

Research Article

Shells and Spheres: An n -Dimensional Framework for Medial-Based Image Segmentation

Aaron Cois,¹ John Galeotti,¹ Robert Tamburo,¹ Michael Sacks,² and George Stetten¹

¹749 Benedum Hall, University of Pittsburgh, Pittsburgh, PA 15261, USA

²306 CNBIO, 300 Technology Drive, University of Pittsburgh, Pittsburgh, PA 15219, USA

Correspondence should be addressed to George Stetten, stetten@pitt.edu

Received 6 October 2009; Revised 1 February 2010; Accepted 12 April 2010

Academic Editor: Guo W. Wei

Copyright © 2010 Aaron Cois et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We have developed a method for extracting anatomical shape models from n -dimensional images using an image analysis framework we call *Shells and Spheres*. This framework utilizes a set of spherical operators centered at each image pixel, grown to reach, but not cross, the nearest object boundary by incorporating “shells” of pixel intensity values while analyzing intensity mean, variance, and first-order moment. Pairs of spheres on opposite sides of putative boundaries are then analyzed to determine boundary *reflectance* which is used to further constrain sphere size, establishing a consensus as to boundary location. The centers of a subset of spheres identified as *medial* (touching at least two boundaries) are connected to identify the interior of a particular anatomical structure. For the automated 3D algorithm, the only manual interaction consists of tracing a single contour on a 2D slice to optimize parameters, and identifying an initial point within the target structure.

1. Introduction

The framework of Shells and Spheres described in this paper is based on a set of spheres called a *sphere map*. A sphere map consists of exactly one sphere centered at each image pixel, whose radius can be adjusted. Calculations denoted as *Variable-Scale Statistics* (VSSs) are performed on populations of pixels within spheres, as well as populations of adjacent and overlapping spheres. Memory and computational requirements are kept reasonable by storing only a relatively small, fixed number of VSSs at every pixel, many of which can be updated incrementally when growing or shrinking spheres. The ultimate goal of adjusting radii is to produce a sphere map in which each sphere is as large as possible without crossing a boundary. In the optimized sphere, the sphere map is thus equivalent to what is commonly known as a *distance map* [1], that is, the distance to the nearest boundary. Though the task is trivial with binary images, where definitive boundaries are known, it presents a challenge when boundaries are difficult to determine due to noise and tissue inhomogeneity. Our approach is well suited to this challenge, with the caveat that

the correctness of segmentation of real images is generally subjective.

Many conventional methods for image processing consider a region of fixed size and shape, usually referred to as a *kernel*, especially when used for convolution. Other common approaches define dynamic regions adjoining boundaries using deformable contours [2] or level sets [3]. Our approach, instead, uses a set of spheres whose individual radii are optimized using VSS operators to achieve maximum discrimination between adjoining image regions. Not only do such spheres provide highly representative populations for boundary detection, but those spheres that touch at least two boundaries are also *medial*, providing a basis for medial feature extraction. Unlike Gaussian blurring, commonly used in multiscale analysis [4], Shells and Spheres preserves sharp boundaries with increasing scale. This paper presents the notation and basic operators of Shells and Spheres for computing VSS. Using this framework, a wide variety of algorithms for sphere map optimization are possible, and we present two such algorithms, a trivial one for noiseless images and a practical one for real medial images with noise. We then extend this algorithm to include methods to identify

boundary and medial locations, followed by an application of that extended algorithm to image segmentation.

A primary goal of the research presented here is to extract medial ridges from images. The lineage of the medial approach may be traced to the medial axis (otherwise known as the symmetric axis or skeleton) introduced on binary images by Blum and developed [5–7]. Pizer extended the medial axis to gray-scale images, producing a graded measure called *medialness*, which links the aperture of the boundary measurement to the radius of the medial axis to produce what has been called a *core*. A core is a locus in a space whose coordinates are position, radius, and associated orientations [8, 9]. Methods involving these continuous loci of medial primitives have proven particularly robust against noise and variation in target shapes [10]. Models including discrete loci of medial primitives have also provided the framework for a class of active shape models known as *deformable m-reps* (sampled medial representations) [11], as well as a statistical approach using pairs of detected boundary points known as *core atoms* developed previously by Stetten and Pizer [12]. An excellent review of medial approaches and theory may be found in a recent text by Siddiqi and Pizer [13].

The following sections will present the fundamental framework and basic operators of Shells and Spheres, as well as a detailed description of the two algorithms just mentioned. A summary of the experimental validation already published elsewhere is also included, and a brief discussion.

2. Shells and Spheres Framework

We begin by defining our notation. Shells and Spheres is inherently n -dimensional. For brevity, we use the term *sphere* regardless of image dimension, instead of the dimension-specific terms *disk* (*circle*), *sphere*, or *hypersphere*. Figures are presented in 2D for ease of illustration.

Since the framework of Shells and Spheres is used to gather statistics on dynamic populations of pixels, we adopt a hybrid form of notation derived from standard set theory and statistics. We denote vectors by lowercase bold-faced letters (\mathbf{x}), scalars by lowercase italic letters (r), and sets by uppercase letters (S). We use \mathbb{Z} to denote the set of all integers, and $\Omega \subset \mathbb{Z}^n$ to denote the set of all pixel locations in a sampled n -dimensional image.

Given an n -dimensional image with intensities $f(\mathbf{x})$ for $\mathbf{x} \in \Omega$, we define a *sphere map*, which assigns the radius $r(\mathbf{x})$, the radius of a sphere centered at pixel \mathbf{x} . We define a sphere to be an n -dimensional neighborhood of pixels that lie within a radius r of a center point. We use an integer value for r , such that a sphere of radius r centered at a pixel \mathbf{x} is given by

$$S_r(\mathbf{x}) = \{\mathbf{y} : \text{round}(|\mathbf{y} - \mathbf{x}|) \leq r, \mathbf{y} \in \Omega\}, \quad (1)$$

where $S_r(\mathbf{x})$ is a sphere of radius r centered at image pixel \mathbf{x} , and \mathbf{y} is a pixel within that sphere. Note the shorthand notation for the subscript r , meaning $r(\mathbf{x})$, the radius of the particular sphere at \mathbf{x} as given by the sphere map. In some instances, the reader will encounter an example with a different subscript, such as $S_1(\mathbf{x})$, meaning a sphere of radius

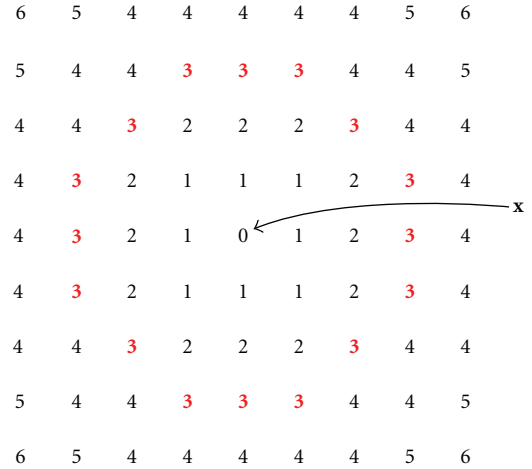


FIGURE 1: Each pixel is shown as a number indicating its integer distance from the central pixel. If we denote the central pixel as \mathbf{x} , then pixels labeled n are members of the set $H_n(\mathbf{x})$. For example, the pixels labeled “3” (shown in bold) comprise the shell $H_3(\mathbf{x})$.

1, irrespective of $r(\mathbf{x})$. By definition, $\mathbf{x} \in S_r(\mathbf{x})$ for all \mathbf{x} , even when $r(\mathbf{x}) = 0$, and hence $S_r(\mathbf{x})$ is always nonempty.

