Accurate and Robust Centerline Extraction from Tubular Structures in Medical Images

Jianfei Liu and Kalpathi Subramanian

Abstract. Extraction of centerlines is useful in interactive navigation and analysis of objects in medical images, such as the 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) accurate, (2) noise tolerant, (3) computationally efficient, and (4) preferably do not require an accurate segmentation. We propose a new centerline extraction method that employs a Gaussian type probability model to estimate the boundaries of medical objects. The model is computed using an integration of the image gradient field. Probabilities assigned to boundary voxels are then used to compute a more robust distance field, that is less sensitive to noise. Distance field algorithms are then applied to extract the centerline. Noise tolerance of our method is demonstrated by adding Gaussian, Poisson and Rician noise to these datasets, and comparing results to traditional distance field based methods. Accuracy of our method was measured using two datasets with known centerlines, (1) a synthetically generated sinusoidally varying cylindrical dataset, and (2) a radiologist supervised segmented head MRT angiography dataset. Average errors for the cylinder dataset using our method was 0.5-0.8 voxels vs. 0.7-2.0 voxels using the traditional distance transform method; for the MRT dataset, it was 0.5-0.7 voxels vs 2.0-3.0 voxels for the traditional method. Additionally, experiments with six datasets were performed, (1) a second head MRT angiography dataset, (2) an aneurysm dataset, and (3) four colon datasets. Results of our approach illustrate the robustness of our centerline extraction method, in terms of the smoothness as well as reduced artifacts, such as spurious branches. Finally, the stability of our centerline is evaluated by measuring its sensitiveness to initialization and segmentation

Jianfei Liu
Charlotte Visualization Center, Department of Computer Sicence,
University of North Carolina at Charlotte, Charlotte, NC, 28223
e-mail: jliu1@uncc.edu

Kalpathi Subramanian
Charlotte Visualization Center, Department of Computer Sicence,
University of North Carolina at Charlotte, Charlotte, NC, 28223
e-mail: krs@uncc.edu

Z.W. Ras and W. Ribarsky (Eds.): Advances in Information & Intelligent Sys., SCI 251, pp. 139–162.
springerlink.com

parameters (on the head MRT dataset), and found to vary on the average between 0.2-0.4 voxels. Running times of our algorithm are on the order of 1-7 minutes for datasets ranging from $256 \times 256 \times 256$ to $409 \times 409 \times 219$ voxels.

1 Introduction

The extraction of *centerlines* is useful in 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[1], that have precise definitions and requirements[2] 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* and *accurate* centerline extraction from medical images and volumes, especially 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 modified 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 a number of publicly available medical imaging datasets. These experiments were done by adding significant amounts of Gaussian, Poisson and Rician noise to the datasets, as these have been shown to be relevant to medical imaging data from CT, MRI[29]. Accuracy is quantified using both synthetic models, and an MRT angiography dataset with a radiologist supervised segmentation. We also evaluate our technique by comparison to traditional distance field methods. We have successfully tested this approach on medical imaging datasets from blood vessel geometry in the brain, and CT datasets of the 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.

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 probabilistic distance field based centerline extraction method and present experimental 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 to boundary, DFB). Such distant maps have been used to accurately represent binary (or segmented) volumes to control aliasing artifacts[4], extract skeletons [5]. 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[6], 3-4-5 chamfer metric[7], or 10-14-17[8]. Exact voxel distances ($1-\sqrt{2}-\sqrt{3}$), assuming unit cube voxels) have also been used[2].

A number of researchers have used a combination of distance fields and Dijsktra'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.[8] used this approach but modify the shortest path voxels to the maximal DFB voxels orthogonal to the path, while Zhou[6] chooses among voxel clusters with the same DFS distance. Bitter et al.[10, 9] 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.[2] 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. In other words, the centerline is usually highly sensitive to the accuracy of the boundary location; it is this fact that our proposed method attempts to overcome by (1) using a better boundary estimate, and (2) using smoothing and object scale towards a more robust method.

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 $\mathbf{x}_0$,

$$I(x_0+\delta x_0,\sigma) \approx I(x_0,\sigma)+\delta x_0^T \Delta_{0,\sigma}+\delta x_0^T H_{0,\sigma}\delta x_0 \tag{1}$$

where $\Delta_{0,s}$ and $H_{0,s}$ are the gradient and Hessian of the image at x_0 at a scale s. Secondly, scale space theory[11, 12] relates scale to derivatives, which can be defined as convolution with derivatives of Gaussians:

$$\frac{\partial}{\partial x}I(x,s)=\sigma^{\gamma}I(x)*\frac{\partial}{\partial x}G(x,\sigma) \tag{2}$$

where G is a Gaussian with zero mean and deviation of σ. Parameter γ, introduced by Lindeberg[13] helps define normalized derivatives and provides the intuition for the use of scale in analyzing image structures.

In [14], 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 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 multi-scale, can handle vessels of varying cross-section, given the results of Eq. 2, that relate scale to boundary position.

Aylward [15] 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.[16] 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 at al.[17, 18], and tubular structure detection [19].

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[20] and more recently, using topological characteristics derived from repulsive force fields[21]. Radial basis functions [22] 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 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[23]; 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

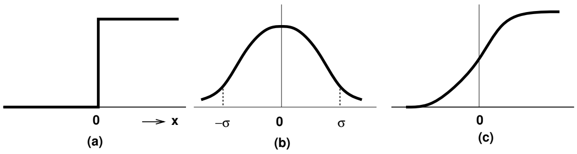


Fig. 1 (a) Step edge: ideal boundary, (b)Change in gradient magnitude (approximated by a Gaussian), (c) integral of gradient: blurred edge (also error function)

for complex datasets, several examples of which can be found in [24], and which we have also used to test our methodology.

