Reliable Path for Virtual Endoscopy: Ensuring Complete Examination of Human Organs

Taosong He, *Member*, *IEEE*, Lichan Hong, *Member*, *IEEE Computer Society*, Dongqing Chen, and Zhengrong Liang

Abstract—Virtual endoscopy is a computerized, noninvasive procedure for detecting anomalies inside human organs. Several preliminary studies have demonstrated the benefits and effectiveness of this modality. Unfortunately, previous work cannot guarantee that an existing anomaly will be detected, especially for complex organs with multiple branches. In this paper, we introduce the concept of reliable navigation, which ensures the interior organ surface is fully examined by the physician performing the virtual endoscopy procedure. To achieve this, we propose computing a reliable fly-through path that ensures no blind area during the navigation. Theoretically, we discuss the criteria of evaluating a reliable path and prove that the problem of generating an optimal reliable path for virtual endoscopy is NP-complete. In practice, we develop an efficient method for the calculation of an effective reliable path. First, a small set of center observation points are automatically located inside the hollow organ. For each observation point, there exists at least one patch of interior surface visible to it, but that cannot be seen from any of the other observation points. These chosen points are then linked with a path that stays in the center of the organ. Finally, new points inside the organ are recursively selected and connected into the path until the entire organ surface is visible from the path. We present encouraging results from experiments on several data sets. For a medium size volumetric model with several hundred thousand inner voxels, an effective reliable path can be generated in several minutes.

Index Terms—Virtual endscopy, reliable path, reliable navigation, camera control, NP-complete, visibility.

1 Introduction

Virtual endoscopy is an integration of medical imaging and computer graphics technologies, leading to a computerized alternative to the traditional clinic endoscopy for examining the interior structure of human organs. From high resolution patient data acquired by CT or MR scanners, virtual endoscopy reconstructs a 3D virtual model of the hollow organ and allows a physician to navigate inside the model to detect abnormalities. Compared to the thousands of conventional endoscopy procedures performed each year, the advantages of virtual endoscopy are manifold. It is noninvasive, cost-effective, and free of risks and side effects such as perforation and infection. It could also improve the diagnostic sensitivity and specificity and be used to explore body regions not accessible to conventional endoscopy [1], [2], [3].

To realize these significant promises of virtual endoscopy, many prototype systems have been developed for a variety of clinical applications, including colonoscopy [4], bronchoscopy [5], angioscopy [6], and others. As a crucial component of such a system, the navigation technique decides how the physician controls the position and orientation of a virtual camera to examine the inner surface of the organ. Earlier work [2], [7], [8] has focused on planned navigation that precomputes a fly-through path to automatically move the camera from one end of the organ to another end. The whole navigation process is generated as an offline animation by manually [8] or automatically [2], [7] specifying the camera parameters at each key frame. Obviously, such a technique prevents the physician from examining an anomaly in detail since no interaction is possible. On the other hand, by taking advantage of the rapid improvement and proliferation of graphics hardware, free navigation aims to allow a physician to interactively control the camera position and orientation, generating the resulting scene in real time [9]. The problem of this approach is that the physician is required to control the camera at each frame to navigate through the virtual model by himself. Like exploring an unknown territory without any map or guidance, this could be both time consuming and challenging. More recent work [4] has combined the benefits of both planned and free navigations. It adopted the guided navigation scheme [10] to achieve a more intuitive and efficient exploration. With a guided navigation, the camera moves automatically from a source point to a target point along a fly-through path generated (on the fly) by a navigation planner. When necessary, the physician can take over the control at any time by interactively and easily adjusting the camera position and direction. The switch between these two modes is seamless, resulting in a smooth navigation.

Unfortunately, all these navigation techniques suffer a serious problem: A physician could not be sure that he has visualized all the existing abnormalities after a virtual

- *T. He is with the Network & Service Management Research Department, Bell Labs, Lucent Technologies, Room 2D-412, 600-700 Mountain Ave., Murray Hill, NJ 07974. E-mail: taosong@research.bell-labs.com.*
- *L. Hong is with EnFashion, 2001 Spring St., Redwood City, CA 94063. E-mail: lichan_hong@yahoo.com.*
- *D. Chen and Z. Liang are with the Department of Radiology, School of Medicine, State University of New York at Stony Brook, 4L-120/Health Sciences Center, Stony Brook, NY 11794-8460. E-mail: {dchen, jzl}@clio.rad.sunysb.edu.*

Manuscript received 12 May 2000; revised 5 Jan. 2001; accepted 11 Jan. 2001. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number 112083.

endoscopy procedure has been performed. This is because the fly-through path generated by a navigation planner cannot guarantee that each interior surface area is visible to at least one point on the path. The flexible camera manipulation provided by the interactive navigation would only alleviate, but not solve, this "blind area" problem since it depends on the physician's maneuver to cover the entire surface. In fact, conventional endoscopy encounters a similar problem. During the procedure, a physician relies on his training, experience, and analysis to determine which parts of the organ surface need to be examined and does his best to cover those areas. This makes it possible to miss an area of diagnostic significance, particularly when the organ has a complex structure. For example, in a bronchoscopy, a physician could have missed an anomaly that is hidden within a small but twisted bronchial passage. Although the success of either a conventional or a virtual endoscopy always depends on the expertise and judgments of the physician performing the procedure, it is desirable to reduce the possibility of such a kind of human negligence as much as possible.

Since we can perform extensive preplanning before navigating inside the virtual organ, the existence of a digital model in virtual endoscopy provides an edge to conventional endoscopy for solving the blind area problem. However, to our knowledge, this issue of avoiding a "blind area" has never been explicitly identified and addressed by previous path planning techniques. A possible reason is that virtual endoscopy research has been focused on simulating the clinical procedure, instead of exploiting the unique advantages provided by the virtual environment and medical imaging techniques. In this paper, we propose a new navigation scheme, *reliable navigation*—its preliminary version was presented at the 1999 IEEE Medical Imaging Conference—for exploring the inner surface along the hollow passages of virtual organs. Our major contribution is the introduction of *reliable path*, i.e., a virtual path which stays near the center of the organ and from which the entire interior surface is visible. Restricting the camera to moving along the reliable path suffices to ensure that the physician could have full coverage of the inner surface during the examination.