A *shell* is a set of all pixels whose distance to the center rounds to a given radius, defined for radius r as

$$H_r(\mathbf{x}) = \{\mathbf{y} : \text{round}(|\mathbf{y} - \mathbf{x}|) = r, \mathbf{y} \in \Omega\}. \quad (2)$$

Shells are nonoverlapping such that for concentric shells,

$$H_p(\mathbf{x}) \cap H_q(\mathbf{x}) = \emptyset, \quad p \neq q. \quad (3)$$

Additionally, shells are space filling, and thus a sphere of radius r may be formed from a union of shells,

$$S_r(\mathbf{x}) = \bigcup_{k=0}^r H_k(\mathbf{x}). \quad (4)$$

Figure 1 illustrates the distribution of pixels in a series of concentric shells surrounding a central pixel in a 2D image. Each pixel is labeled with its integer radius from the central pixel (labeled “ \mathbf{x} ”). The shell $H_3(\mathbf{x})$ is shown as the pixels labeled “3” in bold.

Figure 2 shows an image containing two noiseless objects with pixel intensities of 1 and 9, respectively. Note that pixels in this case are represented by their intensity. The boundary between the image objects is identified by a straight dashed line. Pixel \mathbf{x} is surrounded by a concentric set of four shells $H_0(\mathbf{x})$, $H_1(\mathbf{x})$, $H_2(\mathbf{x})$, and $H_3(\mathbf{x})$, shown separated by dashed circles. Shell $H_3(\mathbf{x})$ is truncated by the edge of the image. The union of all four shells is $S_r(\mathbf{x})$, shown enclosed by a solid circle, also truncated by the edge of the image, with a radius governed by the value of $r(\mathbf{x}) = 3$ in the sphere map. Similarly, on the other side of the boundary, pixel \mathbf{y} with an intensity value of 9 has three shells whose union $S_r(\mathbf{y})$ has a radius $r(\mathbf{y}) = 2$. Both $S_r(\mathbf{x})$ and $S_r(\mathbf{y})$ touch but do not cross the boundary and are therefore correctly optimized.

The correctly optimized sphere map of the image in Figure 2 is shown in Figure 3, with each pixel represented by

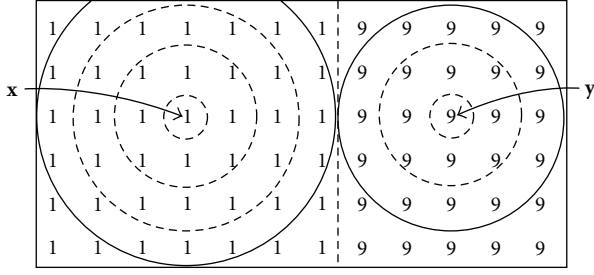


FIGURE 2: Noiseless image with boundary between two objects. Correctly scaled spheres $S_r(\mathbf{x})$ with $r(\mathbf{x}) = 3$ and $S_r(\mathbf{y})$ with $r(\mathbf{y}) = 2$ touch, but do not cross, the boundary. Numbers indicate pixel intensity.

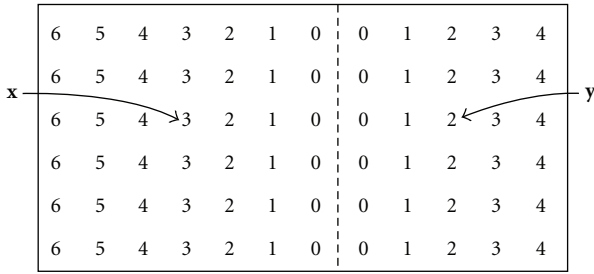


FIGURE 3: Correctly optimized sphere map of the image in Figure 2. Numbers indicate the integer radius of the sphere at each pixel. Pixels \mathbf{x} and \mathbf{y} are labeled as before.

the radius of the sphere centered at that pixel. Note the linear increase in sphere radius with distance from the boundary and the fact that the radius equals zero adjacent to the boundary.

3. Variable Scale Statistics

We derive a number of statistics at pixel \mathbf{x} (and every other pixel), calculated on the intensities of pixels within the sphere $S_r(\mathbf{x})$. Since these statistics depend on the radii of the spheres, we call them Variable Scale Statistics (VSS). We denote as *primary statistics* those VSS at \mathbf{x} calculated using only the population of pixels within $S_r(\mathbf{x})$. Higher-order *secondary statistics* are VSS derived from multiple spheres.

4. Primary Statistics

The primary statistics at pixel \mathbf{x} concern only the population of pixels within the sphere $S_r(\mathbf{x})$. Thus the *mean* at pixel \mathbf{x} is the mean intensity of all pixels within the population $S_r(\mathbf{x})$ at its current radius, defined as

$$\mu(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})|} \sum_{\mathbf{y} \in S_r(\mathbf{x})} f(\mathbf{y}), \quad (5)$$

where $|S_r(\mathbf{x})|$ is the number of pixels in $S_r(\mathbf{x})$ and $f(\mathbf{y})$ is the image intensity at pixel \mathbf{y} .

The *variance* at pixel \mathbf{x} is defined as

$$\sigma^2(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})| - 1} \sum_{\mathbf{y} \in S_r(\mathbf{x})} [f(\mathbf{y}) - \mu(\mathbf{x})]^2, \quad |S_r(\mathbf{x})| > 1. \quad (6)$$

Variance may only be calculated when $|S_r(\mathbf{x})| > 1$, since when $r(\mathbf{x}) = 0$ there is only one pixel in $S_r(\mathbf{x})$ and variance is not defined. The standard deviation $\sigma(\mathbf{x})$ of the intensities of the set of pixels within the sphere centered at pixel \mathbf{x} is simply the square root of the variance.

The *first-order moment* of intensities within $S_r(\mathbf{x})$ is given by

$$\mathbf{m}(\mathbf{x}) = \sum_{\mathbf{y} \in S_r(\mathbf{x})} (\mathbf{y} - \mathbf{x})f(\mathbf{y}). \quad (7)$$

Due to the finite extent of an image's domain Ω , a sphere may be truncated by one or more edges of the image (e.g. $S_3(\mathbf{x})$ in Figure 2). Unlike conventional kernels, which usually require pixel values outside the image to be arbitrarily defined, our spherical sets simply exclude such locations from calculations. Thus, truncation will not adversely affect $\mu(\mathbf{x})$ or $\sigma(\mathbf{x})$, other than by reducing the sample size; it will not bias the result by introducing some arbitrary values for pixels outside the image. The first-order moment of a truncated sphere, however, does exhibit an edge effect, due to its asymmetrical pixel distribution. We compensate for this by defining a measure, *VSS gradient*, which corrects the first-order moment of intensities to remove the edge effect. Given the center of mass of pixel locations in sphere $S_r(\mathbf{x})$,

$$\mathbf{c}(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})|} \sum_{\mathbf{y} \in S_r(\mathbf{x})} \mathbf{y}, \quad (8)$$

the VSS gradient $\nabla f(\mathbf{x})$ is defined as

$$\nabla f(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})|} [\mathbf{m}(\mathbf{x}) - \mu(\mathbf{x})(\mathbf{c}(\mathbf{x}) - \mathbf{x})]. \quad (9)$$

Note that for nontruncated spheres $\mathbf{c}(\mathbf{x}) = \mathbf{x}$ and VSS gradient is equivalent to the moment vector normalized to the number of pixels, $\mathbf{m}(\mathbf{x})/|S_r(\mathbf{x})|$. For truncated spheres, the VSS gradient does not suffer from the edge effects typical of convolution kernels used to measure gradient (see Figure 4).

All of the above *primary statistics* depend only on the pixels within a given sphere. As a sphere grows or shrinks, its primary statistics can be computed incrementally by adding or removing shells, significantly reducing computational load during sphere map optimization.