2.2 Boundary Model

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 [25]) illustrates a step edge and its blurring by a Gaussian. While Kindlmann [25] 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

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

where $f'(x)$ is centered around the point x, K is a normalizing constant and σ 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, $\mathbf{g}$ or $-\mathbf{g}$) 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, $\mathbf{g}$ or $-\mathbf{g}$
3. Beginning from position $x = 0$, move along $\mathbf{g}$ or $-\mathbf{g}$ to determine $-\sigma$ and σ 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

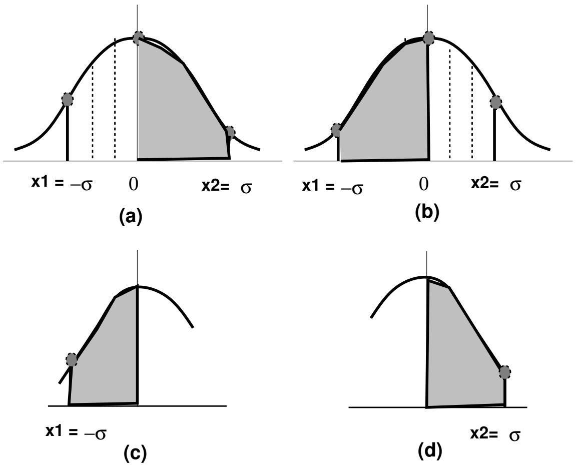


Fig. 2 (a) $K_1 < K_2$: shaded area on the right is larger, (b) $K_1 > K_2$: shaded area on the left is larger, (c,d) Only K_1 or K_2 can be estimated

the object boundary. In other words, the points at which $-\sigma$ and σ 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 right 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 $\mathbf{g}$ or $-\mathbf{g}$ 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 σ, 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. Its 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}.

$$P(v_x) = \begin{cases} 0, & \text{if } P_{avg} < T_{bgrnd} \\ 1, & \text{if } P_{avg} > T_{obj} \end{cases} \tag{9}$$

- 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 and continuous description of the boundary, and encoding the uncertainty associated with it. Specifically, at the boundary, the probability is close to 0.5 (equally likely to be object or background), points leading towards the interior of the object tend towards 1.0, while points leading away from the object (and into the background) tend towards 0.0. We use these probabilities 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.

Fig. 3 displays colored distance field images in one slice of a synthetically generated dataset (shown in Fig. 7, with added Gaussian noise, $\sigma = 40$); intensity variation from blue to red represents increasing distance value. Left image illustrates the distance field using the traditional distance transform, while the right image shows the slice using the probabilistic distance transform. Notice that the reddish hues (large distances close to the centerline) are more continuous in the right image, while the distance field is highly corrupted in the left image, due to the noise. This in turn affects the centerline, as can been from a simple example in Fig. 6.

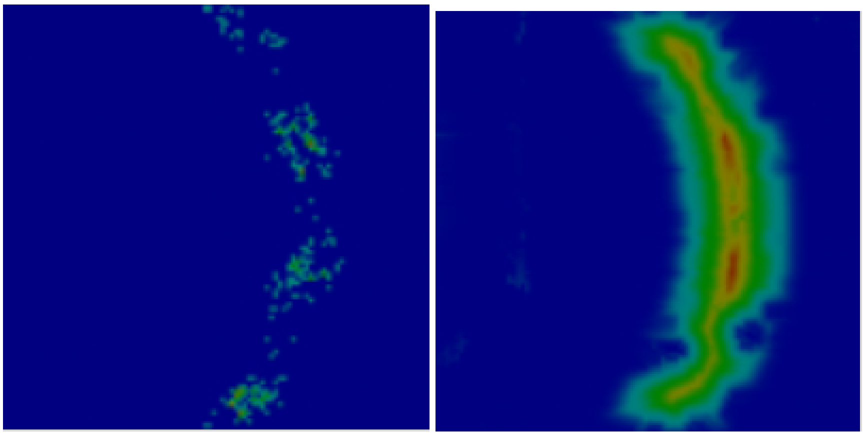
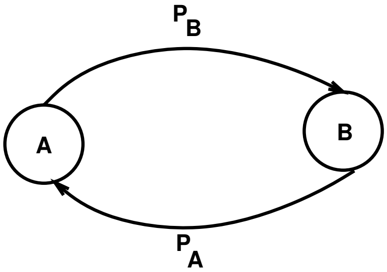


Fig. 3 Comparison of Distance Fields:Center slice of distance volume where color map from blue to red represents increasing distance. *Left:*Using traditional distance transform. *Right:* Using probabilistic distance transform.

Fig. 4 Modified distance field computation.



The use of the probability function makes distance computation and propagation different from traditional distance fields. Consider Fig. 4. P_A and P_B represent the probabilities assigned to points A and B. We compute the distances D_A and D_B, from voxels B and A respectively are calculated as follows:

$$D_A = D_B + P_A D(B,A) \tag{10}$$
$$D_B = D_A + P_B D(A,B) \tag{11}$$

In other words, the distance D_A from another voxel B is given by the sum of the stored distance D_B at B and the distance between A and B weighted by the estimated boundary probability of B. Thus, voxels with smaller probabilities (partially in the background) have a smaller contribution. This is a key factor to the stability of the centerline, especially in the context of a noisy boundary. Note that the traditional distance field algorithm assumes $P_A = P_B = 1$.

Using the above formulation, we compute the distance field using the approach of [5]. 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][1]. Currently we use the voxel with the largest DFB (distance from boundary) as the root of the minimum spanning tree (MST) (as detailed in [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, stability and robustness of our algorithm, we have tested our centerline extraction method on both synthetic as well as publicly available medical imaging datasets. Accuracy was measured quantitatively on two datasets whose exact centerline is known, (1) a synthetic dataset (Fig. 7), and, (2) a radiologist supervised segmentation of a a head MRT dataset (Fig. 8. This is followed by experiments on six medical imaging datasets with added noise to illustrate the robustness of our method. Finally, we illustrate the stability of the extracted centerline in response to variations in initialization segmentation parameters. In all of these experiments, we compare both quantitatively and qualitatively our probabilistic centerline extraction method to traditional distance transform method.

3.1 Implementation

Our centerline extraction algorithm has been implemented in C++ on Linux workstations. We have used the Insight toolkit (ITK)[26] for some of the image processing operations and the Visualization Toolkit(VTK) [27] for displaying the results. All interaction is provided using the Fast and Light Toolkit (FLTK)[28][2].

3.2 Noise Generation

There are three common types of white noise observed in medical images: Gaussian, Poisson, and Rician [29]. We assume the noise is uncorrelated[30]. Among these

[1] We had to slightly modify this algorithm as the flowchart seemed to have some missing conditions.

[2] ITK, VTK and FLTK are open source toolkits that run across a number of different platforms.

