Available online at www.sciencedirect.com

SCIENCE DIRECT®

Medical Image Analysis 9 (2005) 209–221

MEDICAL IMAGE ANALYSIS

www.elsevier.com/locate/media

Flux driven automatic centerline extraction

Sylvain Bouix [a,b,*], Kaleem Siddiqi [c], Allen Tannenbaum [d]

[a] Department of Psychiatry, Boston VA Healthcare System, Harvard Medical School, Boston, USA
[b] Surgical Planning Laboratory, MRI Division, Brigham and Women's Hospital, Harvard Medical School, USA
[c] Center for Intelligent Machines and School of Computer Science, McGill University, Montreal, Canada
[d] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA

Received 25 January 2004; received in revised form 9 June 2004; accepted 21 June 2004
Available online 18 October 2004

Abstract

We present a fast, robust and automatic method for computing centerline paths through tubular structures for application to virtual endoscopy. The key idea is to utilize a skeletonization algorithm which exploits properties of the average outward flux of the gradient vector field of a Euclidean distance function from the boundary of the structure. The algorithm is modified to yield a collection of 3D curves, each of which is locally centered. The approach requires no user interaction, is virtually parameter free and has low computational complexity. We validate the method quantitatively on a number of synthetic data sets with known centerlines and qualitatively on colon and vessel data segmented from CT and CRA images.

Keywords: Virtual endoscopy; Centerlines; Skeletons; Medial surfaces; Medial curves; Validation

1. Introduction

With the availability of high resolution CT and MR images of tubular structures such as the bronchial tree, the gastrointestinal tract, blood vessels and arteries, there is an increasing interest in the field of virtual endoscopy (Jolesz et al., 1997). The main motivation is to complement the often painful and costly invasive procedure of endoscopy, which involves navigating through such structures in a patient with an endoscope, in order to generate views of internal surfaces for diagnostic purposes and surgical planning. Recent advances include the use of an interactive segmentation tool assisted by haptics for improved centerline extraction (Harders et al., 2002) and the use of cubic panoramas for visualization (Tiede et al., 2002). Virtual endoscopy typically involves three steps: (1) the medical images are processed in order to segment the surfaces of structures of interest, (2) flight paths are generated within these structures in order to obtain potential camera viewing positions and (3) given the flight paths, perspective views of the internal surfaces of such structures are generated using conventional graphics rendering techniques.

There has been a great deal of work in the computer vision and medical imaging literature on the development of algorithms for surface segmentation. Whereas research on segmentation remains active, a number of such algorithms in the literature, e.g., those based on deformable models (Kass et al., 1987; Malladi et al., 1995; Kichenassamy et al., 1995; Caselles et al., 1995) work well on hollow structures because they typically have high contrast boundaries. [1] Similarly, the third step in virtual endoscopy, the rendering of views, is essentially a graphics problem for which a number of software packages such as the Visualization Tool Kit

* Corresponding author.
E-mail addresses: sylvain@bwh.harvard.edu (S. Bouix), siddiqi@cim.mcgill.ca (K. Siddiqi), tannenba@ece.gatech.edu (A. Tannenbaum).

[1] There are of course important exceptions, such as very narrow blood vessels viewed in MR angiography.

1361-8415/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.media.2004.06.026

(VTK) provide standard routines that can be used. However, the second step, that of generating flight paths automatically or semi-automatically, has received relatively less attention. The main requirement for an endoscopic flight path is that it should provide an interior view of the object such that local anatomical structures are clearly visible. Intuitively, this criterion can be satisfied by centering the flight path inside the structure of interest. In the absence of prior information about the data set being processed, centering the path locally is an appropriate strategy since the walls of the structure are then equidistant from the path.

The development of an automatic centerline extraction algorithm is the focus of the current paper. Our main goal is to design an algorithm which is robust and accurate and requires no human interaction. Moreover, we would also like the method to be computationally efficient. Surface segmentation methods for tubular structures based on deformable models are usually fast and the segmented surfaces can be rendered in real time with consumer grade graphics cards. The availability of an efficient algorithm for extracting fly throughs could thus allow a clinician to visualize such data within a few minutes of its acquisition.

We begin with a brief review of approaches to finding centerline paths in tubular structures in the literature. This review is necessarily not exhaustive. It is based on a selection of representative methods and includes a comparison of their features.

A first class of methods attempts to find centerlines of tubular structures as they are manifest directly in intensity (MR or CT) images. These methods do not assume that the surfaces of such structures have first been extracted. Three representative approaches in this category are those of Aylward and Bullitt (2002), Deschamps and Cohen (2001), and Wink et al. (2004). Aylward and Bullit present a centerline tracking approach which is based on a characterization of intensity ridges in 2D and 3D images. Centerlines of tubular structures are identified using properties of the Hessian matrix. They pay particular attention to the validation of their method, both quantitative and qualitative, demonstrating its robustness under parameter changes, changes in scale and simulated image acquisition noise. The method is an iterative one, where the centerline is continuously extended in the estimated direction of its local orientation.

Deschamps and Cohen relate the problem of finding centerline paths to that of finding paths of least action in 3D intensity images (Deschamps and Cohen, 2001). This leads to a form of the well-known eikonal equation where a front is propagated in the image with a speed determined by a scalar potential that depends upon location in the medium. The framework aims to infer the boundaries of tubular structures in a first stage, using a standard surface evolution method. The potential function is then designed to take into account a Euclidean distance function from the boundary, so that the minimal paths are centered. Beyond the requirement that the user must specify the starting and end points of a particular path, the algorithm requires little user interaction. The flow is implemented using fast marching schemes and is hence computationally efficient.

Wink et al. have recently presented an approach to centerline extraction, applied in the context of vessel tracking, which combines features of the above two approaches. More specifically, they use the vesselness measure proposed by Frangi et al. (1998), which is based on properties of the Hessian matrix, to characterize putative vessel centerline locations. They then formulate the problem of finding paths between user selected points as a minimum cost path problem which they solve computationally using wavefront propagation. Their method has been validated qualitatively in the presence of stenoses and imaging artifacts.

A second class of methods aims to find centerlines of tubular structures which have first been segmented from 3D MR or CT intensity images. Representative methods in this category are the approaches of (Bitter et al., 2001; Paik et al., 1998; Ge et al., 1999). The method of Bitter et al. (2001) is related to Deschamps and Cohen's minimal path approach, but is formulated in a discrete setting. A graph is first built from a coarse approximation of a 3D skeleton. Each edge of the graph is assigned a weight which is a combination of Euclidean distance from a user defined source node and Euclidean distance from the boundary of the object. A centerline is then extracted using Dijkstra's shortest path algorithm on this graph.

Paik et al. (1998) suggest a different approach where a minimal path is found on the surface of a tubular structure and is later centered. The user chooses the two end points of a desired path. The voxels on the surface of the object closest to each of the user specified points are then obtained. A path on the surface which minimizes the distance between these two points is then found (a type of geodesic). This path is then centered in the structure by applying a modified form of an iterative thinning procedure.