5. Secondary Statistics

We define *secondary statistics* as higher order VSS derived by combining multiple spheres to perform more complex analyses of neighborhoods. One such neighborhood that is useful in these computations,

$$S^{-1}(\mathbf{x}) = \{\mathbf{y} : \mathbf{x} \in S_r(\mathbf{y})\}, \quad (10)$$

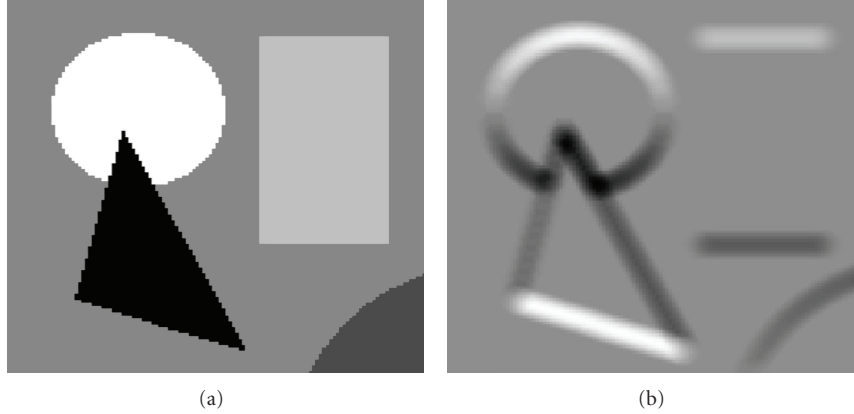


FIGURE 4: (a) Synthetic noiseless image. (b) Corresponding vertical component of the VSS gradient (radius = 3 for all spheres). Note there are no edge effects, such as those seen with conventional gradient calculations.

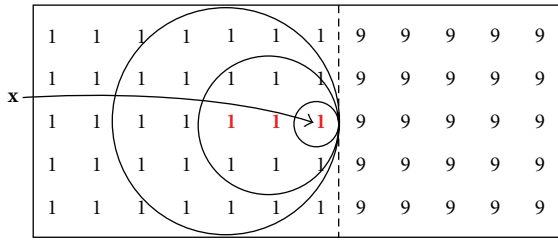


FIGURE 5: The $S^{-1}(\mathbf{x})$ set of spheres that contain pixel \mathbf{x} , adjacent to the boundary between two noiseless regions (same as Figure 2) with respective intensities of 1 and 9.

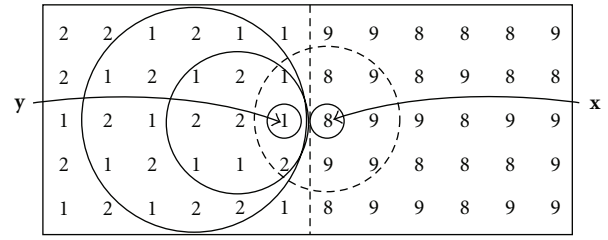


FIGURE 6: Image with noise. Pixel \mathbf{x} is deterred from extending its sphere across the boundary because its mean is an outlier in the population $S^{-1}(\mathbf{y})$.

is the set of all pixels whose spheres contain \mathbf{x} . The -1 superscript in $S^{-1}(\mathbf{x})$ is used to impart the flavor of an inverse function. Since spheres adjust their radii individually, the number of spheres that contain a given pixel varies widely. However, since it is always true that $\mathbf{x} \in S_r(\mathbf{x})$, it is also always true that $\mathbf{x} \in S^{-1}(\mathbf{x})$; every pixel is contained by its own sphere.

Given a correctly optimized sphere map, $S^{-1}(\mathbf{x})$ will consist entirely of pixels from the same object as pixel \mathbf{x} . Figure 5 shows members of one such $S^{-1}(\mathbf{x})$ set, consisting of three pixels (bold), whose spheres each contain \mathbf{x} . Notice that all three spheres touch but do not cross the boundary, so they are correctly optimized. (There would be other spheres as well in a correctly optimized $S^{-1}(\mathbf{x})$.)

Secondary statistics are derived from populations of spheres such as $S^{-1}(\mathbf{x})$. Thus, $\mu_\mu(\mathbf{x})$, the *mean of means* at pixel \mathbf{x} is defined as

$$\mu_\mu(\mathbf{x}) = \frac{1}{|S^{-1}(\mathbf{x})|} \sum_{\mathbf{y} \in S^{-1}(\mathbf{x})} \mu(\mathbf{y}), \quad (11)$$

which is the mean of the mean intensities for all the spheres in $S^{-1}(\mathbf{x})$. In a noiseless image containing distinct homogeneous regions, the correctly optimized sphere map yields values for $\mu_\mu(\mathbf{x})$ identical to the original intensity image $f(\mathbf{x})$. In a noisy image, as will be shown, $\mu_\mu(\mathbf{x})$ yields a an image with reduced noise but with sharp boundaries.

Likewise, $\sigma_\mu(\mathbf{x})$, the standard deviation of the mean intensities for the spheres in $S^{-1}(\mathbf{x})$ is defined as

$$\sigma_\mu(\mathbf{x}) = \left[\frac{1}{|S^{-1}(\mathbf{x})| - 1} \sum_{\mathbf{y} \in S^{-1}(\mathbf{x})} [\mu(\mathbf{y}) - \mu_\mu(\mathbf{x})]^2 \right]^{\frac{1}{2}}, \quad |S^{-1}(\mathbf{x})| > 1. \quad (12)$$

Note that, as with $\sigma(\mathbf{x})$, the above definition of $\sigma_\mu(\mathbf{x})$ is given only for $|S^{-1}(\mathbf{x})| > 1$, since at least two samples are required for variance or standard deviation to be meaningful. For noiseless images, a correct sphere map will yield a value of zero for $\sigma_\mu(\mathbf{x})$ at every pixel.

We can now compute $z_\mu(\mathbf{x} | \mathbf{y})$, the *z-value* for the sphere \mathbf{x} to belong to the set of spheres that already include \mathbf{y} . We define $z_\mu(\mathbf{x} | \mathbf{y})$ as

$$z_\mu(\mathbf{x} | \mathbf{y}) = \frac{|\mu(\mathbf{x}) - \mu_\mu(\mathbf{y})|}{\sigma_\mu(\mathbf{y})}, \quad (13)$$

to provide a measure of how well $S_r(\mathbf{x})$ fits into the current $S^{-1}(\mathbf{y})$ set. The justification is that in an optimized sphere map, if $S_r(\mathbf{x})$ were to justifiably contain pixel \mathbf{y} , then $\mu(\mathbf{x})$ should fall within the distribution of means for all spheres that already contain \mathbf{y} . This concept is illustrated in Figure 6,

which shows pixel \mathbf{x} attempting to extend its sphere across the boundary to include pixel \mathbf{y} . We have included noise in the image to demonstrate that a high z -value could be used to stop the growth of $S_r(\mathbf{x})$ at the boundary, even in the presence of such noise. It should be noted that the utility of this statistic is dependent on a reasonable initialization of the sphere map, such that the percentage of spheres already not crossing boundaries is high enough to lend statistical validity to $\mu_\mu(\mathbf{y})$ and $\sigma_\mu(\mathbf{y})$.

6. Noiseless Sphere Map Optimization Algorithm

Armed with the primary and secondary statistics just described, we are ready to develop algorithms to optimize a sphere map, that is, to set the correct radius of each sphere $S(\mathbf{x})$ in an image such that it reaches, but does not cross, the nearest boundary. We will develop two such algorithms here. The first is designed to optimize a sphere map on a noiseless image, a far simpler task than optimizing a sphere map for a real image.

A noiseless image consists of a set of objects, each containing a set of pixels of uniform intensity. Figure 7(a) shows an example of a synthetically generated noiseless image. Object boundaries can easily be determined in such images by detecting any change in intensity between pixels. Our noiseless algorithm functions simply by growing each sphere $S(\mathbf{x})$ in the image until it contains a pixel \mathbf{y} in its next outer shell, $\mathbf{y} \in H_{r+1}(\mathbf{x})$, with a different intensity value than that of pixel \mathbf{x} , resulting in a nonzero variance $\sigma^2(\mathbf{x})$. This process produces a sphere map containing the correct radii of all spheres within the image, shown in Figure 7(b), whose intensity represents $r(\mathbf{x})$. Notice the bright ridges within and between objects, representing medial ridges, where spheres touch at least two boundary points. The noiseless algorithm is useful for finding medial ridges in cases where an unambiguous segmentation has already been obtained. We have used it, for example, on manual segmentations of the amygdala in the magnetic resonance (MR) images of the human brain, to determine medial parameters of shape for diagnosis of depression [14, 15].

