulting in erroneous computation of the gradient. A common solution to

this problems is to perform one-sided computations to avoid crossing the singularity, however one-sided computations usually exhibit larger bias. Here we chose to perform a two-sided computation of the gradient as we are not interested in its value close to the singularity as we are adopting a one-sided process for the computation of the momentum field. The experiments will show, that even with this possible instability due to the possibility of crossing a singularity in the computation of the gradient, the momentum field is well conserved outside the skeletal branches resulting in a well localized skeleton.

As for the laplacian of the distance map, i.e., the divergence of the velocity field, we compute it using a discretization of the divergence theorem around the convex hull of the 6-neighborhood of each point. Note that even if the leaves are not guaranteed to be at the same level, and thus we cannot guarantee to have a complete 26- or 18- neighborhood, due to the octree construct we always have at least a 6-neighborhood. Doing a linear approximation of $\vec{F}(\vec{x})$ over the faces of the convex hull, we can approximate the flux

$$\Phi_U(\vec{x}) = \int_{\delta U} \vec{F}(s) \cdot \vec{n}(s)\, ds \approx \sum_{t=1}^{8} \frac{1}{3} A_t \vec{n}_t \cdot \left(\sum_{\vec{p} \in V_t} \vec{F}(\vec{p}) \right) , \tag{4}$$

where U is the convex hull of the 6-neighbors of $\vec{x}$ and A_t, $\vec{n}_t$, and V_t are respectively the area, the normal, and the set of vertices of the (triangular) faces of U. Due to the divergence theorem, we have $\int_U \nabla \cdot \vec{F}(\vec{x})\, dx = \Phi_U(\vec{x})$, from which we obtain the following discretization for the divergence:

$$\nabla \cdot \vec{F}(\vec{x}) \approx \frac{\Phi_U(\vec{x})}{|U|} \approx \frac{\sum_{t=1}^{8} \frac{1}{3} A_t \vec{n}_t \cdot \left(\sum_{\vec{p} \in V_t} \vec{F}(\vec{p}) \right)}{|U|} . \tag{5}$$

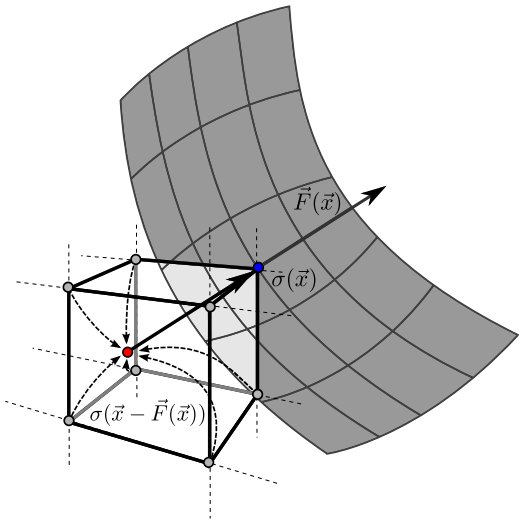


Figure 2: Integration of the density along the boundary path.

3.3. Integration of the Momentum Field

Once the distance, gradient and Laplacian have been computed, we can integrate the density in the newly subdivided skeletal points.

It is of key importance that the density integration is carried out only on those points that have a complete 26-neighborhood, i.e., those with a homogeneous neighborhood. The voxels with a non-homogeneous neighborhood, on the other hand, will simply inherit the value of the density and divergence fields of their parent node. The reason for this is that an inhomogeneous neighborhood induces a higher discretization error to the direction of the gradient which will severely affect the accuracy of the integration step. Thus, before refining the skeleton to a higher resolution level, we perform a dilation of the skeletal voxels in order to guarantee that all their children will indeed have a complete neighborhood. Then, after the refinement, there will be a 1-voxel thick boundary of voxels with non-homogeneous neighbor-

hood that will be children of the dilation voxels, rather than of the skeletal voxels. Note that this dilation can simply be considered a part of the last thinning/dilation step of the refinement of the previous level, which will be described later.

In order to compute the momentum field over the interior of the shape we need to solve Eq. 2. A common approach in this case is that of solving the linear system obtained by rewriting Eq. 2 as a system of difference equation. The problem here is that the skeleton is a set of singularities of momentum field, i.e., we expect the density field to have different values at opposite sides of a medial surface. Consequently, the linear system has no solution. Even looking for an approximate solution using a gradient descent method would result in oscillations near the skeleton, so a different approach is needed.

As proposed by Torsello and Hancock [1], we decide to integrate the equation in the time domain. The critical point is to ensure that when we compute the log-density σ of boundary points at time t we reference only the values of σ calculated at points already crossed by the inward-evolving boundary. In order to do so, we opt to find a numerical solution of Eq. 2 using a Crank-Nicolson approximation [32].

Assume that there exists a family of surfaces $\vec{B}_t$ representing the inward evolution of the boundary $\vec{B}$, that can be locally parametrized as $\vec{B}_t(u,v)$ around any point $\vec{x}$. Then, we have

$$\sigma(\vec{B}_t(u,v)) = \sigma(\vec{B}_{t-1}(u,v)) + \frac{1}{2}[\nabla \cdot \vec{F}(\vec{B}_t(u,v)) + \nabla \cdot \vec{F}(\vec{B}_{t-1}(u,v))] \quad (6)$$

In the spatial domain, if $\vec{x} = \vec{B}_t(u,v)$ we have $\vec{B}_{t-1}(u,v) \approx \vec{x} - \vec{F}(\vec{x})$,

which, substituted into Eq. 6, yields

$$\sigma(\vec{x}) = \sigma(\vec{x} - \vec{F}(\vec{x})) + \frac{1}{2}[\nabla \cdot \vec{F}(\vec{x}) + \nabla \cdot \vec{F}(\vec{x} - \vec{F}(\vec{x}))] \tag{7}$$

Unfortunately the point $\vec{x} - \vec{F}(\vec{x})$ is not guaranteed to belong to the cubic lattice, so we actually need to interpolate it using the values at the eight vertices of the cube containing it. Once again we should ensure that the interpolation doesn't cross the medial surfaces. Luckily, $\vec{x}$ is the last of the eight vertices visited by the evolving boundary, so this requirement is met. Thus we can safely use the trilinear interpolation which yields

$$\begin{aligned}\sigma(\vec{x}) = \Big(&\sigma(\vec{x} - \vec{F}(\vec{x})) - (1 - |F_1|)(1 - |F_2|)(1 - |F_3|)\sigma(\vec{x}) \\ &+\tfrac{1}{2}[\nabla \cdot \vec{F}(\vec{x}) + \nabla \cdot \vec{F}(\vec{x} - \vec{F}(\vec{x}))]\Big)/(1 - (1 - |F_1|)(1 - |F_2|)(1 - |F_3|))\end{aligned} \tag{8}$$

where, F_1, F_2, and F_3, are the three components of $\vec{F}(\vec{x})$ and, due to the fact that we use trilinear interpolation, $\sigma(\vec{x} - \vec{F}(\vec{x})) - (1 - |F_1|)(1 - |F_2|)(1 - |F_3|)\sigma(\vec{x})$ does not depend on the value of $\sigma(\vec{x})$. As Fig. 2 shows, the point $\vec{x} - \vec{F}(\vec{x})$ does not belong to the cubic lattice. We then interpolate it using the values of the log-density on the eight corners of the cube containing the point. Note that $\vec{x}$ is the last of the eight vertices which is visited during the boundary evolution, and thus we are guaranteed that all the points that we use for the interpolation are on the same side of the medial surface.

