



## Multi-region Delaunay complex segmentation

S.J. Williams<sup>a,b,\*</sup>, M. Hlawitschka<sup>c,a</sup>, S.E. Dillard<sup>d</sup>, D. Thoma<sup>b</sup>, B. Hamann<sup>a</sup>

<sup>a</sup> Institute for Data Analysis and Visualization, Department of Computer Science, University of California, Davis, United States

<sup>b</sup> Los Alamos National Laboratory, United States

<sup>c</sup> Scientific Visualization Group, Universität Leipzig, Germany

<sup>d</sup> Pacific Northwest National Laboratory, United States

### ARTICLE INFO

#### Article history:

Available online 10 April 2012

#### Keywords:

Medial axis

Segmentation

Surface reconstruction

Feature extraction

### ABSTRACT

We focus on the problem of segmenting scattered point data into multiple regions in a single segmentation pass. To solve this problem, we begin with a set of potential boundary points and use a Delaunay triangulation to complete the boundaries. We then use information from the triangulation and its dual Voronoi complex to determine for each face whether it resembles a boundary or interior face, allowing a user to choose a specific segmentation by keeping only faces where our parameter is above a threshold. The resulting algorithm has time complexity in  $O(nd)$ , where  $n$  is the number of Delaunay simplices.

© 2012 Elsevier B.V. All rights reserved.

## 1. Introduction

Segmentation is crucially important in traditional image processing, computer vision, and scientific data processing and analysis. In the absence of application-specific constraints, segmentation is an ill-defined problem, so a generic segmentation method is not feasible. In medical image processing, segmentation is used, for example, to identify tumors, and in comparative diagnosis techniques. In computer vision, segmentation underlies many object detection and tracking methods. In the broader context of scientific data processing and visualization, our particular focus area, segmentation is the first step in data analysis and improves the quality of visualization results.

In many cases the definition of a “good” segmentation depends on the user’s data set and specific scientific question. We therefore present a semi-automated technique that provides the user with a set of possible segmentations, allowing the user to select the one they consider most acceptable. This interactive approach implies that all algorithms that are involved in the user feedback loop must be highly efficient.

Our research is motivated by the needs of materials scientists concerned with simulation data analysis and processing in nanoscience and nanotechnology applications. At a higher level, data and phenomena at the nanoscale can be studied directly through electron microscopy and indirectly through computer simulation. The molecular dynamics simulations typically used in materials science output scattered point data, i.e. with each point representing a single atom, while microscopy typically generates image data. The latter can be converted into the former (for example, by using edge detection), so we only consider the case of scattered point data. We consider segmentations consisting of an arbitrary number of regions without assuming or enforcing topological restrictions.

\* Corresponding author at: Institute for Data Analysis and Visualization, Department of Computer Science, University of California, Davis, United States.

E-mail addresses: [sjwill@ucdavis.edu](mailto:sjwill@ucdavis.edu) (S.J. Williams), [hlawitschka@ucdavis.edu](mailto:hlawitschka@ucdavis.edu) (M. Hlawitschka), [scott.dillard@pnl.gov](mailto:scott.dillard@pnl.gov) (S.E. Dillard), [thoma@lanl.gov](mailto:thoma@lanl.gov) (D. Thoma), [hamann@cs.ucdavis.edu](mailto:hamann@cs.ucdavis.edu) (B. Hamann).

## 2. Related work

In general, segmentation is an ill-posed problem—there is no way to define an optimal segmentation—so many different methods have been developed to address a variety of data types and applications. We briefly survey only a subset of these methods that are related to our own method. Segmentation methods can be broken into a few broad categories: those that operate on image data and those that operate on general point sets, also known as *scattered point data*. Further, the output of a segmentation method may be a *binary* segmentation, consisting of only two regions or material types, or a *multi-region* segmentation, consisting of many material types. Note that a segmentation with many connected components can still be handled as a series of binary segmentations, but this requires fully segmenting the image once per desired region.

The watershed transform (Vincent and Soille, 1991) is a well-known multi-region segmentation algorithm that operates on images. The watershed transform searches for low-valued basins separated by high-valued ridges. While the watershed transform can be computed efficiently, it is not innately robust to noise. Much research had as its focus techniques to mitigate the watershed's tendency to create too many segments in the presence of noise, by merging neighboring regions (Haris et al., 1998), using iterative blurring (Kim and Kim, 2003), or minimizing boundary energy (Park and Keller, 2001).

Another approach to multi-region image segmentation is the normalized cut algorithm (Shi and Malik, 2000), in which an image is treated as graph with neighboring pixels connected by similarity-weighted edges, and the graph is cut along low-valued edges. While the normalized cut algorithm provides flexibility in the choice of the weighting function, the algorithm is comparatively inefficient.

