

Topology, Accuracy, and Quality of Isosurface Meshes Using Dynamic Particles

Miriah Meyer, *Student Member, IEEE*, Robert M. Kirby, *Member, IEEE*, and Ross Whitaker, *Member, IEEE*

Abstract—

This paper describes a method for constructing isosurface triangulations of sampled, volumetric, three-dimensional scalar fields. The resulting meshes consist of triangles that are of consistently high quality, making them well suited for accurate interpolation of scalar and vector-valued quantities, as required for numerous applications in visualization and numerical simulation. The proposed method does not rely on a local construction or adjustment of triangles as is done, for instance, in advancing wavefront or adaptive refinement methods. Instead, a system of dynamic particles optimally samples an implicit function such that the particles' relative positions can produce a topologically correct Delaunay triangulation. Thus, the proposed method relies on a *global* placement of triangle vertices. The main contributions of the paper are the integration of dynamic particles systems with surface sampling theory and PDE-based methods for controlling the local variability of particle densities, as well as detailing a practical method that accommodates Delaunay sampling requirements to generate sparse sets of points for the production of high-quality tessellations.

Index Terms—Isosurface extraction, particle systems, Delaunay triangulation.



1 INTRODUCTION

The problem of surface meshing has been studied extensively in a wide array of applications and contexts by a range of disciplines, from visualization and graphics to computational geometry and applied mathematics. Existing approaches for tackling the surface meshing problem can generally be distinguished by answers to the following questions: How is the original surface *specified*? What is the *representation* of the output mesh? What are the metrics for measuring the *quality* of the mesh? This paper deals with a particular application, which is **the generation of nearly-regular triangle meshes from medical or biological data either for visualizing interpolated quantities or for conducting numerical simulations.**

Generating polygonal reconstructions of biological data is important in a variety of contexts. For example, in bioelectric field problems there is evidence of better source localization when inverse problem solution techniques employ geometry and material properties that conform to those of the patient or subject [41]. Similarly, research in cardiovascular fluid dynamics and image guided medical simulations rely increasingly on models created from images of real vasculature [39, 11]. Recent work proposes using image driven geometry for the analysis of biomolecular functional models [47]. Accurate and compelling visualizations of simulated quantities over surfaces, such as the visualization of flow on manifolds [42], also require high-quality geometry. In the context of mesh generation, visualization and simulation are related in the way they demand not only accurate geometric approximations but also representations that provide for accurate interpolation of other physical quantities across the surface.

The dependence of biomedical applications on image data drives several aspects of this work. First, a surface is not represented in a parameterized form but rather as a volumetric constraint, such as a level-set in a binary or gray-scale mask resulting from an image segmentation. Thus, we focus on isosurfaces. Second, images have finite resolution and cannot capture small, sharp features beyond the resolution of the imaging device, which is limited by inter-sample distance and the point-spread function of the measurement process. Thus, the

proposed method trades geometric accuracy for mesh quality and topological consistency, resulting in numerically useful meshes.

The strategy described in this paper combines work from several disparate fields, establishing a *global approach* to meshing isosurfaces of volume data. This approach generates high quality, closed surface meshes (*i.e.* water-tight) that adapt triangle size to approximate closely the isosurface geometry. Although the proposed method is computationally expensive, the production of consistently higher quality meshes than other approaches makes the method useful for applications that demand very regular triangulations. Starting with a smooth reconstruction of the volume data, the method computes the curvature and medial axis of an isosurface, which are then used to construct a Lipschitz continuous measure of *local feature size*, a fundamental geometric quantity that governs the minimal sampling rate of a surface. A dynamic particle system then positions a large number of samples with inter-point distances that respect this local feature size. The resulting point samples are triangulated using the Delaunay-based meshing algorithm of Dey and Goswami [13], an algorithm that relies on sampling densities that exceed the minimal sampling rate. Thus, this paper proposes a new meshing pipeline while also addressing the question of how, in practice, to make use of the fundamental work in surface sampling theory by detailing an algorithm for reliably achieving specific sampling densities.

The remainder of the paper proceeds as follows: the next section provides a brief overview of some recent work in the literature that relates to isosurface meshing and several techniques that are integral to the proposed method. This is followed by sections that describe the meshing pipeline, present results and comparisons, and finish with some conclusions.

2 RELATED WORK

Early work on isosurface meshing focuses on efficiently generating approximate meshes, used mostly for visualization. The well-known *marching cubes* algorithm [46, 25] robustly produces first-order approximations quickly, but can construct incorrect topologies that do not accurately adhere to the isosurface geometry, and may not be water-tight. Furthermore, quality measures of triangle regularity can be arbitrarily poor while vertex valences vary greatly, independent of the input surface geometry. Improvements to marching cubes are numerous, including improvements for better geometric approximations and closed meshes [30].

One general class of strategies for achieving higher quality surface polygonizations is to start with a mesh that is either coarse or of low quality and, through some combination of mesh refinement, edge-

• M. Meyer, R.M. Kirby, and R. Whitaker are with the Scientific Computing and Imaging Institute at the University of Utah, E-mail: miriah,kirby,whitaker@sci.utah.edu.

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007. Published 14 September 2007.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

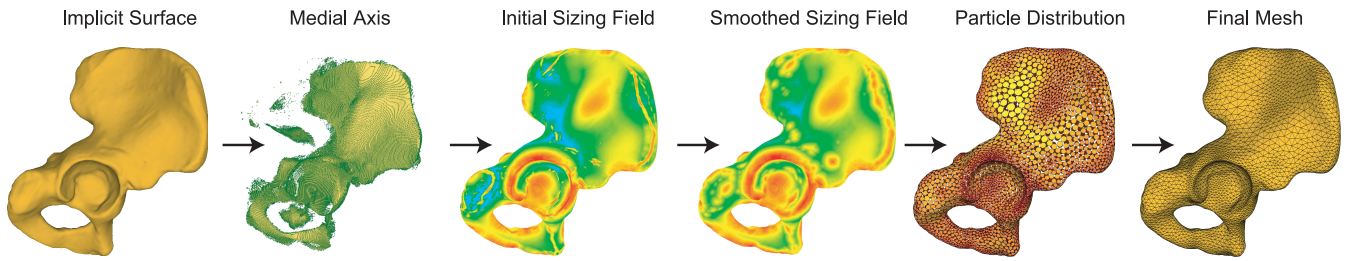


Fig. 1. The proposed mesh generation pipeline using a dynamic particle system. First, a medial axis is computed from a distance transform of an implicit surface; next, an initial sizing field is built from the local feature size and radius of curvature; a smoothed sizing field is then generated by limiting the gradient of the initial sizing field; particles sample the sizing field and distribute themselves accordingly; and finally, the particles are triangulated using a Delaunay surface reconstruction algorithm.

swapping, or vertex repositioning, incrementally improve the mesh geometry and triangle quality. For example, Velho [43] proposes a curvature-based refinement method for improving the geometric accuracy of a marching cubes mesh, but does not fully address the issue of triangle quality. Wood *et al.* [45] propose another strategy that first constructs a coarse, topologically correct mesh that is then smoothly refined, producing higher quality triangles. Other researchers have proposed refinement algorithms without the need for a base mesh [9]. In general, however, the strategy of refining a mesh to improve triangle quality produces the inefficiency of a great many samples that are dictated by mesh quality rather than the geometry of the underlying surface (for convergence rates see [9]).