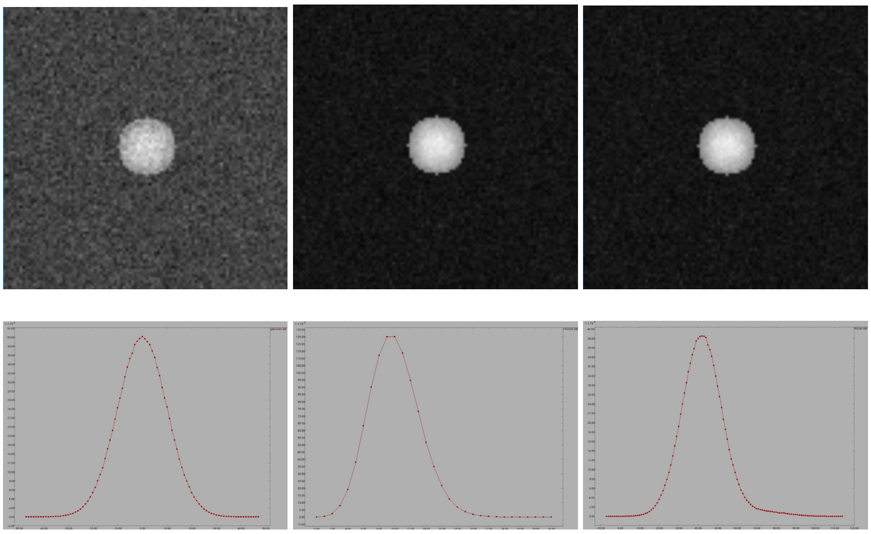


Fig. 5 Noisy Disk images with Gaussian, Poisson, and Rician noise and their associated distributions. *Left column:*:Gaussian noise with $\mu = 0, \sigma = 10$, *Middle column:* Poisson noise with $\sigma = 10$, *Right column:* Rician noise with $\mu = 2, \sigma = 10$

models, Gaussian noise is the most common in medical images, resulting from the central limit theorem, which states that the sum of many random variables generates a signal with a Gaussian PDF. This was implemented using ITK Gaussian generator filter in our experiments. Poisson noise is common in CT images, which are generated by accumulating photons over a detector. We modified the non-uniform random variable code in [31]'s library to generate Poisson noise, using chop down sample methods. Finally, many estimation experiments [29, 32, 33, 34] in MRI demonstrate that the data in a magnitude image is Rician distributed, because computation of a magnitude image is a non-linear operation. We used the method described in [29] to generate Rician noise, by integrating the magnitudes of two Gaussian noise PDFs with the same deviation (see pages 138-139 in [35]).

We generated synthetic data volumes in a manner similar to [15], with background intensity set to 100 and the object intensity ranging from 150 to 200, satisfying a parabolic profile. Fig. 5 displays three noisy images and their associated distribution. The three types of noise were used in our experiments to corrupt the volume data, so as to measure their impact on the centerline algorithms; in particular, our goal was to understand their accuracy, robustness and stability under these conditions.

3.3 Experiments: Accuracy Analysis

We have used a synthetically generated volume of a curved, sinusoidally shaped cylinder ($100 \times 100 \times 102$ voxels, Fig. 7), as well as a radiologist supervised segmented head MRT dataset ($256 \times 320 \times 128$ voxels, Fig. 8) to evaluate the accuracy of our probabilistic centerline extraction method, and compare it with the traditional distance transform method. We added the three types of noise to both datasets, with $\sigma = 20$ and 40 for the sinusoidal cylinder dataset, and $\sigma = 10$ and 20 for the head MRT data. As described in [15], $\sigma = 40$ and above represents a worst case scenario, even for medical images. We computed the gradient magnitude field (using *itk::GradientMagnitudeRecursiveGaussianImageFilter*) with $\sigma = 10$ for the cylinder dataset and $\sigma = 0.5$ for MRT dataset.

Three accuracy measures similar to [15], were computed from the extracted centerline, as follows:

- *Average Error*: This represents the mean distance between corresponding points between 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. Fig. 7 illustrates the results on this dataset with added Gaussian noise (top row), Poisson noise (middle row) and Rician noise (bottom row), with noise deviation, $\sigma = 40$ for all images. For the Rician noise, the mean, $\mu = 2$. 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, especially for Gaussian and Rician corrupted data. For spatial perspective, we also output the isosurface

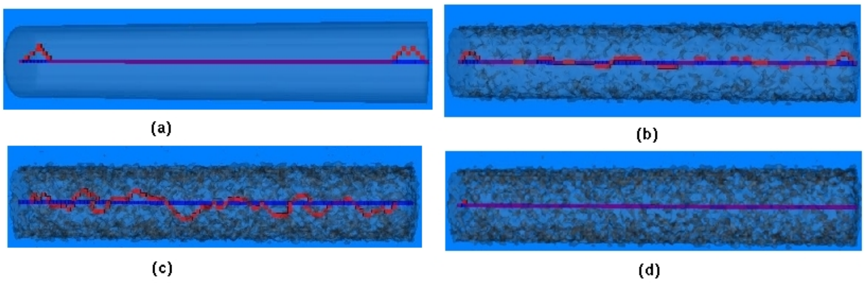


Fig. 6 Centerline of a straight cylinder. (a, b, c) Traditional distance field method with noise, $\sigma = 0, 10, 20$, (d) probabilistic distance field method with noise, $\sigma = 20$

