Robust Centerline Extraction from Tubular Structures in Medical Images

Jianfei Liu Kalpathi Subramanian

Charlotte Visualization Center
Department of Computer Science
The University of North Carolina at Charlotte
Charlotte, North Carolina, USA

ABSTRACT

Extraction of centerlines is useful to analyzing objects in medical images, such as lung, bronchia, blood vessels, and colon. Given the noise and other imaging artifacts that are present in medical images, it is crucial to use robust algorithms that are (1) noise tolerant, (2) computationally efficient, (3) accurate and (4) preferably, do not require an accurate segmentation and can directly operate on grayscale data. We propose a new centerline extraction method that employs a Gaussian type probability model to build a more robust distance field. The model is computed using an integration of the image gradient field, in order to estimate boundaries of interest. Probabilities assigned to boundary voxels are then used to compute a modified distance field. Standard distance field algorithms are then applied to extract the centerline. We illustrate the accuracy and robustness of our algorithm on a synthetically generated example volume and a radiologist supervised segmented head MRT angiography dataset with significant amounts of Gaussian noise, as well as on three publicly available medical volume datasets. Comparison to traditional distance field algorithms is also presented.

Keywords: virtual endoscopy, centerline, noise, medical imaging, skeleton, probability model

1. INTRODUCTION

The extraction of centerlines is useful to routine medical image analysis tasks, such as navigating the interiors of colon, blood vessels, lungs and other tubular structures. Centerlines are a special case of medial surfaces (or skeletons) that have been studied extensively. Review articles on these and their related algorithms have recently appeared, that have precise definitions and requirements of these structures for a variety of applications that span navigation, image/volume registration, animation, morphing, recognition and retrieval.

Our focus in this work is toward robust centerline extraction from medical images and volumes, specifically within noisy environments. The approach we present does not require an accurate segmentation of the object; we estimate the boundary probabilistically using an integration of the image gradient field. The computed probability field is then used to build a more robust distance field, after which we extract the object centerline using existing distance field based algorithms. We demonstrate the power and usefulness of our model by testing it on a set of synthetically generated volumes, as well as on publicly available medical imaging datasets. These experiments were done using significant amounts of gaussian noise added to the datasets. Accuracy is quantified using the synthetic model as well as the segmented medical dataset, and for comparison to traditional distance field methods. We have successfully tested this approach on medical imaging datasets from blood vessel geometry in the brain, and a CT dataset of a human colon. We also have used interactive tools to qualitatively verify the accuracy of our centerline in the medical datasets, in the absence of an accurate segmentation.

Further author information: (Send correspondence to Kalpathi Subramanian, krs@uncc.edu)
Jianfei Liu: E-mail: jliu1@uncc.edu, Telephone: 1 704 687 8641
Kalpathi Subramanian: E-mail: krs@uncc.edu, Web: www.cs.uncc.edu/~krs; Telephone: 1 704 687 8579
We will begin with a look at centerline extraction methods that are directly relevant to the work presented here, specifically those based on distance fields and image characteristics, and briefly mention other methods. We will then develop our probability model and present preliminary results.

**Distance Field Methods.** These methods use a distance function, which is a signed function from each data point, and most often, referring to the distance-to-closest surface (or distance from boundary, DFB). Such distant maps have been used to accurately represent binary (or segmented) volumes to control aliasing artifacts, extract skeletons. In centerline extraction algorithms, an additional distance, distance-from-source, DFS, which represents the distance from a source point has also been employed. Various distance metrics have been used in these algorithms, such as 1-2-3 metric, 3-4-5 chamfer metric, or 10-14-17. Exact voxel distances \((1 - \sqrt{2} - \sqrt{3})\), assuming unit cube voxels) have also been used.