For scattered point data, many binary segmentation methods have been developed to solve the problem of reconstructing a surface from a set of sample points. A series of methods based on Delaunay triangulations were developed in the previous decades. Edelsbrunner and Mücke defined the  $\alpha$ -shape of a point set to be the subset of the Delaunay edges, faces and tetrahedra that were “smaller” than the parameter  $\alpha$  (Edelsbrunner and Mücke, 1992). The  $\alpha$ -shape construction has been applied to modeling molecular structures in chemistry and biology (Edelsbrunner et al., 1998). Amenta et al. developed the crust algorithm (Amenta et al., 1998; Amenta and Bern, 1999) and the subsequent co-cone algorithm (Amenta et al., 2000), which both identify a subset of faces of the Delaunay triangulation as the volume boundary, followed by the power crust algorithm which constructs from the Delaunay triangulation a power diagram, a generalization of the Voronoi diagram, which contains the output surface among its faces (Amenta et al., 2001). These methods can be shown to satisfy theoretical guarantees about the quality of the reconstructed surface based on the local feature size. Dey and Goswami also provide a provably good method for noisy point sets, which is based on many of the same concepts (Dey and Goswami, 2004).

These Delaunay-based surface reconstruction methods are often restricted to binary segmentations, wherein the boundary of a single object or material type is constructed. Our method builds upon these previous methods to construct boundary surfaces between multiple objects corresponding to distinct material types.

## 3. Background

Before discussing our approach, we provide a short review of Delaunay triangulations and Voronoi diagrams, and some of their useful properties; these are covered in more depth by, among others, de Berg et al. (2008).

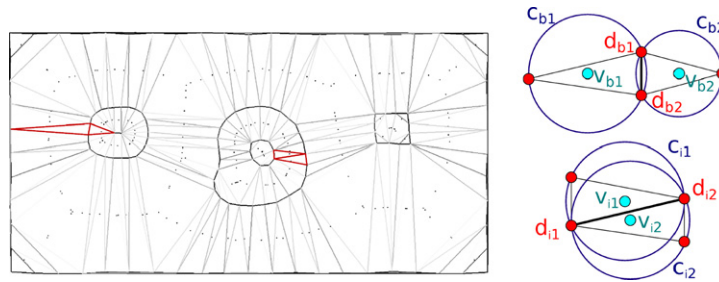
The Delaunay triangulation is a complete triangulation of a point set such that the circumscribing sphere, or *circumsphere*, of each simplex contains no points in its interior. Note that the term “triangulation” is independent of dimension, so in three dimensions a triangulation refers to a tetrahedral mesh. The dual of the Delaunay triangulation is the Voronoi diagram, a decomposition of space into cells such that each cell contains exactly one point from the input set, called a *site*, and all points in the strict interior of a cell are closer to the site in that cell than to any other site. A point where at least  $d + 1$  Voronoi cells meet is called a Voronoi vertex. Each Voronoi vertex is the center of the circumsphere of its dual Delaunay simplex, and the edge between neighboring Voronoi vertices is the dual of a shared face between two neighboring Delaunay simplices.

The *medial axis* of a point set  $S$  is the closure of the set of points that are equally close to multiple points in  $S$ . The medial axis is often used as a feature skeleton: the interior medial axis of a geometric object has exactly one connected component, which traces out its general shape. In the case of a point set, the medial axis is given by the edges of the Voronoi diagram: points inside a Voronoi cell all have exactly one closest point, while points on the boundary between Voronoi cells have multiple closest points. If the points are sampled from a continuous surface, the Voronoi vertices approximate the medial axis of the underlying surface (Amenta et al., 1998), though the quality of that approximation can get arbitrarily bad in three or more dimensions. See Fig. 1 for an example Delaunay triangulation along with its dual Voronoi vertices.

Computing a general Delaunay triangulation in three dimensions is known to be in  $O(n^2)$  with respect to the number of points being triangulated. However, the question of bounding the number of simplices produced by points sampled from a surface is an active research area, due primarily to the importance of the problem to surface reconstruction. Erickson (2005) argued that the Delaunay triangulation of points on a smooth surface has complexity  $O(n^{\frac{3}{2}})$ . With stricter sampling criteria, Attali et al. (2003) find that such a Delaunay triangulation can get as small as  $O(n \log n)$ . This is important to note, as the complexity of our algorithm depends on the size of such a triangulation.

## 4. Voronoi clustering

The objective of any segmentation approach is to partition an input image or space into two or more regions, where the space enclosed by one region belongs only to one object, and each object contains only one region. We use the term



**Fig. 1.** The overlap criterion applied to the edges of a Delaunay triangulation (top), with examples of two simplices sharing a boundary face and two simplices sharing an interior face, highlighted in red, respectively on the left and right sides of the triangulation. The highlighted simplex pairs are shown in a simplified schematic view below. In these diagrams, Delaunay vertices (input points) are colored red, while Voronoi vertices (circumcenters) are colored cyan. The pair of simplices sharing a boundary face are on top-right, while the pair of simplices sharing an interior face are on bottom-right. Input points on the boundary ( $d_{b1}$  and  $d_{b2}$ ) are close together, causing a short face. On the other hand, interior faces connect input points on opposite sides of the region ( $d_{i1}$  and  $d_{i2}$ ), causing interior faces to be long. The circumcircles of triangles sharing a boundary face therefore overlap very little ( $c_{b1}$  and  $c_{b2}$ ), while circumcircles of triangles sharing an interior face have large overlap ( $c_{i1}$  and  $c_{i2}$ ). A pair of triangles with well-separated circumcenters will also have well-separated Voronoi vertices ( $v_{b1}$  and  $v_{b2}$ ), while the Voronoi vertices of a triangle pair with overlapping circumcenters will be close together ( $v_{i1}$  and  $v_{i2}$ ). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

*segmentation* to refer to either a partitioning of space in this manner, or the process of generating such a partitioning, and the term *region* to refer to a single object identified by a segmentation.