Given this formulation, we can integrate the value of the log-density over the interior of the shape, starting from the most external voxels inwards. At the first level the most external voxels will be the boundary boxes, which have a unit density, and thus a null log-density. At all other steps, the external voxels will be the voxels with irregular neighborhood that inherit the log-density from their parents. Once the log-density has been integrated, we can

proceed to compute the divergence of the momentum field in each point of the interior of the shape. The value of $\nabla \cdot (\rho\vec{F})(\vec{x})$ is given by approximating Eq. 1 as follows

$$\begin{aligned}\nabla \cdot (\rho\vec{F})(\vec{x}) &= \Delta\sigma e^{\sigma(\vec{x})-\frac{1}{2}\Delta\sigma} \\ &+\tfrac{1}{2}\left[\nabla \cdot \vec{F}(\vec{x}-\vec{F}(\vec{x}))e^{\sigma(\vec{x}-\vec{F}(\vec{x}))} + \nabla \cdot \vec{F}(\vec{x})e^{\sigma(\vec{x})}\right]\end{aligned} \tag{9}$$

where $\Delta\sigma = \sigma(\vec{x}) - \sigma(\vec{x}-\vec{F}(\vec{x}))$. Note that, since the equations introduced in this section are to be evaluated at different levels of resolution, the integration step is actually dependent on the corresponding voxel size.

3.4. Skeleton Extraction

With the divergence information to hand, we can select the voxels that are likely to contain skeletal points and that will be further subdivided to form the next level in the octree. The skeleton extraction is based on a thinning process guided by the value of the divergence of the momentum field at each voxel.

3.4.1. Divergence Driven Thinning

In [33] Torsello and Hancock show that the field $\rho\vec{F}$ is conservative outside skeletal branches, while its flux through a 1-voxel circle centered on a skeletal point is proportional to $\mathrm{d}l/\mathrm{d}s$, i.e., the ratio between the boundary length $\mathrm{d}l$ and the skeletal segment length $\mathrm{d}s$. This means that theoretically, skeletal branches can be detected by checking voxels with negative divergence of the momentum field. However, adopting any spatial discretization to compute the flux results in a spread-out of the divergence-based signal.

Following Torsello and Hancock, we thin the shape by iteratively removing boundary points in decreasing order of divergence. That is to say that

without any further control on the thinning process we might actually end up introducing holes in the skeleton or even splitting it into disjoint parts.

Recall that one of the key properties of the skeleton is that of having the same topology of the original shape. While for some approaches like the Voronoi-based ones this comes at no cost, the voxel-based methods should always take into account whether if the removal of a voxel would disconnect the shape, introduce a hole or erode it by deleting the endpoints. Unfortunately, when dealing with volumetric objects, ensuring that this property holds is not always an easy task. Hence, in this paper we resort to the voxel classification of Malandain et al. [34], which allows us to efficiently identify removable voxels by exploring the connectivity of their neighborhood. More precisely, Malandain et al. show how to classify a 3D point $\vec{x}$ in a cubic lattice by computing two features. Let $N_n(\vec{x})$ denote the n-adjacent neighbors of $\vec{x}$. Then $C^*(\vec{x})$ and $\bar{C}(\vec{x})$, are defined as follows.

Definition 1. *$C^*(\vec{x})$ is the number of the 26-connected components 26-adjacent to $\vec{x}$ in $B \cap N^*_{26}(\vec{x})$, where B is the set of object points.*

Definition 2. *$\bar{C}(\vec{x})$ is the number of the 6-connected components 6-adjacent to $\vec{x}$ in $W \cap N_{18}(\vec{x})$, where W is the set of background points.*

With this result to hand, we can easily identify the simple points of the medial surface [34], i.e., those points whose removal does not alter the topology of the object. We can then proceed with the thinning process by iteratively removing all simple points in decreasing order of divergence. More precisely, the conditions for a point to be removed are that 1) it is simple, 2) it is not an endpoint and 3) it is characterized by a negative divergence of the

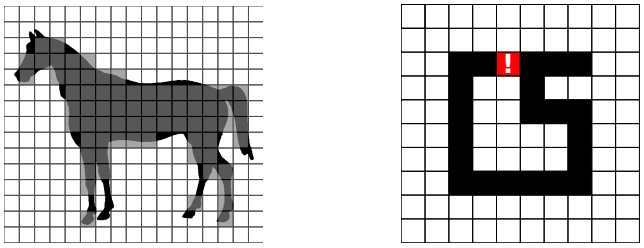


Figure 3: The dilation process is needed to regain details lost at lower levels, although care must be given not to change the shape topology. The left figure shows a two-dimensional example where the discretization of a horse shape results in the loss of those details that are too fine to be captured by the chosen discretization grid. The right figure shows a voxel (marked with an exclamation mark) whose addition would alter the object topology.

momentum field. Note, however, that due to the errors introduced by the discretization of the shape, after the first thinning process the medial surface can be two-voxel thick in certain regions. To ensure thinness at the highest resolution level we further thin the shape by removing all those points that are simple but not endpoints of the surface, regardless of their divergence. Following [3], we decide to restrict our definition of an endpoint to a 6-neighborhood. In this case, it can be shown that a necessary condition for a point to be an endpoint is to have three 6-adjacent background voxels [3].

3.4.2. Skeleton Dilation

With the proposed hierarchical approach, once a voxel is flagged as non skeletal at any level, all its descendants will inherit the property. A problem with this is that fine details might be lost at coarser level, resulting in parts of the skeleton that will be missing at all levels (see Fig. 3). Further, note that the skeletal voxels detected at the coarsest level are not even guaranteed to be connected and, since all further processing is topology preserving, a

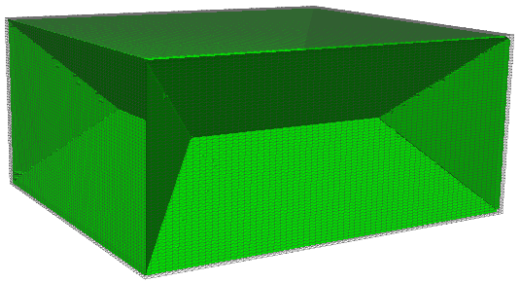


Figure 4: A box shape and its medial surface.

disconnected skeleton will remain disconnected at all levels.

We address the latter problem by keeping only the largest component, while the missing detail is addressed by dilating the skeleton after it has been computed at each new level. This way, once the voxels are small enough to capture the detail, the skeleton will regrow into the missing parts. Note that since the dilation adds new voxels to the current medial surface, we need to ensure that the topology is preserved, thus we dilate only into voxels that would become simple after the dilation (see Fig. 3).

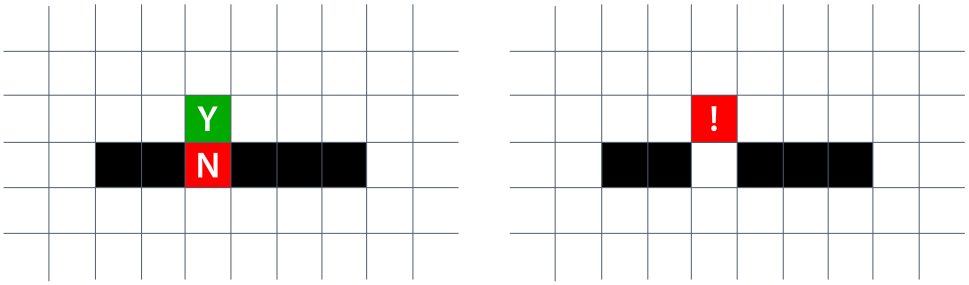
Let V denote the set of voxels before we start thinning the current level of the tree, and let U be the subset of V formed by the boundary voxels of V. We then thin V to reveal the skeletal voxels as previously described. After the thinning step, we check if some voxel $v \in U$ has been selected as skeletal. If that is the case, we dilate it and we compute D, $\vec{F}$, $\nabla \cdot \vec{F}$, ρ, $\nabla \cdot (\rho \vec{F})$ on the dilated set. Then, we apply the thinning process again. The dilation-thinning process is iterated until the thinned skeleton contains no boundary voxels. This process gives us an adaptive dilation which adds only new candidate skeletal voxels with a large value of $\nabla \cdot (\rho \vec{F})$ and thus can be skeletal. Fig. 4 shows the special case of a box shape, together with the

Figure 5: Dilating the skeleton recovers details lost in the coarser levels.

extracted medial surface. Initially, the whole set of voxels in the interior of the cube belongs to V, while the boundary voxels on the faces, edges and vertices of the cube belong also to U. Because of the negative value of the divergence, the voxels on the edges of the cube will survive the first thinning step, and thus will be selected as skeletal. Since these voxels belong to U, they will be dilated, as explained above. Note that U will also be updated in order to include the new dilated boundary of V. However, the following thinning iteration will remove all the voxels in U, and the dilation-thinning process will finally converge. Note that during all these steps we always ensure that the topology of the object is not altered by adding or removing only simple points.

