Mason: Morphological Simplification

Jason Williams and Jarek Rossignac
Georgia Institute of Technology

Abstract

The traditional rounding and filleting morphological filters are biased. Hence, as r grows, the rounding $R_r(S)$ of S shrinks and the filleting $F_r(S)$ grows. A shape S is r-regular when $R_r(S) = F_r(S) = S$. The combinations $F_r(R_r(S))$ and $R_r(F_r(S))$ produce nearly r-regular shapes, but retain a bias: $F_r(R_r(S))$ is usually smaller than S and $R_r(F_r(S))$ is larger. To overcome this bias, we propose a new filter, called Mason. The r-mortar $M_r(S)$ of S is $F_r(S) - R_r(S)$, and the stability of a point P with respect to S is the smallest value of r for which P belongs to $M_r(S)$. Stability provides important information about the shape's imbedding that cannot be obtained through traditional topological or differential analysis tools. $F_r(R_r(S))$ and $R_r(F_r(S))$ only affect space in $M_r(S)$. For each maximally connected component of $M_r(S)$, Mason performs either $F_r(R_r(S))$ or $R_r(F_r(S))$, choosing the combination that alters the smallest portion of that component. Hence, Mason acts symmetrically on the shape and on its complement. Its output is guaranteed to have a smaller symmetric difference with the original shape than that of either combination $F_r(R_r(S))$ or $R_r(F_r(S))$. Many previously proposed shape simplifications algorithms were focused on reducing the combinatorial storage or processing costs of a shape at the expense of the smoothness and regularity or altered the shape in regular portions that did not exhibit any high frequency complexity. Mason is the first shape simplification operator that is independent of the particular representation and offers the advantage of preserving portions of the boundary of S that are regular at the desired scale.

1 Introduction

Our goal is to analyze complex shapes that may contain handles, through-holes, bridges, constrictions, cavities, and connected components of varying sizes, as well as a combination of sparsely and densely filled regions of space. We propose to do this by evaluating how shapes change as we apply different degrees of *regularization.* For a given scale r, we would like to simplify a given shape S to one that is r-regular, i.e., both the shape and its complement are infinite unions of balls of radius r. An r-regular shape has no thin or highly curved features, such as cracks, gaps, branches, or corners that a ball of radius r rolled along the boundary of S could not reach. Regularizing a shape may change its topology, for instance by merging nearby connected components or fusing a handle into a larger shape. By varying r, we can determine the stability of geometric and topological properties under regularization.

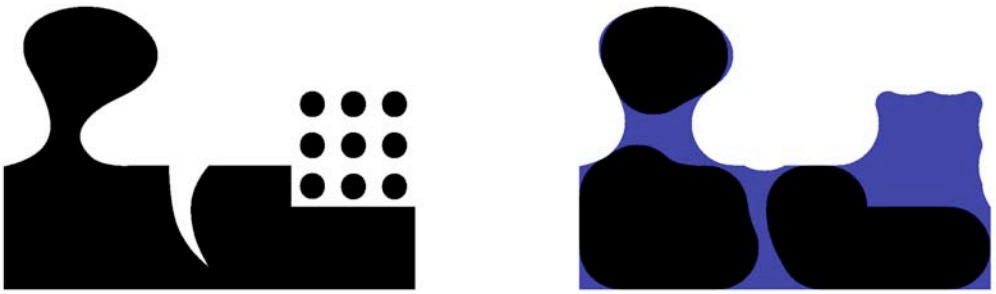

Figure 1: A shape (black) on the left, and its core (black), mortar (gray), and anticore (white) on the right.

One contribution of this work is our introduction of the r-mortar as a tolerance zone of controlled scale for simplification or regularization. If we define the *stability* of a point as the radius of the largest ball that contains it whose interior lies either completely within the shape or its complement, we can define the *r-mortar* of a point set S as the set of points whose stability with respect to S is less than r. Unlike other tolerance zones, the mortar contains only the details of a shape at the desired scale; away from those details, in the r-regular portions of space, it degenerates to the boundary of the shape. Hence, restricting simplification to a given r-mortar ensures that it does not affect the already-regular portions of the shape's boundary. For a given value of r, we define the points outside the mortar, which are not affected by simplification, as part of the shape's *core* if they lie within the shape, and as part of its *anticore* if they lie in its complement. Figure 1 shows an original shape in black on the left, and its mortar (blue), core (black), and anticore (white) on the right.

Several morphological simplification operators leave the r-regular portions of the boundary of a shape unchanged. In particular, the rounding and filleting operations [1, 2, 3] can be combined to produce shapes that are nearly regular and only differ from the input shape in the mortar. However, the results are biased to change the shape's volume depending on which operation is applied first. Filleting first and then rounding tends to add volume, while rounding and then filleting removes it. The complementary character of these biases can be expressed by describing the filters as dual of one another: the complement of applying one filter to a shape is equal to applying the other filter to the shape's complement. Our second contribution is a new filter we call *Mason* that selects which of these two filters should be applied locally. Mason retains the desirable properties of the original combinations of rounding and filleting, but is self-dual, so that it acts symmetrically on the shape and its complement. The volume of the symmetric difference between the input and output of Mason is always less than or equal to those of its components.

2 Prior Art

We have referred to the morphological regularization that Mason and combinations of rounding and filleting perform as a form of shape simplification. There are many other shape processing techniques that can also be classified as simplifications. Some of these

techniques reduce the size of the input's representation, some reduce the topological complexity of the shape, and some smooth or fair the shape.

Most representations of 3D shapes combine shape primitives, such as vertices or polygons, into a graph structure. Storage requirements, transmission delays, and processing and rendering costs grow with the number of the primitives needed to represent the desired shape with sufficient accuracy. Because the complexity of the digital representations of 3D models that are manipulated in many engineering, scientific, or architectural applications has grown much faster than transmission and processing speed, a significant amount of research has been devoted to the development of effective algorithms that produce lower-complexity approximations (LODs) of an original shape. Most of these techniques focus on polygonal, and often triangular, shapes. They reduce the triangle count by clustering vertices [4] or by collapsing edges [5, 6, 7] or both [8, 9]. Typically, these methods seek to minimize the discrepancy between the original shape and its approximation for a desired triangle count reduction. The discrepancy is most often measured by estimating a bound on the Hausdorff distance between the two shapes [4, 10], the maximum of the normal displacement [6], or mean square normal displacement from the original shape [7] or from the result of the previous simplification step [11]. These metrics tend to distribute the discrepancy (error) uniformly over the surface shape. Other metrics based on the normal field, such as [12], allow less error near details than in other regions. Both types of metrics allow modifications to the original shape in areas that were smooth and simple. Although some of the effects on geometry that can be obtained by varying the output size are analogous to those that can be obtained by varying the radius used in morphological simplification, the analogy is limited. Highly simplified triangle meshes, do not resemble a union of large spheres.

In contrast to these simplification methods, our objective is not to reduce storage or triangle count of a representation. In fact, the formulation of Mason, and hence of its effect, is independent of any particular representation used. Its objective is to "simplify" the shape itself, not digital representation of the shape.

Another notable difference is that at a given simplification scale, Mason will leave unchanged the regular, smooth areas of the boundary of the shape, and will only affect the portions of space where the boundary is close to its medial axis. More precisely, consider an original shape S, its approximation A produced by a triangle reduction simplification technique, and its simplification M produced using Mason. Assume that the tolerance imposed on the discrepancy between A and S ensures that no point of A is further from the boundary of S than a distance r. (Such a one-sided Hausdorff distance is often used to define the error of a simplification.) Now, assume that M is produced by a Mason simplification with the same scale r. We have shown that no point of the boundary of M lies further than r from the boundary of S. Furthermore, Mason does not change the shape outside of a tolerance zone or envelope [13] defined as the Minkowski sum of the boundary of S with a ball of radius r, centered at the origin. Hence, M is guaranteed to be as close to S as A is, if such an asymmetric Hausdorff distance is used to measure the discrepancy. But Mason further restricts this tolerance zone to the mortar by reducing its thickness to zero around the regular portions of the boundary of S. By doing so, we guarantee that the portions of S that are already simple (at the desired scale) are not affected by Mason simplification. In addition to enforcing this mortar constraint,

we seek to minimize the volume of the symmetric difference between M and S, as Alliez et. al. did in [14].