7. Real Image Segmentation Algorithm

Real images contain noise and therefore require greater finesse and complexity in establishing the optimal sphere map. The Shells and Spheres framework lends itself to the design of many different algorithms for this purpose. We present here in detail one such algorithm that was the subject of the masters and doctoral dissertations of [16, 17].

The algorithm takes the form of a six-step process, with Steps 1–4 optimizing the sphere map, Step 5 finding medial pixels, and Step 6 producing a segmentation of the target object. Step 1 uses VSS gradient to detect boundaries and creates an initial approximation of the correct sphere map, from which acceptable statistical values can be obtained for use in subsequent steps. Step 2 utilizes the pronounced discrepancy in pixel variance between spheres that have

incorrectly grown across boundaries and those that have not, to reduce the size of incorrect spheres, placing them correctly within their appropriate image objects. Step 3 introduces specialized boundary indicators, known as *outposts*, designed to stop spheres from crossing into a new image object, using information extracted from population testing between spheres in adjacent objects. These outposts influence the radii of nearby spheres, resizing them to adhere to a consensus on boundary location. Step 4 revisits variance calculation using the current sphere map, which is more accurate than the sphere map previously available. The new variance measure is applied to spheres, encouraging them to fully grow within their respective image objects, effectively smoothing the radius image and sharpening its boundaries (as defined by all spheres of radius 0 or 1). Step 5 identifies medial pixels, that is, those whose spheres touch at least two boundaries. Given a seed point within the target object, Step 6 locates the nearest medial pixel and connects neighboring medial pixels, combining their corresponding spheres to produce a segmentation. The following sections describe each step in detail.

Step 1 (VSS Gradient-Based Radius Approximation). As previously illustrated, it is trivial to optimize the sphere map for a noiseless image by growing spheres until a new intensity value is detected. When analyzing real images, however, this approach will fail, because intensity variation due to noise may be indistinguishable from an object boundary, especially within small spheres. For illustrative purposes, we consider the previous synthetic image with Gaussian noise added, shown in Figure 8.

Steps 2, 3, and 4 depend on there already existing a sphere map that is at least somewhat accurate, because those steps make use of secondary VSS, based on collections of existing spheres. For Step 1 to accomplish this first attempt at an accurate sphere map, all spheres are first set to $r(\mathbf{x}) = 0$ and are then allowed to grow until a persistent increase in VSS gradient magnitude (9) is detected over a continuous range of sphere sizes. Unlike conventional gradient measured with a fixed-scale kernel, the VSS gradient depends locally on $r(\mathbf{x})$ and is based on the first-order moment of intensity normalized by the number of pixels in the sphere. Thus the VSS gradient can be expected to increase monotonically as a sphere grows past a boundary, since the first-order moment favors the outer pixels. A persistent increase in VSS gradient for g consecutive steps is sought, indicating that a boundary has been crossed, whereupon $r(\mathbf{x})$ is reset to the radius just before the increase in VSS gradient began. Empirically, it has been found that a value of $g = 5$ performs well for both MRI and CT cardiac data to achieve a reasonable first approximation of the optimized sphere map. We found Step 1 to be effective at growing spheres past tissue inhomogeneity and noise. However, spheres may not stop exactly on the boundary because of the effect of noise on the detection of consistent gradient increase. Using VSS gradient to govern sphere growth can also fail completely for a sphere that encounters two opposing boundaries simultaneously, as their contributions to the gradient may cancel.

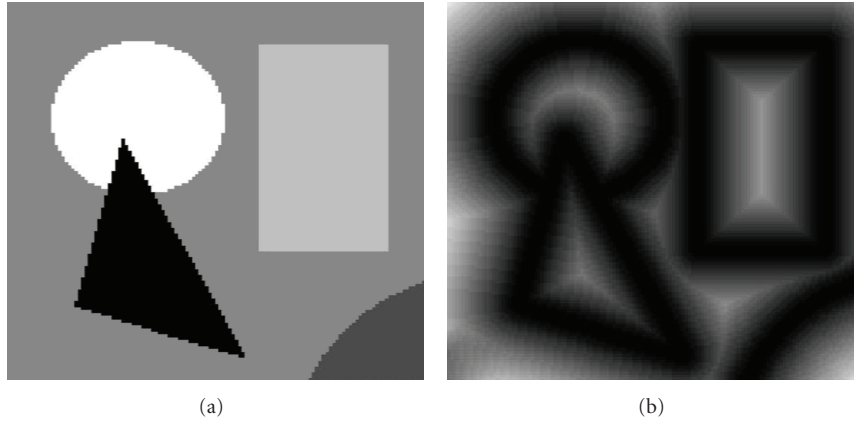


FIGURE 7: (a) An example noiseless image. (b) The sphere map created by the noiseless algorithm, represented as an image where each pixel \mathbf{x} has an intensity value equivalent to the radius of its sphere $S(\mathbf{x})$.

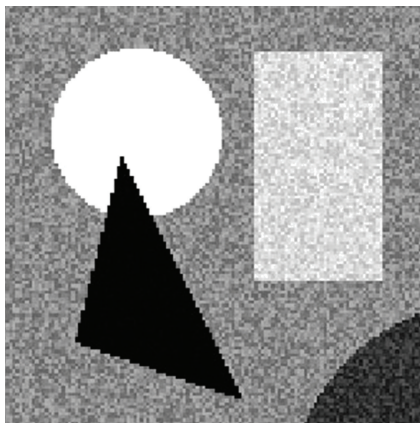


FIGURE 8: Synthetic test image from Figure 7(a) with gaussian noise added.

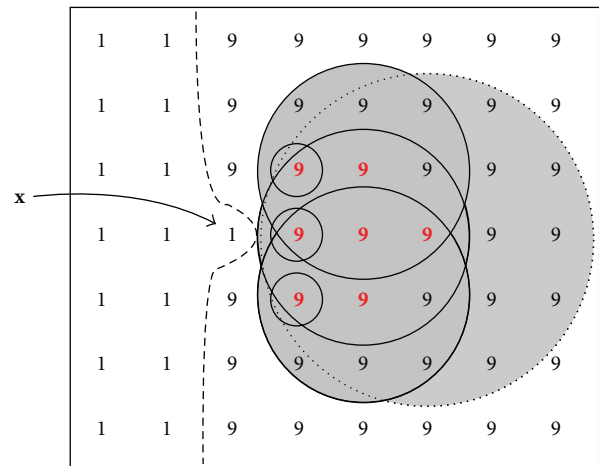


FIGURE 10: Illustration of $K^{-1}(\mathbf{x})$ containing 7 pixels (bold), each of whose spheres would place its reflector across the boundary at \mathbf{x} .

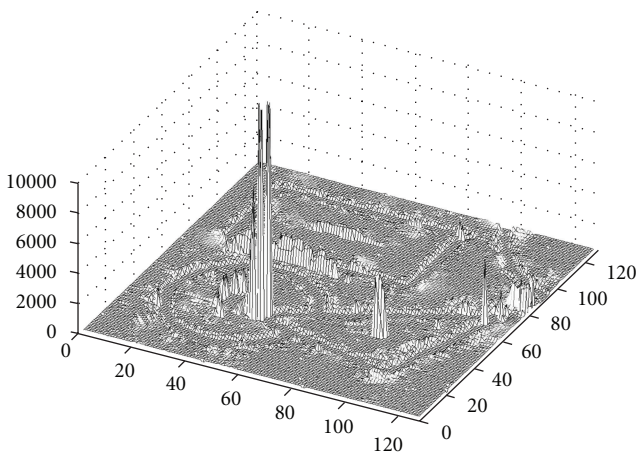


FIGURE 9: A height map of the variance $\sigma^2(\mathbf{x})$ after Step 1 is applied to the image in Figure 8, showing large spikes where spheres encounter two boundaries at once, canceling VSS gradient and growing too large.