Camera control is another crucial component of reliable navigation. For a physician to actually visualize the entire surface, he would have to look in all directions from each point on the reliable path. However, a straightforward implementation to satisfy this requirement could make the navigation difficult, if not impossible, to control and interpret. To solve this problem, we have developed a new camera control scheme and briefly discuss it at the end of this article. The details of our solution will be presented in an upcoming paper.

This paper is organized into four additional sections. In Section 2, we introduce the concept of reliable path and discuss the theoretical complexity of computing an optimal reliable path. Section 3 presents an efficient algorithm to generate an effective approximation to the optimal reliable path. In Section 4, we demonstrate the effectiveness of reliable navigation by applying our system to several experimental data sets. Finally, we conclude with future work in Section 5.

2 Reliable Path

The essence of our approach is to predesign a fly-through path that ensures the removal of any blind area during the navigation. We call such a path a "reliable path" since we can rely on it for not missing any important information. Accordingly, a reliable navigation for virtual endoscopy is defined as the procedure that utilizes a reliable path for a complete examination of the patient's organ. In this section, we first establish the concept of reliable path, including the criteria of evaluating a reliable path and the definition of the optimal reliable path. We then discuss the algorithms for generating a near-optimal reliable path. Finally, we compare reliable path with the popular skeleton approach and demonstrate the fundamental differences between these two path generation methods.

2.1 Optimal Reliable Path

Conceptually, a reliable path in a virtual organ is defined as a set of connected points inside the model from which the entire interior surface can be seen. Note that we do not simply choose a minimum set of points with full surface coverage since the selected points might be disconnected, while a fly-through path has to be continuous.

Strictly speaking, a reliable path only needs to satisfy the requirement that each interior surface area of the organ is visible at least once to the physician moving along the path. As a result, given the virtual model, there usually exists a large number of reliable paths. For example, a path connecting all the internal voxels of a model is always reliable, as well as a path connecting all the internal voxels whose neighbors include at least one surface voxel.

On the other hand, since the purpose of a reliable path is to guide the navigation, a practical reliable path should also satisfy the other desired properties of a fly-through path. First, the path should never penetrate through the surface. This requirement is inherently satisfied by the definition of reliable path since all the points on the path are inside the model. A more restrictive requirement is that the path should stay away from the surface so that the camera can obtain a wide view of the organ surface [2]. In other words, a good reliable path needs to stay near the center of the model as much as possible. In previous work, this has been achieved through the calculation of a center line between two user-specified source and target points using techniques such as boundary peeling [2] or potential field [4]. Unfortunately, there is no guarantee for the reliability of such a center line. Finally, the path should be of short length to increase the efficiency of examination.

Based on these extra requirements, the vast majority of reliable paths, such as the fully connected internal voxel set, are bad choices for guiding navigation and cannot be used in practice. Among the feasible solutions, we define an optimal reliable path in a virtual organ as a minimum set of connected points inside the model from which the entire interior surface can be seen. Intuitively, a path as defined would stay away from the organ surface as desired. This is because those points near the center of the model usually

cover larger surface areas than those near the surface and we are looking for a minimum set of connected points to cover the entire surface. A path of minimum length also alleviates the problem of frequently turning the camera to traverse small branches during the navigation, leading to a shorter examination time and making the whole procedure more efficient.

In the continuous space, given an organ model D, its interior region R, and its boundary area ∂D, we have $D = R + \partial D$. The optimal path is defined as a continuous curve $P \subset R$ of minimum length satisfying the following condition

$$\forall(v_0 \in \partial D) \rightarrow \exists((v_1 \in P) \wedge (\overline{v_0 v_1} \in R \cup \{v_0\})), \qquad (1)$$

where $\overline{v_0 v_1}$ denotes the line segment connecting v_0 and v_1. In other words, P is a curve from which each point on ∂D is visible. Unfortunately, given the discreteness of acquired data in virtual endoscopy, an analytical solution to P is usually not achievable since there a simple function to describe ∂D is lacking. Although trilinear interpolation function can be used to approximate the surface inside each voxel, it generates a piecewise linear function for the whole model, making it difficult to calculate P.

To solve this problem, we redefine the optimal reliable path in the discrete space as follows: From the acquired 2D slices, we reconstruct a 3D volume V. Next, we perform a segmentation of the examined organ, classifying the voxels into a surface voxel set B, an internal voxel set I, and an exterior voxel set E. Note that different notations are used to distinguish the discrete case from the continuous one. For each voxel $v \in I$, there is an associated set of visible voxels $S_v \subset B$ that can be seen from v. Thus, the optimal reliable path is defined as a connected set $C^* = \{v_0, v_1, , v_{n-1}\}$ with a minimum number of voxels n so that each voxel of B belongs to at least one visible voxel set $S_{v_i}, 0 \leq i < n$. The connectivity between two voxels is usually defined by 26-connectivity. Note that, based on the definition, there could be more than one optimal reliable path for a certain model.

Unfortunately, as we shall prove in the Appendix, the optimal reliable path problem is NP-complete. In other words, it is not feasible for us to find the reliable path with minimal length. The best we can hope is therefore to find some *near-optimal* solutions in polynomial time. Mathematically, the optimization level of a path can be defined as $|C|/|C^*|$, where C is the voxel set on the path and C^* is the voxel set on the optimal path. In practice, the criteria of evaluating a reliable path are also based on the navigation requirements. Generally, we can classify all the reliable paths for a model into three categories:

- Optimal reliable paths: Although they are the most effective navigation paths, we cannot afford to calculate them.
- Feasible reliable paths: These are the paths that can be used as navigation guidance. They are reliable, centralized, relatively short, and smooth. Here, the smoothness of a path can be roughly defined as the total amount of curvatures along the path.
- Impractical reliable paths: These are the paths that cannot be used for guiding the navigation.

Both the feasible and impractical reliable paths can be regarded as approximations to the optimal reliable paths and there is no strict boundary between them. Fortunately, we are not required to actually classify each reliable path. Instead, we are looking for efficient algorithms which generate effective reliable paths.