A number of researchers have used a combination of distance fields and Dijkstra’s algorithms (shortest path, minimum spanning tree) in order to extract the object centerline; the primary idea in these schemes is to transform the object voxels (identified in a preprocessing step) into a weighted graph, with the weights being defined by the inverse of the computed distance metric. Then Dijkstra’s algorithm is applied to find the shortest path between specified end points. Chen et al. used this approach but modify the shortest path voxels to the maximal DFB voxels orthogonal to the path, while Zhou chooses among voxel clusters with the same DFS distance. Bitter et al. use a heuristic that combines the DFS and the DFB distances, with the latter being considered a penalty aimed at discouraging the “hugging corner” problem, that is typical of shortest path based approaches. Finally, Wan et al. propose a method that also uses both DFS and DFB distances, but emphasizes the latter to keep the centerline close to the center of the tubular structure. They also use a priority heap which always keeps the voxels close to the center at the top of the heap.

There are two strengths to distance field based methods, (1) outside of the distance field calculation, centerline extraction algorithm is itself quite efficient, and, (2) the centerline is guaranteed to be inside the structure. However, all of these methods begin with a binary image, and for medical images, this means an accurately segmented image. This, in of itself, is a significant task, given that the original images can be considerably noisy (depending on their modality) and of poor contrast, and the presence of interfering organs can make this task even harder.

**Image Characteristics.** Methods in this category have been used in analyzing tubular structures in medical images, in particular, blood vessel geometry. They are based on two properties of images, (1) use of second order derivatives, and (2) multi-scale analysis. Second order structure of an image is defined by the Hessian matrix, for instance, from a Taylor series expansion about a point \(x_0\),

\[
I(x_0 + \delta x_0, \sigma) \approx I(x_0, \sigma) + \delta x_0^T \Delta_{0,\sigma} + \delta x_0^T H_{0,\sigma} \delta x_0
\]

where \(\Delta_{0,\sigma}\) and \(H_{0,\sigma}\) are the gradient and Hessian of the image at \(x_0\) at a scale \(s\). Secondly, scale space theory relates scale to derivatives, which can be defined as convolution with derivatives of Gaussians:

\[
\frac{\partial}{\partial x} I(x, s) = \sigma^\gamma I(x) \ast \frac{\partial}{\partial x} G(x, \sigma)
\]

where \(G\) is a Gaussian with zero mean and deviation of \(\sigma\). Parameter \(\gamma\), introduced by Lindeberg helps define normalized derivatives and provides the intuition for the use of scale in analyzing image structures.

In the work of Frangi et al., the Hessian is used in detecting blood vessels from angiographic images. Eigen vectors of the Hessian matrix are used to determine the principal directions of the vessel structure; in particular, the direction with the smallest eigen (absolute) value points along the vessel axis, while the remaining two (orthogonal) direction vectors are along the vessel cross-section. This forms the basis for vessel detection, which when combined with scale, can handle vessels of varying cross-section, given the results of Eq. 2, that relate scale to boundary position.

Aylward formulated these ideas in proposing a centerline extraction method for blood vessel structures. Their approach was to identify and track ridges within angiographic images. Their method uses dynamic scale
enhancements to handle changes in vessel geometry, as well as perform well in the presence of noise. Wink et al.\textsuperscript{15} also use a multi-scale representation, however, they convert their multiscale “centeredness” measure to a cost (by choosing the largest response across a range of scales) and extract the centerline by computing the minimum cost path using Dijkstra’s algorithm. Potential to cope with severe stenoses was illustrated. Finally, ridge analysis in images has also been studied in detail by Eberly et al.,\textsuperscript{16,17} and tubular structure detection.\textsuperscript{18}

Computing second order derivatives followed by eigen value analysis can be expensive, especially for very large medical objects. Nevertheless, secondary structure properties provide useful information for image analysis and we are looking into approaches to minimize computation and make these techniques more scalable.

Other Methods. A number of other methods have been proposed, including those based on field functions to extract skeletons. Examples of these include the use of potential functions\textsuperscript{19} and more recently, using topological characteristics derived from repulsive force fields.\textsuperscript{20} Radial basis functions\textsuperscript{21} have also been used. Some of these methods work in continuous space, which can potentially move the centerline out of the object, however, they are more flexible, smoother and are less sensitive to noise due to averaging effects. Our proposed method also takes advantage of this property, as we integrate over a smooth gradient field of the image. Another class of algorithms is based on thinning;\textsuperscript{22} in general, these algorithms are quite expensive, but they are indeed quite robust.

