

Interactive prostate shape reconstruction from 3D TRUS images

Tomotake Furuhashi, Inho Song, Hong Zhang, Yoed Rabin and Kenji Shimada*

Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

(Manuscript Received June 22, 2014; Revised September 8, 2014; Accepted September 11, 2014)

Abstract

This paper presents a two-step, semi-automated method for reconstructing a three-dimensional (3D) shape of the prostate from a 3D transrectal ultrasound (TRUS) image. While the method has been developed for prostate ultrasound imaging, it can potentially be applicable to any other organ of the body and other imaging modalities. The proposed method takes as input a 3D TRUS image and generates a watertight 3D surface model of the prostate. In the first step, the system lets the user visualize and navigate through the input volumetric image by displaying cross sectional views oriented in arbitrary directions. The user then draws partial/full contours on selected cross sectional views. In the second step, the method automatically generates a watertight 3D surface of the prostate by fitting a deformable spherical template to the set of user-specified contours. Since the method allows the user to select the best cross-sectional directions and draw only clearly recognizable partial or full contours, the user can avoid time-consuming and inaccurate guesswork on where prostate contours are located. By avoiding the usage of noisy, incomprehensible portions of the TRUS image, the proposed method yields more accurate prostate shapes than conventional methods that demand complete cross-sectional contours selected manually, or automatically using an image processing tool. Our experiments confirmed that a 3D watertight surface of the prostate can be generated within five minutes even from a volumetric image with a high level of speckles and shadow noises.

Keywords: Shape reconstruction; Prostate; TRUS; Ultrasound; Image processing

1. Introduction

The three-dimensional (3D) shape data of the prostate plays a key role in surgical interventions for diagnosing and treating prostate cancer, including biopsy, brachy therapy, high-frequency ultrasound (HIFU), and cryosurgery. For example, the prostate volume has been used in conjunction with prostate-specific antigen (PSA) for cancer diagnosis, and this volume can be calculated more accurately from 3D shape data than by using planimetry measurements (i.e., cross-sectional imaging across a plane of the prostate). The 3D-shape data is pivotal also in the pre-surgical planning, intra-operative image-based guidance, and post-surgical monitoring. In brachytherapy and cryosurgery, planning the placement of radioactive seeds or cryoprobes is a layout/coverage optimization problem that requires the 3D-shape data of the prostate. For HIFU, planning the sequential application of localized heating is a coverage optimization problem of a different kind that essentially requires the same 3D-shape data of the organ.

In addition to the conventional usage for cancer diagnosis and therapy, the 3D prostate-shape data can also be used in a

computer-assisted cryosurgical training system. Based on the previous work on the thermal simulation of a cryosurgical procedure and cryoprobe layout optimization [1, 2], the current research team has been building a computer-assisted trainer for cryosurgeons [3]. In this system, the 3D-shape data of the prostate is necessary for the visualization of the anatomy and cryoprobes, the calculation of the temperature distribution during a cryosurgical procedure, and the visualization of frozen regions against the target region.

In order to generate the 3D-shape data of the prostate accurately and efficiently for the computer-assisted training system for cryosurgery, a two-step semi-automated method has been developed. The proposed method takes as input a 3D transrectal ultrasound (TRUS) image – a set of parallel 2D cross-sectional images positioned in 3D space – and generates a 3D surface model of the prostate. 3D TRUS images can be generated by translating or rotating a 1D ultrasound transducer along an axis [4]. The 3D TRUS images used in this study were generated by Fenster and co-workers using a proprietary imaging system [5]. To make the output 3D-shape data suitable for the thermal simulation performed in the computer-assisted training system, the 3D-shape data must be watertight.