2.2 Approximation Algorithms

As shown in the Appendix, the optimal reliable path is closely related to the minimum set cover problem. The best known approximation to the minimum set cover problem is, at each iteration, from the unselected subsets, choosing the subset U that covers the most remaining uncovered elements [11]. This simple greedy algorithm achieves a ratio of $H(max|U| : U \in F)$ to the optimal solution [12], where $H(d)$ denotes the dth harmonic number, $H(d) = \sum_{i=1}^{d} 1/i$, and F is the family of subsets. It can be implemented in such a way that it runs in $O(\sum_{U \in F} |U|)$.

There are two simple ways to directly extend the greedy minimum set cover algorithm for generating a reliable path in the discrete space. One method is, from the internal voxels that are connected to the already selected path, choose the voxel that covers the most remaining uncovered surface voxels. In other words, we keep the path connected at each iteration. The other method is to first apply the greedy algorithm, then connect the resulting voxels at the end. Note that no ratio upper bound to the optimal solution has been established for either of the methods. To find the voxel with the maximal coverage, both methods must calculate the visible boundary voxel set S_v for each voxel $v \in I$. Unfortunately, as we shall discuss in Section 3.2, the visibility test is an expensive process. Since there could easily be millions of internal voxels in a virtual organ, path generation requiring the visibility tests for all the internal voxels is very time consuming.

To avoid this problem, we have developed a new algorithm for the efficient generation of an effective reliable path. However, before we discuss this algorithm in the next section, it is worthwhile to clarify the concept of reliable path by comparing it with a widely used approach, skeleton.

2.3 Skeleton vs. Reliable Path

Skeleton, or medial axis, was first introduced for studying biological shape [13]. It can be defined as the set of centers of maximal balls, i.e., of balls in a region P that are themselves not enclosed in another ball in P [14]. As an intuitive method for shape representation, skeleton has found applications in a broad range of areas, including pattern recognition and solid modeling. Recently, researchers have also started to apply skeleton as the basis for constructing navigation paths for virtual endoscopy (e.g., [15]).

From the perspective of navigation, skeleton has a nice property that its points by definition stay in the center of the model. Another interesting property is that, since all internal points are contained in at least one maximal ball, the union of the maximal balls corresponds to the interior region. At the same time, since there is at least one maximal ball touching each point on the boundary, the envelope of the maximal balls corresponds to the region boundary. In

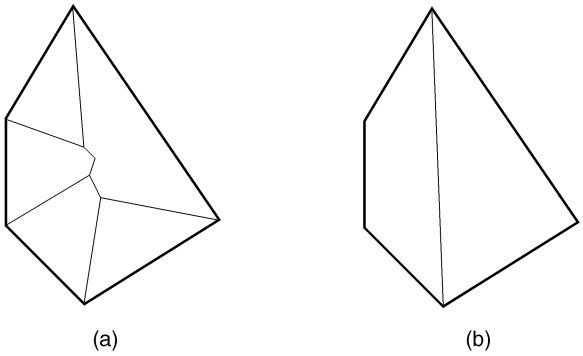


Fig. 1. (a) A 2D skeleton as the reliable path for a convex polygon; (b) an alternative reliable path leading to a smoother navigation.

other words, the entire interior surface of a virtual organ should be visible to its skeleton, i.e., skeleton is a reliable path. Mathematically, the reliability of the skeleton S of a model D can be proven by contradiction as follows: Given D, its boundary area ∂D, and its interior area R, we have $D = \partial D + R$. Let us assume that there exists a point v_0 on the boundary ∂D that cannot be seen from S. This is equivalent to saying that there is another point $v_1 \in \partial D$ such that $v_1 \neq v_0$ and $v_1 \in \overline{cv_0}$, where c is the center of the maximal ball B enclosing v_0 and $\overline{cv_0}$ is the line segment connecting c and v_0. Since B is convex and both c and v_0 are included in B, v_1 is inside B. Based on the definition of maximal ball, the interior area of B is included in R. Therefore, $v_1 \in R$. This is contradictory to the fact that $v_1 \in \partial D$.

Unfortunately, skeleton generally is an impractical reliable path. In other words, it is not a good approximation to the optimal reliable path and should not be directly used for navigation. The main reason is that the definition of skeleton leads to many branches on the path, especially near areas of small cavities on the surface. Following these branches during navigation would add many small turns for the physician, making the fly-through much more difficult to control and interpret; cf. Fig. 1. Another reason is that a skeleton of a 3D model could include surfaces which should not be directly used as part of a fly-through path. For example, the skeleton of a rectangular box includes a face in the middle that is difficult and not necessary to traverse for navigation. Finally, the calculation of 3D skeleton is a difficult problem, especially for large data sets. In the discrete space such as the 3D volume used in virtual endoscopy, some efficient algorithms have been proposed to approximate skeletons (e.g., [16], [15]). This causes another problem that these approximated results no longer preserve the skeleton's reliability.

As a shape descriptor, skeleton was not originally designed for navigation. For skeleton applications like model reconstruction, it is essential to retain, on the resulting medial axis, detailed information about the model, such as the small branches. On the other hand, a well-designed reliable path tries to avoid sharp turns and small branches as long as all the surface detail is visible to the path. The fact that skeleton, by strict definition, happens to be a reliable path does not mean that we are proposing another approximation algorithm to compute skeleton. For example, a volumetric model can be seen as sampling of the original object. Amenta et al. [17] have developed a method to guarantee that a skeleton approximation calculated for densely sampled surface is within a distance bound to the correct skeleton. Compared to a reliable path, their results have no reliability guarantee. More importantly, those results are generally not as good as a good reliable path for navigation because of too many unnecessary small features. On the other hand, a reliable path is usually not a good approximation to skeleton and cannot be used as the basis for shape recognition or reconstruction. In fact, from the perspective of fly-through path design, a good reliable path is more closely related to the center line (e.g., [2]) than to the skeleton since center line is a much better navigation path than skeleton. The fundamental differences between our approach and that of center line are the introduction of reliability during navigation and the incorporation of this concept into effective navigation path planning.

3 Path Generation

In practice, the reliability of a navigation path and its calculation time are more important than its optimization level. We have developed an efficient algorithm to generate a near-optimal, smooth, yet still reliable path. In this section, we first present the key steps of this algorithm, then discuss an important component of the algorithm, the visibility test.