To begin with, we require a sampling of points from the boundaries between objects. Locating boundary points in simulation data relies on information specific to the data set and its underlying scientific application. In the case of our example data set, shown in Fig. 7a, region boundaries occur when the local alignment of atoms becomes irregular, indicated by each atom having fewer neighbors due to less efficient packing. Atoms within such irregularities thus serve as the boundary points used as input to our algorithm.

In order to impose structure on these scattered boundary points, our next step is to compute their Delaunay triangulation. This will be a complete triangulation of the convex hull of the input points. A segmentation of such a triangulation can be viewed from two equivalent perspectives: first, by assigning each simplex to a region, such that each region is represented by a simplicial complex filling that region, or second, by classifying faces as interior or boundary, and defining each region as the space inside a closed envelope of boundary faces.

The chief observation driving our approach to classifying faces, as shown in Fig. 1, is that Voronoi vertices are well-separated across boundary faces and close together across interior faces. This led us to develop the *overlap criterion*: the more overlap between the circumcircles of neighboring simplices, the more likely their shared face lies on the interior of a region, also illustrated in Fig. 1. For two simplices sharing a face, this criterion is quantified as how much the bigger circumcircle of the pair would need to grow or shrink until it just contains the Voronoi vertex (circumcenter) of both simplices. A large overlap means that the circumcircles would have to shrink quite a bit to just contain both, while small overlap might require the circumcircles to grow to contain both. How much growth or shrinking is required depends on the ratio of the sampling density to the size of the regions.

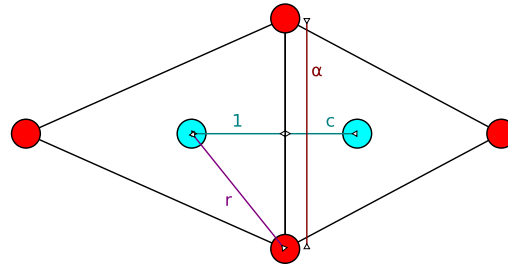
## 5. Overlap criterion

Consider two simplices that share a face, such as those in Fig. 2. First, we define the *local feature size* of each simplex as the signed distance from its Voronoi vertex to the shared face. This distance is positive if the circumcenter is on the same side of the shared face as the simplex's non-shared vertex. (In this case, both local feature sizes are positive.) Because each Voronoi edge is normal to its dual Delaunay face, the sum of the local feature sizes is equal to the length of the Voronoi edge.

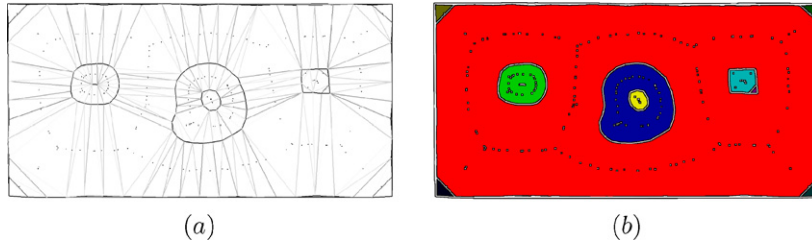
In order to make the overlap criterion independent of scale, we normalize the two-simplex system to the larger local feature size. Thus, the larger local feature size is 1, while the smaller local feature size is some value  $c \leq 1$ . For clarity,  $c$  represents the ratio between the local feature sizes; the larger  $c$ , the more similar the local feature sizes. Furthermore, as we will see, as  $c$  decreases, the likelihood that a face is interior increases, so the large faces of obtuse simplices (having negative values for  $c$ ) are highly likely to be interior.

The other major variable is  $\alpha$ , a measure of the sampling density. Because of the normalization,  $\alpha$  is a multiple of the larger local feature size, so an  $\alpha$  of 1 means that the sampling distance is equal to the larger local feature size, while an  $\alpha$  equal to  $c$  means the sampling distance is equal to the smaller local feature size. Formally,  $\alpha$  is the diameter of the circumsphere of the shared face. For two-dimensional triangles (having a one-dimensional face) this is simply the length of the shared face, while for three-dimensional tetrahedra (having a two-dimensional face) it is the diameter of the circumcircle of the triangle between the two tetrahedra. Finally,  $r$  is the normalized radius of the larger circumsphere.

The overlap criterion can now be stated mathematically. The overlap criterion is a specification of how much the larger circumcircle must grow or shrink to contain both Voronoi vertices, so we define this as a free variable  $\gamma$ . The larger circumsphere contains both Voronoi vertices if  $\gamma r = 1 + c$ . However, we would rather define the criterion in terms of



**Fig. 2.** Two neighboring triangles, with all the variables required to construct the overlap criterion. This figure is normalized to the larger local feature size, so that value is 1. The other variables are implicitly a ratio of the local feature size.  $\alpha$  is the sampling density with respect to the larger local feature size,  $r$  is the radius of the larger circumsphere (also with respect to local feature size), and  $c$  is the ratio between the larger and smaller local feature sizes.