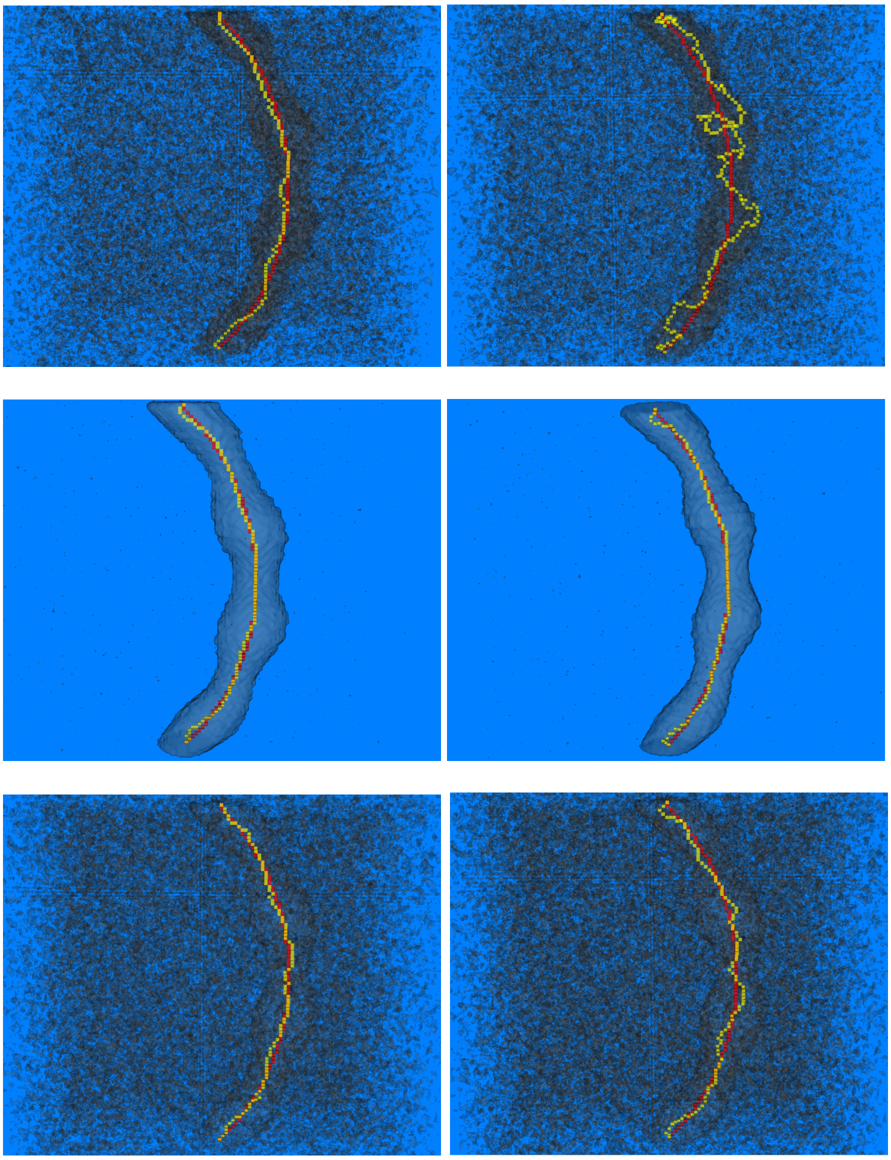


Fig. 7 Results: Synthetic Dataset Centerline Extraction Under Gaussian (top row), Poisson (middle row) and Rician Noise (bottom row). Noise level, $\sigma = 40$. Ideal centerline(in red) overlaid on top of extracted centerline(in yellow). *Left column:* Using our probabilistic distance transform method, *Right column:* Using traditional distance transform method.

Table 1 Accuracy of Sinusoidal Cylinder. Comparison of the probabilistic distance transform method vs. traditional distance transform method at two noise levels, $\sigma = 20, 40$.

Noise Type	Error Measures: Prob. Dist. Transf./Trad. Dist. Transf					
	$\sigma = 20$			$\sigma = 40$		
	Average	Maximum	% Pts within 1 voxel	Average	Maximum	%Pts within 1 voxel
Gaussian	0.8/2.0	2.2/5.4	98.0/48.7	1.3/5.7	3.0/11.2	73.5/9.1
Poisson	0.5/0.7	2.23/3.0	97.1/95.1	0.5/0.7	2.2/4.1	98.1/91.5
Rician	0.5/0.7	2.2/3.0	97.1/95.2	0.8/1.7	2.0/4.2	95.1/53.2

(via the Marching Cubes algorithm[36]) of the object at a threshold of 150 (where the boundary begins). 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 shows three accuracy measures for the cylinder dataset with $\sigma = 20, 40$. We can see that the average error using our method is between 0.5-0.8 voxels, in comparison to 0.7-2.0 voxels using the traditional distance transform method. This demonstrates that our probabilistic approach is more noise tolerant than traditional distance transform methods.

We next compare our method with the traditional distance transform method on the segmented MRT head dataset. In order to accurately estimate the results, a part of the relatively thick trunk was chosen as the experimental volume. We first use the traditional distance transform to extract the centerline from the segmented medical data by specifying start and end points. 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, with $\sigma = 20$. 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. 8 displays the results; the ideal centerline (in red) is overlaid with the extracted centerline (in yellow). The the three rows of images correspond to datsets with added Gaussian, Poisson and Rician noise, respectively. The left column of images illustrates results using the traditional distance transform, while the right column shows results using the probabilistic distance transform method. Notice the regions marked A and B in the figures; due to segmentation errors, the traditional method does not recognize the horizontal bend of the blood vessel and builds the centerline as if there are two separate vessels. Our probabilistic method computes the correct centerline, primarily due to the gradient tracking procedure, that handles very small vessel structures(the region around points A and B were at most 3 voxels wide) in a more robust fashion.

Table 2 displays the computed accuracy measures the for MRT head dataset, with the three different noise types at $\sigma = 20$. Average errors of our method are between 0.5-0.7 voxels, vs. 2-3 voxels using the traditional distance form method. Maximum