With this improvement, we are able to recover small details that might have been lost during the first discretizations, as well as longer skeletal segments. Fig. 5 shows how critical this procedure is. The eagle model in the figure clearly needs a very dense voxelization in order to capture details such as the claws, or even entire parts such as the wings. With the proposed approach, one can simply start from a lower and less computationally intensive resolution and then refine the extracted skeleton to a certain desired resolution.

Finally, once the iterated dilation-thinning process gives us the final skeleton, we perform one final dilation step to ensure the presence of a complete



(a) Before Thinning (b) After Thinning

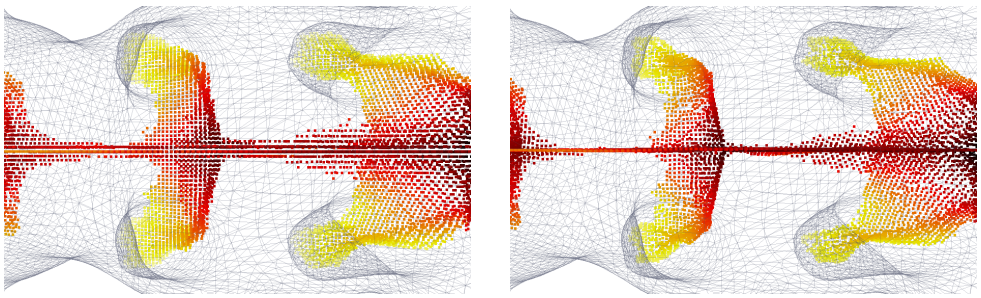
Figure 6: The final iteration of the thinning procedure removes all the simple points which are not endpoints. In this way, however, it can introduce small bumps on the surface, as shown in (b). Here we would like to remove the vertex marked with **Y**, but since this voxel satisfies the endpoint condition it cannot be deleted.

26-neighborhood around the new set of voxels on which we need to compute ρ and $\nabla \cdot (\rho \vec{F})$. At the last resolution level, the final dilation process is substituted with the endpoint-driven thinning that gives us a 1-voxel thick medial surface.

3.5. Medial Surface Alignment

At the end ot the thinning process, we obtain the set of voxels most likely to contain the medial axis, thus placing vertices at the center of the voxels, and deriving the mesh connectivity from the adjacency information of the voxels, will result in a fine approximation of the medial surface in the form of a triangulated mesh. There are, however two sources of noise that limit the quality of the extracted surface, but that can effectively be addressed with a post-processing step.

The first is an artifact due to the limited control over the order in which the thinning process eliminates the voxels. The final iteration of the thinning



(a) Before Alignment (b) After Alignment

Figure 7: Due to the voxelization, the centers of the voxels are very likely to be displaced with respect to the true underlying medial surface (left). Hence, the medial surface alignment procedure is needed to achieve a better approximation of the skeleton (right). Here the color of each voxel is proportional to its distance to the shape boundary.

procedure removes all the simple points which are not endpoints, however, thinning order, and the topology and endpoints preservation rules might prevent us from choosing the correct skeletal voxels as candidate for elimination, while preferring some adjacent voxel which are not endpoints and whose removal doesn't alter the object topology (see Fig. 6). As a consequence, depending on the spatial order of the thinning, we might introduce little bumps on the surface. Due to their formation process, these bumps can be detected easily by comparing their distance to the surface to that of a nearby voxels. Let $d(v)$ be the distance of candidate point v from the shape's surface, let $\vec{F}(v)$ be the gradient of the distance map in v, and let w be the neighbor of v in the direction of $\vec{F}(v)$, i.e., closest to the line $v + t\vec{F}(v)$. If $d(w) > d(v)$ then we v is a bump and we simply remove v from the set of skeletal voxels and mark w as skeletal.

The second limit is a result of the discrete nature of the grid: the centers of the skeletal voxels will be actually slightly displaced with respect to the

true underlying medial surface. We address this issue by allowing the final vertices to move within the voxel from the central position to one that is most likely to lie in the skeletal surface, resulting in a higher precision skeletal mesh even at low voxel resolution (see Fig. 7).

Hence, given a voxel v, we compare the orientation of its velocity field (gradient of the distance transform) with that of its 26-neighbours, in order to determine which voxels lie on the other side of the medial surface. We call this set O_v. Note that thanks to the previous refinement step, we are sure that at least one of v's neighbours will indeed lie on the other side of the medial surface. With the set of voxels to hand, we proceed by computing for each voxel $w \in O_v$ belonging to this set the intersection between the true medial surface and the line connecting w and v. Let s_v and s_w be the surface points closest to v and w respectively, we look for the point $p_w = \alpha v + (1-\alpha)w$ along the line connecting v to w, for which $||p_w - s_v|| = ||p_w - s_w||$, i.e., is equidistant from the closest surface points. This point p_w is likely to be very close to the medial surface, but it displacement from the original position is not limited to the direction of inward motion of the surface and has also a tangential component. We eliminate this by interpolating the position over all the neighbors in O_v.

Fig. 8 illustrates the interpolation process. Let $O_v = \{w_1, \cdots, w_k\}$ and let $p_1, \cdots, p_k$ be the corresponding estimated points on the medial surface, we interpolate between their position using Shepard's inverse distance weighting method [35]. Shepard's interpolation method is a generalized barycentric interpolation approach designed for sparse data. It reconstruct the position

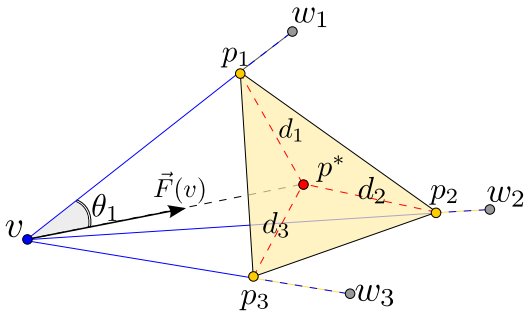


Figure 8: The location of the realigned skeletal point is estimated performing an inverse-distance weighted interpolation of the points p_i obtained finding the bitangent point along the lines connecting v to its neighbors on the other side of the skeletal surface.