The comprehensibility of our input 3D TRUS image ranges from low to medium due to noises such as speckles

*Corresponding author. Tel. : +1-412-268-3614, Fax. : +1-412-268-2908

E-mail address: shimada@cmu.edu

© Society of CAD/CAM Engineers & Techno-Press

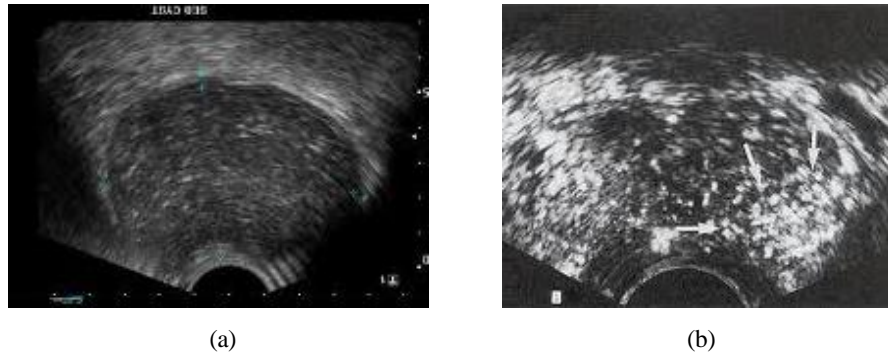


Figure 1. Typical TRUS images – part of an image is often not comprehensible due to speckle and shadow noises: (a) medium quality TRUS image, (b) low quality TRUS image.

and shadows, as shown in Figure 1. Poor image quality is a common weakness of ultrasound imaging, compared to magnetic resonance imaging (MRI) and computerized tomography (CT). Although MRI and CT images are clearer and more comprehensible, those imaging modalities are more time-consuming and expensive, and U.S. medical insurance companies typically do not reimburse for them. TRUS imaging is thus the choice of practice, which is used as input for our 3D prostate-shape generation. The proposed two-step method for reconstructing 3D shapes is specifically designed to work well with low quality TRUS images, where only partial prostate contours are comprehensible. The first step of the proposed method relies upon the user's partial contour specification – the user is asked to select 10-15 easier-to-comprehend images, on which the user specifies partial or full contours. This can be completed within several minutes. Then, a final watertight 3D prostate shape is automatically generated within 20-30 seconds by fitting a deformable template shape to the specified contours.

The proposed method is simpler and quicker than the conventional semi-manual method that requires the user to specify a set of complete closed contours in many more images, even when some images may be of poor quality. Figure 2

shows an example of such a conventional method of 3D shape generation using a commercial product, 3D Doctor [6]. The input was a 3D image consisting of 69 2D TRUS images. It took over 30 minutes for the user to specify a closed full contour for each of the 69 images. The system then created a 3D watertight shape automatically by joining adjacent cross-sectional contours, as illustrated in Figure 2(b). Although this commercial product offers an automated contour generation based on manually specified contours on several images, the method did not work well for our 3D image due to its poor image quality. The final 3D shape is not smooth due to inconsistent contour specifications on adjacent cross-sectional images. While the output shape captures the general size and shape of the prostate, the shape is not natural due to the noisy surface texture caused by non-coherent contour specification, making the 3D shape data unsuitable for surgical planning and training applications.

There have been several published methods on automated or semi-automated reconstruction of 3D prostate shapes [20-23]. These methods typically create contours by using automated image processing and/or active contours -- physically based curve models to create a contour that is of a specified level of smoothness. After a set of contours is specified, some

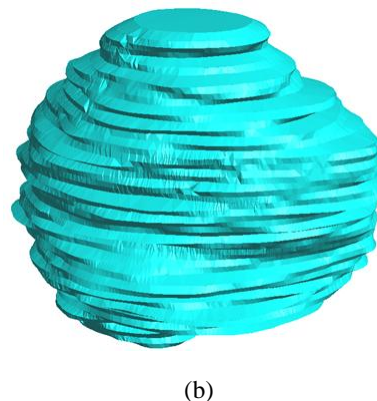
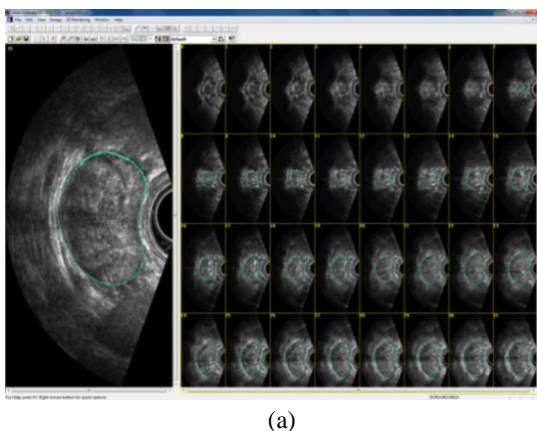