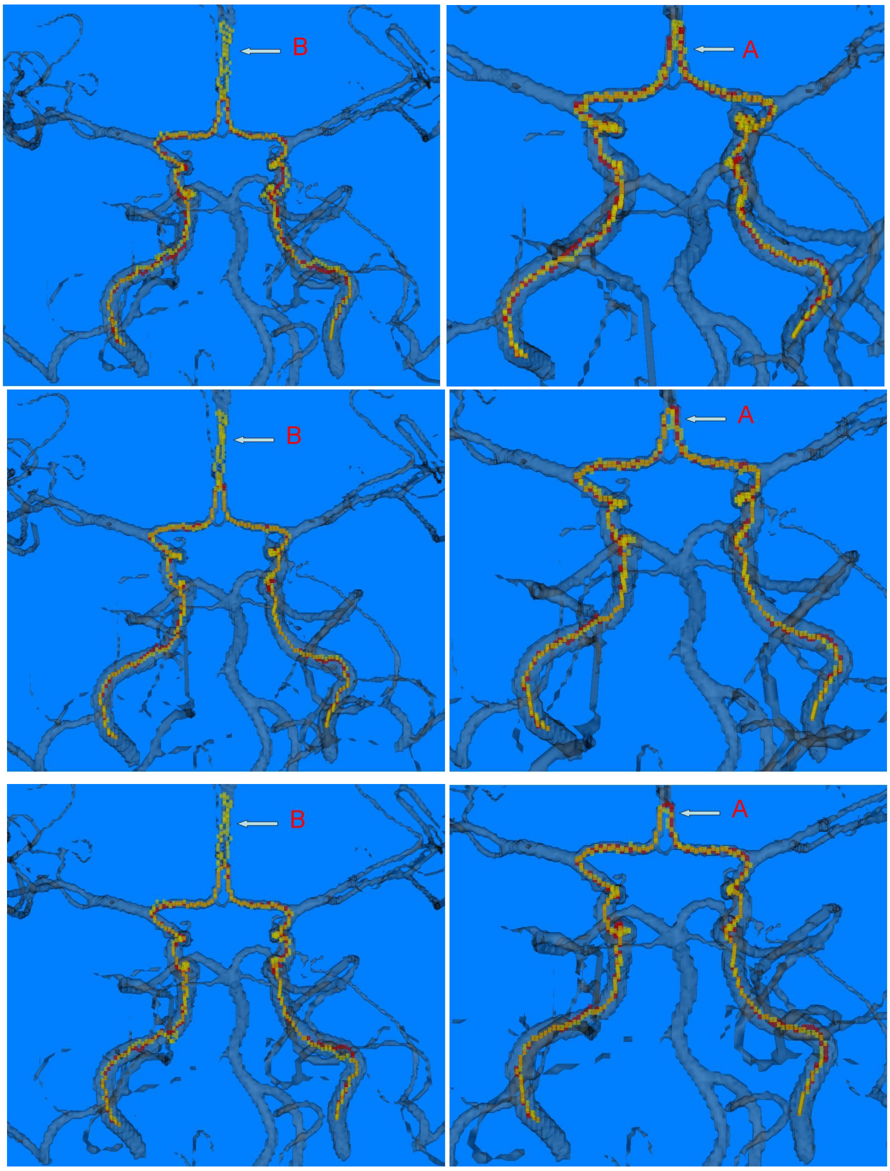


Fig. 8 Comparison between traditional distance transform vs. probabilistic distance transform approach on segmented head MRT data, with added Gaussian (top row), Poisson (middle row) and Rician (bottom row) noise. Ideal centerline (in red) is overlaid on the extracted centerline (in yellow). Images in the left column are generated using traditional distance transform method, while images in the right column are using the probabilistic distance transform method.

Table 2 Centerline accuracy of head MRT data at noise level 20.

Noise Type	Measures		
	Trad. Dist. Transform/Prob. Dist. Transform		
	Average Error	Maximum Error	% Points Within 1 Voxel
Gaussian	3.0/0.7	5.2/1.6	75.6/97.4
Poisson	2.0/0.5	4.8/1.6	85.7/98.9
Rician	2.2/0.5	4.8/1.7	81.4/99.2

errors are also much smaller, 1.6-1.7 vs. 4.8-5.2 voxels, and over 95-99% of voxels are within 1 voxel, vs. 75-85% for the traditional method.

3.4 Experiments: Medical Data

Additionally, we have tested our algorithm with two medical volume datasets available from the archive at University of Tuebingen[24], and four colon datasets available from the National Library of Medicine, [37]. We describe our experiments with these datasets next, in the presence of Gaussian, Poisson and Rician noise.

Fig. 9 displays the results of the aneurysm dataset with no added noise (top left), Gaussian (top right), Poisson(lower left) and Rician noise (lower 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. 10 shows the results of a second head MRT data with Gaussian, Poisson and Rician 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 [14]. 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 four colon datasets, part of the large archive at the National Library of Medicine[37]. Fig. 11 illustrates centerlines extracted from three of these datasets. In Fig. 12, we illustrate the effects of adding noise to one of these datasets. The upper left image illustrates the dataset with no added noise, and the remaining three images with Gaussian(upper right), Poisson(lower left) and Rician (lower right) noise added, at a noise level of $\sigma = 20$. In these images, the centerline of the noisy dataset (in yellow) is overlaid on the

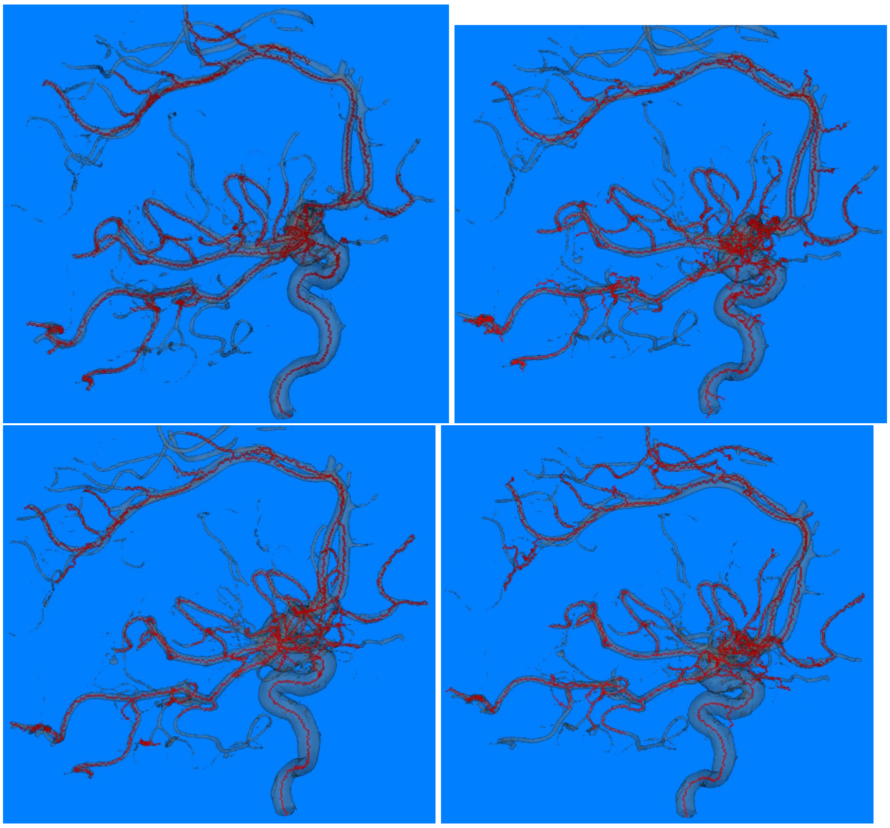


Fig. 9 Aneurysm Dataset with no added noise (top left), Gaussian (top right), Poisson (lower left) and Rician noise (lower right), $\sigma = 50$.

centerline of the clean dataset (in red). Mid sections of the 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 3 shows errors measured on one of the colon datasets when noise is added to the datasets; errors are calculated with respect to the centerline extracted from the clean dataset. Average errors range from 0.3 to 2.3 voxels, while the maximum error ranges from 2.2-4.3 voxels. As can be expected, errors increase with increasing noise levels.

Table 4 illustrates the running times for the six medical dataset examples with Gaussian noise. Probability function construction times range from 1-7 minutes in the Dell Dimension 4550 desktop (Pentium 4 2.66GHz, 1G RAM). Similar to the synthetic datasets, running times can be further improved by more properly handling volume boundary effects.