Step 2 (Variance-Constrained Radius Reduction). After Step 1, three possible states exist for each sphere: The sphere can be too large, too small, or the correct size (i.e., it touches the nearest boundary but does not cross it). The most glaring error in the sphere map after Step 1 is the presence of large-scale spheres that have incorrectly grown past boundaries. As previously mentioned, this type of error typically occurs when a growing sphere contacts multiple boundaries at once, which indicates that the sphere lies on the medial manifold (the locus of centers of spheres lying within an object that touch at least two boundaries). In such a case, the contributions to the VSS gradient from multiple boundaries may cancel, allowing the sphere to grow much larger than its correct radius. Such spheres, luckily, have variance values that are orders of magnitude higher than those that remain within a single object.

Figure 9 shows a height map of variance values $\sigma^2(\mathbf{x})$ for all spheres after applying Step 1 to the synthetic image with

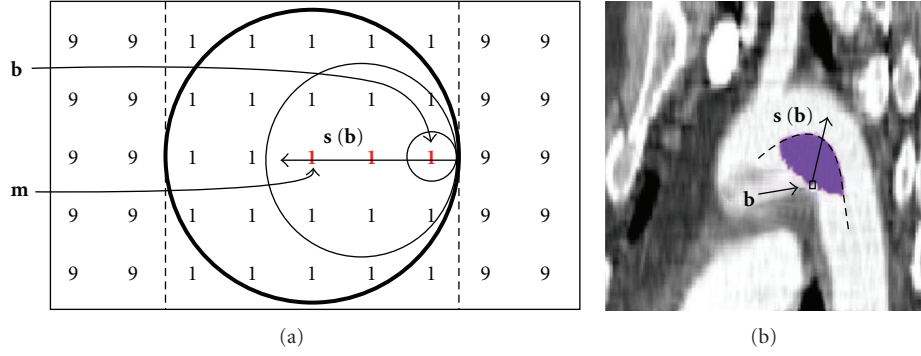


FIGURE 11: (a) Diagram of an object with intensity 1 between two regions of intensity 9, showing set $S^{-1}(\mathbf{b})$ of pixels (bold) whose spheres contain pixel \mathbf{b} . This set produces an $\mathbf{s}(\mathbf{b})$ vector (see text) along which the furthest bold pixel \mathbf{m} is the center of a medial sphere (circle in bold) touching both boundaries (dashed lines). (b) Image of a 2D slice through a CT scan of the aorta with contrast showing an actual $S^{-1}(\mathbf{b})$ set (purple), the resulting $\mathbf{s}(\mathbf{b})$ vector, and the medial manifold (dashed curve) on which the furthest sphere along $\mathbf{s}(\mathbf{b})$ must lie.

added noise shown in Figure 8. The flat “floor” section of the height map is the variance of the uniform noise added to the image, which has a value near 200. Spheres crossing boundaries, however, have a much larger variance than spheres that correctly remain within the object boundaries, extending to a value of nearly 16000. This large dynamic range can be exploited to shrink the spheres that grow too large. All spheres with a variance above a certain threshold α_s are shrunk by decrementing $r(\mathbf{x})$ until $\sigma^2(\mathbf{x}) < \alpha_s$. (The subscript “s” stands for *shrink*.) The threshold α_s is defined as

$$\alpha_s = \mu_{\sigma^2} + \beta_s \sigma_{\sigma^2}, \quad (14)$$

where μ_{σ^2} and σ_{σ^2} are the mean and standard deviation, respectively, of the variance of all the spheres in the current sphere map. The positive constant β_s thus represents the number of standard deviations above the mean permitted for a sphere’s variance without the sphere being required to shrink. The exact value of this parameter is not particularly critical, as the difference between the spheres that grew much too large and the others is very large. We found typical value for β_s to be 0.2, although as will be described below the value was optimized for individual data sets. Reducing the radius of pixels with extremely high variance corrects a majority of the spheres that have incorrectly grown past boundaries. This global threshold for $\sigma^2(\mathbf{x})$ is not ideal, because it assumes a constant expected variance throughout the image. This expectation is likely untrue, given factors such as tissue inhomogeneity and nonuniform noise, and ongoing work is exploring ways around this problem. It should also be noted that $\sigma^2(\mathbf{x})$, and thus the threshold α_s , depend on the current sphere map, $r(\mathbf{x})$, which is not yet fully optimized at this step. This deficiency is addressed by returning to variance in Step 4, once a more accurate $r(\mathbf{x})$ is available.

Step 3 (Reflectance and Outposts). Following Step 2, many spheres are correctly sized and face each other across boundaries. This sets the stage for the use of secondary VSS to differentiate regions on opposite sides of those boundaries. Likely boundary candidates are identified for a given sphere

at location \mathbf{x} by finding a pixel \mathbf{y} in its $H_{r+1}(\mathbf{x})$ shell with a high value for $z_\mu(\mathbf{x} | \mathbf{y})$. As already discussed, and illustrated in Figure 6, such a sphere will detect growth past a boundary by finding itself unlike the $S^{-1}(\mathbf{y})$ population of spheres containing the pixel \mathbf{y} just across the boundary. The sphere at \mathbf{x} is said to place a *reflector* at such a location, a metaphorical construct denoting a vote by the sphere at \mathbf{x} for the pixel \mathbf{y} as being across the nearest boundary. Note that we do not vote for pixel \mathbf{y} as a boundary itself. Instead, pixels on each side of a detected boundary are marked by each other as being across the boundary. Thus, referring again to Figure 6, $S_r(\mathbf{x})$ could place a reflector at pixel \mathbf{y} . The set of reflectors placed by a given sphere $S_r(\mathbf{x})$ is denoted, $K(\mathbf{x})$. In the present algorithm the constraint $|K(\mathbf{x})| = 1$ is applied, limiting each sphere to placing only one reflector, for reasons discussed below. This constraint leads to the definition of $K(\mathbf{x})$ as

$$K(\mathbf{x}) = \left\{ \mathbf{y} : \mathbf{y} = \operatorname{argmax}_{\mathbf{y} \in H_{r+1}(\mathbf{x})} z_\mu(\mathbf{x} | \mathbf{y}) \right\}. \quad (15)$$

If a boundary exists just beyond the outer shell of $S_r(\mathbf{x})$, it will be located at the pixel \mathbf{y} for which the highest $z_\mu(\mathbf{x} | \mathbf{y})$ is calculated. If multiple object boundaries exist just beyond the outer shell of $S_r(\mathbf{x})$, the boundary producing the largest $z_\mu(\mathbf{x} | \mathbf{y})$ will be marked with a reflector. Each pixel may contain reflectors placed by a number of spheres. The set of spheres that have placed reflectors at a pixel \mathbf{x} is defined as

$$K^{-1}(\mathbf{x}) = \{ \mathbf{y} : \mathbf{x} \in K(\mathbf{y}) \}, \quad (16)$$

invoking the same inverse notation used for $S^{-1}(\mathbf{x})$ in (10). A set of spheres placing their reflectors across a boundary at pixel \mathbf{x} is shown in Figure 11. The number of reflectors $|K^{-1}(\mathbf{x})|$ at a given pixel is referred to as the *reflector count*. For example, in Figure 10, the reflector count $|K^{-1}(\mathbf{x})| = 7$. A reflector placed by \mathbf{y} at \mathbf{x} has an inherent direction governed by the vector $(\mathbf{y} - \mathbf{x})$. The vector sum of the directions of all of the reflectors at \mathbf{x} is denoted, the *reflectance* $\mathbf{k}(\mathbf{x})$, defined by

$$\mathbf{k}(\mathbf{x}) = \frac{1}{|K^{-1}(\mathbf{x})|} \sum_{\mathbf{y} \in K^{-1}(\mathbf{x})} \frac{\mathbf{y} - \mathbf{x}}{|\mathbf{y} - \mathbf{x}|}. \quad (17)$$