**Fig. 3.** (a) The Delaunay triangulation of a point set, with faces darkened according to the likelihood they form a boundary between regions. Voronoi vertices are included as black points. Note that Voronoi vertices are well-separated across region boundaries. (b) The segmentation resulting from a specific overlap criterion threshold.

only local feature size and sampling density. This is simple, as by the Pythagorean Theorem,  $r = \sqrt{1^2 + (\frac{1}{2}\alpha)^2} = \sqrt{1 + \frac{1}{4}\alpha^2}$ , making the overlap criterion  $\gamma\sqrt{1 + \frac{1}{4}\alpha^2} = 1 + c$ . Solving for  $\gamma$  gives us the final form of the overlap criterion:

$$\gamma = \frac{1 + c}{\sqrt{1 + \frac{1}{4}\alpha^2}}$$

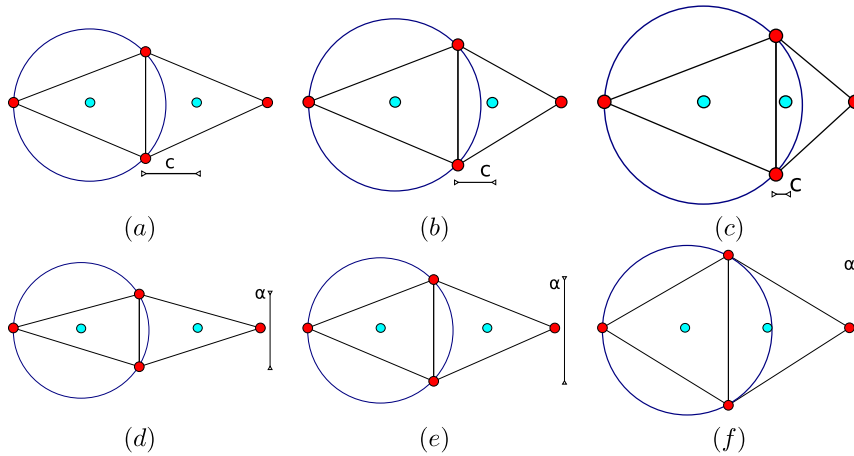
The higher the required  $\gamma$  to meet the overlap criterion, the further apart the Voronoi vertices of the simplices sharing that face, so the more likely it is to be a boundary face. Fig. 3a visualizes  $\gamma$ : darker faces have higher  $\gamma$  and are therefore more likely to be boundary faces. By specifying a  $\gamma$  threshold, above which faces are boundary and below which faces are interior, results in a specific segmentation, as in Fig. 3b.

The relationship between the two parameters ( $c$  and  $\alpha$ ) and whether a shared face is boundary or interior is visualized more fully in Fig. 4. As the math indicates,  $\gamma$  decreases (a face is more likely to be on a region interior) when either  $c$  decreases or  $\alpha$  increases. In the case of  $c$ , overlap is driven by the larger circumsphere, while  $c$  is the ratio of the smaller local feature size to the larger. Because of this definition, as  $c$  changes, the radius (and positioning with respect to the shared face) of the larger circumsphere remains the same. However, a decrease in  $c$  brings the smaller simplex’s Voronoi vertex closer to the larger circumsphere, reducing the amount the circumsphere must grow to contain both and making the shared face more interior. On the other hand,  $\alpha$  affects both sources of overlap: if the positions of the two points (one from each simplex) not involved in the shared face remain fixed, increasing  $\alpha$  necessarily increases the size of both circumspheres, and since those circumspheres “open out” more toward the shared face, their Voronoi vertices are both drawn to the shared face, bringing them closer together (and as a result, closer to the opposing circumsphere).

The criterion considers only information local to each face (and the two simplices sharing it). Each simplex consists of  $d + 1$  faces (where  $d$  is the dimension), so each simplex can only share  $d + 1$  faces. Computing the overlap criterion for a single face requires constant time. Thus, if there are  $n$  simplices in the data set, the time required to compute the overlap criterion for an entire data set is  $O(nd)$ . For the common case of three dimensions, then, the running time of our segmentation is linear on the number of Delaunay simplices.

## 6. Noise

We consider two kinds of noise, which we will call *additive*, when additional points are included that do not lie on an object boundary, and *subtractive*, when the boundary is undersampled. At a high level, the effect of additive noise is that it can potentially add new regions, while subtractive noise can potentially merge regions. Also, both forms of noise can impact the shape of the region boundaries. We assume for this discussion that we have a sampling of points from the boundaries of some objects; that a Delaunay triangulation of the points produces faces that follow all object boundaries (i.e., the sampling is sufficiently dense); that the diameters of all the circumcircles of all the faces following the object boundaries



**Fig. 4.** How changing  $c$  and  $\alpha$  affects whether a shared face is more like a boundary or interior face. (a)–(c): The more the simplices are mismatched in local feature size (if  $c$  is reduced), the closer their Voronoi vertices, reducing  $\gamma$  and making the shared face more interior. (d)–(f): Increasing the sampling distance (increasing  $\alpha$ ) makes the circumferences bigger while bringing the Voronoi vertices closer together, reducing  $\gamma$  and making the shared face more interior.

are not greater than  $\alpha_g$ ; that the ratio of local feature sizes on opposite sides of all boundary faces is not less than  $c_g$ ; and that there is a unique  $\gamma$  value that correctly distinguishes boundary from interior faces. That is, assume the ratio between boundary point sampling and local feature size is such that the lowest- $\gamma$  boundary face and highest- $\gamma$  interior face have very similar  $\gamma$  values. We call this point set the *ground set*.