Several mesh decimation techniques [4, 15, 16, 17] are capable of reducing the topological complexity of a shape (i.e. its genus and the number of connected components of its boundary). More recently, topology-simplifying techniques [18, 19] have been developed to cope with the topological complexity in 3D models that results from scanning errors or isosurface extraction from noisy volumetric data. These recent techniques are not guided by the objective of reducing the triangle count, and unlike regularization, they preserve geometric detail. Other techniques use separate mesh decimation and topology simplification stages. The mechanisms used for topology simplification in several of these techniques are closely related to regularization. [20] employs an operation very similar to morphological closing with a cube, while [21] utlilizes rounding and filleting. Unlike Mason, however, these approaches are not self-dual. [22] applies a lowpass filter to binary (voxelized) volume data and then thresholds it. This has qualitative properties similar to regularization: it can fill in concavities, round off convexities, and merge or eliminate small connected components. However, lowpass filtering erodes convex portions of the shapes and hence affects the shape in already regular portions. Attempts to compensate for such erosion effects through scaling or by using a variable threshold can displace regular contours, which could affect the shape outside of the mortar.

Another body of work concerns fairing or smoothing shapes. Aesthetic or functional constraints have motivated car body or ship hull designers to deform (fair) curved surfaces in order to minimize an energy function [23, 24, 25, 26]. These deformations attempt to minimize curvature variations or integral square curvature. Like regularization, minimizing integral square curvature tends to decrease maximum curvature. The computer graphics literature contains several methods for smoothing meshes, typically to remove acquisition artifacts from laser range scanning or isosurface extraction. Some techniques seek to preserve the sharp features that regularization would remove [27, 28, 29]. Others [30, 31, 32] eliminate sharp features, but like the energy-minimizing techniques they differ from regularization in that they do not simplify topology and they typically preserve thin structures.

[33] describes an approach to shape simplification based on computing a surface's medial axis, pruning it, and then reconstructing the surface from the simplified axis. If we associate each point on the medial axis with the radius of the ball defining it, then reconstructing a shape from the portion of the axis associated with a radius greater than or equal to r is the same as rounding it with radius r. The technique in [33], however, does not prune based on radius, so the reconstructed shapes can retain edges and thin parts that rounding would remove.

3 Definitions and Properties

3.1 Minkowski operators

Our method expands on basic mathematical morphology operators. Classic texts on the subject include [34] and [35]. Although mathematical morphology generalizes to lattices (partially ordered sets whose subsets each have a greatest lower bound and a least upper bound,) we restrict ourselves to regularly sampled sets and discuss implementation results for binary image and volume data.

Many operations in mathematical morphology can be defined in terms of the Minkowski sum and difference. The *Minkowski sum* of two sets A and B, denoted $A \oplus B$, is $\bigcup_{b \in B} A + b$; it is the result of taking the union of copies of B (which is called the structuring element) translated to every point in A. The *Minkowski difference*, denoted $A \ominus B$ is $\bigcap_{b \in B} A - b$; it is the set of points p such that B translated to p lies completely within A. An alternative definition is $A \ominus B = \overline{\overline{A} \oplus \hat{B}}$, where $\hat{B}$ is B reflected about the origin. Contemporary works on Minkowski algebra include [36] and [37].

The Minkowski sum and difference are not inverses. $(A \ominus B) \oplus B$, for instance, is the union of all translated copies of B that fit in A. For fixed B, we can write $(A \ominus B) \oplus B$ as an operator $\Phi(A)$. $\Phi(A)$ is an *opening*, which is defined as an operation that is translation-invariant $(\Phi(A + b) = \Phi(A) + b)$, idempotent $\Phi(\Phi(A)) = \Phi(A)$, increasing $(\Phi(A) \subseteq \Phi(C) \Leftrightarrow A \subseteq C)$, and antiextensive $(\Phi(A) \subseteq A)$. The dual operator $\Psi(A) = \left(A \oplus \hat{B}\right) \ominus \hat{B}$ is the complement of all the copies of B that fit in $\overline{A}$. It is a *closing*, which is defined as an operation that is translation-invariant, idempotent, increasing, and extensive $(\Psi(A) \supseteq A)$.

3.2 Growing, shrinking, rounding, and filleting operators

We are particularly interested in the case where the structuring element B is a ball b_r of radius r centered at the origin . Thus, we define S *grown* by r to be $S\uparrow_r = S \oplus b_r$, S *shrunk* by r to be $S\downarrow_r = S \ominus b_r$, S *filleted* by r to be $F_r(S) = S\uparrow_r\downarrow_r$ and S *rounded* by r to be $R_r(S) = S\downarrow_r\uparrow_r$ [1, 2, 3] . (For simplicity, we will omit the subscript r when dealing with only a fixed radius for all operations.) $S\uparrow$ is the union of all balls whose centers lie in S, $S\downarrow$ is the centers of all balls that lie in S, $R(S)$ is the union of all balls that lie in S, and $F(S)$ is the complement of the union of all balls that lie in $\overline{S}$ (Figure 2). $R(S)$ is an opening, and so it is antiextensive, while $F(S)$ is a closing, so it is extensive. Hence $R(S) \subseteq S \subseteq F(S)$.

3.3 Regularity, Core, Anticore, and Mortar

We define the *stability* of a point with respect to a shape S as the radius of the largest ball containing the point that lies completely within S or its complement. Note that our definition associates the stability measure with all points of space.

Consider a given value of r. If a point in S has stability greater than or equal to r, we say that it is *inner-regular*; otherwise, it is *inner-irregular*. Similarly, a point in the complement of S is *outer-regular* if its regularity is greater than or equal to r and *outer-irregular* otherwise. Note that the set of all inner-regular points is $R(S)$, while the set of outer-regular points is $\overline{F(S)}$.

We refer to the inner-regular points as the *core* of S and the outer-regular points as the *anticore*. Points that are neither in the core nor the anticore lie in the *mortar* $M(S)$. The mortar is the union of the inner-irregular set $S - R(S)$ (which includes the branches,