3.1 Efficient Algorithm

As shown in Fig. 2, the idea of our algorithm is to repeatedly incorporate into the path C new observation points covering maximal areas of previously invisible surface. Essentially, we apply a greedy method similar to the minimum set cover approximation algorithm [11] previously discussed. The key difference is that, instead of sorting the internal voxels according to the sizes of their visible voxel sets, we adopt a measurement which is much easier to compute. In order to speed up the path computation, at each step, we also keep the set of candidate voxels to be included in the path as small as possible.

The measurement we have chosen is based on the Euclidean distance from each internal voxel to the closet surface voxel, which can be efficiently calculated in the volumetric space through distance transform [18]. Intuitively, voxels with higher distance values are closer to the center of the model. However, from the viewpoint of reliable path generation, "center" is a relative concept that is closely related to the local geometry. Moreover, since the visible voxel sets of neighboring voxels could largely overlap, selecting them simply based on their distance values is not effective. To avoid this problem, we take into account the relationship between each voxel and its neighbors by first considering those voxels whose distance values are at local maxima. The motivation is to include a minimum set of local maxima into the path to achieve a majority of surface coverage and greatly reduce the complexity of the remaining steps.

In Fig. 2, most of the computing time is spent on calculating the visible set T_v for each voxel v that has been

1. Reconstruct volume V from 2D slices;
2. Classify V into: external E, internal I, surface B;
3. Perform Euclidean distance transform on I;
4. Initialize L with all the local maximum voxels;
   Sort L in the order of decreasing distance values;
5. $P = \emptyset; U = B$;
   Repeat until $L == \emptyset$
   $v = L.head$; $L = L - v$; get visible set T_v for v;
   if $(T_v \cap U \neq \emptyset)$
   $P = P \cup \{v\}; U = U - T_v$;
6. Connect P into $C \subset I$ using the shortest path algorithm;
7. $P' = C - P$; $U = U - \bigcup_{v \in P'} T_v$; $P = \emptyset$;
8. Repeat until $U == \emptyset$
   select $v \in U$; get $K \subset I$ that is visible from v;
   select $v' \in K$ with the highest distance value;
   $P = P \cup \{v'\}; U = U - T_{v'}$;
9. Connect $C \cup P$ into $C \subset I$ using the shortest path algorithm;
10. Output C.

Fig. 2. The pseudocode of the reliable path generation algorithm.

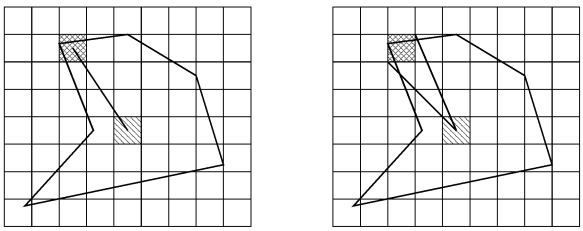
traversed. Our main objective is to reduce the number of traversed voxels. In the following, we take a close look at some key steps of the algorithm to discuss how this goal has been achieved:

- At Step 4, the list of local maximum voxels L is only a small portion of the internal set I. This makes the sorting at Step 4 acceptable.
- At Step 5, we calculate the visible sets covered by local maxima. As we have discussed, usually local maxima themselves cover a majority of surface voxels. As a result, the uncovered set U after Step 5 is much smaller than the original surface set B. Note that we only include those local maximum voxels that are visible to some uncovered surface voxels.
- At Step 6, we link the selected local maxima into a connected path using the Dijkstra's shortest path algorithm [19]. The cost at each internal voxel is assigned as a monotonic decreasing function of its distance value. This is to satisfy the short length requirement of a well-designed reliable path, as we have discussed in Section 2.
- At Step 7, we use the newly added voxels (which are not necessarily local maxima) on the path to reduce $|U|$.
- At Step 8, the size of uncovered set U is much smaller than the internal set I. By traversing U instead of $I - C$, we dramatically reduce the number of iterations at Step 8 and make sure $|U|$ is reduced by at least one element at each iteration. As we shall discuss in Section 3.2, the number of internal voxels visible to a surface voxel is usually much smaller than the internal voxel set I, making it efficient to find v' at Step 8. Meanwhile, since v' has the highest distance value, it is close to the center and generally covers a larger surface area than any of the other candidates.
- At Step 9, we again apply the Dijkstra's shortest path algorithm [19] to connect voxels. In this way, the final graph C at Step 10 is generally close to the center of the model and the total length of C is relatively short.

Essentially, we have presented a two-stage algorithm. The first stage (Steps 3-6 of Fig. 2) establishes a priority list of the center voxels we prefer to include in the path and covers a majority of the surface using the selected center voxels. The second stage (Steps 7-9 of Fig. 2) processes the uncovered surface voxels and guarantees that the resulting path is both reliable and effective. Note that our algorithm does not take into consideration noise presented in data sets. Instead, it assumes a noise-free data set. In contrast to skeleton, our reliable path generated is generally less sensitive to noise introduced during data acquisition and segmentation. This is because visibility, instead of local geometry, is applied for determining the path voxels. However, an extra number of small turns and forks could still be generated along the path when the noise brings small and twisted branches on the surface. To solve this problem, some noise treatment, such as boundary smoothing, can be performed as preprocessing in the segmentation stage.

3.2 Visibility Test

A key contribution of our algorithm is the efficient calculation of the visible voxel set for a selected voxel (cf., Steps 5, 7, and 8 in Fig. 2). This visibility test plays a core role in ensuring the reliability of a generated path. For a volumetric model, a voxel is a quantum unit of volume representing a "cell" instead of a point [20]. For virtual endoscopy, such a cell is usually a rectangular box. This could lead to different definitions of visibility between two voxels. In our system, we have chosen a visibility definition that simplifies the computation and guarantees that the accuracy of the test is up to the discrete resolution of the



(a) (b)

Fig. 3. Two-dimensional example of comparison between different visibility definition: (a) Two marked voxels are visible to each other when the line segment connecting centers does not intersect surface; (b) a more conservative definition requires the entire second voxel visible to the center of the first one.

volumetric data. Essentially, voxel v_0 and voxel v_1 are visible to each other if and only if the line segment $\overline{c_0c_1}$ does not intersect the surface, where c_0 and c_1 are the centers of v_0 and v_1, respectively; cf., Fig. 3a. More conservative visibility definitions can be used and techniques such as the visibility skeleton [21] can be applied accordingly for the test; c.f., Fig. 3b. Generally, those techniques are more complicated and time consuming than our method. On the other hand, our visibility test can be easily replaced with new implementations to accommodate specific requirements.