We first address the case of additive noise. For this situation, we take the ground set and add one additional point, called the noise point. The more obvious effect of the noise point is that it can introduce a “dimple” on one of the regions. For the dimple case, assume the noise point is within a distance of  $\alpha_g$  from exactly  $d - 1$  boundary points. Also, assume that the local feature sizes of the objects separated by these boundary points are quite large in the vicinity of the boundary points. The triangulation will contain a simplex comprised of the  $d - 1$  boundary points and the noise point, due to the fact that by the way this point set is constructed, the circumsphere of such a simplex will contain no other points. Also, for all  $d + 1$  faces of this simplex,  $\alpha \leq \alpha_g$ , while  $c$  remains very large, so all faces of this simplex will have high  $\gamma$  values. Thus, this simplex will most likely be its own region. In comparison to a segmentation of the ground set, this simplex will appear “cut out” of whatever region contains the noise point.

Additive noise can also cause a region to be split apart, though how this problem manifests depends on the specifics of the data set, as well as its dimension. For a straightforward example, though, consider a two-dimensional data set containing an axis-aligned rectangle with rounded corners, and with much greater width than height. As above, the maximum distance between two points is  $\alpha_g$ . Also, assume the height of the rectangle is  $2 \cdot \alpha_g$ . Now, add a noise point at the center of the rectangle. The triangulation will now contain edges of length  $\alpha_g$  crossing from the top-center of the rectangle to the noise point, and from the noise point to the bottom-center of the rectangle. (This also assumes that there are boundary points directly above and below the noise point.) This will split the rectangle into at least two regions, and possibly more depending on the layout of the points.

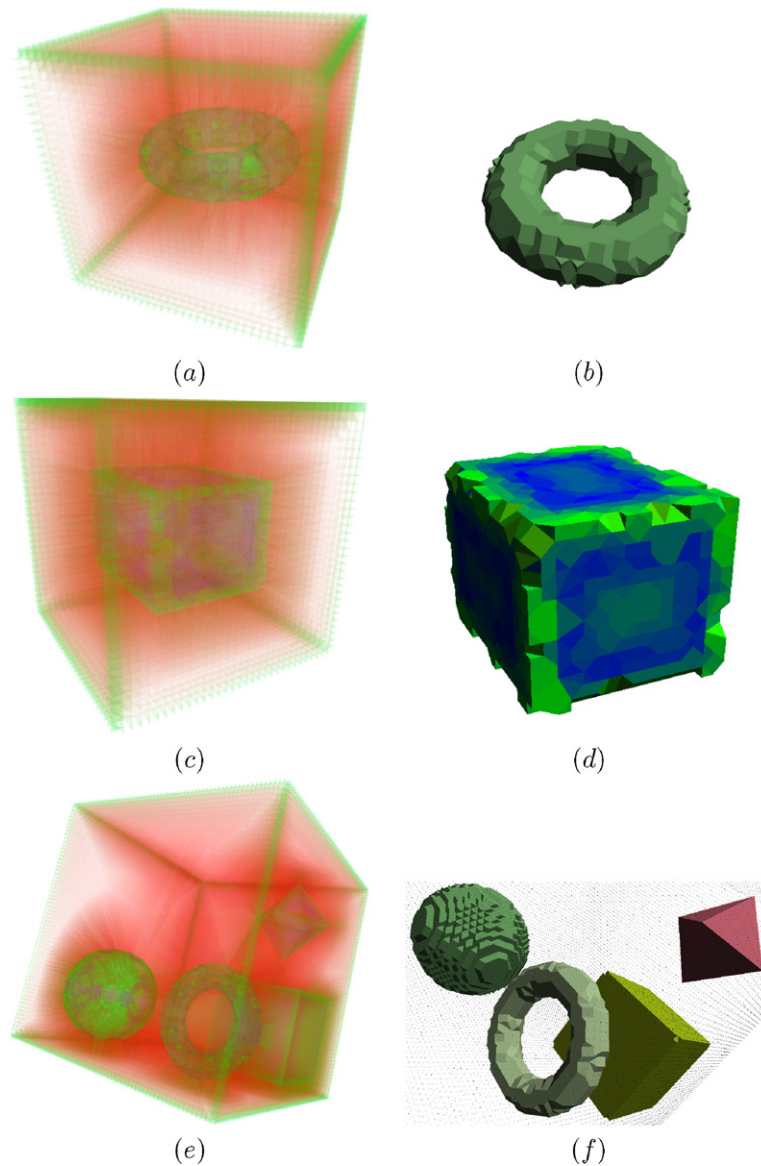
As for the case of subtractive noise, if the  $\gamma$  threshold used for a segmentation is left constant, it can cause regions to merge. If on the other hand the  $\gamma$  threshold is lowered to compensate, then more faces may be treated as lying on the boundary and regions can be split apart. For this discussion, we assume constant  $\gamma$  threshold; the effect of allowing the  $\gamma$  threshold to change is shown in the next section.