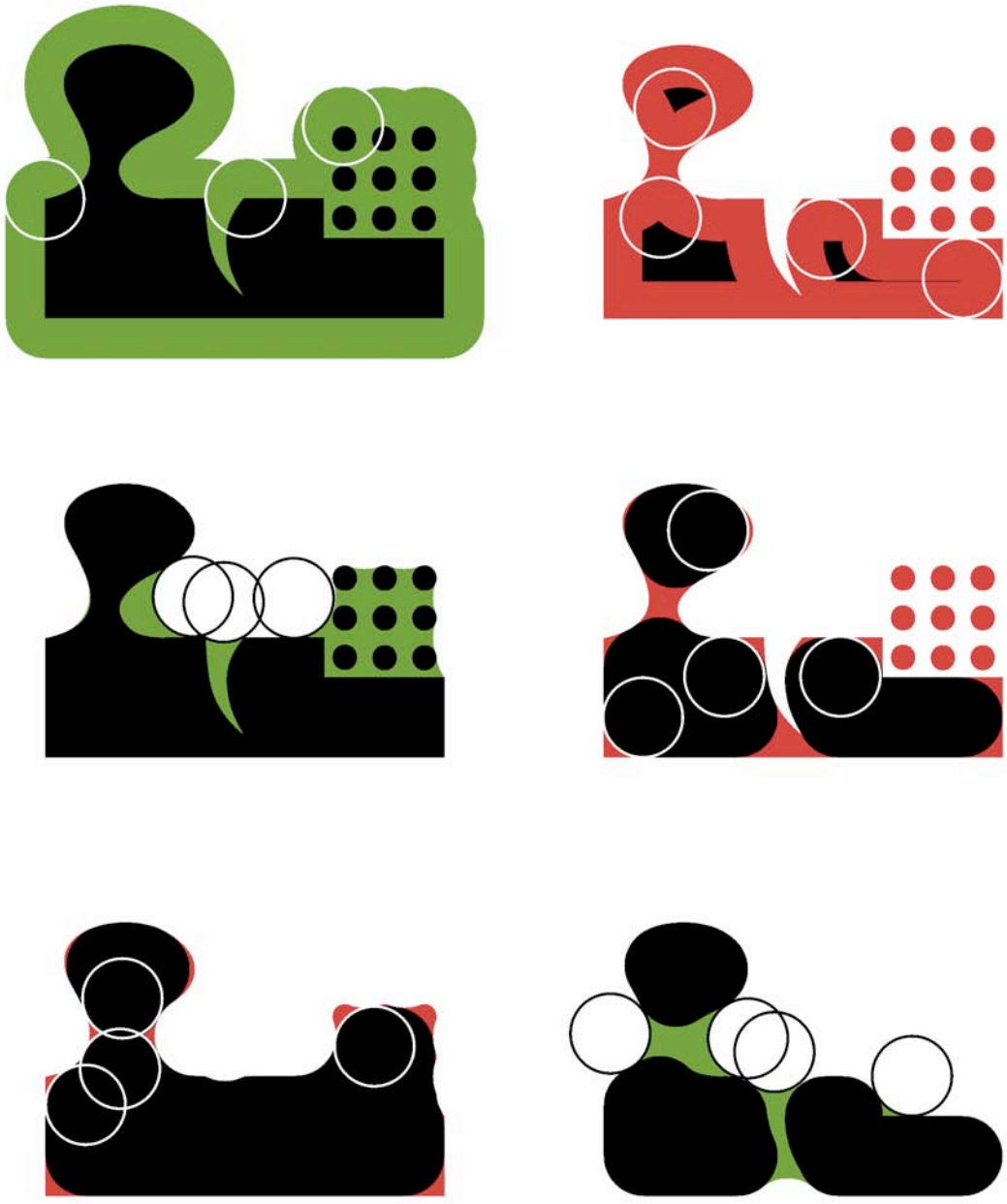

Figure 2: Morphological operations applied to the set S from Figure 1. The material the filters add is shown in green, while the material they remove is shown in red. (Top left) $S\uparrow$ (Top right) $S\downarrow$ (Middle left) $F(S)$ (Middle right) $R(S)$ (Bottom left) $R(F(S))$ (Bottom right) $F(R(S))$

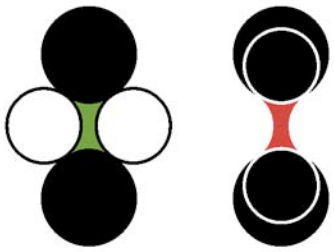

Figure 3: (Left) $S = R(S) = R(F(S))$ is inner-regular but not outer-regular. The outer-irregular points that filleting would add are shown in green. (Right) $F(S) = F(R(S))$ is outer-regular but not inner-regular. The inner-irregular points that rounding would remove are shown in red.

convex corners, and small components of S,) and the outer-irregular set $F(S) - S$ (which includes the gaps, concave corners, and holes.) We study simplification filters that are restricted to only affect the mortar, so that points in the core of the input shape are guaranteed to lie in the output, while points in the anticore are guaranteed to lie in the output's complement.

For a given r, we say that a set is *inner-regular* if $S = R(S)$ and that it is *outer-regular* if $S = F(S)$. A set is *regular* if it is both inner- and outer-regular, in which case $R(S) = S = F(S)$, and the mortar is empty. This definition of regularity for sets is used in [38], which inspired our definitions of stability, which generalizes the concept of regularity to all points of space.

Because rounding and filleting are idempotent, $R(S)$ and $R(F(S))$ are guaranteed to be inner-regular, while $F(S)$ and $F(R(S))$ are guaranteed to be outer-regular. In practice, $R(F(S))$ and $F(R(S))$ are regular almost everywhere. Intuitively, applying both rounding and filleting removes both the inner-irregularities and outer-irregularities from the original shape. However, it is possible for the second operation applied to introduce irregularities. For example, Figure 3 shows a set S for which neither $F(R(S))$ nor $R(F(S))$ is regular. Despite these limitations, a combination of rounding and filleting is among the most effective ways to eliminate irregular features.

3.4 Bias of $R(F(S))$ and $F(R(S))$ Filters

The $R(F(S))$ and $F(R(S))$ filters remove irregular portions of S, eliminating small holes and merging isolated components. Furthermore, they leave the regular portion of S unchanged. Hence, they are excellent candidates for shape simplification. Unfortunately, as illustrated in Section 5, each one of these two operators has a bias that can result in a significant change in the area of S. Because the mortar is $F(S) - R(S)$, filleting first completely fills in the mortar, while rounding first completely clears it out. The second operation simply regularizes the mortar's boundary. Consequently $R(F(S))$ is biased to

have more area in 2D or volume in 3D than S, while $F(R(S))$ is biased to have less. For instance, consider Figure 2. Note that $R(F(S))$ fills in the region containing the small circles, while $F(R(S))$ clears it out.

In this paper, we propose to eliminate this bias and choose between filling and clearing a region based on which causes the least amount of change to the model. We measure the change in terms of the area affected by simplification. Formally, if S' is the result of applying a particular filter to S, we want to minimize the cost of this simplification, which we define as the area of the symmetric difference between S and S'. For binary images, this amounts to counting the number of pixels that change color. Which filter causes the least change may vary at different locations in the image. Therefore, we want the flexibility of choosing between the two filters on a region-by-region basis.

3.5 Local Decisions

We cannot make the choice of the filter for each individual point. Therefore, our strategy is to segment the space into a suitable set of regions. For each region, we compute the number of pixels that would change status if we were to use *R(F(S))* and the number of pixels that would change status if we were to use *F(R(S))*. We then replace the region with the result produced by the one of these two filters that corresponds to the smaller count of altered pixels. Unfortunately, arbitrary space segmentations (such as the cells of a regular partition) will usually lead to sharp irregularities at the inter-region boundaries. For example, a regular lattice of 16x16 pixel regions may result in checkerboard patterns inside the mortar. In the following section, we explain how we identify more suitable candidate regions that avoid most irregularities.

3.6 Regions as Connected Components of the Mortar

The following two theorems justify our approach. In what follows we assume that all F and R operations are performed using the same radius r.

Theorem 1: $R(S) \subseteq R(F(S)) \subseteq F(S)$

Proof: Because $S \subseteq F(S)$ and $A \subseteq B \Leftrightarrow R(A) \subseteq R(B)$, $R(S) \subseteq R(F(S))$. Because $R(A) \subseteq A$, $R(F(S)) \subseteq F(S)$. Therefore $R(S) \subseteq R(F(S)) \subseteq F(S)$.

Theorem 2: $R(S) \subseteq F(R(S)) \subseteq F(S)$

Proof: The proof is similar to the proof of Theorem 1. Because $A \subseteq F(A)$, $R(S) \subseteq F(R(S))$. Because $R(S) \subseteq S$ and $A \subseteq B \Leftrightarrow F(A) \subseteq F(B)$, $F(R(S)) \subseteq F(S)$. Therefore $R(S) \subseteq F(R(S)) \subseteq F(S)$.

Combining them, we can conclude the following:

Theorem 3: The result of applying a combination of rounding and filleting to S is guaranteed to differ from S only in the mortar.