Lastly, an approach which is most closely related to the one presented in the current paper, but is specialized to the case of objects with simple topology such as a colon, is that of Ge et al. (1999). First, a 3D skeleton of the binary volume is generated using a fast topological thinning algorithm. Then loops and spurious branches are pruned from the skeleton using graph search techniques. The result is then modified to yield a centered path between two user specified end points. In this paper, we present an approach to finding centerline paths that is based on a recent skeletonization algorithm we have developed (Siddiqi et al., 2002). The algorithm utilizes an average outward flux measure to distinguish skeletal points from non-skeletal ones and combines this measure with a topology preserving thinning procedure. We present an overview of the skeletonization algorithm in Section 2 and then modify it to find centerline paths in tubular structures of arbitrary topology in Section 3. We then carry out a quantitative validation of the method on a variety of synthetic data sets with known centerlines, as well as illustrate its performance on more complex data including colons and vasculature in Section 4.

The advantages of our framework may be enumerated as follows:

(1) The approach is based on an algorithm whose theoretical properties have been thoroughly justified (Siddiqi et al., 2002; Damon, 2003). Other methods in the literature which use topological thinning often use heuristics for the selection of anchor points.
(2) The algorithms developed here lend themselves to the use of quite standard numerical implementations. Furthermore, the framework finds all centerline paths in volumetric tubular structures having arbitrary topology. Most other methods are designed to find a single centerline path at a time and have numerical implementations that are more complex.
(3) The approach is fully automatic and requires no user interaction. Several of the other methods do require the user to select at least the end points of a particular centerline path.
(4) The method has been validated on synthetic data sets with known centerlines, and has shown to provide results very close to the ground truth. To our knowledge most methods which have been developed for segmented tubular structures have not been validated quantitatively using ground truth centerlines.
(5) The method has a theoretical bound on its worst case complexity which is O($n \log n$), where n is the number of voxels in the interior of the structure. However, our experiments indicate that in practice its complexity is typically linear in n. Hence, the approach is computationally very efficient. For several of the other methods explicit computational complexity estimates are difficult to obtain.

2. Average outward flux based medial surfaces

This section presents an overview of the skeletonization algorithm introduced by Siddiqi et al. (2002). Beginning with a binary volume this algorithm produces as its output a digitized version of the 3D skeleton or medial axis, which is defined to be the locus of centers of maximal inscribed spheres. The term axis is somewhat misleading because in 3D most points of this structure lie on 2D manifolds. The local geometry of such points is illustrated in Fig. 1. Hence, in what follows we shall refer to the 3D skeleton as the *medial surface*.

2.1. The Hamilton–Jacobi formulation

Consider Blum's grassfire flow,

$$\frac{\partial S}{\partial t} = \hat{\mathbf{N}}, \tag{1}$$

acting on a closed 3D surface S, such that each point on its boundary is moving with unit speed in the direction of the inward normal $\hat{\mathbf{N}}$. In physics, such equations are typically solved by looking at the evolution of the phase space of an equivalent Hamiltonian system. Let D be the Euclidean distance function to the initial surface $\mathscr{S}_0$. The magnitude of its gradient, $\|\nabla D\|$, is identical to 1 in its smooth regime. With $\mathbf{q} = (x,y,z)$, $\mathbf{p} = (D_x,D_y,D_z)$ and $\|\mathbf{p}\| = 1$, the Hamiltonian system is given by

$$\dot{\mathbf{q}} = (0,0,0), \quad \dot{\mathbf{q}} = (D_x, D_y, D_z), \tag{2}$$

with an associated Hamiltonian function $H = 1 + \|\nabla D\|$. The discrimination of medial from non-medial surface points can be approached by computing the "average outward flux" of the vector field $\dot{\mathbf{q}}$ at a point. This quantity is given by

$$\mathrm{AOF}(\dot{\mathbf{q}}) := \frac{\int_{\delta R} \langle \dot{\mathbf{q}}, \hat{\mathbf{N}}_o \rangle \, \mathrm{d}S}{\mathrm{area}(\delta R)}, \tag{3}$$

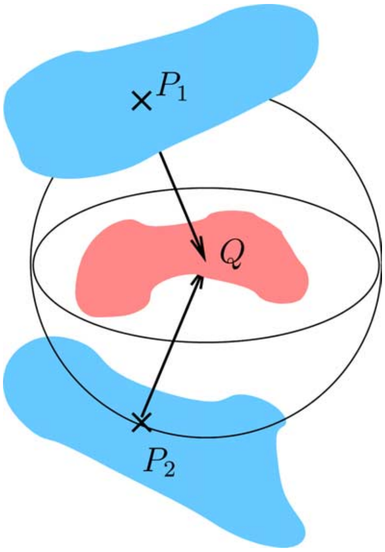

Fig. 1. A medial manifold is shown in pink and the two surface patches to which it corresponds are shown in blue. Each point Q on the medial manifold is associated with two distinct points P_1, P_2 on the object's surface to which it is closest in the sense of Euclidean distance. The object angle α is half of the angle between the vectors $\vec{P_1Q}$ and $\vec{P_2Q}$ in the plane passing through P_1, P_2 and Q. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where dS is a surface area element of the bounding surface δR of a volume R and $\hat{\mathbf{N}}_o$ is the outward normal at each point on the surface. It can be shown that as a spherical volume shrinks to a point not on the medial surface, the average outward flux through its surface approaches zero. In contrast, when such a volume shrinks to a medial surface point, the average outward flux approaches a strictly negative number (Siddiqi et al., 2002; Dimitrov et al., 2003). There is in fact a very precise relationship between the limiting values so obtained and the geometry of the object (Dimitrov et al., 2003). The average outward flux values are proportional to the *sine* of the object angle α, which is half of the angle between the vectors $\vec{P_1Q}$ and $\vec{P_2Q}$ in the plane passing through P_1, P_2 and Q (see Fig. 1). Since these two vectors are perpendicular to the tangent planes to the boundary at P_1 and P_2, respectively, the average outward flux measures the degree to which the medial manifold is parallel (locally) to the corresponding boundaries. More general results on medial integrals have also recently been reported in (Damon, 2003). Thus, the average outward flux measure is an effective way to distinguish between medial and non-medial points. This calculation is discretized, as described in Algorithm 1 (Borgefors, 1984; Siddiqi et al., 1997), and is used to guide a thinning process in a cubic lattice, while taking care to preserve the object's topology.

Algorithm 1.

```
Average Outward Flux.

Compute the distance transform $D$ of the object (Borgefors, 1984) ;
Compute the gradient vector field $\nabla D$ (Siddiqi et al., 1997) ;
Compute the average outward flux of $\nabla D$ using Eq. 3;
for (each cubic lattice point x in the interior of the object) do
    $\mathrm{AOF}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{26} < \hat{\mathbf{N}}_i, \nabla D(\mathbf{x}_i) >$;
    (where $\mathbf{x}_i$ is a 26-neighbor of $\mathbf{x}$ and $\hat{\mathbf{N}}_i$ is the outward normal at $\mathbf{x}_i$ of the
    unit sphere centered at $\mathbf{x}$)
```

2.2. Preserving topology

We refer the reader to the survey paper of Kong and Rosenfeld (1989) for a review of concepts in digital topology. A cubic lattice point is a labeled a *simple* point if its removal does not change the topology of an object. Such a point has the property that its removal: (i) does not create a hole, (ii) does not create a cavity and (iii) does not disconnect a connected component. The classification of a simple point turns out to be one case of a more general categorization of a point $\mathbf{x}$ in a cubic lattice introduced by Malandain et al. (1993). This categorization is based on two numbers:

- C^*: the number of 26-connected components 26-adjacent to $\mathbf{x}$ in $\mathrm{O} \cap \mathrm{N}^*_{26}$,
- $\bar{C}$: the number of 6-connected components 6-adjacent to $\mathbf{x}$ in $\bar{\mathrm{O}} \cap \mathrm{N}_{18}$.