2. METHODS

2.1. Volume Preprocessing

The input volume is first roughly segmented into object voxels and background voxels. In our experiments, we have used thresholding or region growing to isolate the object of interest. However, other more sophisticated operators might be necessary for complex datasets, several examples of which can be found in the Tuebingen archive,\textsuperscript{23} and which we have also used to test our methodology.

2.2. Boundary Model

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure1.png}
\caption{(a) Step edge: ideal boundary, (b) Change in gradient magnitude (approximated by a Gaussian), (c) integral of gradient: blurred edge (also error function)}
\end{figure}

Medical images, by their nature of acquisition and reconstruction are bandlimited; thus, boundaries separating medical structures can be assumed to be blurred by a Gaussian. Fig. 1 (reproduced from\textsuperscript{24}) illustrates a step edge and its blurring by a Gaussian. While Kindlmann\textsuperscript{24} used this model to build transfer functions, our goal here is to define a probability function across the object boundary. Specifically, we define the derivative of the image intensity function $f(x)$, or the gradient, as

$$f'(x) = \frac{K}{\sqrt{2\pi \sigma}} e^{-\frac{x^2}{2\sigma^2}}$$  \hspace{1cm} (3)

where $f'(x)$ is centered around the point $x$, $K$ is a normalizing constant and $\sigma$ represents the deviation. Integrating Eq. 3 results in the familiar blurred boundary, as shown in Fig. 1c.
2.3. Normalizing the Boundary Probability Model

Our next step is to estimate the constant $K$, in order to determine the probability function for voxels close to the boundary. We first evaluate Eq. 3 at $x = 0$ and $x = \pm \sigma$,

$$f'(0) = \frac{K}{\sqrt{2\pi\sigma}} \tag{4}$$

$$f'(-\sigma) = f'(-\sigma) = \frac{K}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}} \tag{5}$$

Thus

$$\frac{f'(\sigma)}{f'(0)} = \frac{f'(-\sigma)}{f'(0)} = e^{-\frac{1}{2}} \tag{6}$$

Consider Fig. 1b. $f'(0)$ occurs when the gradient magnitude attains a maximum, with $(-\sigma, \sigma)$ on either side of it. We use the following procedure to estimate $K$ with respect to each boundary voxel:

1. Starting from each boundary voxel, determine the tracking direction (along the gradient direction, $\vec{g}$ or $\vec{-g}$) that leads to the local maximum; increasing gradient magnitude leads toward the boundary and decreasing magnitude leads away from it.

2. Determine the local maxima of the gradient magnitude by moving along the gradient direction, $\vec{g}$ or $\vec{-g}$.

3. Beginning from position $x = 0$, move along $\vec{g}$ or $\vec{-g}$ to determine $-\sigma$ and $\sigma$ respectively. By using Eqn. 6, we can stop when the ratio reaches approximately $e^{-1/2}$.

4. We know that

$$\int_{-\infty}^{0} f'(x)dx = \int_{0}^{\infty} f'(x)dx = \frac{K}{2} \tag{7}$$

which is approximately

$$\int_{-\sigma}^{0} f'(x)dx = \int_{0}^{\sigma} f'(x)dx = \frac{K}{2} \tag{8}$$

as we are using $K$ to make the area under the Gaussian equal to 1 (in order to convert it into a probability density function). The above equation thus gives us two possible estimates for $K$, denoted $K_1, K_2$. Due to the fact that we are operating in a discrete lattice and the approximations involved in the boundary model, we cannot expect a perfectly symmetric Gaussian shaped variation of gradient across the object boundary. In other words, the points at which $-\sigma$ and $\sigma$ are calculated will usually be at differing distances from $x = 0$. Choose $K = MIN(K_1, K_2)$.

5. There are 5 possible cases:

- $K_1 < K_2$: This is illustrated in Fig. 2a. The shaded area (integral of the gradient) on the left is smaller, which is directly proportional to the estimate of $K_1$

- $K_1 > K_2$: Similarly, as shown in Fig. 2b, the shaded area on the left is larger.