This measure provides the average orientation of the $K^{-1}(\mathbf{x})$ population, which describes a direction generally normal to the boundary, pointing to the center of the region represented by $K^{-1}(\mathbf{x})$. Since it was decided that each sphere will contribute exactly one reflector, reflector density and reflectance are normalized over the image. Therefore, reflector count can be used to differentiate between significant collections of reflectors correctly placed at boundaries and sparse distributions of reflectors incorrectly placed in the interior of objects. To denote pixels containing a significant number of reflectors, the term *outpost* is adopted, since such pixels serve as border markers and face each other across boundaries much the same way that military outposts of opposing armies face each other across the battle line. The set of all pixels in an image chosen to be outposts is denoted by P . In the present algorithm this set is found in two steps. First, the set of *primary outposts* P' is established, containing all pixels with zero radius and at least κ reflectors, that is,

$$P' = \{\mathbf{x} : |K^{-1}(\mathbf{x})| \geq \kappa, r(\mathbf{x}) = 0\}. \quad (18)$$

For the results presented in this paper, $\kappa = 4$. To increase the density of outposts along the boundaries, a set of *secondary outposts* P'' is generated, containing all pixels with zero radius that adjoin an outpost in P' and have at least λ reflectors, where $\lambda < \kappa$,

$$P'' = \{\mathbf{x} : |K^{-1}(\mathbf{x})| \geq \lambda, H_1(\mathbf{x}) \cap P' \neq \emptyset, r(\mathbf{x}) = 0\}. \quad (19)$$

For the results presented, $\lambda = 2$. By combining the sets of primary and secondary outposts, the set of all outposts,

$$P = P' \cup P'' \quad (20)$$

is formed. Each outpost $\mathbf{y} \in P$ has a reflectance $\mathbf{k}(\mathbf{y})$. A sphere at \mathbf{x} can distinguish whether a given outpost is on its side of the boundary, constituting a *friendly outpost*, or the other side of the boundary, constituting an *enemy outpost*, based on the direction of the outpost's reflectance. The set of enemy outposts (those with reflectance facing \mathbf{x}) within the sphere of radius $r(\mathbf{x})$ is defined as

$$E_r(\mathbf{x}) = \{\mathbf{y} : \mathbf{y} \in P \cap S_r(\mathbf{x}), \mathbf{k}(\mathbf{y}) \cdot (\mathbf{y} - \mathbf{x}) < 0\}, \quad (21)$$

where the sign of the dot product determines the direction of $\mathbf{k}(\mathbf{y})$ relative to \mathbf{x} . In governing the growth of a sphere, enemy outposts are to be avoided, while friendly outposts can be included. More specifically, enemy outposts should stop the growth of spheres, as they represent a different image region than the one in which the sphere resides, while friendly outposts do not. Step 3 uses the number of enemy outposts to adjust the sphere size as follows: If the pixel contains no enemy outposts in its next shell out, $H_{r+1}(\mathbf{x})$, the sphere grows until it does. That is,

$$\text{if } |E_{r+1}(\mathbf{x})| = 0, \text{ increase } r(\mathbf{x}) \text{ until } |E_{r+1}(\mathbf{x})| > 0. \quad (22)$$

If the number of enemy outposts included in the set $S_r(\mathbf{x})$ is greater than γ , the radius is decreased until this is no longer true, that is,

$$\text{if } |E_{r+1}(\mathbf{x})| > \gamma, \text{ reduce } r(\mathbf{x}) \text{ until } |E_r(\mathbf{x})| \leq \gamma. \quad (23)$$

In the present implementation, $\gamma = 2$. This value prevents lone pixels that have been improperly labeled as outposts from incorrectly causing spheres to shrink. After Step 2, in which incorrectly large spheres have been adjusted to a more correct size, the most pressing problem with the sphere map is the scattered effects of noise on $r(\mathbf{x})$. Step 3 focuses on spheres that have incorrectly stopped growth at image noise, or grown slightly too large across their nearest boundary. The effect of these outpost-driven operations is that significant densities of reflectors placed by correctly-sized spheres along boundaries are used to govern the size of other spheres, sweeping incorrect reflectors from the within objects to the boundaries (since a sphere's reflector is redistributed when its radius is altered). Since each sphere places only one reflector, some pixels along object boundaries may remain unmarked, leading to a somewhat sparse collection of outposts along boundaries. This will not adversely effect the evolution of the sphere map, however, due to another advantage of our spherical operator design. Because most spheres are large relative to the spacing of outposts along the boundary, their growth will be stopped and they tend not to "leak" or "bleed" across boundaries, as some conventional deformable contours are prone to do.

Step 4 (Variance-Constrained Scale Growth). At this point in the analysis, our sphere map has achieved a configuration generally representative of the shapes within the image, but it still retains adverse effects from noise and suboptimal boundary detection. Although Step 3 results in a reasonably accurate $r(\mathbf{x})$, some spheres still may not quite reach boundaries, due to pixels being incorrectly labeled as outposts. These false outposts will stop spheres in the interior of image objects, leading to potential errors in segmentation. As an added measure to force spheres to grow maximally within their objects, we return to variance a second time, calculated in the same manner as in Step 2, but used to grown now rather than to shrink. A second global variance threshold for variance, α_g , similar to α_s described above, is calculated as

$$\alpha_g = \mu_{\sigma^2} + \beta_g \sigma_{\sigma^2}, \quad (24)$$

where μ_{σ^2} and σ_{σ^2} again are the mean and standard deviation, respectively, of the variance throughout the image (the subscript "g" stands for *grow*). The threshold α_g is used to smooth the boundaries in the radius image by forcing spheres to grow up to the actual boundary using the current, more accurate variance $\sigma^2(\mathbf{x})$. The value of $r(\mathbf{x})$ is incremented for all spheres while their internal variance $\sigma^2(\mathbf{x}) < \alpha_g$. This creates a sphere map that more accurately matches the contours of objects in the image. At this point the sphere map is considered optimized.

Step 5 (Medial Pixel Identification). Given the optimized sphere map $r(\mathbf{x})$, the next goal is to extract medial pixels. To facilitate this, a dense set of boundary pixels B is first defined, as those whose spheres have radius 0 or 1,

$$B = \{\mathbf{x} : r(\mathbf{x}) \leq 1\}. \quad (25)$$

The sets $S^{-1}(\mathbf{b})$ for all boundary pixels $\mathbf{b} \in B$ can be used to find pixels on the medial manifold, whose spheres touch

two boundaries while still lying completely within the object. Recall that the $S^{-1}(\mathbf{b})$ set for pixel \mathbf{b} contains all spheres in the sphere map that themselves contain pixel \mathbf{b} (10). Given a correct sphere map, this set will necessarily contain at least one sphere that touches both the boundary that pixel \mathbf{b} borders as well as an opposing boundary across the object region (and also across the sphere) from pixel \mathbf{b} . Figure 11(a) shows such a medial pixel (labeled “ m ”) on the medial manifold of an object of intensity 1, between two regions of intensity 9. To find such medial pixels within $S^{-1}(\mathbf{b})$, we first define an orientation $\mathbf{s}(\mathbf{b})$, roughly orthogonal to the boundary, as the vector sum of the normalized offsets relative to \mathbf{b} for pixels within $S^{-1}(\mathbf{b})$ as

$$\mathbf{s}(\mathbf{b}) = \frac{1}{|S^{-1}(\mathbf{b})|} \sum_{\mathbf{y} \in S^{-1}(\mathbf{b})} \frac{\mathbf{y} - \mathbf{b}}{|\mathbf{y} - \mathbf{b}|}. \quad (26)$$

For each boundary pixel $\mathbf{b} \in B$, the pixel $\mathbf{m} \in S^{-1}(\mathbf{b})$ that is furthest from the boundary along $\mathbf{s}(\mathbf{b})$ is identified as a medial pixel, as depicted in Figure 12(a). The set of all medial pixels M in the image is thus

$$M = \left\{ \mathbf{m} : \mathbf{m} = \underset{\mathbf{y} \in S^{-1}(\mathbf{b})}{\operatorname{argmax}}((\mathbf{y} - \mathbf{b}) \cdot \mathbf{s}(\mathbf{b})), \mathbf{b} \in B \right\}. \quad (27)$$

Figure 11(b) shows an actual $S^{-1}(\mathbf{b})$ set for a pixel \mathbf{b} on the boundary of the aorta in a 2D slice through a computed tomography (CT) scan with contrast. The furthest pixel along vector $\mathbf{s}(\mathbf{b})$ lies on that medial manifold (dashed line). Selecting a single pixel from each $S^{-1}(\mathbf{b})$ set overlooks a potentially large number of additional medial pixels on the outer edge of each set, especially for a concave boundary point such as shown in Figure 12. One can, however, be certain that each $S^{-1}(\mathbf{b})$ set contains a minimum of one medial pixel, as the center of the largest sphere in the direction roughly orthogonal to the boundary. The set M derived taking advantage of this fact is a sparse but reliable set of pixels on the various medial manifolds within the image.