Here O is a 26-connected object, N^*_{26} is the 26-neighborhood of $\mathbf{x}$ without $\mathbf{x}$ and N_{18} is the 18-neighborhood of $\mathbf{x}$ including $\mathbf{x}$. Simple (and hence removable) points can be identified by the conditions $C^* = 1$ and $\bar{C} = 1$.

The basic strategy now is to guide the thinning of the object by the average outward flux measure computed over a very small spherical neighborhood, while taking care to only remove simple points. The thinning process terminates when all surviving points are not simple, or have an average outward flux below some chosen (negative) value, or both. Unfortunately the result is not guaranteed to be a thin set, i.e., one without an interior. This last constraint can be satisfied by defining an appropriate notion of an end point in a cubic lattice. In $\mathbb{R}^3$, if there exists a plane that passes through a point $\mathbf{x}$ such that the intersection of the plane with the object includes an open curve which ends at $\mathbf{x}$, then $\mathbf{x}$ is an end point of a 3D curve, or is on the rim or corner of a 3D surface. This criterion can be discretized easily to 26-connected digital objects by examining 9 digital planes in the 26-neighborhood of $\mathbf{x}$ (Pudney, 1998). The thinning process proceeds as before, but the threshold criterion for removal is applied only to end points, as described in Algorithm 2.

Algorithm 2.

```
Topology Preserving Thinning: Medial Surface.

for (each cubic lattice point x on the boundary of the object) do
    if (x is simple) then
        insert(x, maxHeap) with AOF(x) as the sorting key for insertion;
while (maxHeap.size > 0) do
    x = HeapExtractMax(maxHeap);
    if (x is simple) then
        if (x is an end point) and (AOF(x) < Thresh) then
            mark x as a medial surface (end) point;
        else
            Remove x;
            for (all neighbors y of x) do
                if (y is simple) then
                    insert(y, maxHeap) with AOF(y) as the sorting key for in-
                    sertion;
```

3. Extracting centerline paths

The medial surface of a cylindrical structure approaches a 3D curve centered in the object's interior. Using the framework we have just described we shall now develop algorithms to compute centerline paths in single tubular 3D structures (such as colons) as well as more complex objects composed of segments that are tubular (such as vessels). Our strategy is to thin the medial surface to obtain a structure composed only of curves (the medial curve) and to then prune the result to obtain well centered paths. A related though distinct algorithm for obtaining medial curves from digital manifolds was developed in (Borgefors et al., 1998). We begin by describing the process to obtain the medial curve from a medial surface for an object with one connected component. [2]

3.1. The medial curve

The medial surface extraction algorithm (Algorithm 2) can be modified to produce a medial curve. Recall that during the thinning procedure a point **x** that is an end point of a 3D curve or is on the rim or corner of a 3D surface will not be removed if its average outward flux is strongly negative. In a cubic lattice, an end point of a 26-connected curve is defined as a point **x** whose 26-neighborhood $O \cap N^*_{26}$ only contains one point that is in the interior of the object. If the criterion for anchoring points is applied only to end points, every surface point will eventually be removed leading to a medial curve. Whereas this curve will lie on the medial surface, it will not necessarily be locally centered within a tubular object when the thinning is ordered by the average outward flux. This is essentially because this measure does not encode the actual distance to the boundary, but rather the angle between the vectors $\vec{P_1Q}$ and $\vec{P_2Q}$ (see Fig. 1) as discussed earlier in Section 2.1. A straightforward way to obtain centerline paths is to order the thinning by the Euclidean distance function to the object's boundary, such that points which survive are as far away from the boundary as possible. This approach is similar in spirit to the use of a Euclidean distance function in (Paik et al., 1998; Deschamps and Cohen, 2001).

Algorithm 3 presents the thinning procedure to extract the medial curve. Two steps have been modified to Algorithm 2: the sorting key during the insertion of new points in the heap and the end point criteria.

The resulting medial curve is a 26-connected connected structure in a cubic lattice which is no more than one voxel thick (except possibly at branch points). Thus, each point on a medial curve can be labeled by examining its 26-neighborhood: (i) end points have only one neighbor, (ii) curve points have exactly two neighbors and (iii) junction points have at least three neighbors.

Algorithm 3.

Topology Preserving Thinning: Medial Curve.

```
for (each cubic lattice point x on the boundary of the object) do
    if (x is simple) then
        insert(x, maxHeap) with $-D(\mathbf{x})$ as the sorting key for insertion;
        ($D$ is the Euclidean distance function (Borgefors, 1984).)
while (maxHeap.size > 0) do
    x = HeapExtractMax(maxHeap);
    if (x is simple) then
        if (x is an end point of a 3D curve) and (AOF(x) < Thresh) then
            mark x as a medial curve (end) point;
        else
            Remove x;
            for (all neighbors y of x) do
                if (y is simple) then
                    insert(y, maxHeap) with $-D(\mathbf{y})$ as the sorting key for insertion;
```

3.2. Pruning

Once a centered medial curve is obtained some degree of pruning has to be carried out, depending on the type of output desired. If the goal is to extract a single centerline path, as is the case for the colon, one can use Dijkstra's single source shortest path algorithm (Dijkstra, 1959). Here the idea is to compute the shortest path from every end point on the medial curve to every other end point and to then pick the longest shortest path as the centerline path. This procedure is described in Algorithm 4.

Algorithm 4.

A single centerline path.

```
Label the medial curve as end points, curve points and branch points;
max = $-\infty$;
for (all end points) do
    Compute the single source shortest path of the medial curve from the end point;
    Find the longest shortest path $c$;
    if (length(c) > max) then
        max = length($c$);
        centerline = $c$;
```

On the other hand, if the goal is to extract several centerline paths, as is the case for blood vessels or airways, one can prune by removing all branches connected to end points whose length is lower than some threshold. A reasonable choice for this threshold is the minimum expected diameter of the tubular structure. Our numerical experiments indicate that the results are not sensitive to the precise value of this parameter because arteries and airways are typically much longer than they are wide. The details are presented in Algorithm 5.

[2] For objects with more than one connected component the same process can be applied to each component in turn.

Algorithm 5.

```
Several centerline paths.

repeat
    Label the medial curve as end points, curve points and branch points;
    Compute the length of branches connected to end points;
    for all such branches do
        if $length(branch) < Thresh$ then
            remove branch from medial curve;
until all branches have length $> Thresh$;
```

3.2.1. Summary of algorithms

The use of the algorithms presented thus far can be summarized as follows. To extract a single centerline path through a data set, one applies Algorithms 1, 3 and 4. This provides a 26-connected digital curve between two endpoints. Examples of this type are presented in Fig. 2. To extract centerline paths for structures with more complex topology, one applies Algorithms 1, 3 and 5. This provides a set of 26-connected digital curves, each connecting an end point to a branch point or two distinct branch points. Examples of this type are presented in Fig. 4. Both types of structures can be represented as a *centerline graph* where the end points and branch points comprise the nodes and the digital curves connecting any two nodes the edges.