- $K_1$ cannot be determined (Case 3) and $K_2$ cannot be determined (Case 4): These two cases can happen, if there are interfering structures that prevent calculating one of the estimates (detected by a sudden increase in gradient integral). In this case, we choose the computed estimate, $K_1$ or $K_2$.

- Neither $K_1$ nor $K_2$ can be estimated: This is rare, as in this case, the boundary is poorly defined and the presegmentation has performed a poor job of obtaining the rough object boundary.
2.4. Probability Assignment for Near-Boundary Points

Once the normalizing constant $K$ has been determined, our next step is to assign probability values to voxels close to the boundary. For this, we need to determine a starting point prior to computing probabilities. The probability is the integral of the gradient (area under the Gaussian) divided (normalized) by $K$. The probability will be 0.5 at the peak ($x = 0$) and decrease or increase on either side of estimated boundary (toward the background/object respectively). Assume that the voxel positions corresponding to $-\sigma, \sigma$ are respectively $x_1, x_2$. We again need to treat each of the five cases above:

- **$K_1 < K_2$:** In this case, we choose $K = K_1$, and the starting point is $x = x_1$, corresponding to $-\sigma$, as shown in Fig. 2a. Probability $P(x_1) = 0.0$, and we move along $\vec{g}$ or $-\vec{g}$ toward the object boundary (increasing gradient magnitude), where $P(0) = 0.5$. At each step, the probability is computed and assigned to the corresponding voxel. Process ends when the probability reaches 1.

- **$K_1 > K_2$:** In this case (Fig. 2b), $K = K_2$ and the starting point is $x = x_2$ corresponding to $\sigma$, with $P(x_2) = 1.0$. In this case, we move toward $x_1$. However in this case, the integrals are decreased (as the probability is decreasing) at each step. The process terminates at a point $x_1'$ such that $x_1 < x_1'$, with $P(x_1') = 0$.

- **$K_1$ estimate only:** Here $K = K_1$, $K_2$ cannot be estimated, and $x_2$ is unknown. In this case (Fig. 2c), we begin with $P(x_1) = 0$ and continue assigning voxel probabilities until the process terminates at $x_2'$, prior to $x = \sigma$.

- **$K_2$ estimate only:** Here $K = K_2$, $K_1$ cannot be estimated, and $x_1$ is unknown. In this case (Fig. 2d), we begin with $P(x_2) = 1$ and continue assigning voxel probabilities until the process terminates at $x_1'$, prior to $x = -\sigma$.

- **Neither $K_1$ or $K_2$ is available:** In this case, we do nothing. It is quite possible voxels affected by this boundary voxel might be assigned by a neighboring boundary voxel at a later point.

2.5. Probability Assignment of Non-Boundary Points

The previous procedure computes the probabilities for the boundary voxels and voxels close to the boundary. We also need to assign probabilities for the remaining voxels, so as to facilitate the distance field computation (as described the following sections). Note that our presegmentation roughly classified all voxels as either background or object voxels. We begin with this assignment (0 or 1) as an initial probability value and proceed to perform local neighborhood operations to correct these values, where necessary, as follows:
• For each unassigned voxel, $v_x$ on the object, compute the average probability, $P_{avg}$ within its 26 connected neighborhood. $P_{avg}$ is thresholded against a background threshold $T_{bgrnd}$, and an object threshold, $T_{obj}$.

$$P(v_x) = \begin{cases} 0, & \text{if } P_{avg} < T_{bgrnd} \\ 1, & \text{if } P_{avg} > T_{obj} \end{cases}$$

(9)

• If $(T_{bgrnd} < P_{avg} < T_{obj})$, the voxel’s probability is determined by looking at a fixed number of local neighbors (we use 2) along the gradient direction on either side of the voxel.

2.6. Distance Field Construction

As mentioned earlier, the principal goal of building the probability function is to have a more accurate description of the boundary. We exploit this in building a distance field that is more accurate and of higher precision. In particular, the boundary voxels will have non-zero distances, in contrast to traditional distance fields where all distances at the boundary start out with zero. Secondly, distance computation and propagation is also different.