Thus we can use the maximally connected components, M_i, of the mortar as regions and replace each one of them with either $M_i \cap F(R(S))$ or $M_i \cap R(F(S))$, selecting the one with the smaller cost (i.e. count of altered pixels). This idea leads to a simple and effective algorithm, which we call Mason, discussed in detail in the next section.

4 Implementation Details

4.1 Identifying the Mortar

We first compute $M(S)$, $R(F(S))$, and $F(R(S))$ by combining the results produced by the standard $F(S)$ and $R(S)$ filters. Researchers have proposed several algorithms for computing morphological operations, which vary depending on the representation of the operands and the nature of the structuring element. [39], for instance, addresses the problem of approximately computing the Minkowski sum of two polyhedral models. [1, 2, 3] present methods for computing morphological operations involving CSG models and a ball, while [40] details an algorithm for computing operations between volume data and a box-shaped structuring element. As [41] points out, an efficient technique for computing morphological operations with binary image data and a ball is to grow and shrink by thresholding a Euclidean distance transform of the input. We implement this approach for both images and volumes using Danielsson's approximate vector propagation algorithm [42].

4.2 Identifying the Regions

We will describe the algorithm in terms of images and pixels. The extension to voxels and volume data is straightforward. To identify the connected components of the mortar, we associate a label with each pixel. We assign pixels in the core and anticore the same label, while pixels in the mortar are initially unlabeled. We scan through the image until we reach an unlabeled mortar pixel. We record the pixel in a list with a new label. We then start a breadth-first traversal by labeling the pixel and its four-connected neighbors and adding the neighbors to a queue. Then, until the queue is empty, we de-queue a pixel, label its unlabeled neighbors, and add them to the queue. When this process completes, all the mortar pixels in the same connected component share the same label. We continue the scan and repeat the process with a new label at the next unlabeled pixel. The technique described in [43] can be applied to reduce the storage requirements of this process.

4.3 Selecting the Best Filter for Each Region

For each connected component M_i of the mortar, we traverse the pixels in M_i and compute the count fr of pixels of M_i where S and $F(R(S))$ disagree and the count rf of pixels of M_i where S and $R(F(S))$ disagree. If $fr < rf$, then we copy the pixels from $F(R(S)) \cap M_i$ to our output image $Mason(S)$; otherwise, we copy the pixels from $R(F(S)) \cap M_i$.

5 2D Results

We present the images produced by applying combinations of rounding and filleting, as well as the Mason filter, to an image of a landscape (Figure 4). We provide statistics indicating for each image: the area, the number of pixels changed, the number of black components, and the number of white components. We also measure the number of inner-irregular and outer-irregular pixels (Table 1). The images are 400 by 300 pixels and were simplified with a radius of 5. The inner irregular pixels are shown in green, while

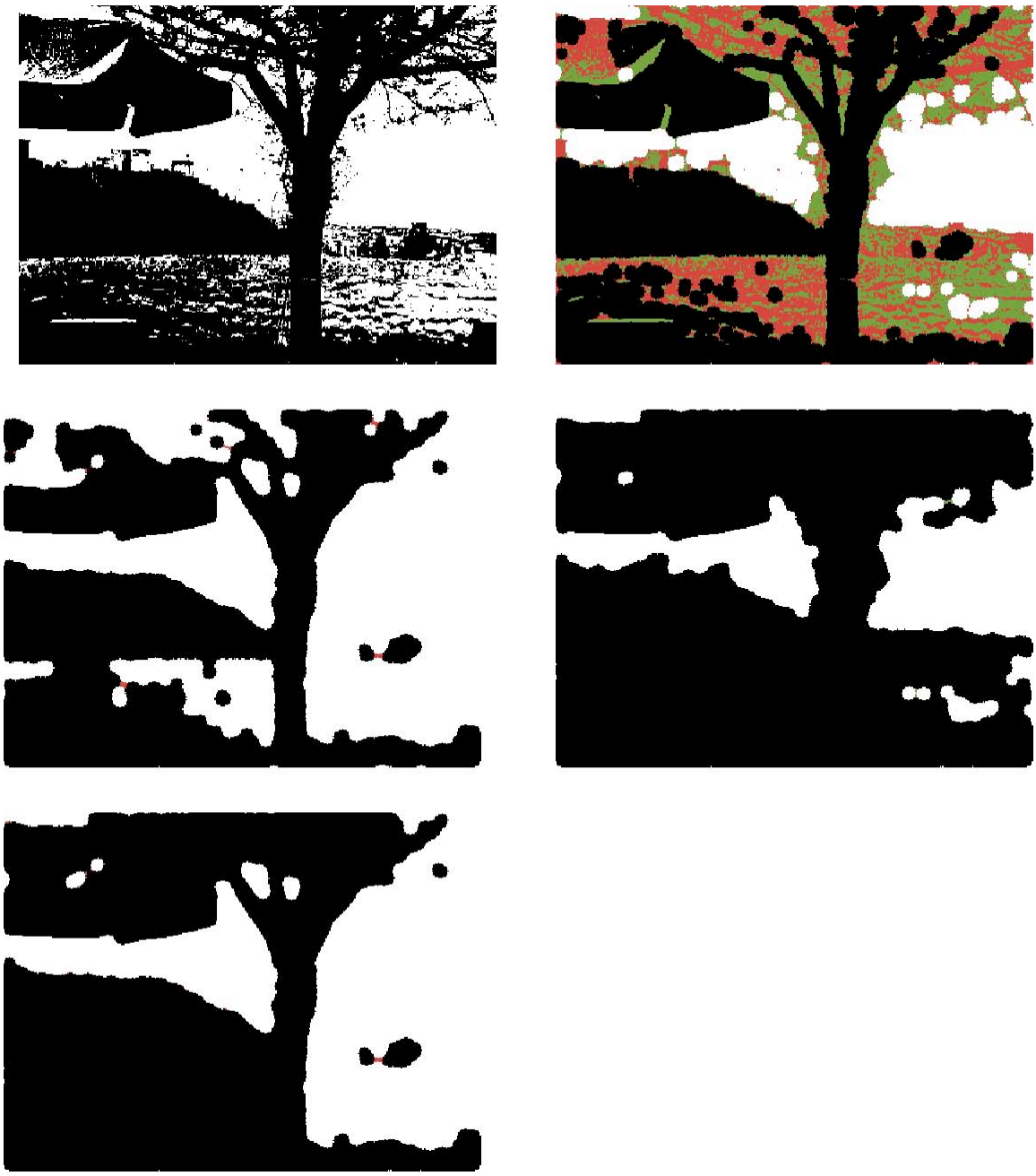

Figure 4: (Top left) The original image S. (Top right) The inner-irregular points of S are colored red, and the outer-irregular points are colored green. The core is black, while the anticore is white. (Middle left) $F\big(R(S)\big)$ Note the thin bridges of inner-irregular points marked in red. (Middle right) $R\big(F(S)\big)$ Note the green bridge of outer irregular points in the upper right portion of the figure. (Bottom left) $Mason(S)$.

	Area	Change	Black	White	Inner	Outer
S	78,195	0	527	711	18,676	18,641
$F(R(S))$	62,808	19,301	6	23	107	0
$R(F(S))$	95,472	18,573	1	33	0	43
$Mason(S)$	75,674	15,071	3	27	48	13

Table 1: Quantitative results for the images in Figure 4. From left to right, the columns are the area of the image (number of black pixels,) the area of the symmetric difference with the original image, the number of black connected components, the number of white components, the number of inner-irregular pixels, and the number of outer-irregular pixels.

	Area	Change	Black	White	Inner	Outer
S	36,967	0	98	177	7,970	7,312
$F(R(S))$	31,873	9,218	8	17	203	0
$R(F(S))$	41,842	8,523	5	20	0	89
$Mason(S)$	38,013	7,922	5	22	203	108

Table 2: Quantitative results for the images in Figure 5. The columns are as in Table 1.

the outer irregular pixels are shown in red. Execution times for this and all other two-dimensional examples are negligible.