An important property of our visibility test is that, for each voxel v on the path, its visible voxel set is calculated without considering the viewing volume at v during the navigation. Instead, we have assumed a 360° field-of-view angle. The reason is that it is very difficult to predict the desired viewing frustum before the actual navigation. Of course, we could fix the viewing direction and angle at each voxel along the flying path and apply the information for calculating the reliable path. However, this approach would severely limit many navigation functionalities, such as zooming and focusing. More importantly, to achieve full inner surface coverage, a physician would have to stop and look at different directions at nearly every point on the reliable path (more precisely, at every point in $C \cup P$ after Step 8 in Fig. 2 in our algorithm). As a result, it is mentally very difficult for him to manipulate the navigation and analyze the result. It could also cause some side effects, such as dizziness and vertigo. Although it is not a practical solution by itself, a 360° view would eliminate these problems. Since we have developed a feasible 360° camera control method that will be addressed in future work, in this section, we discuss the implementation of visibility test under this scheme.

In order for a physician to have a clear view of the examined surface, we restrict the range of sight at each voxel. As a result, there are two steps in the visibility test of voxel v. First, at the center of voxel v, we insert a sphere of radius r representing the range within v's sight. To generate a smooth path, we usually assign r to be one to three times the value of the maximal voxel-to-surface distance in the model. Second, conceptually, we shoot a ray $\overline{vs}$ from v's center to the center of each surface voxel s within the sphere. If the ray does not intersect any other surface voxel, s is visible. For efficiency purposes, prior to the visibility tests we create a template which, for each voxel p inside the sphere, records the relative locations of a list of voxels intersected by the ray shooting from the sphere center to the center of voxel p. For each voxel to be tested, this template is then applied by putting the sphere center at the voxel center.

There are at least four ways to decide whether $\overline{vs}$ intersects a surface voxel. The simplest and most conservative way is to declare an intersection if $\overline{vs}$ intersects the box occupied by the voxel. A more accurate way is to calculate the intersection points between $\overline{vs}$ and the six faces of the voxel box, apply the bilinear interpolation function to compute the density values at the intersections, and, finally, perform a segmentation process to decide whether any of these points represents an intersection with the surface. The most accurate way is to detect whether $\overline{vs}$ intersects the trilinear interpolation function representing the surface inside the voxel. All of these methods have been implemented in our system. Finally, to achieve interactive rendering speed, the organ surface is usually first reconstructed into a polygon mesh using the Marching Cubes algorithm [22]. The intersection test can thus be performed between $\overline{vs}$ and the associated triangles within each surface voxel. We did not use this approach because it is much slower than the others and requires extra bookkeeping.

4 Experimental Results

We have implemented our reliable path generation algorithm and applied it to several data sets. The results have been very encouraging and are summarized in the following. All the experiments were measured on an SGI Octane workstation with a 300MHZ R12000 processor and 640MB memory.

The first experiment used an airway data set, provided by the University Hospital at Stony Brook. With a spiral CT scanner (General Electric Model High-Speed CTI), 2D images were acquired from a female patient of age 35, using protocols of 3mm/1.7:1 pitch, 120kVp, 280mA, and 35cm FOV. The reconstructed data set has a resolution of $512 \times 512 \times 202$, with 1mm thickness between the transverse slices. To extract the major airway from the acquired data, we employed a self-adaptive segmentation method [23]. Fig. 4 shows, from two different angles, the shape of the airway, as well as the reliable path computed by our algorithm. There are $24,571$ internal voxels and $16,958$ boundary voxels in this model. The reliable path consists of 363 voxels. The generation took about 109 seconds, out of which 51 seconds was devoted to performing the Euclidean distance transform.

The structure of the airway as presented in Fig. 4 is relatively simple. In order to display the path inside the model in Fig. 4, we made the airway surface semitransparent. The voxels on the reliable path were colored in red and magnified in size for the purpose of better illustration. Note that, during the visibility tests, we have restricted the range of sight at each voxel. As a result, there are long branches on the path, ensuring that each surface area can be examined from a close distance.

The second data set that we tested is also an airway data (courtesy of Professor Milan Sonka of University of Iowa).

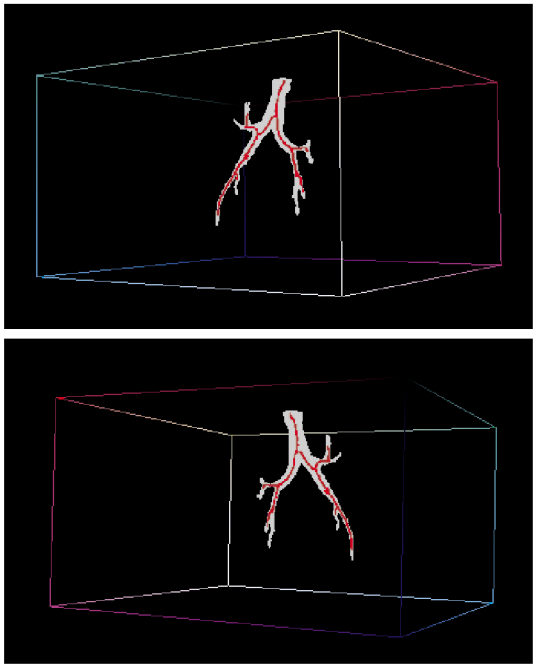


Fig. 4. The shape and reliable path of a simple airway data, shown from two different angles.

This airway structure is much more complicated than that of the first data set due to a different segmentation method. The data resolution is $256 \times 256 \times 166$. Fig. 5 presents, from four different angles, the shape of the airway, as well as the computed reliable path for the model. There are $47,225$ internal voxels and $39,058$ boundary voxels. The path consists of $1,378$ voxels. It took 10 seconds to perform the Euclidean distance transform and the whole path generation consumed about 23 seconds.