3.3. Fly throughs

It is in fact advantageous to view the output data as a centerline graph since this leads to a simple way to generate fly through movies. A depth first search (DFS) of the centerline graph provides a path which visits a tubular structure of complex topology in its entirety and hence can be used to guide a virtual camera. In our implementation we choose to smooth the DFS path with a smoothing spline using Matlab. The spline $\mathbf{s}(t)$ minimizes,

$$p\sum_i w_i(\mathbf{x}_i - \mathbf{s}(t_i))^2 + (1-p)\int \left(\frac{\mathrm{d}^2\mathbf{s}}{\mathrm{d}t^2}\right)^2 \mathrm{d}t,$$

a functional that is a convex combination of two terms; the first is a closeness to data term and the second is a smoothness term. Here w_i is the weight of each data point $\mathbf{x}_i$ and p is in the interval [0,1]. A value of $p = 0$ produces a least squares straight line fit to the data, while a value of $p = 1$ produces a cubic spline interpolant. For the experiments in Section 4, we set w_i to be 1 and p to be 0.4. The smoothing of the DFS path and not the centerline is an important consideration if one wishes to obtain smooth fly throughs, particularly at junction points. A junction point where n branches meet appears n distinct times in the DFS path. Smoothing according to one incoming and one outgoing branch for each instance allows for smoother turns at the junction.

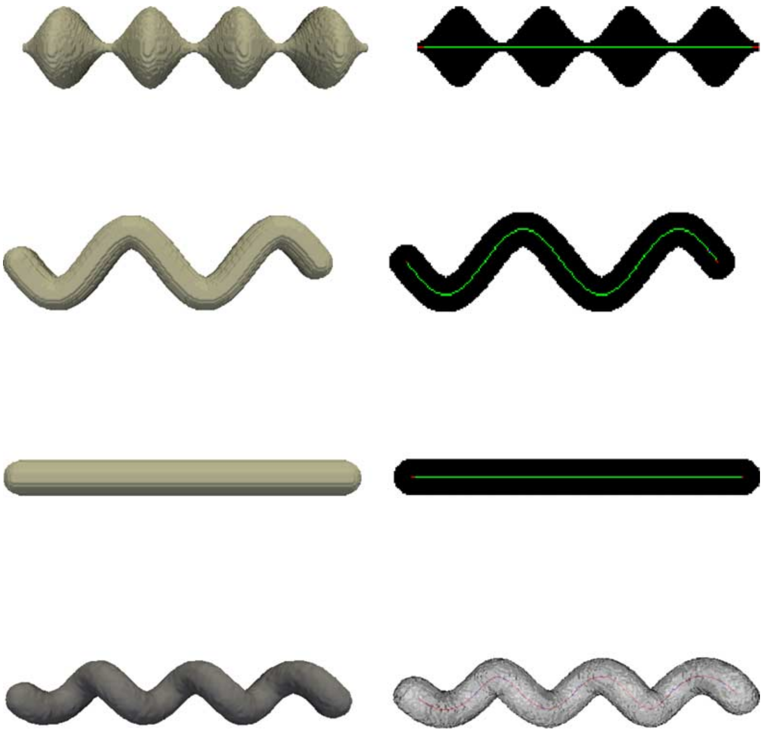

Fig. 2. Left: Synthetic objects with centerlines of fixed shape. Right: The corresponding ground truth and computed centerlines. For each object (black), the overlapping points between the ground truth centerline and the computed one are shown in green, the points belonging to the ground truth centerline but not the computed one are shown in red and the points belonging to the computed centerline but not the ground truth centerline are shown in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In order to create fly through movies we wrote a simple visualization program using standard routines available in the Visualization Tool Kit (VTK). Our strategy was to create perspective views, similar to those used in the work of Deschamps and Cohen (2001). The position and orientation of the camera was determined by selecting an initial reference frame with the camera's optical axis aligned with the tangent to the DFS path and then moving the frame by following the tangent direction. In the bottom row of Figs. 6 and 7, the left image represents a global view of the object with a red ball indicating the current position of the camera, and the right image the corresponding perspective view of internal surfaces. It is important to point out that the DFS path can also be used for more elaborate volume rendering in the context of virtual endoscopy including fly throughs where the reference frame is aligned with the local frenet frame, thus taking torsion into account, or fly throughs with panoramic views, as proposed by Tiede et al. (2002).

4. Experiments

An important consideration in the evaluation of a centerline extraction algorithm is its validation. Approaches which work directly on intensity data have carried out qualitative or quantitative validation studies in the presence of imaging noise and other artifacts (Aylward and Bullitt, 2002; Wink et al., 2004). For methods which have been designed to work on segmented tubular structures, such as ours, quantitative validation on medical images is a challenge because a ground truth centerline is typically not defined. Thus, we have chosen to evaluate our approach quantitatively by creating synthetic tubular structures with known centerlines. A qualitative evaluation is then carried out on medical data sets of varying complexity.

4.1. Quantitative validation on synthetic data sets

We created several ground truth centerlines by either using a fixed shape or by drawing them by hand. For each centerline we then centered a sphere at one location and then translated the sphere along it. The volume swept by the moving sphere thus defined a synthetic tubular structure with a known ground truth centerline. The specific examples we created were:

- *Tube:* A straight centerline with a sphere of constant radius.
- *Sine tube:* A sinusoidal centerline with a sphere of constant radius designed to test the robustness of the method to variations in curvature.

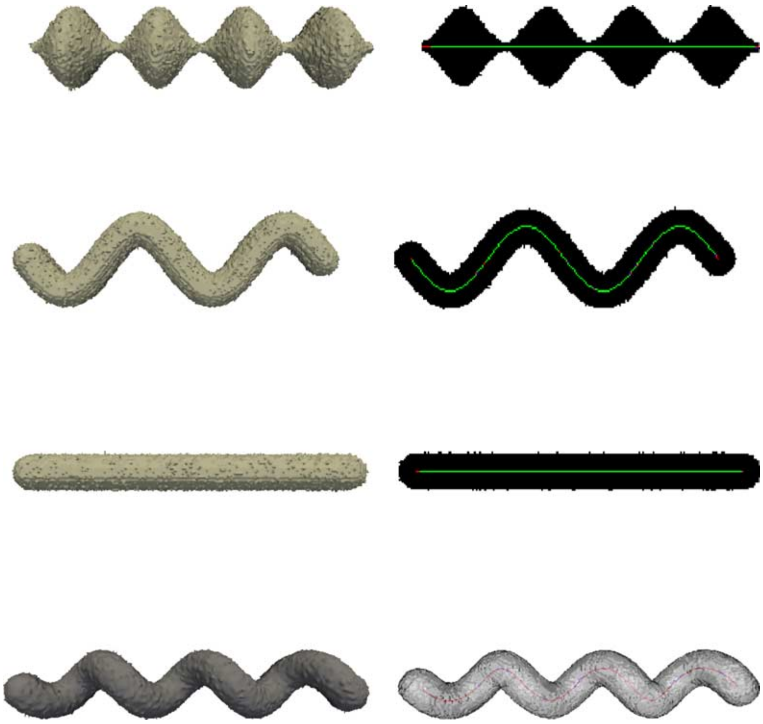

Fig. 3. Left: Synthetic objects with centerlines of fixed shape. Boundary noise was added randomly. Right: The corresponding ground truth and computed centerlines. For each object (black), the overlapping points between the ground truth centerline and the computed one are shown in green, the points belonging to the ground truth centerline but not the computed one are shown in red and the points belonging to the computed centerline but not the ground truth centerline are shown in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- *Helix:* A helical centerline with a sphere of constant radius designed to test the method when both curvature and torsion are present in the anatomical structure.
- *Sausage:* A straight centerline with a sphere of varying radius (a sine function) designed to mimic the highly variable cross section of a colon or the presence of aneurysms in vascular data sets.
- *Tree:* A hand drawn tree-like centerline with a sphere of constant radius designed to test the ability of the method to handle branching structures.
- *Complex:* A hand drawn centerline of complex topology with a sphere of constant radius designed to test the accuracy of the method when both branching structures and loops are present.