Consider the boundary face with the lowest  $\gamma$  value. We still operate under the assumption that the  $\gamma$  threshold required to correctly separate boundary from interior faces is very close to the lowest  $\gamma$  of a boundary face, i.e., the threshold is only slightly less than the  $\gamma$  of the face under consideration. Now assume that due to undersampling, one of the points of that face is thrown out and the triangulation recomputed. Any faces that will be generated using the remaining points (and covering the same section of the boundary as the removed face) must have a higher  $\alpha$  value, meaning lower  $\gamma$  values, and specifically, lowering them below the  $\gamma$  threshold. That will make that section of the boundary instead be treated as interior, causing the regions separated by the removed face to merge.

## 7. Results

We discuss the results of the overlap criterion applied to two types of data. The first data sets consist of simple geometric shapes, with points lying on an integer grid. Additionally, we show the effects of different amounts of sampling of a torus, from very dense to very sparse. The second data set is a molecular dynamics simulation at the atomic scale.



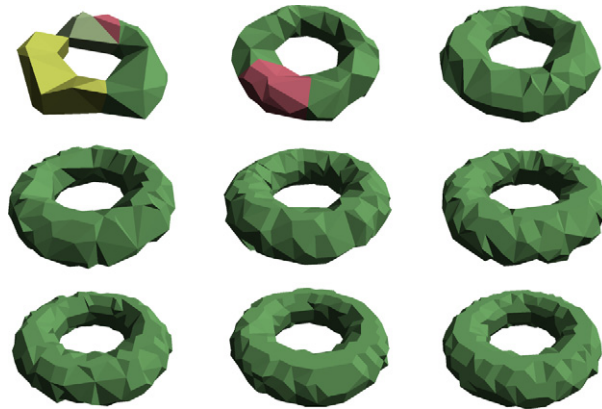


**Fig. 5.** This figure demonstrates the overlap criterion on simple geometric data sets in 3D. (a) A single torus, visualizing  $\gamma$  values for each face. Low values (interior faces) are red and translucent, while high values (boundary faces) are green or blue and opaque. This image is not a volume rendering, but rather a rendering of the entire tetrahedral complex using alpha blending. (b) A single segmentation is chosen by thresholding  $\gamma$ . (The segmentation fills the entire bounding box, but only the torus is rendered.) (c) and (d): Analogous to (a) and (b), respectively. Instead of a torus, we instead show a cube within another bounding box. Faces in the segmentation are colored based on  $\gamma$  value, on the same color scale. The inner cube is intentionally undersampled, to show the “notching” effects of undersampling or missing data points. (e) and (f): Analogous to (a) and (b), respectively, containing four geometric primitives (torus, sphere, cube, and octahedron). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

### 7.1. Geometric shapes

The geometric shapes were defined by directly sampling points on an integer grid and marking points on their surfaces. Additionally, the shapes are surrounded by points sampled from a box representing the boundary of the volume. This is done to ensure that there are at least two regions in each data set. Due to the dense and regular point sampling, as these data sets were artificially created based on mathematical specifications of the shapes being represented, all segmentations were computed using a  $\gamma$  value of 1.

Fig. 5a shows the  $\gamma$  value for each face of a volume containing a single torus. In all renderings of this form,  $\gamma$  is mapped to both opacity and color: for values between 0 and 1,  $\gamma$  is mapped directly to opacity (using standard OpenGL transparency) times a constant scaling factor, and its color is linearly interpolated between red at  $\gamma = 0$  and green at  $\gamma = 1$ . For values above 1, the faces have an opacity of the same constant scaling factor, and color is interpolated between green at



**Fig. 6.** Starting from the torus in Fig. 5b, we demonstrate the effects of undersampling by throwing out points at random. For a worst case, in the top-left, we throw out 90% of the points. This requires us to lower  $\gamma$  until the torus breaks apart into four different regions. In the next example, we throw out 80% of the points, causing the torus to break apart into two regions. The rest of the tori, each successively throwing out 10% fewer points than the last, can be segmented into only one region. While the general shape of the torus is visible in all such images, the shape of the torus improves as more points are kept: lower sampling densities results in pronounced dimples on the torus, while the torus in the bottom-right, throwing out only 10% of the points, looks nearly identical to the original.

$\gamma = 1$  and blue at  $\gamma = 2$ , with all faces with  $\gamma > 2$  also blue and opaque. As expected, the boundary of the torus is green and opaque (indicating high  $\gamma$ ), while the intervening volume is red and translucent (indicating low  $\gamma$ ).

The volume of the segmented torus is 1396 cubic units and its surface area is 985 square units. An actual torus with hole radius  $R = 8$  and tube radius  $r = 3$  has a volume of 1421 cubic units and surface area of 947 square units: the segmented torus has about 1.8% less volume, and about 4% more surface area. The difference—especially the increase in surface area—is due in large part to the integer grid.

Fig. 5c contains a cube within a bounding box. However, the cube is intentionally undersampled to show the effects of missing or incomplete data. Undersampling reduces the  $\gamma$  value of boundary faces, but correctly segmenting an object requires choosing a  $\gamma$  threshold at or below that of all faces on the object boundary. Hence, having undersampled data means the user must choose a lower  $\gamma$  value to differentiate the desired object from neighboring objects. However, reducing the  $\gamma$  threshold also generates more objects (since more faces are considered on a boundary), particularly near corners and edges. The result (which is quite obvious in Fig. 5d) is that the object gets “notched” along edges and at corners.