Consider Fig. 3. $P_A$ and $P_B$ represent the probabilities assigned to points $A$ and $B$. We compute the distances $D_A$ and $D_B$, from $B$ and $A$ respectively are calculated as follows:

$$D_A = D_B + P_A D(B, A)$$

(10)

$$D_B = D_A + P_B D(A, B)$$

(11)

In other words, we scale the distance between the points, $D(A, B)$, by the probability of the point being on the boundary. The traditional distance field algorithm assumes $P_A = P_B = 1$.

Using the above formulation, we compute the distance field using the approach of Gagvani.\(^4\) In our implementation, we use exact voxel distances, $1 - \sqrt{2} - \sqrt{3}$ for isotropic volumes, or the actual voxel distances based on the voxel size.

2.7. Centerline Extraction

Once the distance field has been computed, we can now extract the centerline from the volume. We use a slight variant of the algorithm proposed by Wan et al.\(^2\)∗. Currently we use the voxel with the largest DFB (distance from boundary) as the root of the minimum spanning tree (MST) (as detailed in Wan et al.\(^2\)) in the centerline extraction algorithm. We also keep track of the largest geodesic distance from this starting point (or DFS), which is then used to lead toward the root point, via the chain of links built during the MST construction.

3. RESULTS

In order to evaluate the accuracy and robustness of our algorithm, we have tested our centerline extraction method on both synthetic as well as publicly available medical imaging datasets\(^1\). Accuracy was measured quantitatively on two datasets whose exact centerline is known, (1) a synthetic dataset, and, (2) a radiologist supervised segmentation of a a head MRT dataset (Fig. 4). This is followed by experiments on three medical imaging datasets with added noise to illustrate the robustness of our method.

\(^{1}\)We had to slightly modify this algorithm as the flowchart seemed to have some missing conditions.

\(^{2}\)Color images will be available at http://www.cs.uncc.edu/~krs/publ.html
3.1. Implementation

Our centerline extraction algorithm has been implemented in C++ on Linux workstations. We have used the Insight toolkit (ITK)\(^2^5\) for some of the image processing operations and noise generation, and the Visualization Toolkit (VTK)\(^2^6\) for displaying the results. All interaction is provided using the Fast and Light Toolkit (FLTK)\(^2^7\)‡.

3.2. Experiments: Accuracy Analysis

We have used a synthetically generated volume of a curved, sinusoidally shaped cylinder \((100 \times 100 \times 102\) voxels, Fig. 4), as well as a radiologist supervised segmented head MRT dataset \((256 \times 320 \times 128\) voxels, Fig. 4) to evaluate the accuracy of our probabilistic centerline extraction method, and compare it with the traditional distance transform method. Our parameters of synthetically generated volume are similar to those used in Aylward;\(^1^4\) background intensity level is set to 100 and the object voxels range from 150 at the boundary to 200 at the center of the object. Because Gaussian noise is the most common in medical images, we added it (using ITK) to both datasets, with \(\sigma = 40\) for the sinusoidal cylinder dataset, and \(\sigma = 20\) for the head MRT data. As described in Aylward,\(^1^4\) \(\sigma = 40\) and above represents a worst case scenario, even for medical images.

We computed the gradient magnitude field (using \texttt{itk::GradientMagnitudeRecursiveGaussianImageFilter}) with \(\sigma = 10\) for the cylinder dataset and \(\sigma = 0.5\) for MRT dataset.

Three accuracy measures similar to Aylward,\(^1^4\) were computed from the extracted centerline, as follows:

- **Average Error**: This represents the mean distance between corresponding points from the ideal centerline and extracted centerline. Results can be ambiguous, depending on how the corresponding points are computed; in our implementation, we pick the larger of the two distances computed, starting from each of the two centerlines.

- **Maximum Error**: This represents the maximum distance between two corresponding points.

- **Percent Points Within 1 Voxel**: This represents the percentage of voxels on the extracted centerline that are within 1 voxel of their closest ideal centerline point.