All these objects have the property that the restriction of their medial surface to medial curves comprised of locations with maximal distance to the boundary coincides approximately with the ground truth centerlines.

Fig. 2 shows the different objects with fixed centerlines (left) and the results (right) of applying Algorithms 1, 3 and 4. For each object (black), the overlapping points between the ground truth centerline and the computed one are shown in green; the points belonging to the ground truth centerline but not the computed one are shown in red and the points belonging to the computed centerline but not the ground truth centerline are shown in blue. In Fig. 3 (left), some points were randomly added to the boundary of these objects to simulate acquisition noise or segmentation errors. The computed centerlines are shown in Fig. 3 (right) with the same color coding. Figs. 4 and 5 present the results of applying Algorithms 1, 3 and 5 to the original and noisy versions of the objects with hand drawn centerlines, again with the same color coding. These objects have considerably more complex geometry and topology.

The figures indicate that the overlap between the ground truth and computed centerlines is very strong

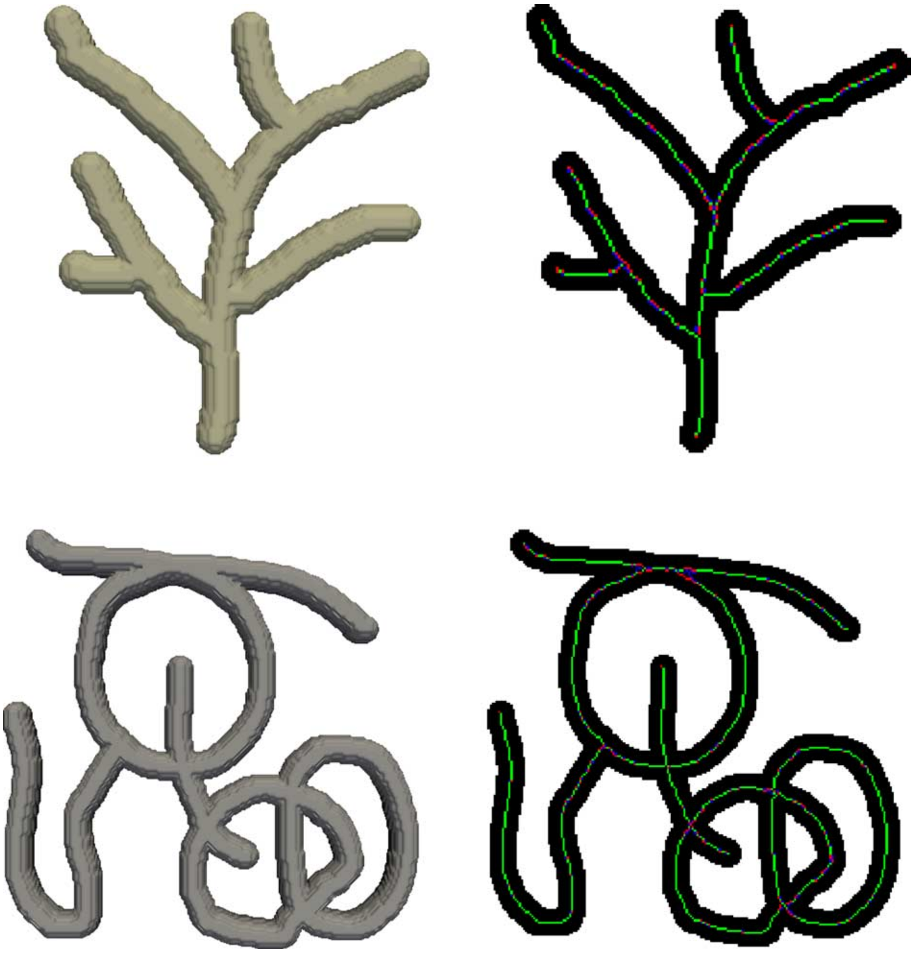

Fig. 4. Left: Synthetic objects with hand drawn centerlines. Right: The corresponding ground truth and computed centerlines. For each object (black), the overlapping points between the ground truth centerlines and the computed ones are shown in green, the points belonging to the ground truth centerlines but not the computed ones are shown in red and the points belonging to the computed centerlines but not the ground truth centerlines are shown in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

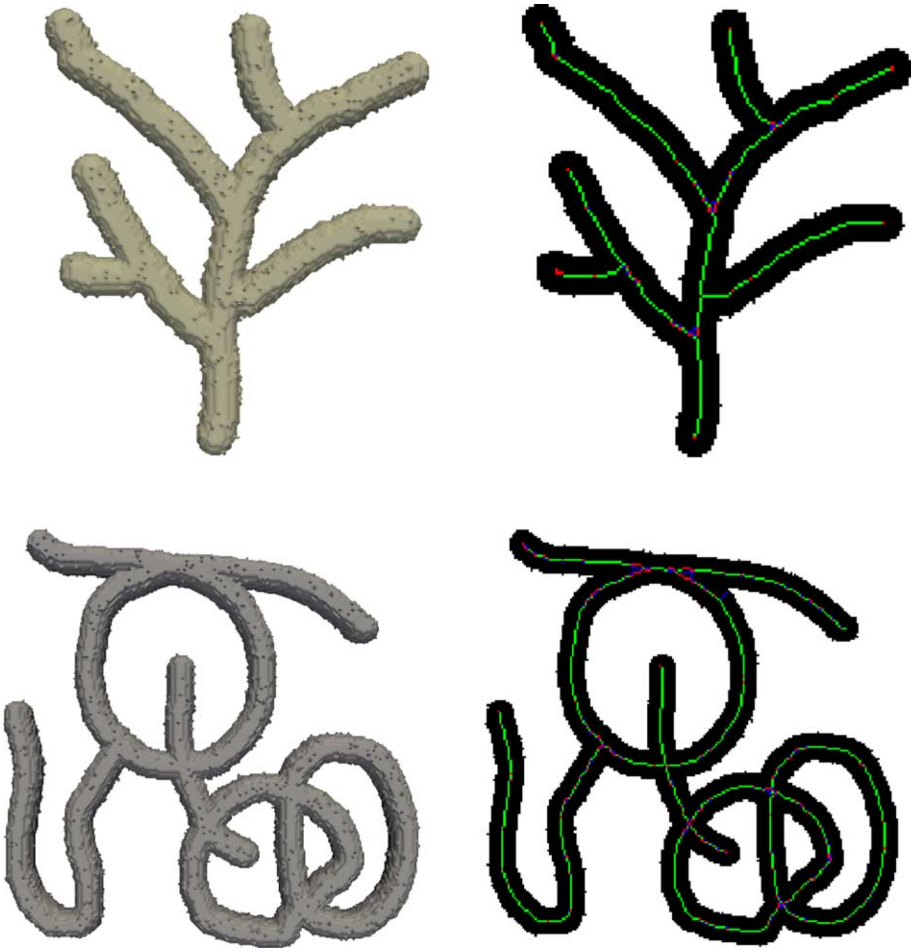

Fig. 5. Left: Synthetic objects with hand drawn centerlines. Boundary noise was added randomly. Right: The corresponding ground truth and computed centerlines. For each object (black), the overlapping points between the ground truth centerlines and the computed ones are shown in green, the points belonging to the ground truth centerlines but not the computed ones are shown in red and the points belonging to the computed centerlines but not the ground truth centerlines are shown in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for every synthetic object considered. A quantitative analysis was carried out by computing the amount of overlap, the average distance and the maximum distance between points on the ground truth and computed centerlines. The results are shown in Table 1 for the original objects and in Table 2 for their noisy versions. The amount of overlap is consistently above 80% except for the noisy helix which has an overlap of 56%. However, the average distance is never more than 0.5 mm (half a voxel) and in most cases is far less. The maximum distance is 5.0 mm, although Figs. 2–5 indicate that the distance is never more than one or two voxels except at end points. These results provide some insight into how the method would behave for real medical data. First, the *sausage* phantom displays the ability of the approach to handle highly variable cross sections, a key factor for colon data and vessel data with aneurysms. Second, the *complex* phantom shows that if the segmentation contains erroneous topology, the centerline can still be computed accurately but will reflect the loops and branches of the segmented object. Third, the *sine tube*, *helix*, *tree* and *complex* examples show that the method can handle structures with curvature and torsion provided that boundaries are relatively smooth.