Figure 2. A prostate 3D geometry reconstructed from a set of manually selected full contours is not smooth and may not be watertight: (a) manual selection of full contours on 69 images, (b) prostate 3D geometry reconstructed by a conventional method.

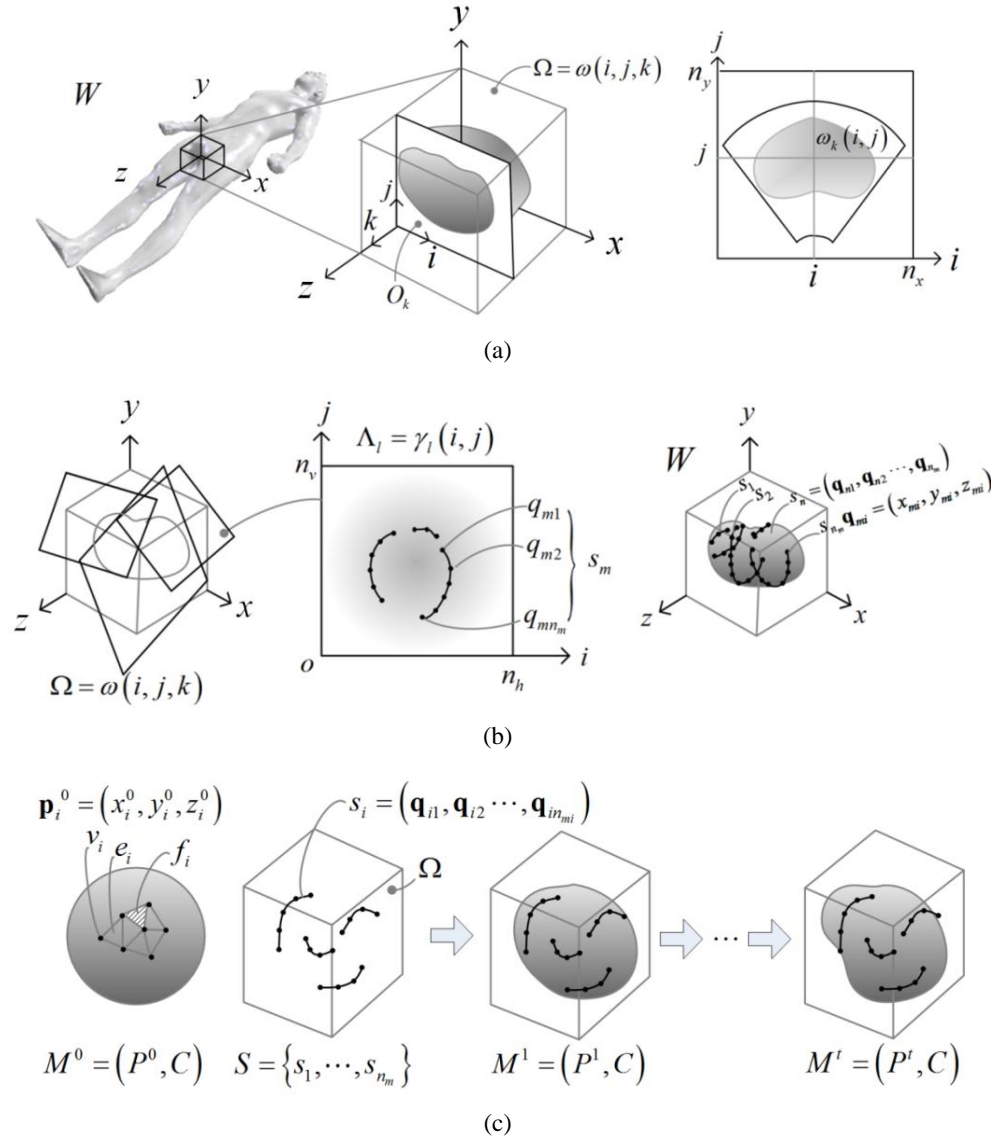


Figure 3. Proposed two-step approach to generating the 3D surface model of a prostate from a 3D TRUS image: (a) input 3D TRUS images, (b) step 1: The user specifies partial contours on cross-sectional images with arbitrary orientations, (c) step 2: The system automatically deforms a template mesh to fit the partial contours