Another general strategy for quality isosurface meshing is to start from one or more seed points and grow triangles in the form of an expanding, or *advancing*, front [23, 16]. The basic approach is quite fast and can produce high-quality triangles, especially when triangle size is adapted to local surface curvature [20, 33]. The advancing front, however, must be able to detect and avoid self-intersections, causing triangle sizes along merging fronts to be determined not only by surface geometry, but also by the geodesic curvature (curvature in the tangent plane) of the moving front. This problem becomes even more acute when the data contains wavefronts that collide from opposite directions, which is unavoidable for certain topologies or shapes. Thus, advancing front algorithms must have built-in heuristics, such as wavefront smoothing [34], or special triangulation schemes (*e.g.* edge swapping) that deal with collapsing or colliding wavefronts, often at the expense of triangle quality in those areas.

A third approach to surface meshing is to generate an unorganized set of surface samples and use algorithms from the computational geometry literature to create a Delaunay tessellation of the points. Early work in the field provides the algorithmic foundations for producing solid Delaunay triangulations in 2D and 3D [32, 15, 10], complemented by literature on the theory and methods for extracting the surface manifold [3, 2, 5, 12]. These methods employ a compelling bottom-up approach for constructing edges and faces from non-local properties of a point set, guaranteeing closed, non-intersecting meshes. Generating the set of surface samples, which determine the topology of the resulting tessellation *and* the quality of the resulting triangles, is difficult and generally treated as either a separate problem, or as part of an adaptive scheme for iteratively improving mesh quality [9, 14].

This paper proposes a surface sampling and triangulation algorithm that relies on fundamental sampling requirements associated with Delaunay surface reconstruction schemes. Amenta *et al.* provide the quantitative requirements, based on surface geometry, for 2D curves [3] and 3D surfaces [2], such that a unique Delaunay tessellation exists from which a subset of edges or faces have a topological equivalence to the underlying surface. After distributing a set of points based on these sampling requirements, we use methods that generate water-tight Delaunay reconstructions from such samples [13, 6] to create a tessellation.

The core of the Delaunay sampling requirements relies on a characterization of surface geometry that depends on non-local information. Given a smooth surface $F \subset \mathbb{R}^3$, a sufficiently dense sampling P is one such that for any point $s \in F$ the Euclidean distance between s

and the closest sample point $p \in P$ is no greater than ϵ times the *local feature size* at s . Any discrete set of surface points P that meets this requirement is an ϵ -sample of F . The current theoretical (3D) results show $\epsilon = 0.06$ is sufficient [2], however, empirical results indicate that the actual bound might be looser, and several authors have conjectured that $\epsilon = 0.5$ may be closer to the necessary bound [4].

The definition of local feature size (*lfs*) is an important aspect of these results and of the proposed algorithm. The *lfs* of a point $s \in F$ is defined as the distance from s to the nearest point on the medial axis (MA) of F . The MA has been heavily studied in the literature [22, 19] in the context of shape modeling, computational geometry, and computer vision. Although the MA has several formal definitions and many interesting and important characteristics, it is sufficient for the proposed method to consider the MA of a surface as (the closure of) the set of points $M \subset \mathbb{R}^3$ such that the nearest point on the surface to $m \in M$ (*i.e.* $\min_{s \in F} |s - m|$) is not unique. Alternatively, several authors define the MA as the set of points where there exists a sphere that does not intersect the surface and is tangent to the surface in more than one location.

The relationships between the MA, local surface geometry, and sampling requirements are important in several ways. First, the cotangency definition implies that the *lfs* of $s \in F$ is *no greater than* the local radius of curvature at that point. The radius of curvature is then an upper bound, as is the sampling condition, and is therefore not a suitable proxy for the *lfs*. For instance, the feature size can be very small on thinly separated, flat objects that have a very large radius of curvature. Second, the *lfs* condition is necessary for establishing the correct topology among an unorganized set of points. If the topology is somehow known *a priori* (*e.g.* via continuity in an advancing front) the sampling density could be much more sparse. However, one common use of a surface mesh is for the construction of a body fitting tetrahedralization [1] where it is important that the triangles, which form the faces of the corresponding tetrahedra, conform to the global solid geometry.

Algorithms for constructing an ϵ -sampling of a surface F are not immediately evident from the sampling theorems or mesh generation algorithms. Although several related schemes propose methods for sampling surfaces with less-strict or slightly different bounds [8, 9], the ϵ -sampling requirement provides guarantees necessary for subsequent simulations due to the side effect of respecting local as well as global object shape. Among the contributions of this paper is a practical scheme for generating sets of surface points that closely conform to the ϵ -sampling requirements.

The proposed method uses a sizing field generated from geometric information of F , the creation of which is described in subsequent sections, to control a system of dynamic particles which form the surface point samples. Dynamic particle systems, first proposed for geometric modeling by Witkin and Heckbert [44], rely on a set of markers that are constrained by an implicit function and move based on a local potential related to neighboring particle distances, resulting in even distributions across the surface. The particle interactions can be modified to produce adaptive distributions by using compact and quasi-scale invariant potential functions along with mechanisms for robustly controlling particle densities [26, 27]. This paper combines these adap-

tive particles with a sizing field based on the lfs to generate surface point samples which are tessellated by the Delaunay-based algorithm TIGHT COCONE [13]. We discuss relevant quality measures for such meshes and demonstrate the effectiveness of the algorithm relative to these metrics.

3 MESH GENERATION WITH PARTICLE SYSTEMS

The goal of the proposed system is to generate nearly-regular triangular meshes of isosurfaces. By adaptively distributing a set of dynamic particles such that their positions conform to an ε -sampling requirement, the particles can be used to generate a Delaunay surface mesh that corresponds to the geometry and topology of the isosurface. To achieve this goal, the proposed method consists of several steps: (1) computation of a MA approximation to determine the local feature size; (2) creation of a sizing field to specify the desired distances between particles; (3) adaptive distribution of particles across the isosurface; and (4) triangulation of particle positions to create a polygonal reconstruction of the isosurface. Figure 1 depicts this pipeline.