Table 1
Validation of the centerline extraction algorithms for the synthetic objects

Name	Overlap (%)	Mean distance (mm)	Maximum distance (mm)
Tube	98	0.02	2.00
Sine tube	98	0.02	2.23
Helix	81	0.23	4.12
Sausage	96	0.10	4.00
Tree	90	0.11	1.41
Complex	84	0.16	2.23

In the presence of noise, high curvature and high torsion, (e.g., the helix with noise) the algorithm does not find the true centerline. The additive noise on highly twisted objects seem to have an impact on the location of the maxima of the Euclidean distance transform

Table 2
Validation of the centerline extraction algorithms for the noisy synthetic objects

Name	Overlap (%)	Mean distance (mm)	Maximum distance (mm)
Tube	98	0.02	2.00
Sine tube	94	0.07	2.83
Helix	56	0.50	1.73
Sausage	96	0.09	5.00
Tree	89	0.12	2.23
Complex	81	0.20	3.00

which is the most probable explanation for the 56% overlap. Nevertheless, the estimated centerline is still very close to the original and is quite acceptable as a flight path for virtual endoscopy. Overall, our results provide quantitative evidence that the proposed algorithms perform well.

4.2. Qualitative validation on medical data sets

We have carried out a qualitative validation of our centerline path extraction algorithms on a 512 × 512 × 400 CT colon data set provided by the Surgical Planning Laboratory of Brigham and Women's Hospital, a 515 × 512 × 286 CT data set of coronary arteries provided by the Cleveland Clinic and a 360 × 330 × 420 computed rotational angiography (CRA) data set of the head from the John P. Robarts Research Institute. The colon and coronary artery data sets were segmented using conformal snake algorithms

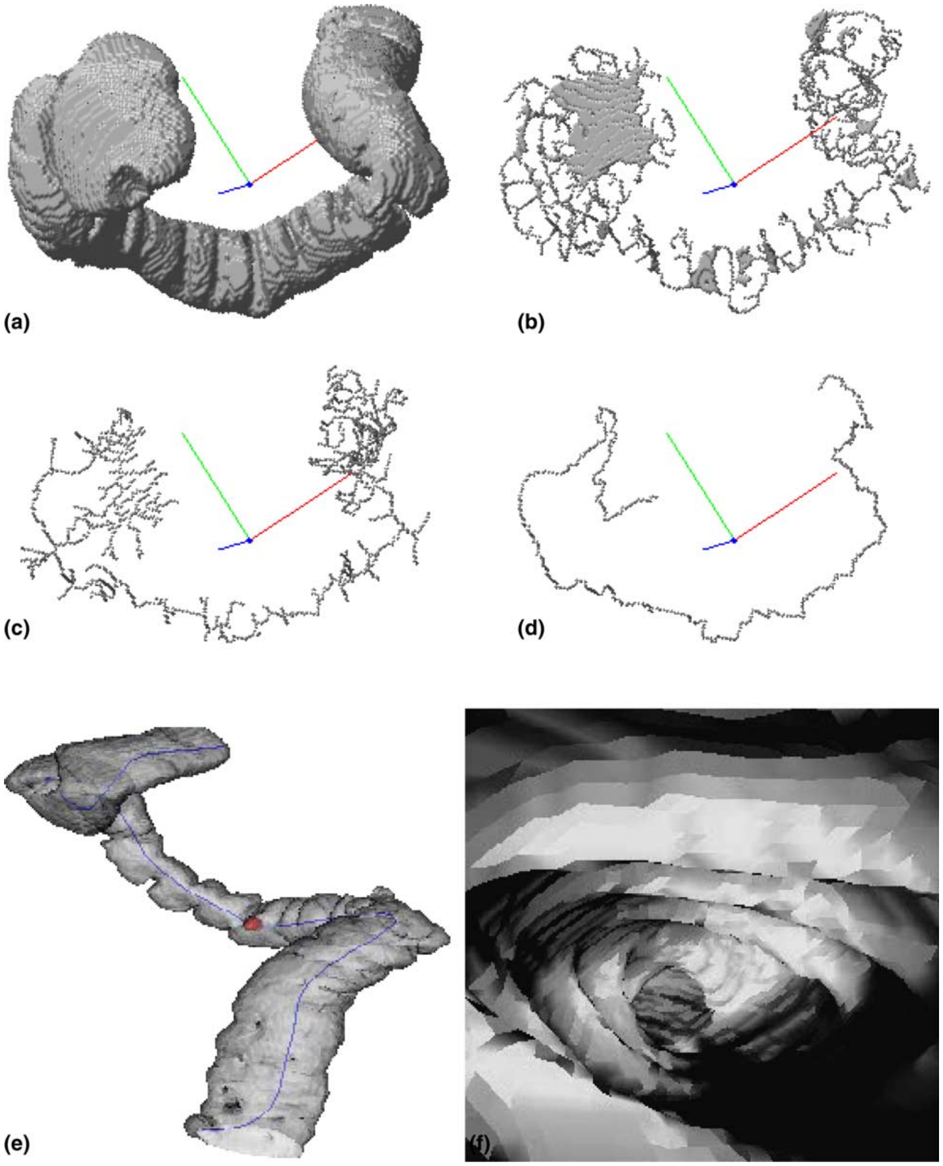

Fig. 6. (a) A segmented colon. (b) Its medial surface. (c) The associated medial curve. (d) The centerline path. (e) The smoothed path shown within the colon. (f) A snapshot from a fly through movie. The entire movie can be viewed at http://www.cim.mcgill.ca/~sbouix/research/data/endoscopy and in the online version of this article, at doi:10.1016/j.media.2004.06.026.

(Caselles et al., 1997; Kichenassamy et al., 1996) with discretizations motivated by the work of Brakke, which are discussed in detail in (Yezzi and Tannenbaum, 2002). The CRA data set was segmented using the flux maximizing flow algorithm of Vasilevskiy and Siddiqi (2002). The segmented images were then cropped around the regions of interest to reduce computational time and memory requirements.

Fig. 6 shows the colon data, its medial surface (Algorithm 2), its medial curve (Algorithm 3), the centerline path (Algorithm 4) and a screen shot of a fly through movie. Fig. 7 shows the arteries, their medial surface (Algorithm 2), the medial curve (Algorithm 3), the centerline paths (Algorithm 5) and a screen shot of a second fly through movie. Fig. 8 illustrates the application of the method on cropped regions from a segmented computed rotational angiography (CRA) data set. Observe that the complex topology (cycles and branches) and geometry of the data are captured accurately by the extracted centerline paths.