Fig. 5e shows a similar visualization, this time with three additional geometric primitives in the same volume: a cube, an octahedron, and a sphere. The cube and octahedron demonstrate that our metric can resolve some sharp edges, but has difficulty with corners. The sphere experienced more pronounced aliasing than the torus, but that aliasing is present in the original data.

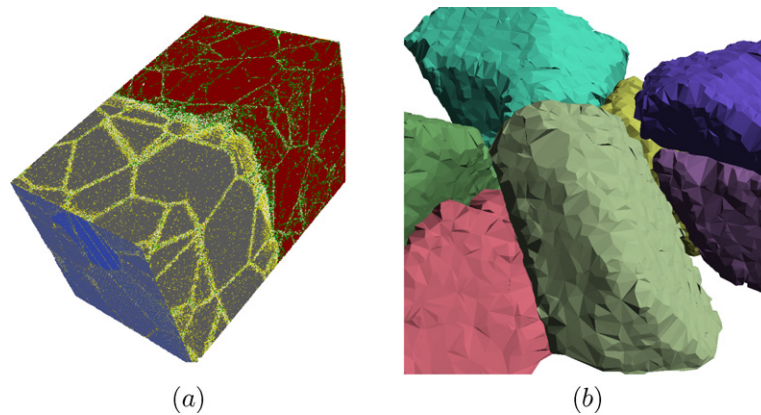
## 7.2. Undersampling

The previous discussion of subtractive noise assumed that the  $\gamma$  threshold remained constant. This was enforced by assuming that the  $\gamma$  threshold used to select the desired segmentation was necessarily very near the lowest  $\gamma$  value of any boundary face. For this section, we instead allow the  $\gamma$  threshold to change in order to get the desired segmentation. In order to make it obvious what constitutes the desired segmentation, we use a data set containing only a torus inside a bounding cube. Beginning with the torus data set used in Fig. 5b, we remove points at random to emulate undersampling. Nine resulting tori are shown in Fig. 6. For the top-left torus, we remove 90% of the points. For the next torus, we removed 80% of the points, and we removed 70% of the points from the next, and so on. For the last torus, then, we only removed 10% of the points.

While the original torus was segmented with a  $\gamma$  threshold of 1.0, these tori were sampled with  $\gamma$  thresholds ranging from 0.8 at high sampling rates, down to 0.5 at low sampling rates. The lowered  $\gamma$  thresholds cause tetrahedra to form their own regions, leaving tetrahedral cuts on the surface of the torus. At very low sampling rates, the  $\gamma$  thresholds are reduced so low that the tori are broken apart into more than one region, indicated by different colors. Also, when 90% of the points are removed, the resulting figure is barely recognizable as a torus.

## 7.3. Molecular dynamics data

A grain in materials science is an area where the arrangement of atoms is locally consistent; the interface between adjacent grains is highly irregular and, therefore, of interest to materials scientists, since grain boundaries are believed to be the weakest locations in a material. In this data set (Kadau et al., 2007), atoms are points in general position and without neighborhood information. Whether an atom lies inside a grain or on a grain boundary was determined by domain scientists based on how many atoms are within a specific (physically-based) distance of the atom in question. In Fig. 7a,



**Fig. 7.** (a) The molecular dynamics data set. The domain is filled with particles representing atoms. At small scales, atoms will tend to align themselves into tightly-packed “grains.” Where two grains meet, the organization breaks down, and these grain boundaries are believed to where cracking and deformation in materials originate. In this image, atoms are colored based on the number of nearby atoms. The number of nearby atoms indicates the packing density: atoms within a grain (colored gray and blue) are packed more efficiently, so they have more neighbors, while atoms on grain boundaries (colored yellow) have fewer neighbors. In the simulation, the material is undergoing a shock that perturbs the atoms; red and green atoms are those that have already undergone the shock, with the shockwave appearing as a yellow front between them. For the purposes of this paper, we restrict ourselves to the non-shocked (gray) side of the data. (b) Seven grains extracted from the molecular dynamics data set using our algorithm. The grain boundaries are several atoms thick, and atoms within the boundary can lie at arbitrary positions within the boundary region. This causes the surfaces of the grains in this segmentation to appear crumpled. These perturbations are an artifact of the inherent uncertainty in attempting to precisely specify a grain boundary. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

atoms believed to be on the interior of a grain are colored gray and blue, while atoms believed to be on a grain boundary are colored yellow. Also, in the simulation, the material is undergoing a shock from one end of the volume; this produces a yellow shockwave, and the even more disorganized atoms in its wake are colored red and green. For this work, we are only concerned with grain identification, so we restrict ourselves to the gray half of the data.

We begin by removing all atoms except those tagged by the simulation as lying on a grain boundary, so that we have a sampling of boundary points as in the prior examples. Our approach reconstructs the grain volumes, as illustrated in Fig. 7b. Grain boundaries are a region rather than a surface, and the boundary atoms are randomly positioned inside the interface region. This effect causes the grain boundaries to also be jagged. Since the segmentation is opaque, due to being a complete triangulation of a three-dimensional volume, only seven grains are shown. For this segmentation, we used a  $\gamma$  threshold of 0.31.