The proposed system specifically targets volumetric data, such as segmentations from MRI or CT scans. From the discretely sampled data stored over a regular grid, we utilize smooth reconstruction filters to generate a continuous, differentiable implicit function F [28]. Specifically, a 4^3 cubic B-spline or Catmull-Rom spline is used for approximating or interpolating the function, respectively, while the first and second derivatives of the cubic B-spline are used for computing the gradient and Hessian. These combinations of filters are ideal for applications that measure curvature values [21]. The geometric information computed for the data (discussed in the following sections) is also stored over a lattice and sampled using reconstruction kernels.

3.1 Local Feature Size

As described in Section 2, the lfs is the distance to the MA of a surface. Accurate computation of the MA is a challenging research problem, and numerous approaches have been proposed for its solution. One approach presented in the computer vision literature is to detect discontinuities (*i.e.* shocks) in a distance transform of a surface. Detecting the shocks, and hence the MA approximation, in the distance transform is numerically tricky due to the discontinuities in the derivatives at these points. Siddiqi *et al.* [37] propose measuring the divergence of the gradient of the distance transform, where high values indicate a significant change in the field, and thus a MA point. Another approach by Persson [31] fits local quadratics over the distance transformation grid, looking for places where these functions intersect. A set of heuristics then determine whether an intersection point should be included in the MA approximation.

For this work we have developed a medial axis detection algorithm for more general implicit surfaces which is moderately robust to free parameters, gives subgrid accuracy, and does not require the thinning or post-processing of similar methods [37]. This scheme relies on the *foot point* method, which is the nearest point p on a surface to a given point q , and can be found using gradient descent by the method described in [17]. Here we consider only the MA proper, and not the singular points where the MA terminates. The line segment defined $\bar{p}q$ is perpendicular to the surface, and every point on $\bar{p}q$ has p as its foot point. This line segment forms a *characteristic*, which is the path of a surface point that moves inward/outward in the direction of the surface normal (to within a sign difference). As we proceed from the surface along an inward or outward characteristic, the foot point of each point along that path remains the starting point for the characteristic *until* the characteristic intersects the MA — once the characteristic intersects the MA, the position of the foot point changes. The algorithm for detecting the MA is as follows. For each point on the grid q find the foot point p , then find the point q_* along the characteristic (away from the foot point) that intersects the current voxel (far face of the cube). Find the footpoint p_* associated with q_* . If the angle between the line segments $\bar{p}q$ and \bar{p}_*q_* is greater than some small threshold α , then the segment $\bar{p}p_*$ crosses the MA. The position of the MA along $\bar{p}p_*$ can be found by using a first-order (tangent plane) approximation to the

surface at the point p_* . We use $\cos(\alpha) = 0.9$ for all of the results in this paper.

After constructing M , a lfs field $\lambda(\mathbf{x})$ is created by finding the distance to the closest medial axis point $m \in M$ at the grid nodes. For efficiency, we restrict this field to the subset of grid nodes that bound the isosurface. The λ field needs only to be a conservative estimate of the distance to the true MA as lfs (and curvature, which will be discussed in the next section) provides an upper bound on the distance between particles in the proposed system; anything less than, or equal to, the true distance will drive the final distribution of particles to be an ε -sampling.

The accuracy of the λ field fundamentally relies on the underlying accuracy of the medial axis detection algorithm, which is itself a sampling problem. Creating the medial axis requires a sampling of the data field, with where those samples are chosen dictating the accuracy of the medial axis detection. The problem thus cycles between choosing samples to detect the medial axis, and using the detected medial axis to determine where the samples should be to accurately capture the true medial axis. For this work, we intentionally break the cycle by relegating the accuracy of the system for determining the lfs , and ultimately achieving a true ε -sampling of the surface, to that of the accuracy of the underlying medial axis detection algorithm. As research into medial axis detection progresses, the proposed sampling method can make use of new algorithms to achieve more accurate results.

3.2 Sizing Field

The sizing field is the mechanism by which the particle system adapts its distribution to meet an ε -sampling requirement. There are two geometric quantities, lfs and radius of curvature, and two parameters, ε and δ , that govern the construction of the field.

To establish an initial sizing field $h_0(\mathbf{x})$, the λ field is compared to the radius of curvature at each grid node in a narrow band around the isosurface. The radius of curvature is calculated as $1/|\kappa_{max}|$ — the absolute value of the maximum curvature can be computed directly from the Hessian of F [21]. The initial sizing field is given as:

$$h_0(\mathbf{x}) = C \min(\lambda(\mathbf{x}), 1/|\kappa_{max}(\mathbf{x})|) \quad (1)$$

where C is a constant based on ε that is discussed and defined in the following paragraphs. Including the radius of curvature in Equation 1 helps to ensure that small surface features that may not have been captured in M appear in the sizing field construction.

Although h_0 contains most of the core geometric information about a surface necessary for describing an ε -sample, it is not suitable on its own for regulating particle distances for two reasons: first, the Delaunay sampling theory indicates that some fraction of the lfs is required for topologically correct reconstructions; and second, to achieve high quality triangles across the entire mesh, the gradient of the sizing field must be limited to ensure smooth, gradual changes in triangle size. These two characteristics are controlled by the user defined parameters ε and δ , which modify h_0 to create an (ε, δ) -sizing field $h(\mathbf{x})$.

Multiplying h_0 by twice the ε parameter, such that $C = 2\varepsilon$, causes the sampling to be a fraction of local feature size, *i.e.* an ε -sampling. The Delaunay sampling requirement [3] specifies ε for some point on the surface *other* than the sample points, thus we include this implied factor of two in C , and the literature indicates that $\varepsilon = 0.5$ may be a loose upper bound for ε . The second parameter, δ , is used to limit the gradient of h_0 such that the values in the resulting field h will not change faster than δ . Thus, δ dictates how quickly the edge lengths of neighboring triangles can change — $\delta < \varepsilon$ will generally produce well-shaped triangles. This limiting produces a δ -Lipschitz field, an important property for smooth triangle gradation that limits how quickly the field can change. To limit the rate of change of h over its grid we use the following discrete operator, operating on a lattice sampling of our field $h_{ijk} = h(\mathbf{x})$ at a grid node position \mathbf{x} , which is shown by Persson [31] to generate a δ -Lipschitz field:

$$h_{ijk}^{n+1} = h_{ijk}^n + \Delta t (\min(0, \delta - \Delta_{ijk}^+)) \quad (2)$$