4.3. Computational complexity and parameters

In theory, the worst case complexity of the algorithm for computing centerline paths is O($n \log n$), because the thinning process is implemented using a heap (Siddiqi et al., 2002). Here n is the number of voxels in the interior of each structure. We plot the actual amount of computation time required to to obtain the centerlines for each structure in Table 3. These results indicate that in practice the complexity is almost a linear function of n, and hence the method is extremely efficient.

The approach we have presented has exactly two parameters: the average outward flux threshold at which

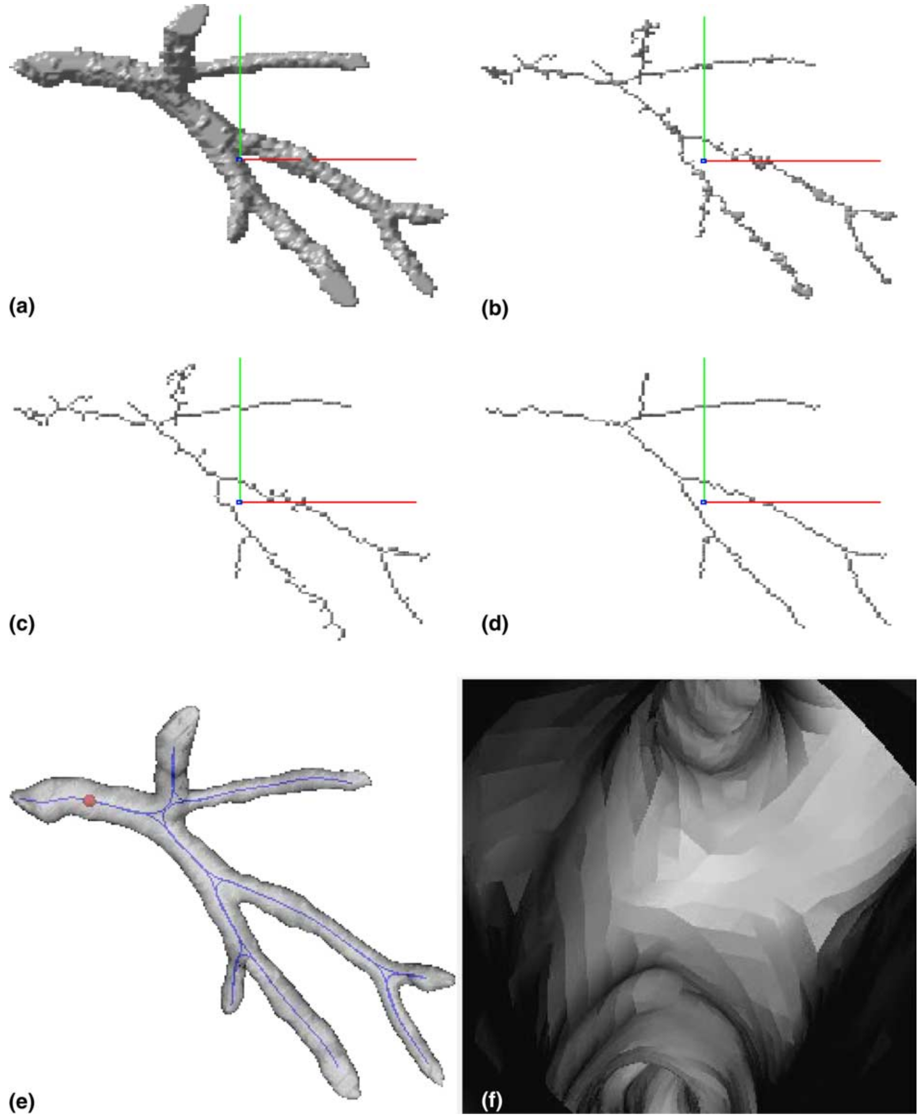

Fig. 7. (a) Segmented arteries. (b) Their medial surface. (c) The associated medial curve. (d) The centerline paths. (e) The smoothed paths shown within the arteries. (f) A snapshot from a fly through movie. The entire movie can be viewed at http://www.cim.mcgill.ca/~sbouix/research/data/endoscopy and in the online version of this article, at doi:10.1016/j.media.2004.06.026.

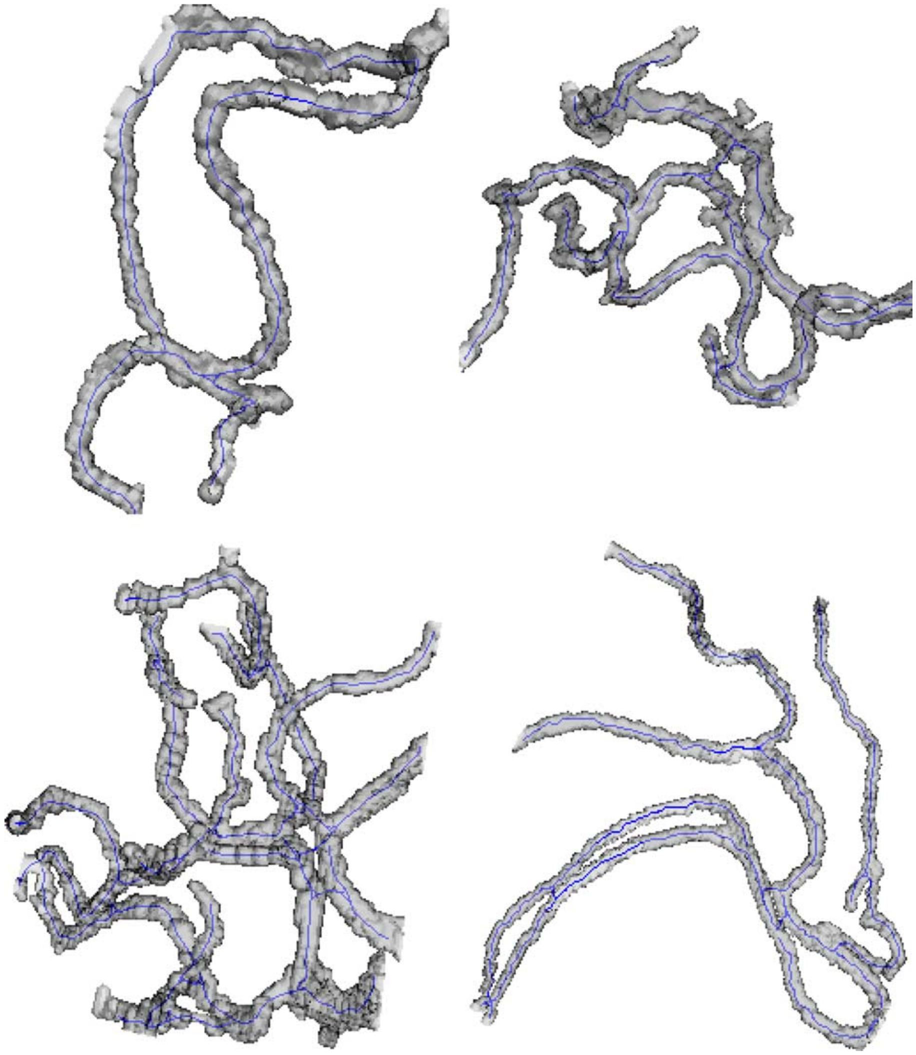

Fig. 8. Centerline paths of cropped regions from a segmented CRA data set of the head.