of a point as a linear combination of the samples p_i

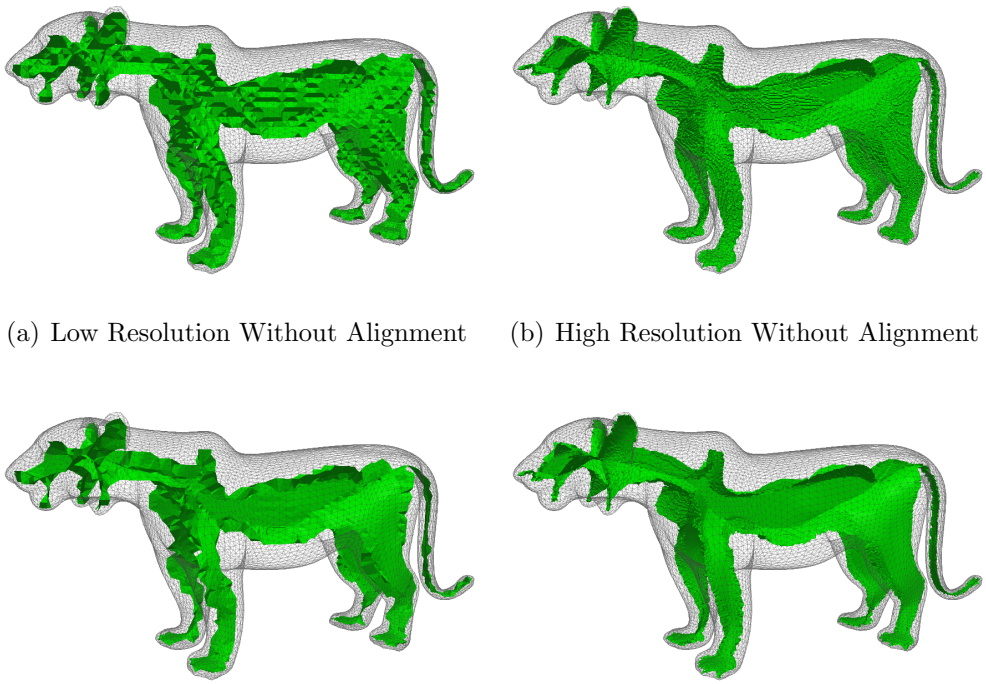
$$p^* = \frac{\sum_{i=1}^{k} w_i p_i}{\sum_{i=1}^{k} w_i} \tag{10}$$

where the weights w_i are a function of the inverse distance d_i of the interpolant p^* to the samples p_i, usually $w_i = \frac{1}{d_i^2}$.

In order to apply Shepard formula we need to estimate the (squared) distances of the points p_i to the interpolant p^*. To this end we make the simplifying assumption that the gradient of the distance map $\vec{F}$ is approximately orthogonal to the medial surface at p^*. Under this assumption we note that $d_i = ||p_i - v|| \sin\theta_i$, where θ_i is the angle between $\vec{F}(v)$ and $v\vec{p}_i$, and thus

$$w_i = \frac{1}{d_i^2} = \frac{1}{||p_i - v||^2 \sin^2\theta_i} = \frac{1}{||p_i - v||^2 (1 - \cos^2\theta_i)} = \frac{1}{||p_i - v||^2 - \left((p_i - v)^T \vec{F}(v)\right)^2}. \tag{11}$$

Fig. 7 shows the result of the alignment procedure on the voxels of a medial surface segment. Perhaps the major advantage of the proposed pro-



(a) Low Resolution Without Alignment (b) High Resolution Without Alignment

(c) Low Resolution With Alignment (d) High Resolution With Alignment

Figure 9: The proposed alignment procedure yields a faster convergence speed, in the sense that we are able to get a good approximation of the real underlying medial surface even at low levels of resolution.

cedure is that it yields a faster convergence speed for the medial surface extraction algorithm. Fig. 9 clearly shows that when we skip the alignment step we need to increase the depth of the hierarchical refinement considerably in order to get a decent approximation of the underlying medial surface. On the other hand, if we align the skeletal voxels as described in this Section we can stop the hierarchical refinement earlier and still get a good result.

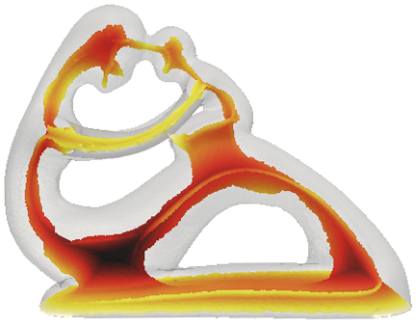
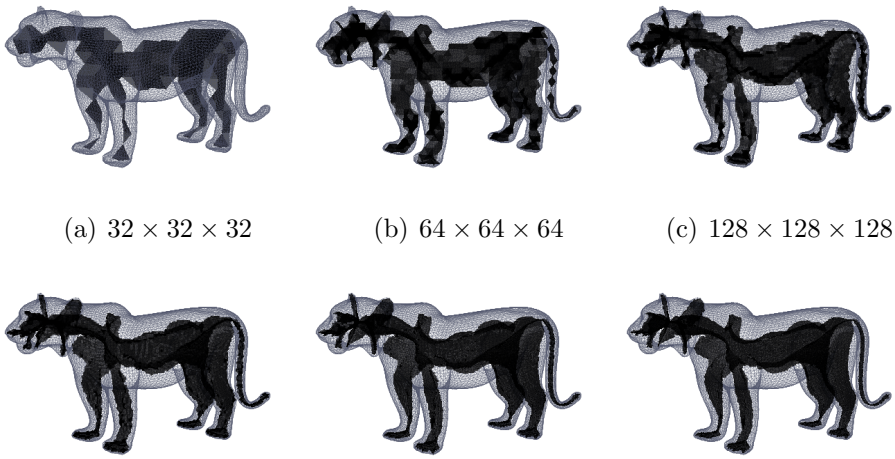


Figure 10: The medial surface of a shape with genus greater than 0.

4. Experimental Results

In this section we evaluate the quality of the proposed algorithm[1] with a wide series of experiments. Here we present quantitative and qualitative comparison with three different approaches, namely the Hamilton-Jacobi algorithm of Siddiqi et al. [18], the multiscale algorithm of Reniers et al. [23] and the Voronoi-based approach of Yoshizawa et al. [5]. Note that the first two methods work on a voxelized 3D shape, while the latter works directly on the mesh. The analysis has been performed on a selection of 40 shapes from the Princeton Shape Benchmark [36] and the SHREC 2010 database [37]. All skeletons are extracted with $res_{min} = 16$ and $res_{max} = 1024$, unless otherwise stated. Fig. 11 shows some sample skeletons extracted at various stages of hierarchical refinement. Note also that the proposed approach works independently of the shape's genus, and our dataset include shapes with genus greater than zero (see for example Fig. 10).



(a) $32 \times 32 \times 32$ (b) $64 \times 64 \times 64$ (c) $128 \times 128 \times 128$

(d) $256 \times 256 \times 256$ (e) $512 \times 512 \times 512$ (f) $1024 \times 1024 \times 1024$

Figure 11: The hierarchical refinement of the medial surfaces. The skeletal points are meshed for ease of visualization.

4.1. Qualitative Evaluation

Here we propose a qualitative evaluation of our algorithm by comparing it with the Voronoi-Based approach of Yoshizawa et al. [5], the Multiscale algorithm of Reniers et al. [23] and the standard Hamilton-Jacobi method. Both the implementations of [5] and [23] were downloaded from the authors websites, while we implemented the Hamilton-Jacobi algorithm simply by dropping the density integration procedure in our framework.