For the cylinder dataset, we know the exact location of the centerline, which enables us to measure the accuracy of our algorithm under various conditions. For the segmented MRT head dataset, we first use the traditional distance transform to extract the centerline from the segmented medical data by specifying start and end points on a a part of the relatively thick trunk. This result is considered as the ideal centerline in our experiments (our implementation closely follows,\(^2\) which is based on locating centerline voxels with the largest distance from the boundary). Then we tested our algorithm on noisy MRT data. The volume was first roughly segmented using thresholding. Both methods were then used on this dataset with the same start and end points(as used to compute the ideal centerline).

Fig. 4 illustrates the results on this dataset with added Gaussian noise with noise deviation, \(\sigma = 40\) for the cylinder dataset and \(\sigma = 20\) for the MRT head dataset. The left column of images are generated using our probabilistic distance transform method, while the images in the right column are generated using the traditional distance transform method. The ideal centerline (in red) is overlaid with extracted centerline (in yellow). Significant errors can be noticed in the images in the right column. For spatial perspective, we also output the isosurface (via the Marching Cubes algorithm\(^2^8\)) of the object. At the higher noise levels, the isosurface adds more and more geometry, making it difficult to perceive the centerline. Hence we have made the isosurface almost fully transparent.

Table 1 displays the computed accuracy measures the for sinusoidal cylinder dataset and MRT head dataset both with gaussian noise, \(\sigma = 20\). Average errors of our method are between 0.7-1.3 voxels, vs. 3.0-5.7 voxels using the traditional distance form method. Maximum errors are also much smaller, 1.6-3.0 vs. 5.2-11.2 voxels, and over 73-97% of voxels are within 1 voxel, vs. 9-75% for the traditional method.

\(^2\)ITK, VTK and FLTK are open source toolkits that run across a number of different platforms.
Figure 4. Comparison between traditional distance transform vs. probabilistic distance transform approach on sinusoid cylinder dataset (top row) at Noise level, $\sigma = 40$ and segmented head MRT (bottom row) data at Noise level, $\sigma = 20$. Ideal centerline (in red) overlaid on top of extracted centerline (in yellow). Left column: Using our probabilistic distance transform method, Right column: Using traditional distance transform method.

Table 1. Centerline accuracy of sinusoid cylinder dataset at noise level 40 and head MRT data at noise level 20.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Measures</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Trad. Dist. Transform/Prob. Dist. Transform</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avg. Error</td>
<td>Max. Error</td>
</tr>
<tr>
<td>Sinusoid Cyl.</td>
<td>5.7/1.3</td>
<td>11.2/3.0</td>
<td>9.1/73.5</td>
</tr>
<tr>
<td>Head MRT</td>
<td>3.0/0.7</td>
<td>5.2/1.6</td>
<td>75.6/97.4</td>
</tr>
</tbody>
</table>
3.3. Experiments: Medical Data

Additionally, we have tested our algorithm with two medical volume datasets available from the archive at University of Tuebingen, and a colon dataset available from the National Library of Medicine. We describe our experiments with these datasets next, in the presence of Gaussian noise.

Fig. 5 displays the results of the aneurysm dataset with no added noise (left), Gaussian (right) at a noise level, $\sigma = 50$. Centerlines were extracted on all vessels connected to the main trunk. As this vascular tree has also a significant number of disconnected structures as well as many extremely small vessels, it is a particular challenging dataset. Here we show the isosurface of the vessels (for spatial perspective) from the clean (no noise) data, as otherwise the centerline is barely visible. Since it’s very hard to judge the results from thin branches, we focus on the resulting centerlines of the trunk. At high noise levels, there are a few spurious branches using our method.

Fig. 6 shows the results of a second head MRT data with Gaussian noise at $\sigma = 20$. The isosurface is extracted from segmented data. The MRT data set has considerably weaker boundaries. As the vessels are just a few voxels wide, for noise levels of $\sigma = 40$ (not shown) and above, the centerline starts to exhibit errors. This can also happen when small blood vessels are extremely close to each other, as encountered by Frangi. Thus, we also qualitatively verify the centeredness of our algorithm using 2D texture mapped planes (not shown), corresponding to axial, sagittal and coronal orientations.