where

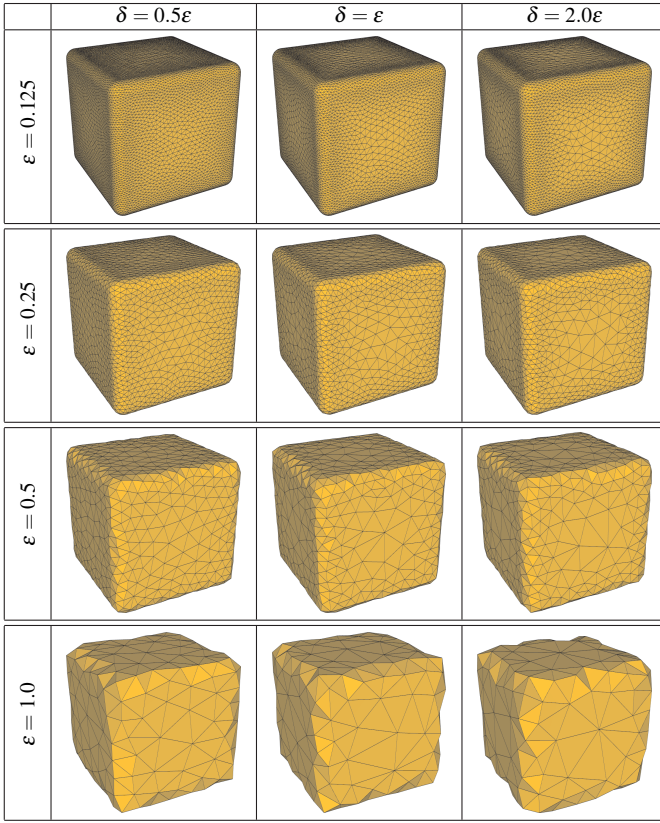


Fig. 2. Illustrative comparison of mesh quality and number of triangles for varying values of ϵ and δ , the user defined parameters in Equations 1 and 2 respectively. The ϵ values vary down the columns while the δ values vary across the rows. Smaller values of ϵ produce more geometrically accurate meshes, while smaller values of δ generate more regular triangulations — smaller values of either parameter inherently increases the overall number of triangles.

$$\Delta_{ijk}^+ = [\max(D_x^- h_{ijk}^n, 0)^2 + \min(D_x^+ h_{ijk}^n, 0)^2 + \max(D_y^- h_{ijk}^n, 0)^2 + \min(D_y^+ h_{ijk}^n, 0)^2 + \max(D_z^- h_{ijk}^n, 0)^2 + \min(D_z^+ h_{ijk}^n, 0)^2]^{1/2} \quad (3)$$

and where D^+ and D^- are the forward and backward difference operators respectively with subscripts denoting the axes along which they are operating. In our implementation of the system we consider the limiting of the gradient to have converged when the maximum relative change of any grid node is less than 10^{-5} .

While the lfs function λ is 1-Lipschitz [32], the inclusion of the radius of curvature causes h_0 to lose this property. We have found, however, that the initial sizing field h_0 is nearly 1-Lipschitz, producing a final sizing field h which is $\min(2\epsilon, \delta)$ -Lipschitz. We have experimented with a range of values for both ϵ and δ , and present an illustrative example in Figure 2 to provide intuition on each parameter's role in the final mesh quality. This example visually emphasizes the balance between geometric accuracy, the number of triangles, and triangle quality. The rounded box is a level-set of a sampled analytic distance transform of box, where the faces are planar, the edges are cylindrical, and the corners are spherical. The surface is reconstructed using a Catmull-Rom spline kernel and the first and second derivative of a cubic B-spline.

3.3 Distributing Particles

Using the particle system framework proposed in [27], a set of dynamic particles can be controlled by h such that their final distribution meets the sampling requirements for F . To do this, we initialize the system with a set of particles, the positions of which are determined

from a marching cubes triangulation to ensure that disconnected parts of the isosurface are seeded with particles. The particles are then projected onto F using a Newton-Raphson gradient descent method. Once on the surface, each particle is associated with an individual potential function which induces inter-particle forces that push them towards lower, local energy states. To control the sampling density, we scale the distances between particles — which determines the magnitude of the inter-particle forces — by the value of h at each particle's position. The distance between particles p_i and p_j becomes:

$$d_{ij} = \alpha_{ij} |\mathbf{x}_i - \mathbf{x}_j| = d_{ji} \quad (4)$$

where α_{ij} is defined by h (evaluated at particle positions \mathbf{x}_i and \mathbf{x}_j using trilinear reconstruction kernels within a voxel) as follows:

$$\alpha_{ij} = \alpha_{ji} = \frac{\beta}{\min(h_i, h_j)} \quad (5)$$

with $\beta = 0.5/\cos(\frac{\pi}{6})$. The particles are then iteratively moved to lower local energy states until the system reaches an equilibrium.

Equation 5 scales the effective distance between particles based on an ideal hexagonal packing across a flat surface where the region of influence of a particle ends at the closest two-ring neighbors. The value of β is derived from this ideal packing, and allows for a population control mechanism to be defined that adds and removes particles based on their energy, driving the system towards an ideal packing. Because h_0 establishes an upper bound on the allowed distance between particles for meeting an ϵ -sampling requirement, using the minimum sizing field value of any pair of particles establishes a conservative sampling, as do the convex, linear interpolation kernels that reconstruct h at arbitrary particle locations.

There is, however, a caveat to producing an ϵ -particle sampling. The particle system uses Euclidean distances to compute inter-particle forces and energies, as opposed to the more accurate (and computationally expensive) geodesic distance. The distance approximation causes particles in a two-ring neighborhood to become influential, adversely affecting the population-control mechanism for obtaining ideal neighborhood packings. This artifact, however, is bounded by ϵ , which allows for a contraction factor g to be introduced to the system to counter-act the effects. Figure 3 shows the distance contraction for particles equally spaced across a circle. The distance between each adjacent particle is d — the distance between p_0 and p_2 , however, is not $2d$ as the ideal packing model assumes. Instead, it is:

$$D = 2d \left(1 - \frac{(d\kappa)^2}{4} \right)^{1/2}. \quad (6)$$

In the proposed system, $d \leq \epsilon/\kappa$ because the lfs is bounded from above by the radius of curvature. Thus, we can bound the contraction parameter, g , as:

$$1 \geq g \geq \left(1 - \frac{\epsilon^2}{4} \right)^{1/2}. \quad (7)$$

As ϵ goes to zero, g approaches one because the surface becomes locally more and more planar as the distance between particles shrinks. These bounds on g attest that the contraction effect cannot get arbitrarily worse as the surface features become smaller; the worst case is bounded by ϵ .

To counteract the contraction of distance to neighboring particles, we inversely scale the sizing field as $h \leftarrow h/g$. We empirically determined a value of g by observing histograms of the ratio of triangle edge lengths to the average of h at each edge's vertices. Over a range of ϵ and δ values for a variety of data sets the shape of the histograms were visually identical except for the tails, which contain a relatively small number of outliers — an example of one histogram is shown in Figure 3. Based on our observations, we determined a conservative estimate of g to be 1.5. The inclusion of g creates a final mesh where the length of virtually every edge is at most $2\epsilon h_0$. These results are detailed in Section 4.