Using semitransparency to display the airway surface in Fig. 5 resulted in partial overlapping of some branches in the pictures. Nevertheless, by looking at the airway from different angles, we can clearly see that there are branches of different sizes in this data. More importantly, the reliable path generated stays in the center of the airway. It is also smooth and insensitive to noise introduced during data acquisition. To traverse the complicated path, we can simply perform a depth-first search of all the path voxels with 26-connectivity. The resulting navigation path is at most twice the length of the total number of path voxels.

Our third experiment was conducted on a blood vessel data set (courtesy of Dirk Bartz at University of Tübingen). This data set was acquired from a 41-year-old patient using a Siemens neurostar biplanar angiograph. It contains a scan of an area close to the base of the skull, where the arteria carotis interna splits into arteria cerebri media and arteria cerebri anterior. The data set has a resolution of $512 \times 512 \times 258$. Fig. 6 presents, from four different angles, the shape of the blood vessel, as well as the computed reliable path for the model. There are $292,773$ internal voxels and $151,064$ boundary voxels. The path consists of $2,741$ voxels. The Euclidean distance transform took 67 seconds. The generation of the reliable path took about

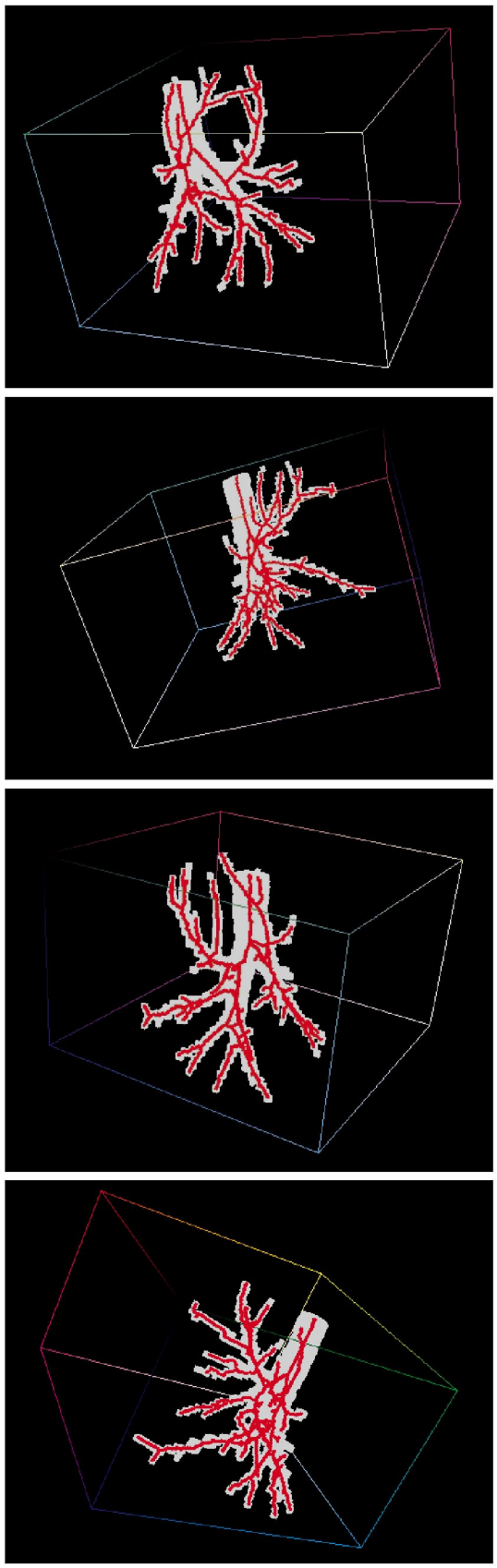


Fig. 5. The shape and reliable path of a complicated airway data, shown from four different angles.

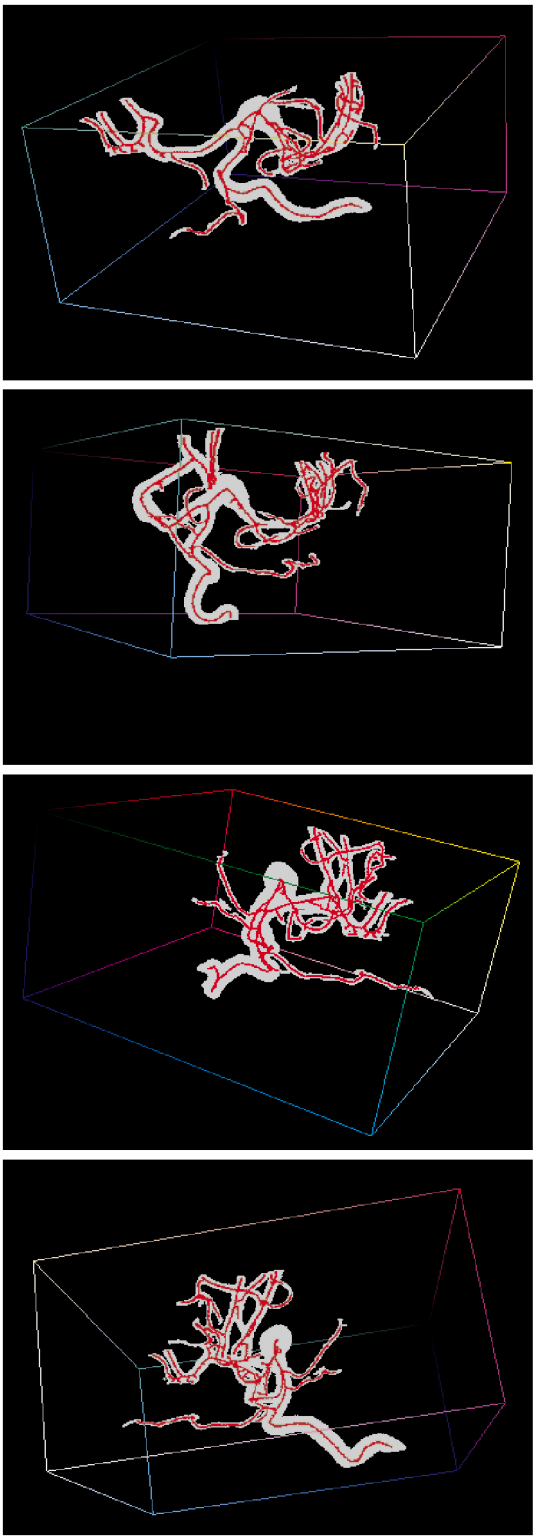


Fig. 6. The shape and reliable path of a blood vessel, shown from four different angles.