Fig. 12 shows a qualitative comparison between the four methods. The Voronoi skeleton is clearly the noisiest one and in most cases fails to provide an acceptable approximation of the medial surface, although it is computationally significantly less expensive than the other algorithms. The Multiscale

[1]Code available at `http://www.cs.bham.ac.uk/~rossil/#Software`

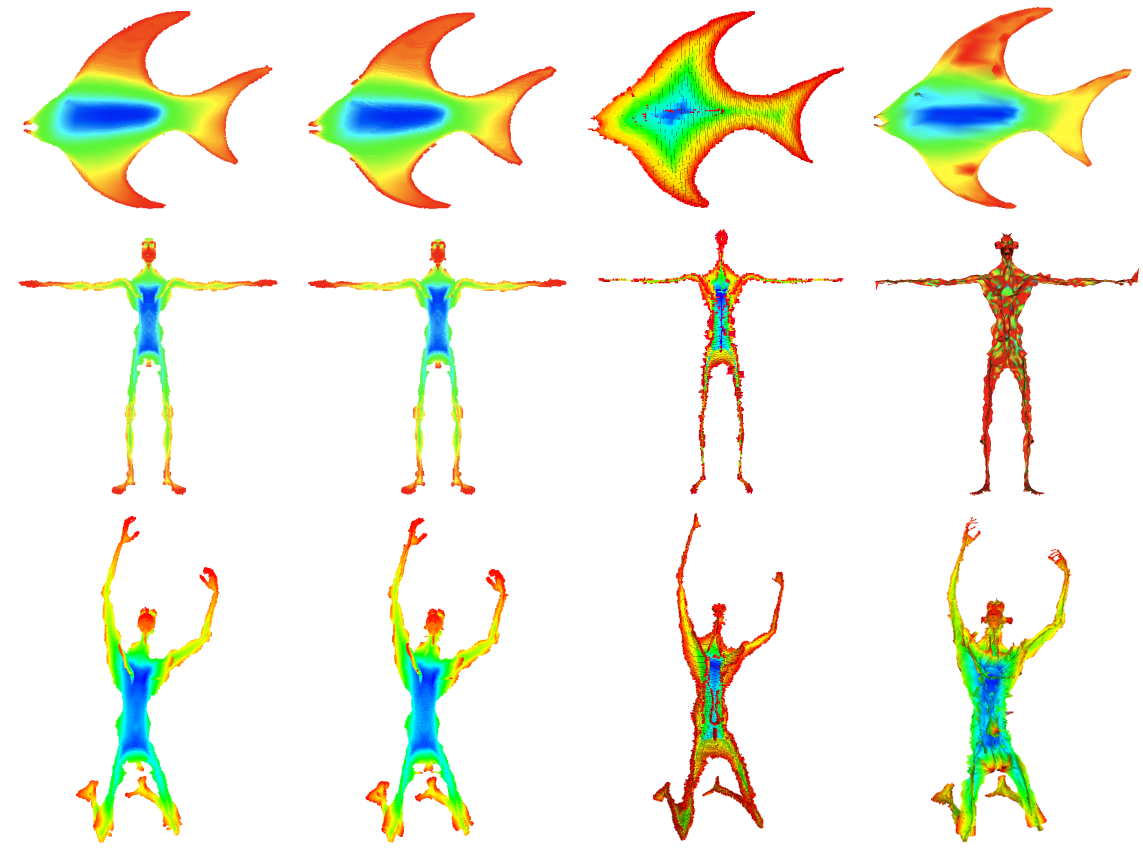
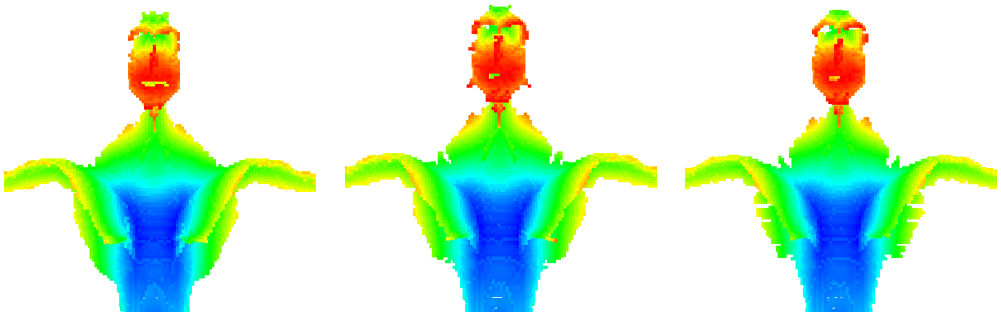


Figure 12: Comparison of our approach against a standard Hamilton-Jacobi algorithm, the Multiscale algorithm of Reniers et al. [23] and the Voronoi-Based approach of Yoshizawa et al. [5]. Note that the voxels are colored according to the distance from the boundary of the shape.

approach on the other hand performs quite well, although due to the complexity of processing a complete voxelization of the shape it was not able to reach the level of detail of our method. Finally, the Hamilton-Jacobi skeletons exhibit a few spurious skeletal segments due to the lack of the correction



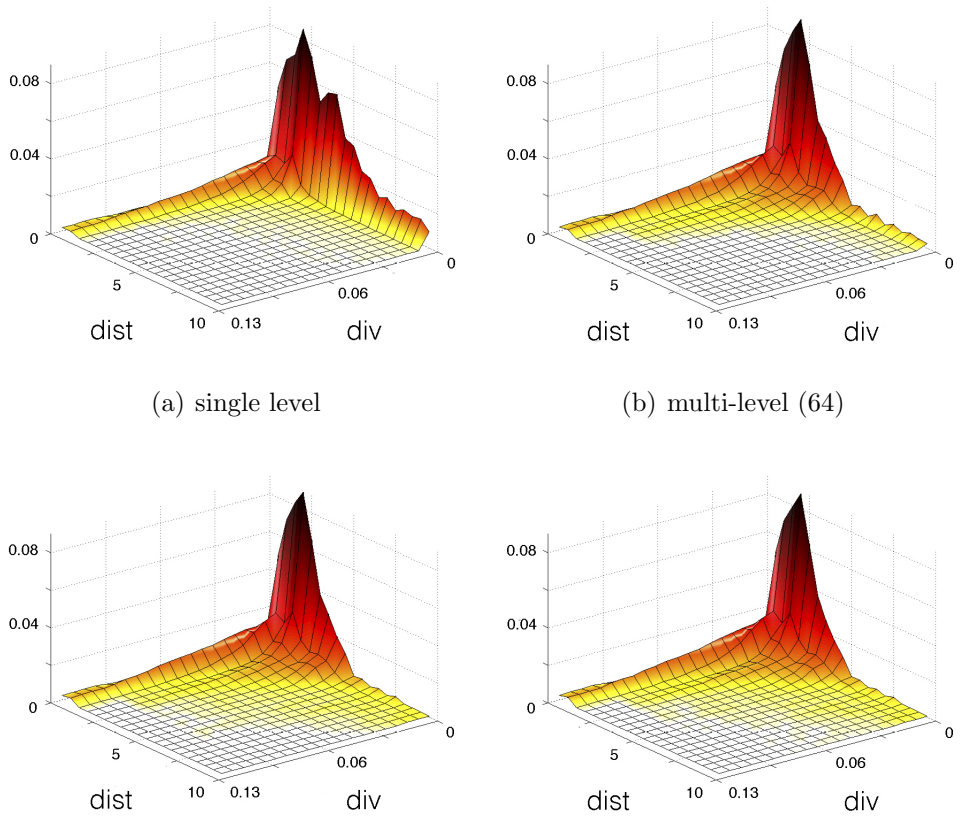
(a) Hierarchical (b) Hamilton Low Threshold (c) Hamilton High Threshold

Figure 13: A magnified view of the head and torso of the medial surface of a human shape. The standard Hamilton-Jacobi algorithm produces spurious segments which can be removed by setting a stricter threshold, although this results in a loss of details of the torso.

of the curvature effects. Fig. 13 provides a magnified view of the torso and head of a selected medial surface extracted with our algorithm and the standard Hamilton-Jacobi method, respectively. As Fig. 13(b) shows, the head of the human shapes contains some spurious segments which are located as expected in the areas of higher curvature. Although setting a stricter threshold eliminates these spurious branches, it also results in a loss of details in the torso, as highlighted in Fig. 13(c).