As the results indicate, $F(R(S))$, $R(F(S))$, and $Mason(S)$ yield similar, and substantial, reductions in the topological complexity and irregularity of S. $Mason(S)$ is neither inner regular nor outer regular, but it has a smaller number of pixels changed and a smaller change in total area than the other filters. As noted earlier, $F(R(S))$ fills in thick connected components of the mortar, while $R(F(S))$ clears them out. $Mason(S)$ chooses between filling and clearing based on which results in the smallest number of pixels changed. For instance, Mason chooses to use $F(R(S))$ to paint white the predominantly white portion of water to the right of the tree and chooses to use $R(F(S))$ to paint black the predominantly black portion of the top left part of the image. In this image, some components contain mostly inner irregular pixels while others contain mostly outer irregular pixels, so Mason's pattern of filling and clearing achieves significantly lower cost than either filling in all the components or clearing them all out.

In some cases, the advantages of Mason over $F(R(S))$ or $R(F(S))$ are small. For example, Figure 5 and Table 2 illustrate a situation where Mason's advantages are less pronounced. In this 300 by 225 pixel data from a slide of lung cells [44], simplified with a radius of 5, most of the connected components of the mortar contain comparable quantities of inner irregular and outer irregular pixels, so all patterns of filling and clearing yield similar numbers of pixels changed.

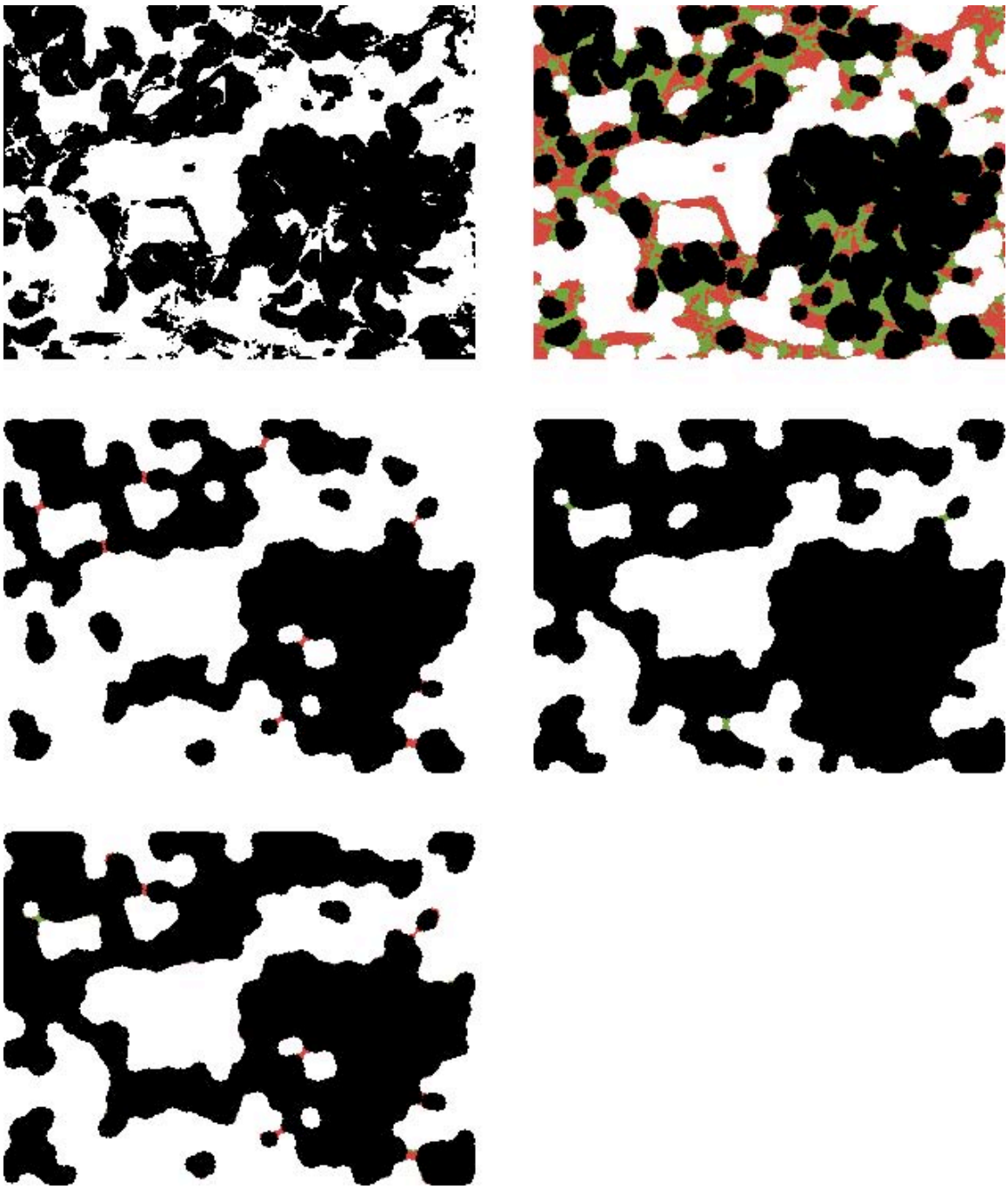

Figure 5: (Top left) The original image S. (Top right) The inner-irregular points of S are colored red, and the outer-irregular points are colored green (Middle left) $F(R(S))$ (Middle right) $R(F(S))$ (Bottom left) $Mason(S)$