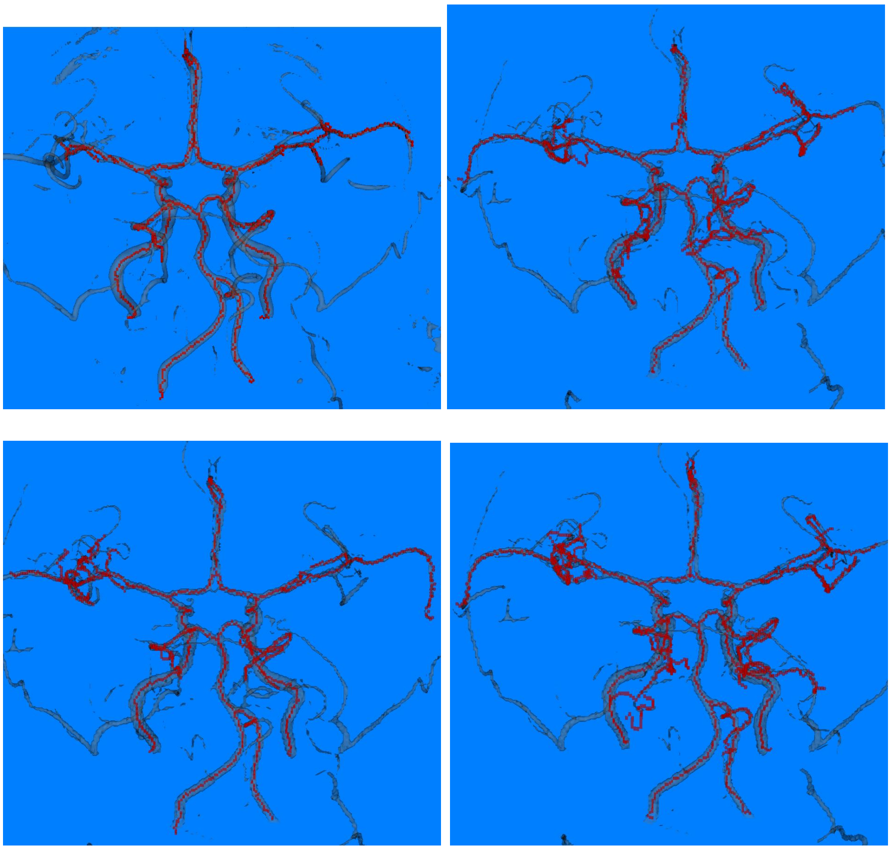


Fig. 10 Aneurysm Dataset. $\sigma = 20$.*(Upper Left:)* No added noise, *(Upper Right:)* With Gaussian Noise, $\sigma = 20$, *(Lower Left:)* With Poisson Noise, $\sigma = 20$, *(Lower Right:)* With Rician Noise, $\sigma = 20$.

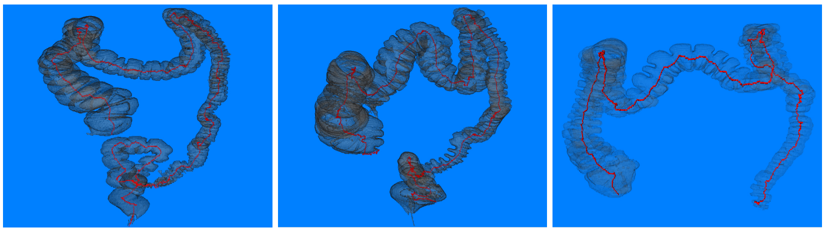


Fig. 11 Centerline Extraction from 3 Colon Datasets.

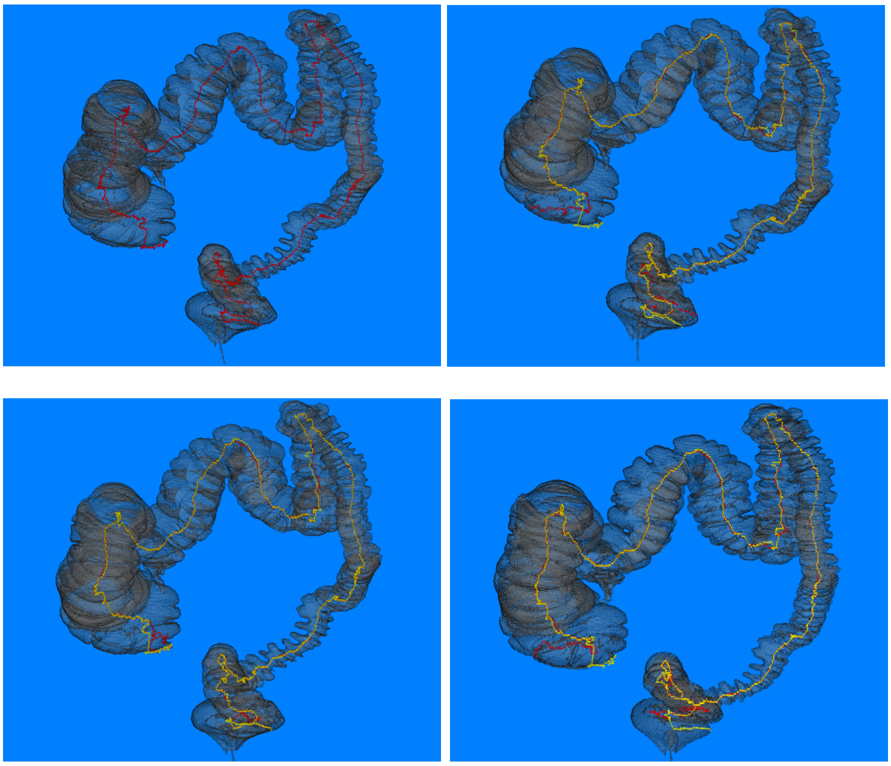


Fig. 12 Results:Analysis of Colon Dataset with Added Noise, $\sigma = 20$. *Upper Left:* Centerline of colon dataset with no added noise, *Upper Right:* With Gaussian noise, *Lower left* With Poisson noise, *Lower right:* With Rician noise. Centerlines of noisy data (in yellow) overlaid on centerline of clean dataset(in red).

Table 3 Error Measures of Colon Dataset with added noise.