4.2. Skeleton Localization

The Hamilton Jacobi framework [18, 19] is based on the principle that the (normalized) flux around an infinitesimal area not containing a skeletal branch is zero, while it is non-zero over the skeleton. This guarantees the divergence-based thinning approach to converge to the exact location of the skeleton points. However, as noted in [1], this analysis is true only for the normalized flux and only in the limit. Adopting any spatial discretization



(a) single level

(b) multi-level (64)

(c) multi-level (32)

(d) multi-level (16)

Figure 14: Distribution of the voxels as a function of both divergence and distance to the skeleton. The starting resolution ranges from $128 \times 128 \times 128$ to $16 \times 16 \times 16$, while the maximum resolution remains fixed at $128 \times 128 \times 128$. Note that the points with non-zero divergence are all located near the skeleton, while the points that are far from the skeleton have a value of the divergence equal to zero. We note a decrease of the total number of points that are located far from the skeleton, which is in line with the decrease of total voxels created. We also observe a little noise due to the propagation of numerical errors, which is typical of hierarchical algorithms.

to compute the normalized flux results in non-zero values also outside the skeleton that is proportional to the curvature of the inward evolving front. This results in a spread-out of the divergence-based signal especially close to skeletal endpoints, severely affecting the localization of the skeletal branches and also resulting in the creation of small spurious branches [1]. The curvature correction process [1], on the other hand, localized the non-zero values of the divergence much better, resulting in better localization and avoiding the creation of spurious branches.

In this section we evaluate the localization properties of the skeletons extracted with our algorithm and we compare it against the standard Hamilton-Jacobi approach. To evaluate the localization properties of the density correction we plot the distribution of the voxels as a function of both divergence and distance to the skeleton. In order to evaluate the loss in localization caused by the hierarchical approach, we compare this distribution for shapes at the same target level but at different starting levels. In particular, the histograms in Fig. 14 plot the average distribution of skeletons extracted at the maximum resolution of $128 \times 128 \times 128$, with starting resolutions going from $128 \times 128 \times 128$ (single level), to $16 \times 16 \times 16$ (multi-level (16)), thus all the skeletons were extracted with varying levels of hierarchical refinement.

First we note that when the hierarchical approach goes through more levels, the points tend to be more concentrated around the skeleton. This is to be expected since there is a decrease in the total number of voxels expanded. In general we see that the proposed algorithm yields a good localization of the skeleton, since the points with non-zero divergence are all located near the skeleton, while the points that are far from the skeleton have a value of

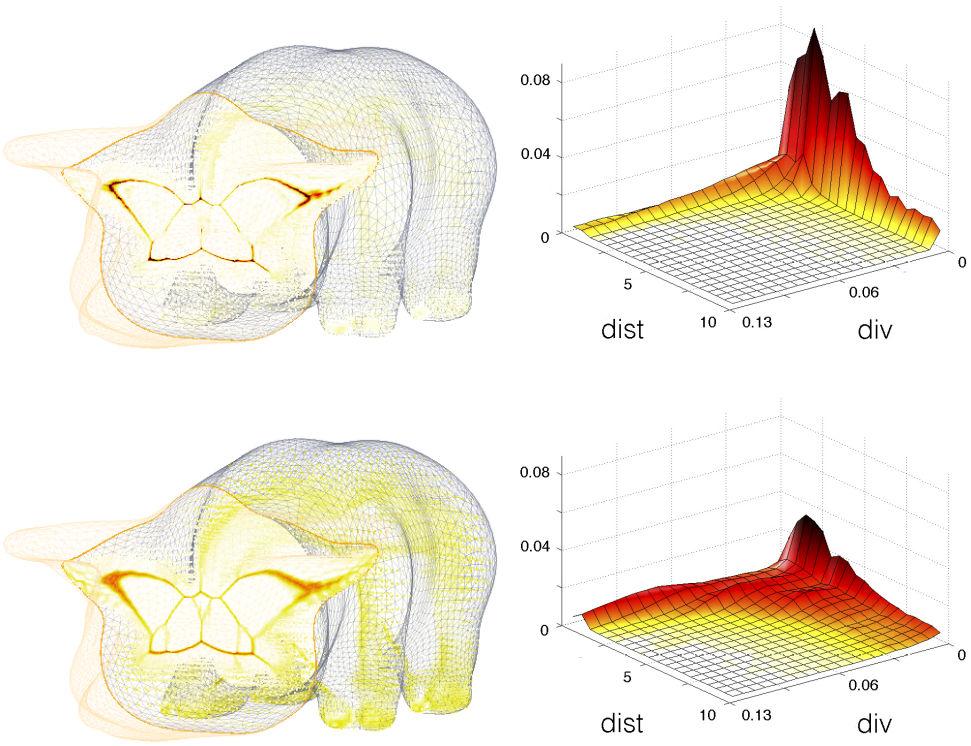


Figure 15: Comparison between the momentum field (top) and the velocity field (bottom). The top histogram shows a good localization of the skeleton, while in the bottom histogram we observe a non-negligible tail of distant points with non-zero divergence.

the divergence equal to zero. However, we do observe a little noise due to the propagation of numerical errors, which is typical of hierarchical algorithms. Nonetheless, the distribution remains tightly peaked, with very few points far from the skeleton with a non-negligible divergence of the momentum field.

Fig. 15 compares the localization of the divergence of the momentum field against that of the velocity field as used by Siddiqi et al. [18]. As previously reported by Torsello and Hancock [1], even in 3D the momentum field localizes the skeleton much more tightly than the velocity field.

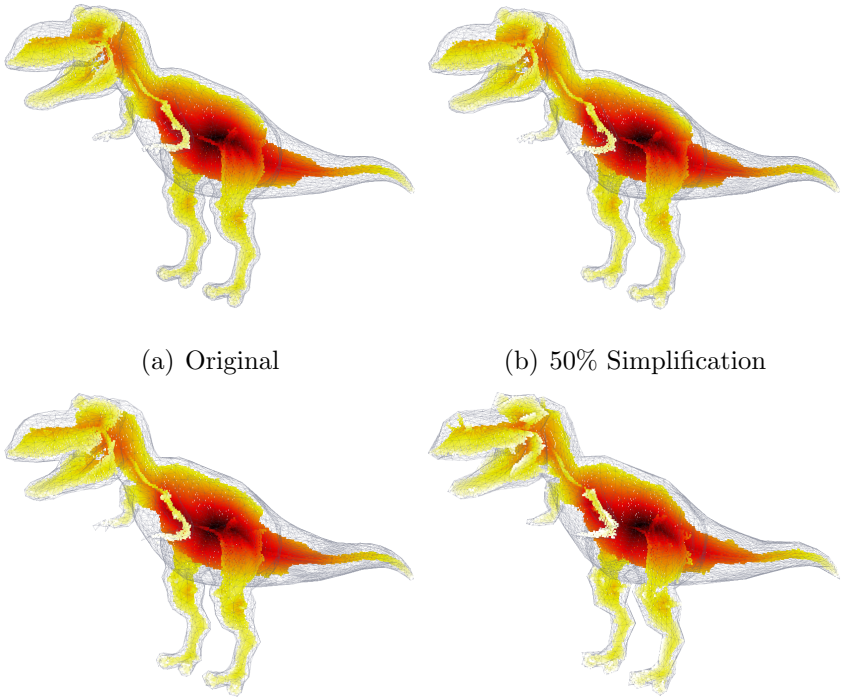
Here we show also a slice of the shape voxelization in order to reveal its interior, where the voxels are colored according to the value of the divergence, i.e., low values correspond to white while high (negative) values correspond to black. Recall that the value of $\nabla \cdot \vec{F}$ in a point p depends on the local boundary curvature and thus its value tends to infinity as p moves closer to a skeleton endpoint, even if p is not skeletal.

As a consequence of this, we observe some blurred areas around the endpoints of the medial surface. On the other hand, in the density-corrected slice we see a much sharper localization of the skeleton.