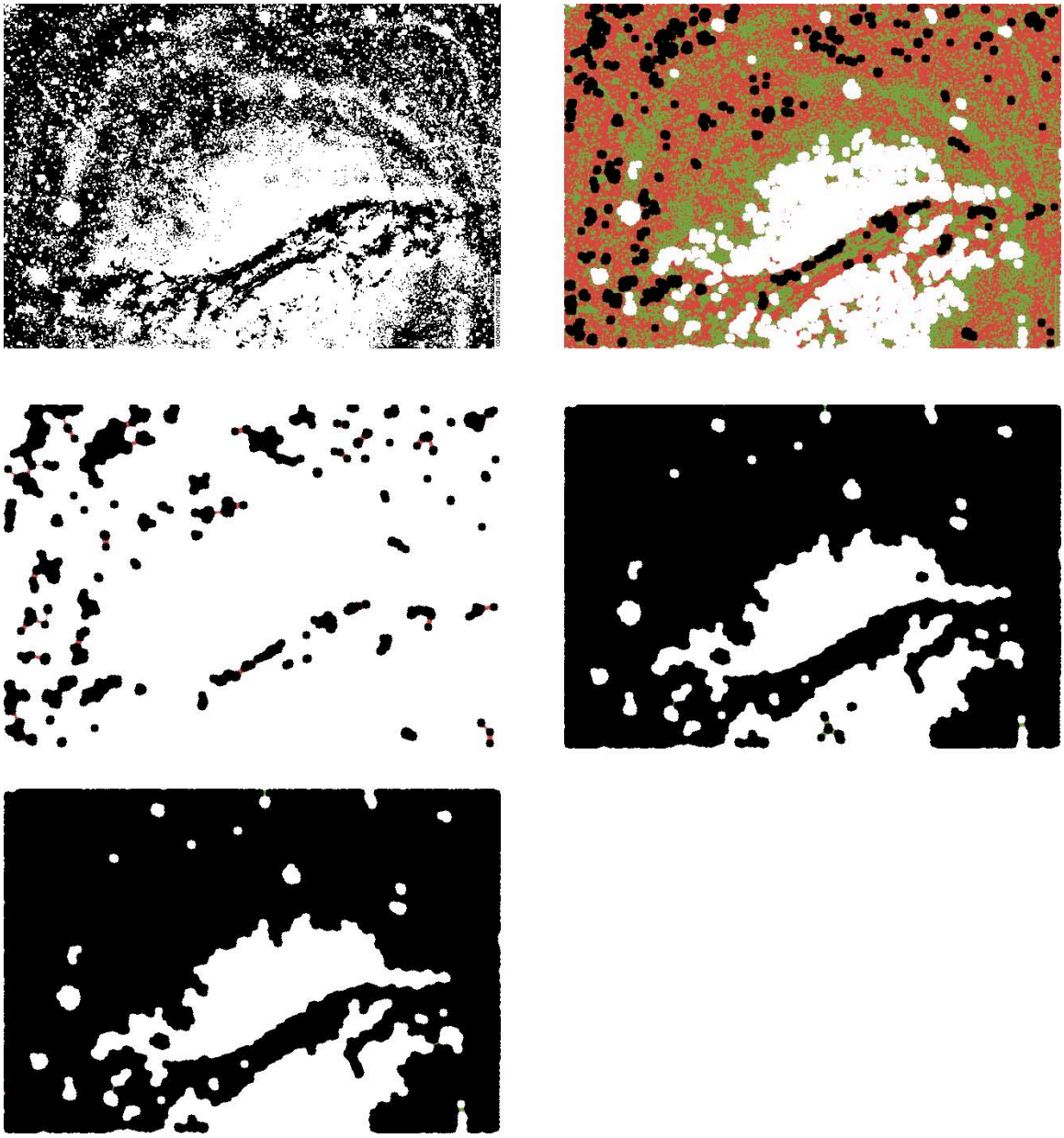

Figure 6: (Top left) The original image S. (Top right) The inner-irregular points of S are colored red, and the outer-irregular points are colored green (Middle left) $F(R(S))$ (Middle right) $R(F(S))$ (Bottom left) $Mason(S)$

	Area	Change	Black	White	Inner	Outer
S	128,956	0	2,212	3,218	101,717	79,769
$F(R(S))$	29,993	100,969	64	6	679	0
$R(F(S))$	204,213	79,225	9	145	0	154
$Mason(S)$	203,329	78,923	3	145	15	107

Table 3: Quantitative results for the images in Figure 6.

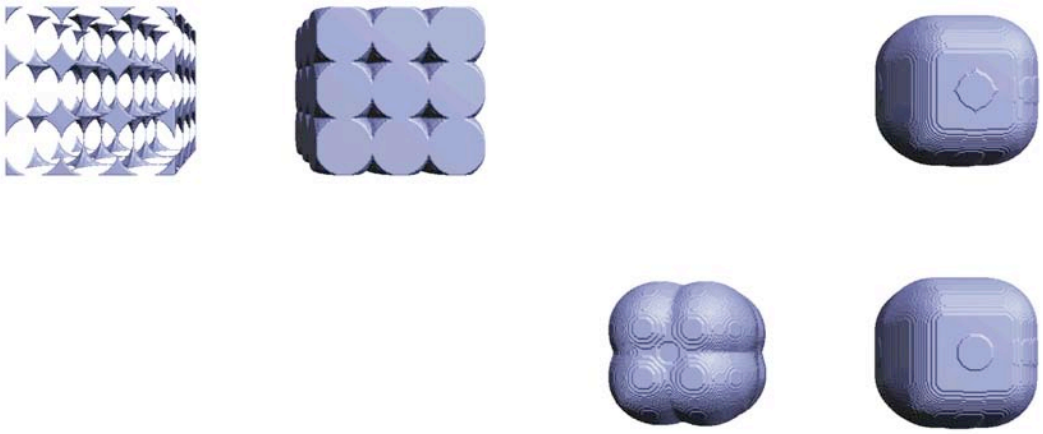

Figure 7: (Top left) Original shape S (Top right) $Mason(S)$ (Bottom left) $F(R(S))$ (Bottom right) $R(F(S))$

Finally, noise in the data may fuse the mortar regions where we would like to use $F(R(S))$ and regions where we would like to use $R(F(S))$ into a single very large component. In such cases, Mason is forced to make a single choice for both kinds of regions. For instance, consider Figure 6 and Table 3, which highlight a limitation shared by $Mason(S)$, $F(R(S))$, and $R(F(S))$. In this 620 by 430 pixel data drawn from an image of a galaxy [45], also simplified with a radius of 5, predominantly white regions with black noise are adjacent to predominantly black regions with white noise, forming a single connected component of mortar. In this case, Mason is forced to make a single decision for the mixed component and hence loses its advantage over $R(F(S))$ and $F(R(S))$. $F(R(S))$ clears out this component, while $R(F(S))$ and $Mason(S)$ fill it in. All three filters result in a large area change.

6 3D Results

Figure 7 presents a synthetic example designed to illustrate the dramatic differences between $Mason(S)$, $F(R(S))$, and $R(F(S))$. The right half of the 425 x 150 x 150 model consists of 27 overlapping balls arranged to almost completely fill the space inside a