of the methods fit a deformable 3D model to the set of contours to yield a watertight 3D prostate shape. In contrast to these conventional approaches which require a specification of full contours in the original 2D TRUS images, our new method presented in this paper allows the user to select the best cross-sectional directions and draw only clearly recognizable partial contours. The user can thus avoid time-consuming and inaccurate guesswork on where prostate contours are located. The proposed method also yields more accurate prostate shapes than conventional methods by avoiding the usage of noisy, incomprehensible portions of the TRUS image.

The 3D prostate shape reconstruction problem presented in this paper can be stated as follows:

Input:

A 3D TRUS image of a prostate: a set of gray-scale 2D TRUS images arranged sequentially in 3D space, forming a volumetric image.

Output:

A 3D surface model of the prostate as a 3D polygonal mesh.

Requirements:

- Target accuracy: 2 mm for contour specification and smooth interpolation
- Processing time: five minutes in total
- Water-tightness of the output surface model

Realistic and natural shape suitable for surgical planning

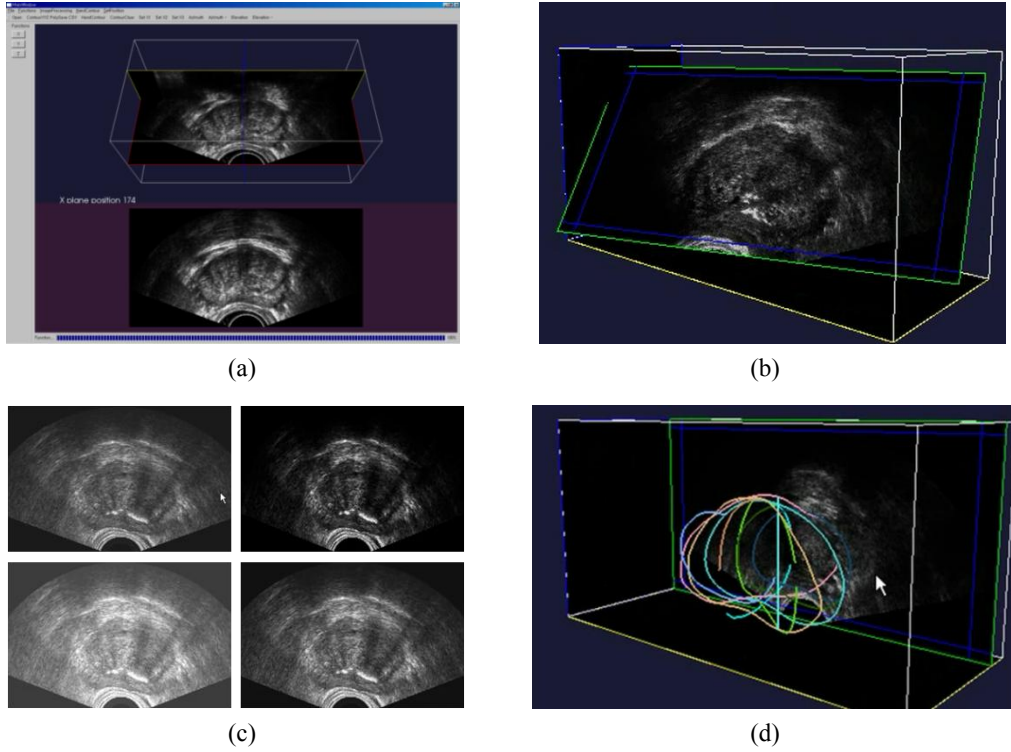


Figure 4. Developed GUI for Step 1 cross-sectional image navigation, adjustment of intensity and contrast, and specification of partial/full contours: (a) 3D (upper) and 2D (lower) cross-sectional image views, (b) cross-sectional views, (c) adjustment of contrast and intensity, (d) specified 3D contours.

and training applications.