Table 3
Computation time required to extract the centerline paths on a Pentium 4, 2 GHz, 1G RAM computer

Name	Dimensions	# of voxels	Time (s)
Tube	256 × 256 × 50	71255	10
Sine tube	256 × 256 × 50	97283	11
Helix	256 × 256 × 50	100702	12
Sausage	256 × 256 × 50	124547	14
Tree	256 × 256 × 50	64949	12
Complex	256 × 256 × 50	167745	17
Colon	256 × 256 × 167	222124	31
Arteries	157 × 117 × 50	14221	3

The complexity is approximately a linear function of the number of voxels in the original object.

end points are preserved in the extraction of the medial curve (Algorithm 3) and the length threshold for the removal of spurious branches to obtain centerline paths (Algorithm 5). The first parameter can be selected such that 25–40% of the average outward flux map has a value less than this threshold. In our experiments a value of −5.0 was used for all computations. The length threshold used in the pruning can be chosen as a function of the thickness of the structures in the data. We scaled the arteries and CRA images such that the maximum diameter of a vessel was about 10 voxels and used this value as the length threshold for pruning the centerline paths. For the colon dataset no branch length threshold is required since the algorithm extracts the longest shortest path between two end points of the medial curve.

5. Conclusions

We have presented a robust and automatic method for finding centerline paths of segmented tubular structures. The framework enjoys a number of the advantages of the medial surface algorithm on which it is based. The homotopy thinning procedure preserves the topology of the object, ensuring a connected centerline for a connected object. It also defines the medial curve as the locus of positions which maximize the Euclidean distance from the object's boundary, thus guaranteeing that the resulting path is centered. Such a strategy is motivated in part by the distance ordered homotopic thinning algorithm of Pudney (1998). Fixing end points using the average outward flux measure has a theoretical justification (Dimitrov et al., 2003) and provides an algorithm which is robust to noise and similarity transformations and also has low computational complexity. The extraction of centerline paths requires no user interaction and is essentially parameter free: the average outward flux threshold is set automatically and the branch length threshold for pruning centerline trees is set to the diameter of the thickest expected tubular structure. We have validated the method quantitatively on a number of synthetic data sets with known ground truth centerlines. Qualitative evaluation has been carried out on a colon data set, a data set of coronary arteries and on CRA vessel data with complex geometry.

Acknowledgements

We are grateful to several colleagues for providing the medical data sets used in our experiments. We like to acknowledge the generous support of NSERC, FQRNT, CFI, AFOSR, NSF, MURI, ARO, and NIH (NAC and NAMIC). We thank the reviewers for their helpful comments.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.media.2004.06.026.

References

Aylward, S., Bullitt, E., 2002. Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. IEEE Transactions on Medical Imaging 21 (2), 61–75.

Bitter, I., Kaufman, A.E., Sato, M., 2001. Penalized-distance volumetric skeleton algorithm. IEEE Transactions on Visualization and Computer Graphics 7 (3), 195–206.

Borgefors, G., 1984. Distance transformations in arbitrary dimensions. CVGIP 27, 321–345.

Borgefors, G., Nystrom, I., Baja, G.S.D., 1998. Skeletonizing volume objects part II: from surface to curve skeleton. In: Proceedings of the SSPR'98, pp. 220–229.

Caselles, V., Kimmel, R., Sapiro, G., 1995. Geodesic active contours. In: Proceedings of the International Conference on Computer Vision, pp. 694–699.

Caselles, V., Kimmel, R., Sapiro, G., 1997. Geodesic snakes. International Journal of Computer Vision 22 (1), 61–79.

Damon, J., 2003. Global geometry of regions and boundaries via skeletal and medial integrals. Technical Report. Department of Mathematics, University of North Carolina at Chapel Hill.

Deschamps, T., Cohen, L.D., 2001. Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. Medical Image Analysis 5 (4), 281–299.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271.

Dimitrov, P., Damon, J.N., Siddiqi, K., 2003. Flux invariants for shape. In: Proceedings of the CVPR'2003, pp. 835–841.

Frangi, A., Niessen, W., Vincken, K.L., Viergever, M.A., 1998. Multiscale vessel enhancement filtering. In: Proceedings of the MICCAI'98, pp. 130–137.

Ge, Y., Stelts, D.R., Wang, J., Vining, D., 1999. Computing the centerline of a colon: a robust and efficient method based on 3D skeletons. Journal of Computer Assisted Tomography 23 (5), 786–794.

Harders, M., Wildermuth, S., Weishaupt, D., Székely, G., 2002. Improving virtual endoscopy for the intestinal tract. In: Proceedings of the MICCAI'2002, LNCS 2489, pp. 20–27.

Jolesz, F.A., Lorensen, W.E., Shinmoto, H., Atsumi, H., Nakajima, S., Kavanaugh, P., Saiviroonporn, P., E.Seltzer, S., Silverman, S.G., Phillips, M., Kikinis, R., 1997. Interactive virtual endoscopy. American Journal of Radiology 169, 1229–1237.

Kass, M., Witkin, A., Terzopoulos, D., 1987. Snakes: active contour models. International Journal of Computer Vision 1, 321–331.

Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A., 1995. Gradient flows and geometric active contour models. In: Proceedings of the International Conference on Computer Vision, pp. 810–815.

Kichenassamy, S., Olver, P., Tannenbaum, A., Yezzi, A., 1996. Conformal curvature flows: from phase transitions to active vision. Archive of Rational Mechanics and Analysis 134, 275–301.

Kong, T.Y., Rosenfeld, A., 1989. Digital topology: introduction and survey. Computer Vision Graphics and Image Processing 48 (3), 357–393.

Malandain, G., Bertrand, G., Ayache, N., 1993. Topological segmentation of discrete surfaces. International Journal of Computer Vision 10 (2), 183–197.

Malladi, R., Sethian, J.A., Vemuri, B.C., 1995. Shape modeling with front propagation: a level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (2), 158–175.

Paik, D.S., Beaulieu, C.F., Brooke Jeffrey, R., Rubin, G.D., Napel, S., 1998. Automated path planning for virtual endoscopy. Medical Physics 25 (5), 629–637.

Pudney, C., 1998. Distance-ordered homotopic thinning: a skeletonization algorithm for 3D digital images. Computer Vision and Image Understanding 72 (3), 404–413.

Siddiqi, K., Bouix, S., Tannenbaum, A., Zucker, S.W., 2002. Hamilton–Jacobi skeletons. International Journal of Computer Vision 48 (3), 215–231.

Siddiqi, K., Kimia, B.B., Shu, C., 1997. Geometric shock-capturing ENO schemes for subpixel interpolation, computation and curve evolution. Graphical Models and Image Processing 59 (5), 278–301.

Tiede, U., von Sternberg-Gospos, N., Steiner, P., Höhne, K.H., 2002. Virtual endoscopy using cubic quicktime-VR Panorama views. In: Proceedings of the MICCAI'2002, LNCS 2489, pp. 186–192.

Vasilevskiy, A., Siddiqi, K., 2002. Flux maximizing geometric flows. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (12).

Wink, O., Niessen, W.J., Viergever, M.A., 2004. Multiscale vessel tracking. IEEE Transactions on Medical Imaging 23 (1), 130–133.

Yezzi, A., Tannenbaum, A., 2002. 4D active surfaces for cardiac analysis. In: Proceedings of the MICCAI'2002, LNCS 2488, pp. 667–673.