270 seconds. Again, Fig. 6 shows that the generated path stays in the center of the main branches of the vessel tree and only forks into small branches when necessary to maintain its reliability.

5 Conclusions and Future Work

In this paper, we have presented a new navigation scheme for virtual endoscopy, reliable navigation. We have identified a major drawback of the existing virtual endoscopy techniques that an anomaly cannot be guaranteed to be examined by the physician. We solved this problem by introducing the concept of reliable path. This is the first known attempt to ensure the complete examination of human organs. We discussed the criteria to evaluate the reliable paths and proposed the definition of optimal reliable path. Based on our proof that the optimal reliable path problem is NP-complete, we developed an efficient approximation algorithm to generate a smooth and near-optimal reliable path in the volumetric space. This path provides an effective navigation guide for exploring a model with complex structure. The efficiency of our path generation algorithm depends on both the resolution of the data sets and the complexity of their inner structures. We have successfully applied the algorithm to several data sets and presented encouraging results.

A reliable path basically provides a "road map" for exploring the virtual model. For the navigation, a camera control mechanism is essential to assist the physician in using the road map. Currently, to our knowledge, no effective navigation scheme has been designed for examining data sets with multiple branches. In addition, the viewing parameters at each point on the path need to be carefully manipulated for full surface coverage. We have developed a new camera control scheme to make sure that the physician actually examines the whole inner surface during the procedure. The details of that approach are beyond the scope of this article and will be presented in an upcoming paper. In general, we restrict the physician to the precomputed path while allowing him to control the speed and direction (backward and forward) of the movement. Additionally, the physician is equipped with a virtual camera with an infinitely zoomable lens so that he can reach any surface area and examine it as closely as possible. A key idea of our method is that the virtual camera has to be specially designed for capturing and presenting a full 360° view.

We are currently working on a variety of techniques to accelerate the process of reliable path computation and improve the quality of the generated path. One direction is to take advantage of the inherent spatial coherence of the volumetric data. We are also conducting experiments on more data sets. Finally, we plan to evaluate our system through clinical studies in the near future.

Appendix

Complexity Analysis

In this appendix, we take a close look at the complexity of the optimal reliable path problem as presented in Section 2.1. If, from the original problem, we remove the requirement that C^* is connected, it becomes the well-known minimum set cover problem, which is NP-complete (Garey and Johnson: SP5 [24]). In the following, we prove that, with the additional connection constraint, it is still NP-complete.

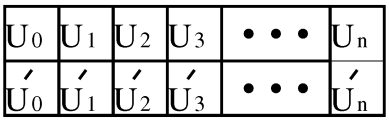


Fig. 7. A simple 2D internal voxel configuration satisfying the connection requirement.

First, it is straightforward to show that optimal reliable path $\in$ NP. Next, if we show that the optimal reliable path is also NP-hard, it is NP-complete. This can be proven by a reduction from minimum set cover. Formally, given a finite set X and a family F of subsets of X such that every element of X belongs to at least one subset in F:

$$X = \bigcup_{U \in F} U, \tag{2}$$

a minimum set cover is a minimum size subset $C \subseteq F$ whose members cover all of X:

$$X = \bigcup_{U \in C} U. \tag{3}$$

For each subset $U_i \in F, 0 \leq i < |F|$, we create a new subset U_i'. Each U_i' contains exactly one new element e_i such that $e_i \notin X$ and $e_i \notin U_j', j \neq i$. Subsequently, we construct a new finite set X' and a new family F' of subsets of X' as follows:

$$X' = X \cup \{e_0, e_1, \ldots, e_{|F|-1}\} \tag{4}$$

$$F' = \bigcup_{0 \leq i < |F|} U_i \cup \bigcup_{0 \leq i < |F|} U_i' = F \cup \bigcup_{0 \leq i < |F|} U_i'. \tag{5}$$

We now construct a 3D volume V based on X' and F'. First, we create a surface voxel set B and establish an one-one mapping function $f : B \leftrightarrow X'$:

$$\forall v \in B \rightarrow (f(v) = e) \wedge (f^{-1}(e) = v), e \in X'. \tag{6}$$

Then, we create an internal voxel set I and establish an one-one mapping function $g : I \leftrightarrow F'$:

$$\forall v \in I \rightarrow (g(v) = U) \wedge (g^{-1}(U) = v), U \in F'. \tag{7}$$

Finally, based on f and g, we map the elements of each subset $U \in F'$ to its associated visible set $S_v, v \in I$:

$$\forall v \in I \rightarrow S_v = \bigcup_{(e \in U)} f^{-1}(e), U = g(v). \tag{8}$$

The next step is to show that the internal voxel set I can be configured in such a way that the optimal reliable path for V is based on the minimum cover set C. To achieve this, we only need to ensure that all the internal voxels corresponding to the new subsets U_i' are connected by themselves and each pair of voxels corresponding to U_i and U_i' are connected. For example, Fig. 7 presents a possible configuration in 2D using 4-connection. For better illustration, $g^{-1}(U_i)$ and $g^{-1}(U_i')$ are simply marked in Fig. 7 as U_i and U_i', respectively. Extending the concept to 3D and other connections is straightforward.

Since a reliable path covers the whole set of surface voxels, the optimal reliable path C^* for V is

$$C^* = C \cup \bigcup_{0 \leq i < |F|} g^{-1}(U_i'), \tag{9}$$

where $C \subseteq F$ is the minimum set cover for F. In other words, we have shown that any instance of minimum set cover can be reduced in polynomial time to an instance of optimal reliable path. This completes the proof that optimal reliable path is NP-complete.

REFERENCES

[1] D. Vining, D. Gelfand, R. Bechtold, E. Scharling, E. Grishaw, and R. Shifrin, "Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality," *Ann. Meeting Am. Roentgen Soc.*, p. 104, 1994.

[2] L. Hong, A. Kaufman, Y. Wei, A. Viswambharan, M. Wax, and Z. Liang, "3D Virtual Colonoscopy," *IEEE Biomedical Visualization,* pp. 26-32, 1995.

[3] R. Robb, "Computed (Virtual) Endoscopy: Development and Evaluation Using the Visible Human Datasets," *Proc. Nat'l Library of Medicine Visible Human Project Conf.,* pp. 19-20, 1996.