4.3. Sensitivity to Mesh Resolution

We now evaluate the sensitivity of the proposed approach to different samplings and sampling densities of the mesh. Given a mesh, we compute 3 increasing simplifications where the number of triangles is decreased respectively to 50%, 25% and 10% (see Fig. 16). For each of these, we extract the medial surfaces using our approach, the standard Hamilton-Jacobi one, the Voronoi-Based approach of Yoshizawa et al. [5] and the Multiscale [23] algorithm. We then compute the average nearest neighbour distance between the voxels of the medial surfaces of the simplified meshes and those of the original medial surface.

Table 1 shows the average cost for different levels of simplification and different skeleton extraction methods. As we can see, our approach yields the minimum average distance, hence showing that it is less sensitive to the mesh resolution than the other methods. Note that under a 50% mesh simplification the Hamilton-Jacobi algorithm performs similarly to our method, as by removing 50% of the triangles the mesh quality is only slightly altered,



(a) Original (b) 50% Simplification

(c) 25% Simplification (d) 10% Simplification

Figure 16: Medial surfaces of increasingly simplified meshes extracted, where the number of triangles is reduced to 50%, 25% and 10% respectively. All the medial surfaces are extracted using the proposed algorithm.

and hence we don't observe the formation of new spurious branches. On the other hand, as we further simplify the mesh, its surfaces becomes less smooth and this in turns yields the formation of some spurious segments which induce a higher average nearest-neighbour distance. As expected, the Voronoi-based approach turns out to be the most unstable. It is known, in fact, that in the case of Voronoi-Based skeletonization algorithms the quality of the extracted medial surface greatly depends on the mesh resolution and on how densely it is being sampled. It is hence clear that by simplifying the

Mesh Simplification	50%	75%	90%
Our Method	0.0009	0.0012	0.0017
Hamilton-Jacobi	0.0008	0.0014	0.0024
Multiscale [23]	0.0004	0.0019	0.0021
Voronoi-Based [5]	0.0032	0.0044	0.0051

Table 1: Average nearest neighbour distance between medial surface of the original shape and its simplified counterparts. Note that our methods is less sensitive to mesh quality when compared to the standard Hamilton-Jacobi approach, the Voronoi-Based approach of Yoshizawa et al. [5] and the Multiscale [23] algorithm.

shape we are inevitably altering the quality of the resulting medial surface, as Table 1 clearly shows. Finally the Multiscale algorithm seems to perform slightly better than us when the number of triangles is decreased by 50%, while for higher levels of mesh simplification our approach is achieving better results.

4.4. Robustness Against Noise

A good skeletonization algorithm should also be able to deal with moderately noisy inputs. To this end, we approximate the skeletonization of the diffused shape by smoothing the distance map as in [1]. Hence, given a voxel and its neighborhood, we update the local value of the distance by interpolating the values of the distance function on its neighbors [38].

Fig. 17 shows the robustness to noise of the proposed approach. The results obtained by our algorithm and the Multiscale one are comparable. Note, though, that in the latter the robustness is achieved thanks to a fine tuning of the importance threshold, comes at the cost of losing some detail

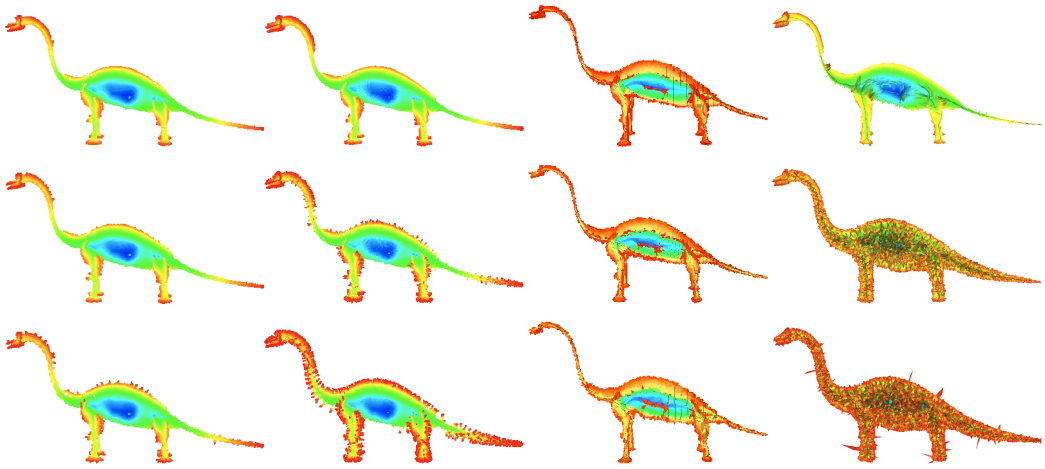


Figure 17: Effects of noise. The first row shows the skeletons extracted from the original object, while the second and the third rows show the skeletons after random vertex displacement of respectively 10% and 20% of the average edge applied to the shape. From left to right: our approach, Hamilton-Jacobi, Multiscale [23] and Voronoi-based [5].

in the finer parts. On the other hand the Voronoi-based algorithm is unable to cope with the noise on the mesh boundary and thus performs much worse than the other approaches. Finally, the presence of noise clearly increases the formation of spurious branches in the Hamilton-Jacobi algorithm.

In order to evaluate quantitatively the robustness to noise, we compute again the average nearest neighbour distance between the medial surface extracted from the original mesh and the medial surfaces extracted from the noisy shapes. The results are shown in Table 2. As the qualitative experiments suggested, the Voronoi-based approach is clearly performing worse than all the other methods, while the Multiscale approach and the proposed algorithm yield similar results, although we know that in the Multiscale ap-

Mesh Noise	10%	20%
Our Method	0.0010	0.0014
Hamilton-Jacobi	0.0013	0.0033
Multiscale [23]	0.0009	0.0018
Voronoi-Based [5]	0.0112	0.0146

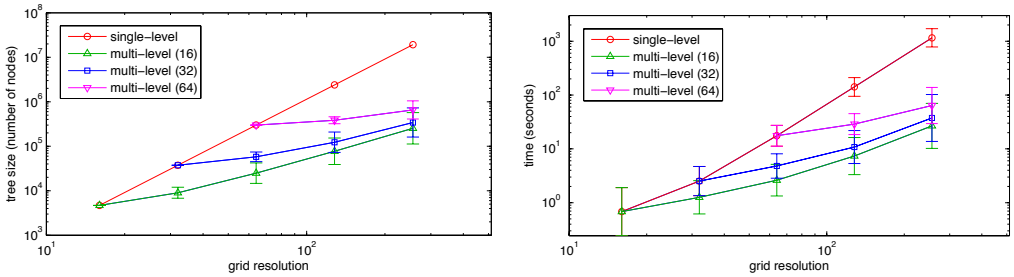
Table 2: Average nearest neighbour distance under increasing mesh noise. Compared to the standard Hamilton-Jacobi approach and the Voronoi-Based approach of Yoshizawa et al. [5], our methods is less sensitive to noise, while it performs similarly to the Multi-scale [23] algorithm.

proach this comes at the cost of losing fine details. Finally, once again the importance of the density correction is highlighted by the decreased performance of the standard Hamilton-Jacobi approach.

4.5. Time and Spatial Complexity

Perhaps the most obvious advantage of our algorithm is the decrease of space and time requirements. As for theoretical complexity, it is governed by the sorting of points with respect to their distance to the boundary that takes place before the density integration, which is $O(n \log(n))$, where n is the number of leaves of the octree. Anyway, while in the case of a complete grid $n = m^3$, where m is the final skeleton resolution, in the proposed approach the growth is only quadratic, i.e., $n = m^2$, since the voxels are refined only around the two-dimensional medial surfaces.