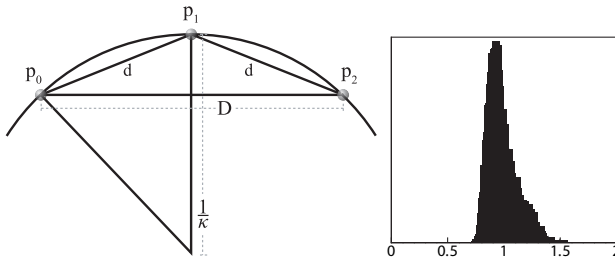


Fig. 3. The effects of curvature cause the two-ring neighboring particles to become closer than $2d$ (left). This effect is bounded by ε , which allows for a scaling parameter to be introduced into the system. We empirically determined this value by studying histograms of the triangle edge lengths versus h , such as that of the pelvis reconstruction (right).

Table 1. Details of each data set, including size of the volume, values for ε and δ , minutes required to distribute the particle system, and resulting number of mesh vertices and triangles.

Data	Volume Size	ε, δ	Time (mins)	Vertices	Triangles
brain	149x188x148	0.5, 0.3	41	91702	182188
pelvis	271x390x282	0.75, 0.5	1	4992	9984
skull	256x256x115	0.5, 0.3	232	212188	424588
vessel	265x265x171	0.5, 0.3	280	287990	576493

3.4 Triangulation

To triangulate a distribution of particles, we use the water-tight Delaunay triangulation method TIGHT COCONE [13] – a free version of the software is available from the author’s website. TIGHT COCONE first builds an initial Delaunay triangulation of the input point set. A manifold surface is then extracted from all the triangles that intersect a thickened surface built from the approximation of the points’ normal vectors. Where the normals cannot be faithfully approximated, holes appear in the extracted mesh. TIGHT COCONE applies a final selection criteria to the original triangulation to fill in these holes.

4 RESULTS AND DISCUSSION

In this section we present results from the proposed system for generating isosurface meshes of a variety of biological data sets. The first two tessellations, shown in Figures 6 and 7, are of a pelvis and brain generated from binary segmentations that have been smoothed with a Gaussian kernel ($\sigma = 1.5$). Figure 8 illustrates a skull extracted from gray-scale CT data, and Figure 9 depicts vasculature represented by the zero-set of a distance transform generated by an anisotropic smoothing algorithm [29]. All four surfaces are reconstructed with approximating cubic B-spline kernels.

The proposed method was run on a P4 3.2GHz CPU with 2.0Gb of memory. The generation of the medial axes and the λ grids took on average about an hour for each data set while limiting the gradient field of h_0 to generate h required several minutes. Run times to distribute the particle systems, along with the resulting mesh dimensions are presented in Table 1. As mentioned in Section 3.4, we use TIGHT COCONE [13] to triangulate the particle distributions — the tessellations required on the order of several minutes.

The stated goal for this work is the generation of isosurface meshes suitable for simulations and the interpolation of data. To measure the quality of our results for these purposes we draw upon ideas established in the finite element literature which characterize the role of mesh quality in the simulation accuracy [18]. Here, we briefly discuss the interplay between geometric accuracy and the error of a finite element solution computed over a mesh.

Given a domain Ω and a partial differential equation (PDE) that operates on a solution u that lives over Ω , the standard finite element method attempts to construct a geometric approximation $\tilde{\Omega} = \mathcal{F}(\Omega)$ consisting of a tessellation of polygonal shapes (e.g. triangles and quadrilaterals for 2D surfaces) of the domain Ω , and to build an ap-

proximating function space $\tilde{\mathcal{V}}$ consisting of piece-wise linear functions based upon the tessellation [18]. Building on these two things, the goal of a finite element analysis is to find an approximation $\tilde{u} \in \tilde{\mathcal{V}}$ that satisfies the PDE operator. The details of how this is accomplished are beyond the scope of this work. The important points, however, are that a finite element analysis must balance geometric error and approximation error while respecting stability constraints (e.g. as discussed by Babuska and Aziz [7]), and that these errors are connected through the tessellation that is generated. The space of functions from which \tilde{u} is generated depends on the type of elements that exist in $\tilde{\Omega}$, thus the quality of the solution approximation is not only related to the accuracy of $\tilde{\Omega}$ for approximating Ω , but also to the geometric properties of the mesh elements. In the L^2 norm, the accuracy of \tilde{u} is bounded by a constant that includes angles of triangular elements. Babuska and Aziz [7] show that if the largest triangle angle is bounded away from 180° , the finite element method converges as the triangle size decreases. Shewchuk [36] notes that small angles are preferable over large angles, so long as the largest angles are not too large, and extends these results to provide functions that guide mesh generation and refinement algorithms toward the production of high quality finite element tessellations. A common quality metric used in the literature for measuring this relationship of element angles is the ratio of the radii of the inscribed circle to the circumscribing circle of a triangle, r_{in}/r_{circ} . This metric penalizes triangles containing small angles, with the worst ratios going to triangles that also contain a large angle.

The proposed system addresses both aspects of geometric quality posed by the finite element method. First, the accuracy of the tessellation for capturing the topology of the domain is guaranteed by the Delaunay reconstruction algorithms for ε -distributions of particles. We quantify the ability of the proposed system to meet this requirement by computing the ratio of triangle edge lengths versus the average of h_0 at the edge vertices. In Figure 4 we present histograms of the results for each data set. The pelvis mesh contains no edges larger than h_0 dictates, and virtually every edge in the other three tessellations meets the sampling requirements defined in h_0 — less than 0.004% of the triangles in the brain and skull meshes, and less than 0.008% in the vessel mesh, contain an edge that falls above the required sampling length. We note that while the particles reliably meet the sampling requirements of h_0 , the correctness of these requirements are ultimately related to the accuracy of the medial axis detection method, as discussed in Section 3.1. These results indicate that the proposed particle-based method is a practical scheme for generating an ε -sample of an isosurface that relates point density solely to the geometry of the surface, and not to the quality of the tessellation which is instead achieved implicitly by the low-energy configuration of particles.

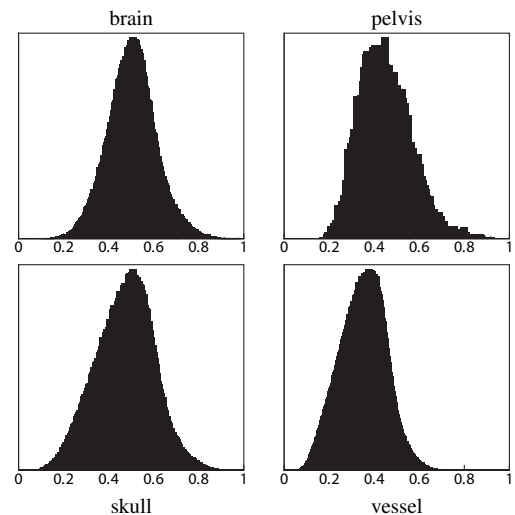


Fig. 4. The edge length versus h ratios for the four data sets. Values greater than 1.0 were encountered at a frequency of less than 0.01% in the brain, skull, and vessel meshes.