[4] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He, "Virtual Voyage: Interactive Navigation in the Human Colon," *Proc. ACM SIGGRAPH,* pp. 27-34, 1997.

[5] R.M. Summers, "Computers in Radiology: Navigational Aids for Real-Time Virtual Bronchoscopy," *Am. J. Radiology,* vol. 168, pp. 1165-1170, 1997.

[6] E. Gobbetti, P. Pili, A. Zorcolo, and M. Tuveri, "Interactive Virtual Angioscopy," *Proc. IEEE Visualization '98,* pp. 435-438, 1998.

[7] W. Lorensen, F. Jolesz, and R. Kikinis, "The Exploration of Cross-Sectional Data with a Virtual Endoscope," *Interactive Technology and the New Medical Paradigm for Health Care,* pp. 221-230, IOS Press, 1995.

[8] G. Rubin, C. Beaulieu, V. Argiro, H. Ringl, A. Norbash, J. Feller, M. Dake, R. Jeffrey, and S. Napel, "Perspective Volume Rendering of CT and MR Images: Applications for Endoscopy," *Radiology,* vol. 199, pp. 321-330, 1996.

[9] M. Brady, K. Jung, H. Nguyen, and T. Nguyen, "Interactive Volume Navigation," *IEEE Trans. Visualization and Computer Graphics,* vol. 4, no. 3, pp. 243-256, July-Sept. 1998.

[10] T. Galyean, "Guided Navigation of Virtual Environments," *Proc. ACM Symp. Interactive 3D Graphics,* pp. 103-104, 1995.

[11] D. Johnson, "Approximation Algorithms for Combinatorial Problems," *J. Computer and System Sciences,* vol. 9, pp. 256-278, 1974.

[12] V. Chvatal, "A Greedy Heuristic for the Set Covering Problem," *Math. Operation Research,* vol. 4, no. 2, pp. 233-235, 1979.

[13] H. Blum, "A Transformation for Extracting New Descriptors of Shape," *Models for Perception of Speech and Visual Form,* W. Wathen-Dunn, ed., pp. 362-380, MIT Press, 1967.

[14] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley-Interscience, 1973.

[15] Y. Zhou and A. Toga, "Efficient Skeletonization of Volumetric Objects," *IEEE Trans. Visualization and Computer Graphics,* vol. 5, no. 3, pp. 196-209, July-Sept. 1999.

[16] D.W. Capson and A.C.-S. Fung, "Connected Skeletons from 3D Distance Transforms," *Proc. Image Analysis and Interpretation,* pp. 174-179, 1998.

[17] N. Amenta, S. Choi, and R. Kolluri, "The Power Crust, Unions of Balls, and the Medial Axis Transform," *Int'l J. Computational Geometry and its Applications,* to appear.

[18] T. Saito and J. Toriwaki, "New Algorithms for Euclidean Distance Transformation of an N-Dimensional Digitized Picture with Applications," *Pattern Recognition,* vol. 27, no. 11, pp. 1551-1565, 1994.

[19] E. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numerische Mathematik,* vol. 1, pp. 269-270, 1959.

[20] A. Kaufman, *Volume Visualization.* Los Alamitos, Calif.: IEEE CS Press, 1991.

[21] F. Durand, G. Drettakis, and C. Puech, "The Visibility Skeleton: A Powerful and Efficient Multi-Purpose Global Visibility Tool," *Proc. ACM SIGGRAPH,* pp. 89-100, 1997.

[22] W.E. Lorensen and H.E. Cline, "Marching Cubes: High Resolution 3D Surface Construction Algorithm," *Computer Graphics (SIGGRAPH '87 Proc.),* vol. 21, pp. 163-169, July 1987.

[23] D. Chen, L. Li, and Z. Liang, "A Self-Adaptive Vector Quantization Method for MR Image Segmentation," *Proc. Int'l Soc. Magnetic Resonance in Medicine Seventh Scientific Meeting and Exhibition,* pp. 22-28, May 1999.
[24] M. Garey and D. Johnson, *Computer and Intractability, A Guide to the Theory of NP-Completeness.* New York: W.H. Freeman and Company, 1979.

Taosong He graduated from Fudan University in 1991, received the MS degree in computer science from the State University of New York at Stony Brook in 1993, and received the PhD degree degree in computer science from the State University of New York at Stony Brook in 1996. He is a researcher working in the Network & Service Management Department at Bell Labs, Lucent Technology. His research interests include network visualization, medical imaging, interactive graphics, virtual reality, volume visualization, and 3D interaction. He is a member of the ACM and the IEEE.

Lichan Hong received the BS degree in computer science from the University of Science and Technology of China in 1990 and the MS and PhD degrees in computer science from the State University of New York at Stony Brook in 1995 and 1997, respectively. From October 1997 to May 2000, he was a member of the technical staff at the Software Production Research Department of Bell Laboratories, Lucent Technologies. Since June 2000, he has been employed as a principal engineer at EnFashion. His current interests include computer graphics, 3D visualization, and physically based simulation. He is a member of the IEEE Computer Society.

Dongqing Chen received the BE degree in applied mathematics (with honors) from the Changsha Institute of Technology, Changsha, China, in 1985 and the PhD degree in probability theory from the Beijing Normal University (with honors), Beijing, China, 1990. He joined the Department of Applied Mathematics, Tsinghua University, Beijing, China, in 1990, where he was promoted to an associate professor in 1993. Since 1998, he has been a research scientist in the Department of Radiology, State University of New York at Stony Brook. His research interests include medical image processing and visualization, pattern recognition, and statistical algorithms.

Zhengrong Liang is currently a professor of radiology, computer science, and biomedical enginerring at the State University of New York at Stony Brook. His major research interests are image reconstruction and segmentation in medical applications. He has contributed significantly to quantitative single photon emission computed tomography by modeling imaging physics and integrating expectation maximumization algorithm into maximum a posteriori probability reconstruction framework. He is one of the few pioneers in developing virtual colonoscopy and has published a series of articles on related topics. He is currently on the editorial board of the *IEEE Transactions on Medical Imaging*, the advisory board of MD OnLine, Inc, and the Board of Directors of Viatronix, Inc.