Fig. 18 shows the memory and time requirements for the extraction of a series of skeletons from a wide variety of shapes. Note that because of the higher memory requirements of the complete discretization, the machine



(a) memory requirement (b) execution time

Figure 18: The plots show the memory and time requirements for the computation of a series of skeleton with different levels of refinement. Our approach clearly outperforms the standard algorithm where the space is completely discretized.

on which the experiments were performed, which is equipped with 20 GB of RAM, couldn't afford resolutions beyond $256 \times 256 \times 256$. On the other hand, using the hierarchical approach we could easily reach resolutions as high as $1024 \times 1024 \times 1024$, which would have required 1,073,741,824 voxels if we were to voxelize the shape uniformly.

5. Conclusion

In this paper we presented a novel algorithm for medial surfaces extraction that is based on the density-corrected Hamiltonian analysis [1]. In order to cope with the exponential growth of the number of voxels, we compute a first coarse discretization of the mesh which is iteratively refined until a desired resolution is achieved. The refinement criterion relies on the analysis of the momentum field, where only the voxels with a suitable value of the divergence are exploded to a lower level of the hierarchy. In order to partially compensate for the discretization errors incurred at the coarser levels, a dilation procedure is added at the end of each iteration. Finally we designed a simple alignment

procedure to correct the displacement of the extracted skeleton with respect to the true underlying medial surface. We evaluated the proposed approach with an extensive series of qualitative and quantitative experiments.

References

[1] A. Torsello, E. R. Hancock, Correcting curvature-density effects in the hamilton-jacobi skeleton, IEEE Transactions on Image Processing 15 (4) (2006) 877–891.

[2] M. Pelillo, K. Siddiqi, S. W. Zucker, Matching hierarchical structures using association graphs, in Proc. 5th ECCV, Vol. 2, Springer-Verlag, (1998) 3–16.

[3] K. Siddiqi, S. Pizer, Medial Representations: Mathematics, Algorithms and Applications, Springer, 2008.

[4] J. Bloomenthal, Medial-based vertex deformation, in: Proc. ACM SIGGRAPH/Eurographics symposium on computer animation, (2002) 147–151.

[5] S. Yoshizawa, A. Belyaev, H.-P. Seidel, Skeleton-based variational mesh deformations, in Proc. EUROGRAPHICS, Vol. 26, (2007) 255–264.

[6] D. Reniers, A. Telea, Skeleton-based hierarchical shape segmentation, in Proc. IEEE Int. Conf. on Shape Modeling and Applications, (2007) 179–188.

[7] S. S. Abeysinghe, T. Ju, W. Chiu, M. Baker, Shape modeling and matching in identifying protein structure from low-resolution images, in Proc. ACM symposium on Solid and physical modeling, (2007) 223–232.

[8] H. Blum, A Transformation for Extracting New Descriptors of Shape, in: W. Wathen-Dunn (Ed.), Models for the Perception of Speech and Visual Form, MIT Press, Cambridge, (1967) 362–380.

[9] S. Chang, Extracting skeletons from distance maps, Int. J. of Comp. Sci. and Network Security 7 (7) (2007) 213–219.

[10] G. Bertrand, A parallel thinning algorithm for medial surfaces, Pattern Recogn. Lett. 16 (1995) 979–986.

[11] W. Deng, S. S. Iyengar, N. E. Brener, A fast parallel thinning algorithm for the binary image skeletonization, Int. J. High Perform. Comput. Appl. 14 (2000) 65–81.

[12] P. Golland, W. E. L. Grimson, Fixed topology skeletons, in Proc. CVPR, (2000) 1010–1017.

[13] F. Leymarie, M. D. Levine, Simulating the grassfire transform using an active contour model, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 56–75.

[14] A. Meijster, J. Roerdink, W. H. Hesselink, A general algorithm for computing distance transforms in linear time, in: Mathematical Morphology and its Applications to Image and Signal Processing, Kluwer, 2000, pp. 331–340.

[15] R. Ogniewicz, O. Kübler, Hierarchic voronoi skeletons, Pattern Recognition 28 (3) (1995) 343–359.

[16] R. Kimmel, D. Shaked, N. Kiryati, A. M. Bruckstein, Skeletonization via distance maps and level sets, Comput. Vis. Image Underst. 62 (1995) 382–391.

[17] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, International Journal of Computer Vision 1 (4) (1988) 321–331.

[18] K. Siddiqi, S. Bouix, A. Tannenbaum, S. W. Zucker, The hamilton-jacobi skeleton, in Proc ICCV, (1999) 828–834.

[19] K. Siddiqi, S. Bouix, A. Tannenbaum, S. W. Zucker, Hamilton-jacobi skeletons, Int. J. Comput. Vision 48 (2002) 215–231.

[20] N. Cornea, D. Silver, P. Min, Curve-skeleton properties, applications, and algorithms, Visualization and Computer Graphics, IEEE Transactions on 13 (3) (2007) 530–548.

[21] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, T.-Y. Lee, Skeleton extraction by mesh contraction, ACM Trans. Graph. 27 (2008) 44:1–44:10.

[22] C. Arcelli, G. Sanniti di Baja, L. Serino, Distance-driven skeletonization in voxel images, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 709–720.

[23] D. Reniers, J. van Wijk, A. Telea, Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure,

IEEE Transactions on Visualization and Computer Graphics 14 (2008) 355–368.

[24] Y. Bai, X. Han, J. L. Prince, Digital topology on adaptive octree grids, J. Math. Imaging Vis. 34 (2009) 165–184.

[25] W. R. Quadros, K. Shimada, S. J. Owen, 3d discrete skeleton generation by wave propagation on pr-octree for finite element mesh sizing, in Proc. 9th ACM symp. on Solid modeling and applications, (2004) 327–332.

[26] M. Hisada, A. G. Belyaev, T. L. Kunii, A 3d voronoi-based skeleton and associated surface features, in: Proc. 9th Pacific Conf. on Comp. Graphics and Applications, 2001.

[27] C. B. Barber, D. P. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, ACM Trans. Math. Softw. 22 (1996) 469–483.

[28] L. Rossi, A. Torsello, An adaptive hierarchical approach to the extraction of high resolution medial surfaces, in Proc. Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, (2012) 371–378.

[29] G. V. D. Bergen, Efficient collision detection of complex deformable models using AABB trees, in Journal of Graphics Tools, 2(4), (1997) 1–13.

[30] J. A. Baerentzen, H. Aanaes, Signed distance computation using the angle weighted pseudonormal, IEEE Transactions on Visualization and Computer Graphics 11 (2005) 243–253.

[31] L. Neumann, B. Csébfalvi, A. König, E. Gröller, Gradient Estimation in Volume Data using 4D Linear Regression, Computer Graphics Forum 19 (3) (2000) 351–358.

[32] J. Crank, P. Nicolson, A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, Adv. Comput. Math. 6 (1) (1996) 207–226.

[33] A. Torsello, E. R. Hancock, A skeletal measure of 2d shape similarity, Comput. Vis. Image Underst. 95 (2004) 1–29.

[34] G. Malandain, G. Betrand, N. Ayache, Topological segmentation of discrete surfaces, Int. J. Comput. Vision 10 (1993) 183–197.

[35] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, In Proc. ACM National Conference (1968) 517–524.

[36] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The princeton shape benchmark, in: Shape Modeling International, 2004.

[37] A. Bronstein, M. Bronstein, U. Castellani, A. Dubrovina, L. Guibas, R. Horaud, R. Kimmel, D. Knossow, E. Von Lavante, D. Mateus, M. Ovsjanikov, A. Sharma, SHREC 2010: Robust Correspondence Benchmark, in Eurographics Workshop on 3D Object Retrieval (3DOR '10), 2010.

[38] T. Ju, S. Schaefer, J. Warren, Mean value coordinates for closed triangular meshes, in: ACM SIGGRAPH 2005 Papers, SIGGRAPH '05, ACM, New York, NY, USA, (2005) 561–566.