Table 2. The minimum and average radius ratios for each data set (min/avg) using the proposed particle-based method (ps), an advancing front scheme (af), and a modified marching cubes algorithm (mc).

	pelvis	brain	skull	vessel
ps	0.40/0.92	0.18/0.94	0.092/0.94	0.0195/0.94
af	0.23/0.94	0.02/0.93	0.006/0.93	0.0007/0.94
mc	0.00/0.66	0.00/0.67	0.000/0.66	0.0000/0.66

The second finite element requirement for generating high quality tessellations is the production of nearly regular triangles. We compute the radius ratios for each data set to measure the quality of the triangles in the resulting meshes — Figure 5 displays these histograms. We present the average radius ratios for each data set in Table 2, along with the minimum (worst) ratio which is important for determining the condition number in a finite element simulation. We also include the radius ratios for meshes generated using a marching cubes [25] algorithm that has been modified to use the same reconstruction kernels as those used in the particle system, and from an advancing front algorithm [33] that has been supplied to us by the authors. The data indicates that the proposed method generates average radius ratios that are nearly identical to the advancing front technique, but consistently produces much better minimum ratios than either alternative triangulation method. The proposed system is able to (globally) produce very regular triangulations due to the natural, low energy, hexagonal packing of particles, avoiding the problems associated with grid-based methods (*i.e.* restriction of vertices to grid edges) or advancing front techniques (*i.e.* detecting and stitching merging fronts) — poorly shaped triangles due to these problems are shown in Figure 8.

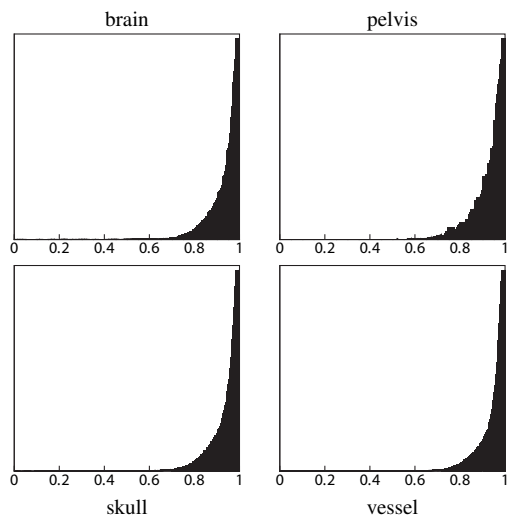


Fig. 5. The radius ratios for the four data sets, all with an average ratio of ~ 0.94 .

A third metric for measuring the quality of triangulations is a measure of the vertex valence of a mesh. For applications such as mesh compression [40] and subdivision surfaces [24] the regularity of the vertex valences across a mesh is important for efficient and accurate results. Regular triangulations tend toward a valence of six for most vertices, similar to the hexagonal properties of particle packing. Vertex valence also indirectly indicates the tendencies of a mesh to contain large and small angles. In Table 3 we summarize the valences of the vertices in our triangulations. The meshes indicate a good affinity for valence-six vertices (63.275%, compared with 44.15% for marching cubes and 71.75% for advancing front), with only a small fraction of vertices exhibiting valences greater than seven or less than five (0.7%, compared with 12.86% for marching cubes and 0.95% for advancing front). These numbers show that the particle-based method outperforms marching cubes while also containing a smaller percentage of extreme valences than the advancing front technique.

Table 3. Vertex valences for each data set, given as a percentage of the total number of vertices.

	< 5	5	6	7	> 7
brain	0.1	17.2	65.3	17.1	0.3
pelvis	0.3	22.3	56.2	20.8	0.4
skull	0.1	16.8	66.0	18.0	0.2
vessel	0.3	16.7	65.6	16.9	0.5

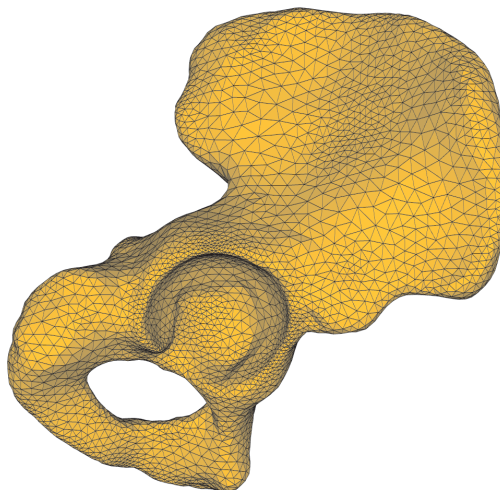


Fig. 6. A tessellation of a pelvis segmentation.

5 CONCLUSION

In this paper we propose a particle-based method for generating high quality tessellations of biological data sets. By creating a sizing field to dictate the density of the particle distributions, this method produces sets of points that can meet the sampling requirements of Delaunay surface reconstruction algorithms for generating topologically accurate tessellations, the accuracy of which depends on the correctness of the underlying medial axis detection algorithm. We present results from a variety of data sets that indicate the proposed method can reliably produce meshes that closely capture isosurface geometry, as well as generate very regular triangulations. We also compare the method against other tessellation techniques and show that the particle-based scheme generates consistently better minimum radius ratios, an important characteristic for reducing geometric error in finite element simulations.

Practically, the proposed method relies on a lower bound for the lfs to ensure that the number of particles does not blow up. In general, however, implicit functions can have arbitrary large curvature, which could result in very high densities of points around small features. The curvature of the implicit surface can be systematically controlled with level-set-based deformations [38, 35], and in practice we imagine this technique would be applied to binary masks which are then smoothed by the reconstruction filter. By controlling the local configurations of binary voxels, *e.g.* through mathematical morphology operations, the curvature of the reconstructed isosurface, and thus the maximum density of particles, can be controllably bounded — this is an area for further investigation.

The creation of high quality surface tessellations also is an interesting starting point for the generation of solid tetrahedralizations. Numerous volumetric mesh generators exist that can build a solid mesh from a surface triangulation, many of them based on Delaunay methods. We are interested in applying results from the proposed method to tetrahedralization schemes.

ACKNOWLEDGEMENTS

The authors thank John Schreiner and Carlos Scheidegger for help with using Afront. This research was supported by ARO grant W911NF-05-1-0395, the NIH Center for Integrative Biomedical Computing NCRN Project 2-P41-RR12553-07, and NSF grant CCR-

0310705.