This work was done on a laptop with an Intel Core Duo at 1.83 GHz and 2 GB of main memory. The data contain 30,137,647 atoms, of which 151,262 were identified by the simulation as lying on a grain boundary. This resulted in 980,289 tetrahedra in the Delaunay triangulation. The triangulation was computed using the Computational Geometry Algorithms Library (CGAL) in 4.7 seconds. Computing the  $\gamma$  of each face took 1.7 seconds. Finally, selecting a different  $\gamma$  threshold to select a different segmentation (and updating internal data structures accordingly) took 0.31 seconds.

## 8. Conclusions

As one of the fundamental problems of image processing, we have been studying data segmentation in order to simplify analysis for our colleagues in materials science. Their data come principally from either nanoscale imaging devices such as electron microscopes or through molecular dynamics simulations. In segmenting the latter, we begin by constructing a Delaunay triangulation of points on the boundaries between objects, and use a new criterion to determine the likelihood that faces lie on object boundaries and either visualize this likelihood measure directly, or threshold it to select a specific segmentation. We have evaluated our implementation of the algorithm for both analytic and real-world data sets; it has shown to work well in practice and segments multiple regions in a computationally efficient manner.

## Acknowledgements

We acknowledge the support of the LANL-UC Davis Materials Design Institute, and especially the support and comments made by Sriram Swaminarayan, Billy Sanders, and Dan Thoma. We thank the National Science Foundation (CCF-0702817) and the Los Alamos National Laboratory, Materials Design Institute for supporting this research.

## References

- Amenta, N., Bern, M., 1999. Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry* 22 (4), 481–504.
- Amenta, N., Bern, M., Kamvyselis, M., 1998. A new Voronoi-based surface reconstruction algorithm. In: *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, New York, NY, USA, pp. 415–421. <http://doi.acm.org/10.1145/280814.280947>.



- Amenta, N., Choi, S., Dey, T., Leekha, N., 2000. A simple algorithm for homeomorphic surface reconstruction. In: Proceedings of the Sixteenth Annual Symposium on Computational Geometry. ACM, New York, NY, USA, pp. 213–222.
- Amenta, N., Choi, S., Kolluri, R.K., 2001. The power crust. In: SMA '01: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications. ACM, New York, NY, USA, pp. 249–266. <http://doi.acm.org/10.1145/376957.376986>.
- Attali, D., Boissonnat, J.-D., Lieutier, A., 2003. Complexity of the Delaunay triangulation of points on surfaces: The smooth case. In: SCG '03: Proceedings of the Nineteenth Annual Symposium on Computational Geometry. ACM, New York, NY, USA, pp. 201–210. <http://doi.acm.org/10.1145/777792.777823>.
- de Berg, M., Cheong, O., van Kreveld, M., Overmars, M., 2008. Computational Geometry: Algorithms and Applications, 3rd edition. Springer. <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/3540779736>.
- Dey, T., Goswami, S., 2004. Provable surface reconstruction from noisy samples. In: Proceedings of the Twentieth Annual Symposium on Computational Geometry. ACM, New York, NY, USA, pp. 330–339.
- Edelsbrunner, H., Mücke, E., 1992. Three-dimensional alpha shapes. In: Proceedings of the 1992 Workshop on Volume Visualization. ACM, New York, NY, USA, pp. 75–82.
- Edelsbrunner, H., Facello, M., Liang, J., 1998. On the definition and the construction of pockets in macromolecules. *Discrete Applied Mathematics* 88 (1), 83–102.
- Erickson, J., 2005. Dense point sets have sparse Delaunay triangulations or ... but not too nasty. *Discrete and Computational Geometry* 33 (1), 83–115. <http://dx.doi.org/10.1007/s00454-004-1089-3>.
- Haris, K., Estradiadis, S.N., Maglaveras, N., Katsaggelos, A.K., 1998. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing* 7 (12), 1684–1699. [citeseer.ist.psu.edu/haris98hybrid.html](http://citeseer.ist.psu.edu/haris98hybrid.html).
- Kadau, K., Germann, T.C., Lomdahl, P.S., Albers, R.C., Wark, J.S., Higginbotham, A., Holian, B.L., 2007. Shock waves in polycrystalline iron. *Physical Review Letters* 98 (13), 135701, 4 pp.
- Kim, J.-B., Kim, H.-J., 2003. Multiresolution-based watersheds for efficient image segmentation. *Pattern Recognition Letters* 24 (1–3), 473–488. [http://dx.doi.org/10.1016/S0167-8655\(02\)00270-2](http://dx.doi.org/10.1016/S0167-8655(02)00270-2).
- Park, J., Keller, J.M., 2001. Snakes on the watershed. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (10), 1201–1205. <http://dx.doi.org/10.1109/34.954609>.
- Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8), 888–905. [citeseer.ist.psu.edu/shi97normalized.html](http://citeseer.ist.psu.edu/shi97normalized.html).
- Vincent, L., Soille, P., 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (6), 583–598. <http://dx.doi.org/10.1109/34.87344>.