Step 6 (Medial Flood-Fill Segmentation). To segment a particular object, a seed pixel $\mathbf{p} \in M$ on that object’s medial manifold is needed. To find it, a sample pixel is manually selected by the user, and a search is conducted for the closest medial pixel by iterating through successive shells moving radially outward from the selected pixel. The first medial pixel encountered is accepted as \mathbf{p} . A flood fill operator is then used to find a connected subset $C \subseteq M$. Pixels belonging in C are found iteratively using a series of sets C_i starting with C_0 , a set containing just the seed pixel \mathbf{p} . At each subsequent step $i+1$, the set C_{i+1} is created by adding medial pixels within a radius m of pixels already in set C_i . More precisely, C_i is defined recursively as

$$\begin{aligned} C_0 &= \{\mathbf{p}\}, \\ C_{i+1} &= \{\mathbf{x} : \mathbf{x} \in M, S_m(\mathbf{x}) \cap C_i \neq \emptyset\}. \end{aligned} \quad (28)$$

For the results presented, radius m was dynamically set to $m(\mathbf{x}) = r(\mathbf{x})/2$ as this causes the algorithm to search halfway

from the medial manifold to the boundary for new medial pixels to include, therefore staying within the designated object. When a final step f adds no new pixels, such that

$$C_f = C_{f-1}, \quad (29)$$

the flood-fill is complete, as the set of connected medial pixels within the object is the current pixel collection, or

$$C = C_f. \quad (30)$$

The union of the set of spheres centered at these medial pixels effectively segments the object by including all of the pixels designated as within the object. These spheres, centered on the medial manifold, extend to all points on the boundary.

8. Results

The present paper is the first disclosure in a journal article of the detailed mathematics of Shells and Spheres. Validation of the particular algorithm just described has already been published [18], so we only summarize those results briefly here. Figure 12 shows a segmentation of the aorta on a 2D slice through a thoracic CT scan with contrast. On the left, the raw CT data is shown to have considerable noise. On the right, the mean of means image $\mu_\mu(\mathbf{x})$ exhibits greatly reduced noise while maintaining sharp boundaries and significant detail. The aorta has been segmented (purple) by initializing a flood-fill operation with a single-seed point near its medial manifold.

Figure 13(a) shows a 3D segmentation of the aorta using the same algorithm (which is inherently n -dimensional). The top portion shows the raw CT data and the bottom shows a surface rendering of the segmented aorta as the union of all the medial spheres.

Figure 13(b) shows the results of segmenting the heart in 3D magnetic resonance (MR) data. In this case, we performed parameter optimization using a single manual tracing on a 2D slice selected from a 3D image data set, to find optimum values for β_s and β_g , used to compute the thresholds for variance-based shrinking and growing in Steps 2 and 4 of the algorithm, respectively. Otherwise, the algorithm is completely automatic.

To test the accuracy of the 3D segmentation, a validation study was conducted to compare it to three manual segmentations. Each segmentation was produced by a different user. Figure 14 shows the Dice Similarity Coefficient (DSC) values for our automated segmentation compared to the manual segmentations, as well as the DSC values for the manual segmentations compared to each other. The DSC produces the value 1 for identical segmentation and 0 for segmentations that do not overlap at all. It can be seen that our automated segmentation matches the manual segmentations with a DSC between 0.83 and 0.86. It should be noted that a DSC of 0.70 is considered excellent agreement in the literature [19], although the definition of sufficient accuracy is, of course, specific to the application. While the manual segmentations produced slightly higher agreement with each other than with the automated segmentation, it is believed that a significant portion of this discrepancy

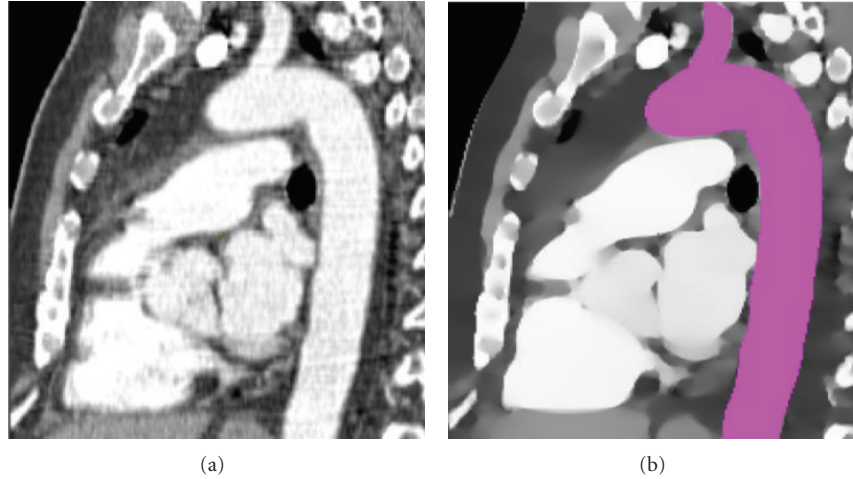


FIGURE 12: (a) Two-dimensional sagittal slice through computer tomography (CT) data of thorax with vascular contrast used in preliminary testing of our algorithm. (b) Mean of means $\mu_\mu(\mathbf{x})$ image showing reduced noise and sharp boundaries, automated segmentation of aorta highlighted (purple) created from a single-seed point.

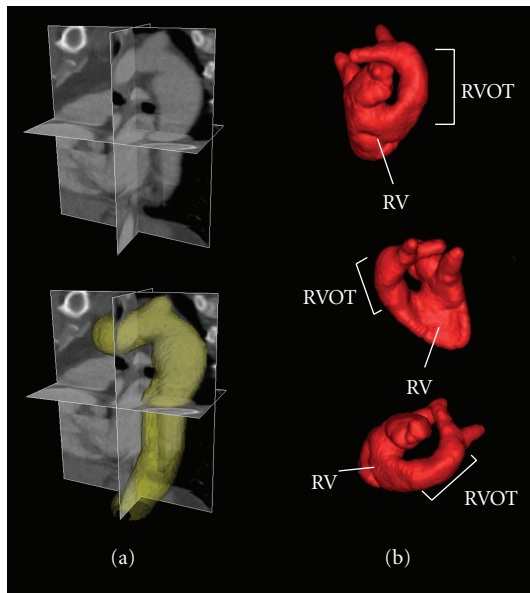


FIGURE 13: (a) 3D contrast-enhanced CT scan of the aorta (top left), and the same scan with an overlaid 3D segmentation (bottom left) achieved using the Shells and Spheres framework. (b) Surface rendering of a Shells and Spheres segmentation of the right heart with labeled Right Ventricle (RV) and Right Ventricular Outflow Tract (RVOT), shown from three different perspectives.

is due to the difficulty for the algorithm in defining the extent of the “right heart” along the continuum of the circulatory system, rather than the boundaries of the vessels themselves. Subject 3 elected to include less of the branching vasculature connected to the main cardiac structures, which led to greater agreement with the automated segmentation and less agreement with the other manual segmentations. Despite the variation in manual segmentations, our system still demonstrated reliable segmentation results.

	Subject 1	Subject 2	Subject 3
S&S	0.83	0.84	0.86
Subject 1	*	0.91	0.88
Subject 2	*	*	0.89

FIGURE 14: Table of DSC values comparing segmentations produced by 3 independent subjects and our Shells and Spheres algorithm.

Comparison with other segmentation techniques is always desirable, but problematic in view of the enormous number of methods that have been developed. We compared our method against two other techniques, an active contour method and intensity thresholding. We present those results in the following section.