REFERENCES

- [1] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun. Variational tetrahedral meshing. *ACM Trans. Graph.*, 24(3):617–625, 2005.
- [2] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. *Discrete and Computational Geometry*, 22:481–504, 1999.
- [3] N. Amenta, M. Bern, and D. Eppstein. The crust and the beta-skeleton: Combinatorial curve reconstruction. *Graphic Models and Image Processing*, 60(2 of 2):125–135, 1998.
- [4] N. Amenta, S. Choi, and R. K. Kamvyselis. A new voronoi-based surface reconstruction algorithm. *Siggraph*, pages 415–421, 1998.
- [5] N. Amenta, S. Choi, T. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *ACM Symposium on Computational Geometry*, pages 213–222, 2000.
- [6] N. Amenta, S. Choi, and R. K. Kolluri. The power crust. In *SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266, New York, NY, USA, 2001. ACM Press.
- [7] I. Babuska and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, 1976.
- [8] J.-D. Boissonnat and S. Oudot. Provably good sampling and meshing of surfaces. *Graph. Models*, 67(5):405–451, 2005.
- [9] S.-W. Cheng, T. K. Dey, E. A. Ramos, and T. Ray. Sampling and meshing a surface with guaranteed topology and geometry. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, pages 280–289, New York, NY, USA, 2004. ACM Press.
- [10] L. P. Chew. Guaranteed-quality delaunay meshing in 3d (short version). In *SCG '97: Proceedings of the thirteenth annual symposium on Computational geometry*, pages 391–393, New York, NY, USA, 1997. ACM Press.
- [11] S. Cotin, C. Duriez, J. Lenoir, P. Neumann, and S. Dawson. New approaches to catheter navigation for interventional radiology simulation. pages 534–542, Palm Springs, California, USA, 26-30 october 2005.
- [12] T. K. Dey and J. Giesen. Detecting undersampling in surface reconstruction. In *SCG '01: Proceedings of the seventeenth annual symposium on Computational geometry*, pages 257–263, New York, NY, USA, 2001. ACM Press.
- [13] T. K. Dey and S. Goswami. Tight cocone: a water-tight surface reconstructor. In *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 127–134, New York, NY, USA, 2003. ACM Press.
- [14] T. K. Dey and J. A. Levine. Delaunay meshing of isosurfaces. In *SMI '07: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*, pages 241–250, Washington, DC, USA, 2007. IEEE Computer Society.
- [15] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.
- [16] E. Hartmann. A marching method for the triangulation of surfaces. *The Visual Computer*, 14(3):95–108, 1998.
- [17] E. Hartmann. On the curvature of curves and surfaces defined by normal-forms. *Comput. Aided Geom. Des.*, 16(5):355–376, 1999.
- [18] T. J. R. Hughes. *The Finite Element Method*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.
- [19] M. W. Jones, J. A. Baerentzen, and M. Sramek. 3d distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):581–599, 2006.
- [20] T. Karkanis and A. J. Stewart. Curvature-dependent triangulation of implicit surfaces. *IEEE Computer Graphics and Applications*, 22(2):60–69, March 2001.
- [21] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization 2003*, pages 513–520, October 2003.
- [22] H. Krim and A. Yezzi, editors. *Statistics and Analysis of Shapes*. Birkhauser Boston, 2006.
- [23] T. S. Lau and S. H. Lo. Finite element mesh generation over analytic curved surfaces. *Computers and Structures*, 59(2):301–310, 1996.
- [24] A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 85–94, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [25] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press.
- [26] M. Meyer, P. Georgel, and R. Whitaker. Robust particle systems for curvature dependent sampling of implicit surfaces. In *Proceedings of the International Conference on Shape Modeling and Applications (SMI)*, pages 124–133, June 2005.
- [27] M. Meyer, B. Nelson, R. M. Kirby, and R. Whitaker. Particle systems for efficient and accurate high-order finite element visualization. *IEEE Transactions on Visualization and Computer Graphics*, In Press, 2007.
- [28] T. Möller, K. Mueller, Y. Kurzion, R. Machiraju, and R. Yagel. Design of accurate and smooth filters for function and derivative reconstruction. In *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization*, pages 143–151, New York, NY, USA, 1998. ACM Press.
- [29] O. Nemitz, M. Rumpf, T. Tasdizen, and R. Whitaker. Anisotropic curvature motion for structure enhancing smoothing of 3D MR angiography data. *Journal of Mathematical Imaging and Vision*, December 2006. published online.
- [30] G. M. Nielson, A. Huang, and S. Sylvester. Approximating normals for marching cubes applied to locally supported isosurfaces. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 459–466, Washington, DC, USA, 2002. IEEE Computer Society.
- [31] P.-O. Persson. Mesh size functions for implicit geometries and pde-based gradient limiting. *Eng. with Comput.*, 22(2):95–109, 2006.
- [32] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *SODA '93: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 83–92, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.
- [33] J. Schreiner, C. Scheidegger, and C. Silva. High-quality extraction of isosurfaces from regular and irregular grids. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1205–1212, 2006.
- [34] J. A. Sethian. Curvature flow and entropy conditions applied to grid generation. *J. Comput. Phys.*, 115(2):440–454, 1994.
- [35] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999.
- [36] J. R. Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures. Technical Report Manuscript, 2002.
- [37] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. The hamilton-jacobi skeleton. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 828, Washington, DC, USA, 1999. IEEE Computer Society.
- [38] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface smoothing via anisotropic diffusion of normals. In *VIS '02: Proceedings of the conference on Visualization '02*, Washington, DC, USA, 2002. IEEE Computer Society.
- [39] C. A. Taylor and M. T. Draney. Experimental and computational methods in cardiovascular fluid mechanics. *Annual Review of Fluid Mechanics*, 36:197–231, 2004.
- [40] C. Touma and C. Gotsman. Triangle mesh compression. In *Graphics Interface*, pages 26–34, 1998.
- [41] R. V. Uitert, C. Johnson, and L. Zhukov. Influence of head tissue conductivity in forward and inverse magnetoencephalographic simulations using realistic head models. *IEEE Transactions on Biomedical Engineering*, 51(12):2129–2137, 2004.
- [42] J. J. van Wijk. Image based flow visualization for curved surfaces. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 17, Washington, DC, USA, 2003. IEEE Computer Society.
- [43] L. Velho. Simple and efficient polygonization of implicit surfaces. *J. Graph. Tools*, 1(2):5–24, 1996.
- [44] A. P. Witkin and P. S. Heckbert. Using particles to sample and control implicit surfaces. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 269–277. ACM Press, 1994.
- [45] Z. J. Wood, P. Schroder, D. Breen, and M. Desbrun. Semi-regular mesh extraction from volumes. In *VIS '00: Proceedings of the conference on Visualization '00*, pages 275–282, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.
- [46] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for soft objects. *The Visual Computer*, (2):227–234, 1986.
- [47] Y. Zhang, G. Xu, and C. Bajaj. Quality meshing of implicit solvation models of biomolecular structures. *Comput. Aided Geom. Des.*, 23(6):510–530, 2006.