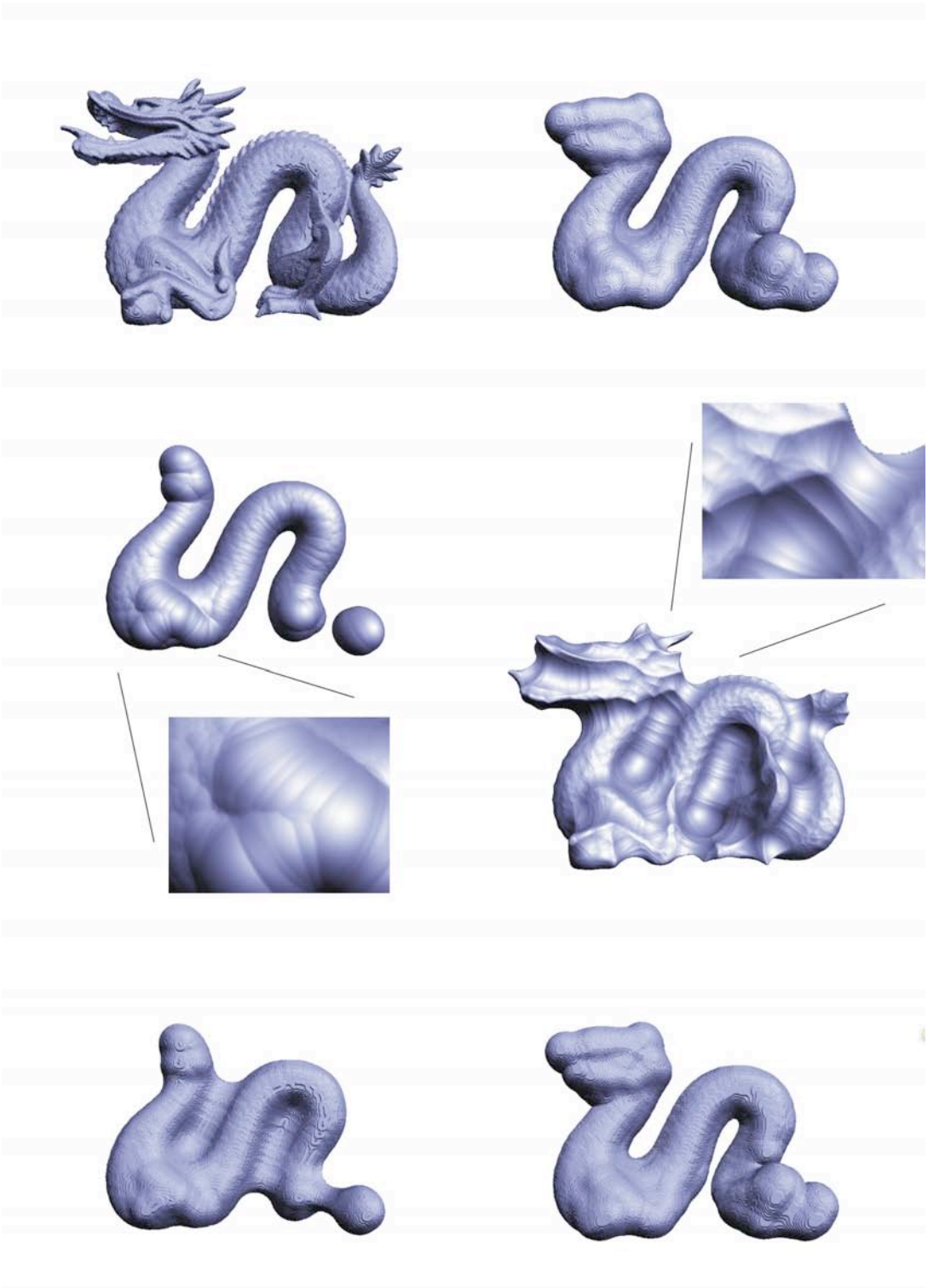

Figure 8: (Top left) Original shape S. (Top right) $Mason(S)$ (Middle left) $R(S)$. The inset shows that $R(S)$ is a union of spheres. (Middle right) $F(S)$. The inset shows that $\overline{F(S)}$ is a union of spheres. (Bottom left) $F(R(S))$ (Bottom right) $R(F(S))$

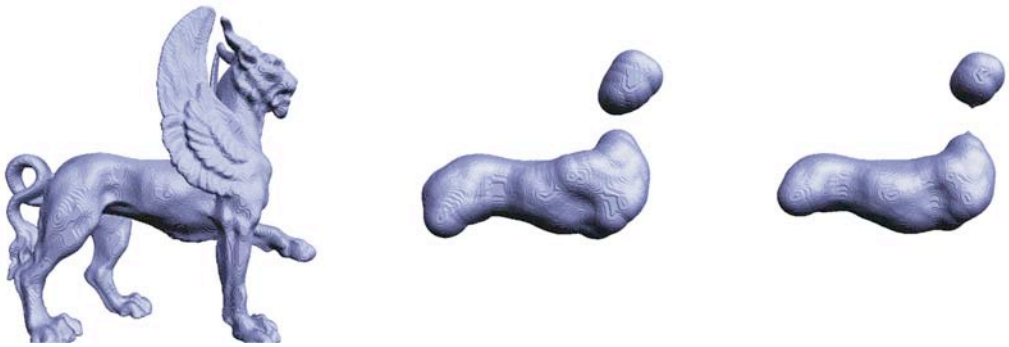

Figure 9: (Left) Original shape S. (Middle) $R(F(S))$ (Right) $F(R(S))$

cube, while the left half is the complement of the right half. The model has a volume of 3,375,000 and was simplified with a radius of 50 voxels . $F(R(S))$ clears out both halves of the model, resulting in a symmetric difference equal to the original volume, while $R(F(S))$ fills them in, resulting in a symmetric difference of 3,068,335 and a total volume of 4,832,105 voxels. $Mason(S)$, by contrast, yields a symmetric difference of 95,308 and a total volume of 2,591,312 voxels.

Although there are several connected components of the Mortar for the model shown in Figure 8, Mason makes the same choice for all of them, and so its result is similar to that of $R(F(S))$. The symmetric differences of Mason and $R(F(S))$ are 1,872,535 and 1,870,597, while $F(R(S))$ has a symmetric difference of 2,336,582 voxels. The original shape is 363 x 231 x 514, with a volume of 7,463,327, and was simplified with a radius of 32 voxels.

The middle images in Figure 8 illustrate the rounding and filleting of volume data. In contrast to the other isosurfaces we show, which are reconstructed from binary volume data, these isosurfaces were extracted from the signed distance field. There are ridges visible in the insets because the distance field is computed from discrete points rather than a continuous surface. Consequently, the rounding is only a union of a finite number of spheres, as is the portion of the complement of the filleting within the data volume.

Figure 9 shows a case where $R(F(S)), F(R(S))$ and $Mason(S)$ are practically identical.

Figure 10 illustrates the result of varying the radius of simplification. The original Buddha appears on the left, and the succeeding images show it simplified with Mason using radii of 16, 32, 48, and 64 voxels.

The computation time for these results is dominated by the calculation of the distance transform, which requires a time proportional to the number of voxels it processses. We estimate that it takes 15 seconds (on an 800 MHz Powerbook G4 with 1 GB RAM) to process 1,000,000 voxels. Our input sizes range from 10-56 million voxels. However, to store the result of the growing operation, we need to pad the models on all sides by the radius of simplification, which can significantly increase the number of voxels we process. Our padded models range from 33-94 million voxels, corresponding to computation times of 8–24 minutes for each distance transform. It takes two transforms to compute $R(S)$ or $F(S)$, and another two transforms to compute $R(F(S))$ or $F(R(S))$

Figure 10: (Left to right) The original shape S and the results of applying Mason with radii of 16, 32, 48, and 64 voxels.

as well. This means we need a total of eight transforms to compute $Mason(S)$, correspondiong to computation times of 1-3 hours. $Mason(S)$ requires an additional time of approximately 30 seconds per 1,000,000 voxels, but it does not require padding, so that the total extra time required by $Mason(S)$ ranges from 5-28 minutes.

7 Discussion

Given a shape S, our problem has been to find a nearly r-regular shape S' such that the symmetric difference $S \otimes S'$ has minimal area and lies completely within the mortar $M(S)$. Ideally our method would produce a fully r-regular shape. However, in some cases there is no r-regular S' such that $S \otimes S' \subseteq M(S)$. For example, consider the set S shown in Figure 3, and let S' be a set such that $S \otimes S' \subseteq M(S)$, which is equivalent to $R(S) \subseteq S' \subseteq F(S)$. Suppose $S' \supset S$. $S' \subseteq F(S)$ implies $R(S') \subseteq R(F(S))$, while $R(F(S)) = R(S)$ by construction. Then $R(S') \subseteq R(S)$, but because $R(S) \subseteq S'$, $R(S) \subseteq R(S')$, because rounding is idempotent. Consequently, $R(S') = R(S)$, while $R(S) = S$ by construction. But because $S \neq S'$, this implies $S' \neq R(S')$, so S' is inner-irregular. Alternatively, if $S = S'$, then because $S \neq F(S)$ by construction, $S' \neq F(S')$, so S' is outer-irregular. As a result, solutions that produce a regular set may have to modify space outside of the mortar.

8 Summary of Contributions and Conclusions

Previously proposed shape simplification techniques used Hausdorff or other error measures for shape simplification that allowed the shape to be altered both in regular and in irregular regions. We have proposed a new tolerance zone, called the mortar, that confines the effects of shape simplification to irregular regions, where the boundary has high curvature or where two distinct portions of the boundary are close to each other. Such a confinement ensures that the low-curvature, regular portions of the boundary are not affected by simplification. We have developed a mathematically simple set-

theoretical definition of the mortar that is independent of the dimension of the underlying space and of the choice of the representation. Furthermore, we have demonstrated that the mortar may be identified using a simple and efficient implementation on discrete models in 2D and 3D. We have proven that standard mathematical morphology operators (combinations of rounding and filleting) only alter the shape inside the mortar, and we have shown that they tend to produce shapes that are regular almost everywhere. We have pointed out that for some portions of the image, one of these operators may be better, while the other is more suitable for other portions. We have therefore proposed to split the image into regions and to select the most suitable filter for each region independently. To ensure that the disparity of filter choices between adjacent regions does not result in unexpected irregularities, we have devised an approach where the regions are identified as the connected component of the mortar. We measure the suitability of a filter as the count of pixels it changes. We have demonstrated that our approach, which we call the Mason filter, produces nearly r-regular shapes with less change to the model than combinations of standard rounding and filleting operations. In addition, Mason is self-dual, so that $Mason(S) = \overline{Mason(\overline{S})}$, while $R(F(S))$ and $F(R(S))$ are dual of each other, meaning that $R(F(S)) = \overline{F(R(\overline{S}))}$. Unlike rounding and filleting, the Mason filter treats positive and negative space symmetrically.