Noise Level		Error Measures (in voxels)		
		Average	Maximum	% points within 1 voxel
Gaussian	$\sigma = 10$	0.3	2.2	95.5
	$\sigma = 20$	0.6	2.7	92.1
	$\sigma = 40$	1.7	4.2	77.8
Poisson	$\sigma = 10$	1.0	4.2	84.6
	$\sigma = 20$	0.4	2.6	91.4
	$\sigma = 40$	0.7	3.6	89.4
Rician	$\sigma = 10$	0.4	2.5	92.0
	$\sigma = 20$	0.8	2.9	86.1
	$\sigma = 40$	2.3	4.3	68.2

Table 4 Running Times(secs): Medical Datasets

Dataset	Resolution (voxels)	Running Times (seconds)		
		Prob. Model Constr./Centerline Extr.		
		$\sigma = 0$	$\sigma = 20$	$\sigma = 25$
Aneurysm	$256 \times 256 \times 256$	114.1/8.0	114.3/8.1	114.3/8.1
MRT	$256 \times 320 \times 128$	70.8/4.5	74.1/13.9	74.1/13.9
Colon1	$409 \times 409 \times 220$	424.7/40.9		
Colon2	$385 \times 385 \times 231$	397.8/56.3		
Colon3	$409 \times 409 \times 212$	412.7/60.6		
Colon4	$409 \times 409 \times 219$	417.9/28.2		

Table 5 Stability of Extracted Centerline: Error Measurements as Segmentation Threshold is Varied from 60 to 90. Noise level $\sigma = 0.5$.

Error Measures	Dataset			
	No Noise	Gaussian	Poisson	Rician
Average	0.4	0.4	0.4	0.3
Maximum	2.1	1.9	2.1	1.5

3.5 *Stability of Centerline*

It is important to understand the stability of the centerline is with respect to initialization and segmentation parameters. In our algorithm, the initial parameters include the threshold used to generate the rough segmentation, noise deviation and for Rician noise, the mean of the distribution. This experiment was performed on the head MRT dataset.

Table 5 illustrates error measures related to varying the initial segmentation threshold from 60 to 90 in steps of 10 units of the scalar field value. We then computed the distance between these centerlines (one pair at a time), which in turn was done by computing the closest distance between corresponding voxels. The results were averaged to determine the average error. The maximum of these distances was reported as the maximum error. Average error ranged from 0.3-0.4 voxels and the maximum error from 1.5-2.1 voxels.

4 Discussion

The primary goal of this work was to extract centerlines of tubular structures in a robust and accurate fashion, without assumptions of knowledge of exact boundaries. Complex datasets such as those used in this work and archived at [24] are considerably challenging to traditional distance field algorithms, which assume a binary (usually thresholded) dataset and hence zero distance values on the boundary.

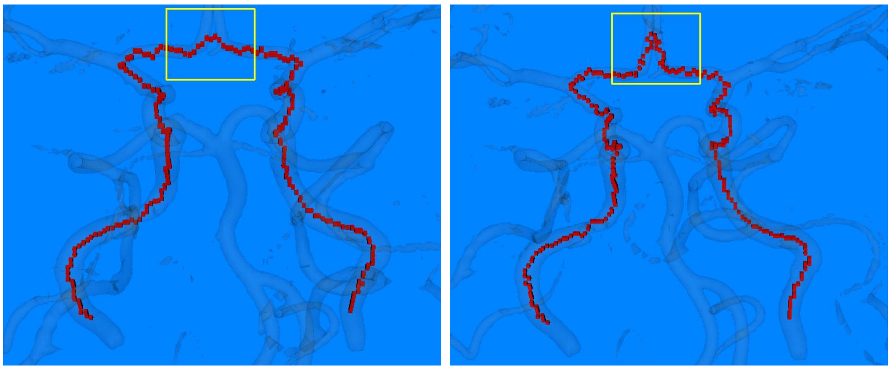


Fig. 13 Effect of Change in Scale. The region marked by the yellow rectangle has two blood vessels very close to each other. *Left:* Using gaussian smoothing with $\sigma = 1.0$ results in neighboring voxels from the two vessels, and the centerline is henceforth joined, *Right:* $\sigma = 0.5$, produces the correct result.

Using a probabilistic model to estimate the boundary encodes its uncertainty, and thus leads to a centerline algorithm that is less sensitive to errors and artifacts that distort the boundary. Moreover, this approach permits us to work directly with the grayscale images and their properties; here we use the gradient field to determine the boundary location by posing this as a search for a local maximum. Existing distance field based methods assume the boundary is easily available, but this implies (at least for medical images) an accurate segmentation, a significant task in of itself. Our approach has a weaker assumption, i.e., the anatomical structures of interest can be roughly separated from the rest of the volume. This is comparable to methods such as [15, 17, 18].

We further explore the robustness of our method by explicitly adding noise that is common in medical images: Gaussian, Poisson and Rician, and demonstrate both qualitatively and quantitatively the strength of this approach over existing distance field based approaches. Here again, exploiting image characteristics produces better results. Traditional distance field based approaches will need to have a robust segmentation method to deal with the image noise, prior to extracting the centerline. This will significantly add to the computation. Otherwise, they tend to produce poor results, as illustrated in our experiments using thresholding to segment the structures in the noisy images.

Approaches that use ridges [17, 18] using eigen value analysis do produce excellent results, but there are two issues, (1) significantly increased computation to determine Hessian and eigen values, and (2) for datasets such as a colon there are generally significant numbers of interior voxels with little or no gradient (large homogeneous regions).

Our approach is efficient, as it requires only the gradient field to be computed for estimating the boundary. Often this has to be computed anyhow, since visualization algorithms require them to compute surface normals. The most expensive part of our method is the distance field computation, but here again, efficient algorithms exist[38].

Finally, some aspects of the centerline extraction algorithm are incomplete, for instance, incorporation of variable scale to adapt to significant changes in object geometry, or, in dealing with extremely small tubular structures. An example is shown in Fig. 13. In the left image, the MRT dataset was filtered by a gaussian with $\sigma = 1.0$. This effectively joins the two parallel blood vessels in the region shown (marked by the yellow rectangle). In the right image, we have reduced the scale to $\sigma = 0.5$ to obtain the correct centerline. In this particular example, the region of interest was adaptively smoothed with the smaller gaussian kernel to illustrate the right result. We are currently looking into ways of efficiently detecting such regions as part of our algorithm, so as to use the appropriate scale based on local image characteristics.

5 Conclusions

We have presented a robust and accurate centerline extraction algorithm that can work directly with gray scale images and a rough segmentation of the structures of interest. The goal of this work was toward analyzing large medical structures. 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. Experiments on both synthetic and clinical datasets illustrate the strength of the proposed method. We have tested our approach for accuracy on both synthetic volume models and segmented medical datasets. Additional experiments were performed on six publicly available medical datasets. Noise tolerance of the method was demonstrated by adding significant amounts of Gaussian, Poisson and Rician noise to the volume data. Stability of the centerline was evaluated with respect to changes in input segmentation parameters. All of our experiments illustrate the strength of this method, in comparison to existing distance field based approaches to centerline extraction.