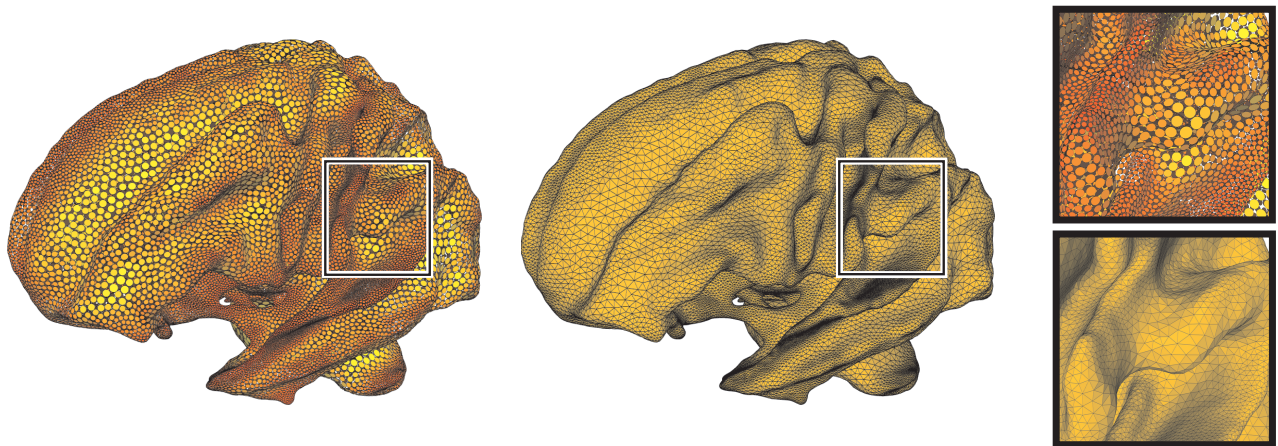


Fig. 7. Particles on the brain and the resulting tessellation. The surface is a reconstruction of a white-matter segmentation.

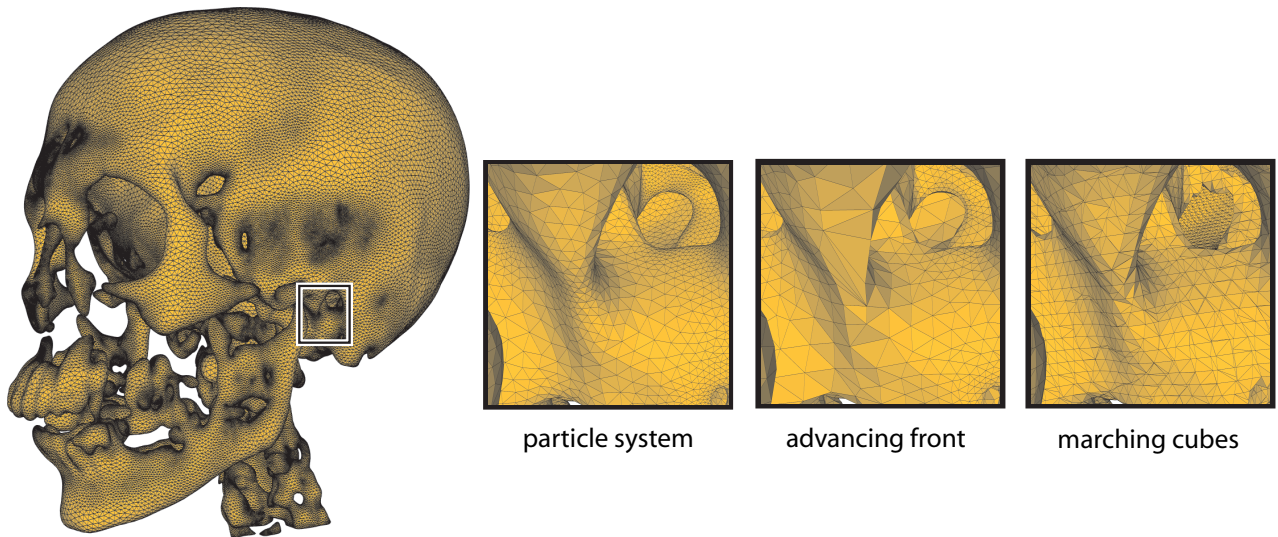


Fig. 8. The skull mesh is generated by reconstructing a level-set of a gray-scale CT image. Close-ups are from triangulations generated using the proposed particle system method, an advancing front technique [33], and a marching cubes algorithm.

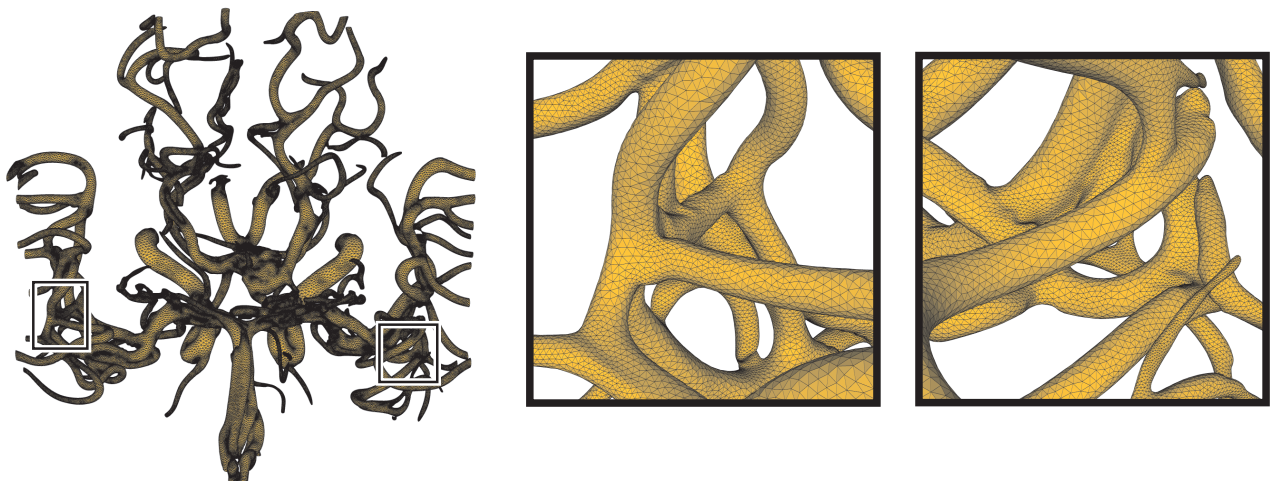


Fig. 9. The vessel mesh represents the zero-set of a distance transform generated using an anisotropic smoothing algorithm [29].