9 Acknowledgements

This research was supported by a DARPA/NSF CARGO grant number 0138420. We thank Alex Powell for his assistance in developing a complete system for performing arbitrary combinations of morphological operators on binary images.

References

[1] J. ROSSIGNAC AND A. REQUICHA, Constant-radius blending in solid modeling, *ASME Computers in Mechanical Engineering* 3 (1984) 65-73.

[2] J. ROSSIGNAC, Blending and offsetting solid models, Ph.D. thesis, University of Rochester, NY, 1985.

[3] J. ROSSIGNAC AND A. REQUICHA, Offsetting operations in solid modeling, *Computer-Aided Geometric Design* 3 (1986) 129-148.

[4] J. ROSSIGNAC AND P. BORREL, Multi-resolution 3D approximations for rendering complex scenes, in: B. Faliciendo and T. Kunii (Eds.), *Geometric Modeling in Computer Graphics*, Springer-Verlag, Berlin, 1993, pp. 455-465.

[5] H. HOPPE, Progressive meshes, *SIGGRAPH 96* (1996) 99-108.

[6] R. RONFARD AND J. ROSSIGNAC, Full range approximation of triangulated polyhedra, *Eurographics 96* (1996) 67-76

[7] M. GARLAND AND P. HECKBERT, Simplifying surfaces with color and texture using quadratic error metric, *IEEE Visualization 98* (1998) 287-295.

[8] D. LUEBKE, View-dependent simplification of arbitrary polygonal environments, Ph.D. thesis, University of North Carolina, Chapel Hill, 1985.

[9] P. LINDSTROM, Out-of-core simplification of large polygonal models, *SIGGRAPH 2000* (2000) 271-278.

[10] P. CIGNONI, C. ROCCHINI, AND R. SCOPIGNO, Metro: measuring error on simplified surfaces, *Eurographics 98* 17, 2 (1998) 167-174.

[11] P. LINDSTROM AND G. TURK, Fast and memory efficient polygonal simplification, *IEEE Visualization 98* (1998) 279-286.

[12] D. COHEN-STEINER, P. ALLIEZ, AND M. DESBRUN, Variational shape approximation, *SIGGRAPH 2004* (2004) 905-914.

[13] J. COHEN, A. VARSHNEY, ET. AL., Simplification envelopes, *SIGGRAPH 96* (1996) 119-128.

[14] P. ALLIEZ, N. LAURENT, AND H. SANASON, Mesh approximation using a volume-based metric, *Seventh Pacific Conference on Computer Graphics and Applications* (1999) 292-301.

[15] D. LUEBKE AND C. ERIKSON, View-dependent simplification of arbitrary polygonal environments. *SIGGRAPH 97* (1997) 199-208.

[16] J. POPOVIC AND H. HOPPE, Progressive simplicial complexes, *SIGGRAPH 97* (1997) 217-224.

[17] C. ERIKSON AND D. MANOCHA, GAPS: General and automatic polygonal simplification, *Proc. of the 1999 Symposium on Interactive 3D Graphics* (1999) 79-88.

[18] Z. WOOD, H. HOPPE, ET. AL., Removing excess topology from isosurfaces, *ACM Transactions on Graphics* 23, 2 (2004) 190-208.

[19] A. SZYMCZAK AND J. VANDERHYDE, Extraction of topologically simple isosurfaces from volume datasets, *IEEE Visualization 2003* (2003) 67-74.

[20] J. EL-SANA AND A. VARSHNEY, Topology simplification for polygonal virtual environments, *IEEE Transactions on Visualization and Computer Graphics* 4, 2 (1998) 133-144.

[21] F. NOORUDDIN AND G. TURK, Simplification and repair of polygonal models using volumetric techniques, *IEEE Transactions on Visualization and Computer Graphics* 9, 2 (2003) 191-205.

[22] T. HE, L. HONG, ET. AL, Controlled topology simplification. *IEEE Transactions on Visualization and Computer Graphics* 2, 2 (1996) 171-183.

[23] G. CELNIKER AND D. GOSSARD, Deformable curve and surface finite elements for free-form shape design, *SIGGRAPH 91* (1991) 257-266.

[24] S. HAHMANN AND S. KONZ, Fairing bi-cubic B-spline surfaces using simulated annealing, in: A. Le Mehaute, C. Rabut, and L. Schumaker (Eds.), *Curves and Surfaces with Applications in CAGD*, Vanderbilt University Press, Nashville, 1997, pp. 159-168.

[25] H. MORETON AND C. SEQUIN, Functional optimization for fair surface design, *SIGGRAPH 92* (1992) 167-176.

[26] G. WESTGARD AND H. NOWACKI, Construction of fair surfaces over irregular meshes. *ACM Symposium on Solid Modeling and Applications* (2001) 88-98.

[27] S. FLEISCHMAN, I. DRORI, AND D. COHEN-OR, Bilateral mesh denoising. *SIGGRAPH 2003* (2003) 950-953.

[28] T. JONES, F. DURAND, AND M. DESBRUN, Non-iterative, feature-preserving mesh smoothing, *SIGGRAPH 2003* (2003) 943-949.

[29] C. BAJAJ AND G. XU, Anisotropic diffusion of surfaces and functions on surfaces, *ACM Transactions on Graphics*, 22, 1 (2003) 4-32.

[30] M. DESBRUN, M. MEYER, ET. AL., Implicit fairing of irregular meshes using diffusion and curvature flow, *SIGGRAPH 99* (1999) 317-324.

[31] G. TAUBIN, Curve and surface smoothing without shrinkage, *Fifth International Conference on Computer Vision* (1995) 852-857.

[32] H. ZHANG AND E. FIUME, Butterworth filtering and implicit fairing of irregular meshes, *11th Pacific Conference on Computer Graphics and Applications* (2003) 502-506.

[33] R. TAM AND W. HEIDRICH, Shape simplification based on the medial axis transform, *IEEE Visualization 2003* (2003) 481-488.

[34] H. HEIJMANS, *Morphological Image Operators*, Academic Press, New York, 1994.

[35] J. SERRA, *Image Analysis and Mathematical Morphology*, Academic Press, 1982.

[36] R. FAROUKI, H. MOON, AND B. RAVANI, Minkowski geometric algebra of complex sets, *Geometria Dedicata* 85 (2001) 283-315.

[37] H. MUHLTHALER AND H. POTTMANN, Computing the Minkowski sum of ruled surfaces, *Graphical Models* 65 (2003) 369-384.

[38] D. ATTALI, r-Regular shape reconstruction from unorganized points, *Computational Geometry* (1997) 248-253.

[39] G. VARADHAN AND D. MANOCHA, Accurate Minkowski sum approximation of polyhedral models, *12th Pacific Conference on Computer Graphics and Applications* (2004) .

[40] J. ROODRIGUEZ, D. AYALA, AND A. AGUILERA, EVM: A complete solid model for surface rendering, in: G. Brunnet, B. Hamann, et. al. (Eds.), *Geometric Modeling for Scientific Visualization*, Springer-Verlag, 2004, pp. 259-274.

[41] O. CUISENAIRE, Distance transformations: fast algorithms and applications to medical image processing, Ph.D. thesis, Universite catholique de Louvain, 1999.

[42] P. DANIELSSON, Euclidean distance mapping, *Computer Graphics and Image Processing* 14 (1980) 227-248.

[43] C. ANDUJAR, Space-efficient connectivity test of n-dimensional images, *Computers & Graphics* 22, 4 (1984) 557-558.

[44] S. FAIRY, available at: http://www.cytology-asc.com/branch/nsw/jun03/cases.html, 2003

[45] R. STENGER, available at: http://www.cnn.com/2002/TECH/space/10/25/galactic.crash, 2002.