Finally, we have tested our method on a colon dataset from the large archive at the National Library of Medicine. Fig. 7 illustrates the effects of adding noise. The left image illustrates the dataset with no added noise, and the right image with Gaussian noise added, at a noise level of $\sigma = 20$. In these images, the centerline of the noisy dataset (in yellow) is overlaid on the centerline of the clean dataset (in red). Mid sections of the
The centerlines deviate at the beginning and the ending regions of the colon; this is due to the differing start and end points used in the respective datasets.

Table 2 illustrates the running times for the three medical dataset examples with Gaussian noise. Probability function construction times range from 1-7 minutes. Similar to the synthetic datasets, running times can be further improved by more properly handling volume boundary effects.

Table 2. Running Times(secs): Medical Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Resolution (voxels)</th>
<th>Running Times (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Prob. Model Constr./Centerline Extr.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>σ = 0</td>
</tr>
<tr>
<td>Aneurysm</td>
<td>256 × 256 × 256</td>
<td>114.1/8.0</td>
</tr>
<tr>
<td>MRT</td>
<td>256 × 320 × 128</td>
<td>70.8/4.5</td>
</tr>
<tr>
<td>Colon</td>
<td>409 × 409 × 220</td>
<td>424.7/40.9</td>
</tr>
</tbody>
</table>

section show very little error. The centerlines deviate at the beginning and the ending regions of the colon; this is due to the differing start and end points used in the respective datasets.

Table 2 illustrates the running times for the three medical dataset examples with Gaussian noise. Probability function construction times range from 1-7 minutes. Similar to the synthetic datasets, running times can be further improved by more properly handling volume boundary effects.

Figure 8. Comparison of using traditional distance fields vs. using probabilistically defined distance fields in centerline extraction. Top three images show an axis-aligned cylinder with its centerline extracted using the algorithm by Wan et al. at noise levels of sigma = 0, 10, 20; bottom figure illustrates the object with a probabilistic boundary, with σ = 20.
4. DISCUSSION

The primary goal of this work was to extract centerlines of tubular structures in a robust and accurate fashion, without assumptions of exact boundaries. Complex datasets such as those used in this work and archived at are considerably challenging to traditional distance field algorithms, which assume a binary (usually thresholded) dataset and hence zero distance values on the boundary. Our main idea in this work is to determine a probabilistic estimate of the boundary location (in the spirit of for computing smooth transfer functions) and use these to modify the traditional distance from boundary. We still take advantage of the speed of distance field based algorithms for centerline extraction.

An example of comparing our method to traditional distance field based centerline extraction is shown in Fig. 8 for a simple example of an axis-aligned cylinder. The top three images illustrate the cylinder with noise levels of $\sigma = 0, 10$ and $20$. We use the method of Wan et al. We see the dramatic deviation of the centerline from the horizontal as the noise level is increased. The bottom image illustrates our algorithm at $\sigma = 20$. As illustrated in Table 1 the errors for the cylinder are very low even at very high noise levels.

We note with interest the use of the Hessian based methods coupled with multi-scale analysis to explore and analyze highly complex medical datasets. These are generally computationally more expensive (as our initial investigations have revealed); this is especially the case if they have to be computed for all object voxels. We are currently looking into using the Hessian in a limited fashion, thus making it possible to use eigen value analysis for larger medical structures.

5. CONCLUSIONS

We have presented a robust and accurate centerline extraction algorithm that can work with gray scale images and a rough segmentation of the structures of interest. The goal of this work is toward analyzing large medical structures of interest. We have presented a probabilistic model to estimate the boundary using an integration of the gradient field. The computed voxel probabilities were then used to build a modified distance field which is then used to extract the centerline of the object. Preliminary experiments on both synthetic and clinical datasets are promising. Accuracy within a voxel has been illustrated using this model. We have also tested our algorithm on three medical imaging datasets, including an aneurysm dataset, head MRT blood vessel dataset, and a colon dataset. Comparisons to traditional distance field algorithms for centerline extraction illustrate the robustness of this method within noisy environments.

The work presented here has focused on a centerline extraction algorithm that can work well in the presence of noise. Several aspects of of this algorithm are incomplete, such as incorporating scale as part of this method to adapt to more significant changes in object geometry. We are currently investigating these issues.

REFERENCES