References

1. Cornea, N.D., Silver, D.: Curve-Skeleton Applications. In: Proceedings of IEEE Visualization 2005, pp. 95–102 (2005)
2. Wan, M., Liang, Z., Bitter, I., Kaufman, A.E.: Automatic Centerline Extraction for Virtual Colonoscopy. IEEE Transactions on Medical Imaging 21(12), 1450–1460 (2002)
3. Gravel, P., Beaudoin, G., De Guise, J.A.: A Method for Modeling Noise in Medical Images. IEEE Transactions on Medical Imaging 23(10), 1450–1460 (2004)
4. Gibson, S.F.F.: Using Distance Maps for Accurate Representation in Sampled Volumes. In: Proceedings of the 1998 IEEE Symposium on Volume Visualization, pp. 23–30 (1998)

5. Gagvani, N., Gagvani, D.: Parameter Controlled Skeletonization of Three Dimensional Objects. In: CAIP-TR-216. Dept. of Electrical and Computer Engineering and CAIP-Center, Rutgers University (1997)
6. Zhou, Y., Toga, A.W.: Efficient Skeletonization of Volumetric Objects. IEEE Transactions on Visualization and Computer Graphics 5(3), 196–209 (1999)
7. Borgefors, G.: Distance Transformations in Digital Images. Computer Vision and Image Understanding 34, 344–371 (1986)
8. Chen, D., Li, B., Liang, Z., Wan, M., Kaufman, A.E., Wax, M.: A tree-branch searching multiresolution approach to skeleteonization for virtual endoscopy. In: SPIE Medical Imaging, vol. 3979, pp. 726–1002 (2000)
9. Bitter, I., Kaufman, A.E., Sato, M.: Penalized-Distance Volumetri Skeleton Algorithm. IEEE Transactions on Visualization and Computer Graphics 7(3), 195–206 (2002)
10. Bitter, I., Sato, M., Bender, M., McDonnel, K., Kaufman, A.E., Wan, M.: CEASER: A smooth, accurate and robust centerline-extraction algorithm. In: Proceedings of IEEE Visualization 2000, pp. 45–52 (2000)
11. Koenderink, J.J.: The Structure of Images. Biological Cybernetics 50, 363–370 (1984)
12. Florack, L.M., ter Haar Romeny, B.M., Koenderink, J.J., Viergever, M.A.: Scale and the Differential Structure of Images. Image and Vision Computing 10(6), 376–388 (1992)
13. Lindberg, T.: Feature Detection with automatic scale selection. Intrernational Journal of Computer Vision 30(2), 79–116 (2002)
14. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale Vessel Enhancement Filtering. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 130–137. Springer, Heidelberg (1998)
15. Aylward, S., Bullitt, E.: Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. IEEE Transactions on Medical Imaging 21(2), 61–75 (2002)
16. Wink, O., Niessen, W.J., Viergever, M.A.: Multiscale Vessel Tracking. IEEE Transactions on Medical Imaging 23(1), 130–133 (2004)
17. Eberly, D., Gardiner, R., Morse, B., Pizer, S., Scharlach, C.: Ridges for Image Analysis. Journal of Mathematical Imaging and Vision 4, 351–371 (1994)
18. Eberly, D.: Ridges in Image and Data Analysis (Computational Imaging and Vision). Springer, Heidelberg (1996)
19. Krissian, K., Malandain, G., Ayache, N., Vaillant, R., Trousset, Y.: Model Based Detection of Tubular Structures in 3D Images. In: 3736. INRIA, Sophia Antipolis, France (1999)
20. Chuang, J., Tsai, C., Kuo, M.: Skeletonization of Three-Dimensional Object Using Generalized Potential Field. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11), 1241–1251 (2000)
21. Cornea, N.D., Silver, D., Yuan, X., Balasubramanian, R.: Computing Hierarchical Curve-Skeletons of 3D Objects. Visual Computer 21(11), 945–955 (2005)
22. Ma, W.-C., Wu, F.-C., Ouhyoung, M.: Skeleton Extraction of 3D Objects with Radial Basis Functions. In: Proceedings of IEEE Shape Modeling 2003, pp. 207–215 (2003)
23. Ma, C.M.: A Fully Parallel 3D Thinning Algorithm and its Applications. Computer Vision and Image Understanding 64(3), 420–433 (1996)
24. Bartz, D.: `http://www.gris.uni-tuebingen.de/areas/scivis/volren/datasets/new.html`
25. Kindlmann, G.: Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. In: Proceedings of the 1998 IEEE Symposium on Volume Visualization, pp. 79–86 (1998)

26. Yoo, T.: Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis. A.K. Peters (2004)
27. Schroeder, W., Martin, K., Lorensen, B.: The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, 3rd edn. Prentice Hall Inc., Englewood Cliffs (2002)
28. Spitzak, B.: The Fast Light Toolkit, `http://www.fltk.org`
29. Gravel, P., Beaudoin, G., De Guise, J.A.: A Method for Modeling Noise in Medical Images. IEEE Transactions on Medical Imaging 23(10), 1221–1232 (2004)
30. Welch, G., Blshop, G.: An Introduction to the Kalman Filter. In: 95041. Dept. of Computer Science, University of North Carolina at Chapel Hill (2001)
31. Agner Fog, `http://www.agner.org/random`
32. Sijbers, J., Den Dekker, A.J., Van Audekerke, J., Verhoye, M., Van Dyck, D.: Estimation of the noise in magnitude MR images. Magnetic Resonance Imaging 16(1), 87–90 (1998)
33. Henkelman, P.M.: Measurement of signal intensities in the presence of noise in MR images. Medical Physics 173, 232–233 (1985)
34. Gudbjartsson, H., Patz, S.: The Rician distribution of noisy MRI data. Magnetic Resonance in Medicine 34, 910–914 (1995)
35. Papoulis, A.: Probability, Random Variables and Stochastic Processes. McGraw-Hill, New York (1984)
36. Lorensen, W.E., Cline, H.E.: Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm. Computer Graphics 21(4), 163–169 (1987)
37. Choi, R.: `http://nova.nlm.nih.gov`
38. Saito, T., Toriwaki, J.I.: New algorithms for Euclidean distance transformations of an N-dimensional digitised picture with applications. Pattern Recognition 27(11), 1551–1565 (1994)