Segmentations of the RVOT in our 3D ovine MRI data sets were produced with a widely used geodesic active contour method implemented in the Insight Toolkit ITK-SNAP software package [20]. As before, the DSC was used to show agreement to the manual 3D segmentations of the RVOT produced by our three experts. Results of comparing the active contour method to our Shells and Spheres method can be seen in Figure 15.

This graph shows the mean segmentation agreement to all expert segmentations over ten ovine MRI data sets for the active contour method and our Shells and Spheres method. The error bars were determined by the Standard Error of the Mean (SEM).

We can see that the Shells and Spheres segmentation system performed slightly better than the active contour method, but the high degree of overlap of the respective SEM values indicates roughly equivalent performance. An independent-samples t -test showed that the two means were not significantly different (P -value = .741). We conclude that Shells and Spheres can match this current clinical state of the art-automated segmentation method. Furthermore,

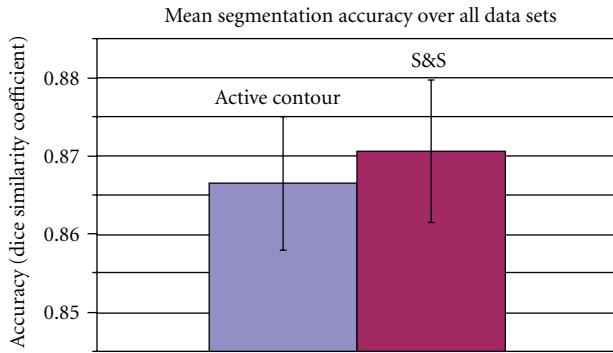


FIGURE 15: Bar graph showing the mean 3D segmentation agreement to all expert segmentations over all data sets for both ITK-SNAP active contour method (blue) and the Shells and Spheres method (maroon). Error bars show the standard error of the mean (SEM).

the minimal manual input required by the Shells and Spheres algorithm at the onset of analysis (tracing a single 2D slice for parameter optimization) represents considerable less time and effort on the part of the human operator than the continual supervision necessary for the active contour method. Additionally, our system is designed to require only skills and expertise inherent to the clinical professional, rather than expecting a medical professional to gain algorithmic or mathematical expertise to effectively perform the active contour segmentation.

Intensity thresholding coupled with a flood-fill from a manually placed seed point was also explored as a common technique for comparison to our segmentation system, but the prevalence of partial-volume effects and tissue inhomogeneity in MRI images made this method incapable of segmenting the RVOT in our data sets, due to bleeding of the floodfill regardless of threshold parameters. Without a high degree of manual postprocessing, this method produced a failed segmentation ($DSC < 0.70$ for all expert segmentations) on each of our MRI data sets.

9. Discussion

The Shells and Spheres framework for image analysis and the associated n -dimensional algorithms described here represent a novel system to facilitate image segmentation. Advantages include preservation of sharp boundaries while including large populations of pixels from both sides of the boundaries for statistical analysis. The primary statistics exhibit no edge effect and can be efficiently computed by adding and subtracting shells while optimizing the sphere map. Because the framework is truly n -dimensional, volumetric segmentation occurs in 3D, not slice by slice, lending a natural anatomical appearance to the visualized surface. Finally, the algorithm presented yields useful medial features for further analysis, which is highly relevant to understanding anatomical shape.

We have presented just two example algorithms. Many others are possible. One that is currently under development

introduces a new secondary statistic, a variation on the Student's t -test, incorporating mean and variance in a way that further sharpens boundary detection. Subsequent determination of the medial manifold uses the divergence of the unit direction to the nearest boundary [21], an adaptation of concepts recently developed by Dimitrov et al. [22] which are particularly robust in the presence of noise and efficient to compute when incorporated into the Shells and Spheres framework.

Acknowledgments

Support for this research came from NIH Grants R21EB 007721 and RO1HL089750, and a contract from the National Library of Medicine for development of the Insight Toolkit (ITK). The authors thank Bahar Fata, Danielle Gottlieb, and John Mayer for their assistance in supplying data for the validation of their algorithms.

References

- [1] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, 1980.
- [2] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [3] J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge Monographs on Applied and Computational Mathematics, no. 3, Cambridge University Press, Cambridge, UK, 1996.
- [4] T. Lindeberg, *Scale-Space Theory in Computer Vision*, Monograph, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.
- [5] H. Blum and R. N. Nagel, "Shape description using weighted symmetric axis features," *Pattern Recognition*, vol. 10, no. 3, pp. 167–180, 1978.
- [6] L. R. Nackman, "Curvature relations in three-dimensional symmetric axes," *Computer Graphics and Image Processing*, vol. 20, no. 1, pp. 43–57, 1982.
- [7] L. R. Nackman and S. M. Pizer, "Three-dimensional shape description using the symmetric axis transform I: theory," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 2, pp. 187–202, 1985.
- [8] C. A. Burbeck and S. M. Pizer, "Object representation by cores: identifying and representing primitive spatial regions," *Vision Research*, vol. 35, no. 13, pp. 1917–1930, 1995.
- [9] S. M. Pizer, D. Eberly, B. S. Morse, and D. S. Fritsch, "Zoom invariant vision of figural shape: the mathematics of cores," *Computer Vision and Image Understanding*, vol. 69, no. 1, pp. 55–71, 1998.
- [10] B. S. Morse, S. M. Pizer, D. T. Puff, and C. Gu, "Zoom-Invariant vision of figural shape: effect on cores of image disturbances," *Computer Vision and Image Understanding*, vol. 69, no. 1, pp. 72–86, 1998.
- [11] S. M. Pizer, P. T. Fletcher, S. Joshi, et al., "A method and software for segmentation of anatomic object ensembles by deformable m-reps," *Medical Physics*, vol. 32, no. 5, pp. 1335–1345, 2005.

- [12] G. D. Stetten and S. M. Pizer, "Medial-node models to identify and measure objects in real-time 3-D echocardiography," *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 1025–1034, 1999.
- [13] K. Siddiqi and S. Pizer, *Medial Representations: Mathematics, Algorithms, and Applications*, Springer, London, UK, 2008.
- [14] R. J. Tamburo, G. J. Siegle, G. D. Stetten, et al., "Localizing amygdala structure differences in late-life depression," in *Proceedings of the 4th IEEE International Symposium on Biomedical Imaging (ISBI '07)*, pp. 340–343, Washington, DC, USA, April 2007.
- [15] R. J. Tamburo, G. J. Siegle, G. D. Stetten, et al., "Amygdalae morphometry in late-life depression," *International Journal of Geriatric Psychiatry*, vol. 24, no. 8, pp. 837–846, 2009.
- [16] C. A. Cois, *Shells and spheres: a novel framework for variable-scale statistical image analysis*, M.S. thesis, Department of Bioengineering, University of Pittsburgh, Pittsburgh, Pa, USA, 2006.
- [17] C. A. Cois, *Variable scale statistics for cardiac segmentation and shape analysis*, Ph.D. dissertation, Department of Bioengineering, University of Pittsburgh, Pittsburgh, Pa, USA, 2007.
- [18] C. Aaron Cois, K. Rockot, J. Galeotti, et al., "Automated segmentation of the right heart using an optimized Shells and Spheres algorithm," in *Proceedings of the 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI '07)*, pp. 876–879, Washington, DC, USA, April 2007.
- [19] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric analysis of white matter lesions in MR images: method and validation," *IEEE Transactions on Medical Imaging*, vol. 13, no. 4, pp. 716–724, 1994.
- [20] P. A. Yushkevich, J. Piven, H. C. Hazlett, et al., "User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability," *NeuroImage*, vol. 31, no. 3, pp. 1116–1128, 2006.
- [21] G. Stetten, S. Horvath, J. Galeotti, G. Shukla, and B. Chapman, "Image segmentation using the student's t-test on adjacent spherical populations of pixels," *SPIE Medical Imaging*, paper #7623-125, February 2010.
- [22] P. Dimitrov, J. N. Damon, and K. Siddiqi, "Flux invariants for shape," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 835–841, Madison, Wis, USA, June 2003.