The rest of this paper is organized as follows. Section 2 presents the outline and the detail of the proposed two-step method. Section 3 shows the results of three experimental cases; the first validates the first step of the proposed method, the second validates the second step of the proposed method, and the third validates the effectiveness of the total process. Section 4 presents the conclusions and future work.

2. Proposed computational method for 3D shape reconstruction

The proposed method takes as input a 3D TRUS image and generates a 3D surface model of the prostate as a 3D polygonal mesh by using the two-step approach illustrated in Figure 3: (1) the user specifies partial contours on cross-sectional images with arbitrary orientations; and (2) the system automatically deforms a template mesh to fit the partial contours.

Throughout the entire process, the position and orientation of the 3D TRUS images, cross-sectional images, partial contours, the initial template geometry, and the final 3D surface geometry are represented in the coordinate system attached to the patient's body. This world coordinate system, denoted as W , is oriented as illustrated in Figure 3(a), that is, the xy -plane represents the traverse plane, the xz -plane the sagittal plane, and the yz -plane the coronal plane. A 3D TRUS image is a sequence of n_k 2D images, O_k , $k = 1, \dots, n_k$, that are paral-

lel to the traverse plane. The transformation from the world coordinate system, W , to each 2D image coordinate system, wO_k , is provided from the 3D TRUS imaging system. A 3D TRUS image is a volumetric image represented by a $3 \times 3 \times 3$ matrix that stores a grayscale intensity of voxels: $\Omega = \omega(i, j, k)$, $i = 1, \dots, n_i, j = 1, \dots, n_j, k = 1, \dots, n_k$, where n_i, n_j , and n_k are the number of voxels in the x, y and z directions respectively. The size of a voxel is defined by its length in the x, y , and z directions: $\Delta x, \Delta y$, and Δz .

The input 3D TRUS image is processed in two steps. The first step, Step 1, is an interactive process where the user navigates through the 3D TRUS image by taking cross sections in arbitrary positions and orientations, freely adjusts the cross-sectional images' contrast and intensity, and specifies partial or full contours of the prostate. As illustrated in Figure 3(b), a set of cross-sectional images is denoted as $\Lambda_l = \gamma_l(i, j)$, $i = 1, \dots, n_h, j = 1, \dots, n_v, l = 1, \dots, n_l$, where n_h and n_v are the number of pixels in the x and y directions respectively, and n_l is the number of cross-sectional images. The coordinate transformation from the world coordinate to cross-sectional image coordinate, Λ_l , is denoted as ${}^w\Lambda_l$. On each cross-sectional image, the user may specify multiple partial contours. All the partial contours are transformed to 3D line segments represented in the world coordinate system. This set of n_m 3D contours is denoted as $S = \{s_1, \dots, s_{n_m}\}$ where each contour consists of a sequence of n_{mi} 3D points, $s_i = (q_{i1}, \dots, q_{in_{mi}})$, $i = 1, \dots, n_{mi}$.

The second step, Step 2, is an automated process where template geometry, represented by a polygonal mesh, is deformed to fit the user-specified contours. The mesh is represented as a pair, $M^0 = (P^0, C)$, where P^0 is a set of n_v points in \mathbb{R}^3 , $P^0 = \{p_1^0, \dots, p_{n_v}^0\}$, and C is topological connectivity information that contains a set of n_v vertices $V = \{v_1, \dots, v_{n_v}\}$, a set of n_e edges $E = \{e_1, \dots, e_{n_e}\}$, and a set of n_f faces $F = \{f_1, \dots, f_{n_f}\}$. Point $p_i = (x, y, z)$ represents the geometric position of Vertex v_i in the template mesh. In Step 2 the template mesh is deformed iteratively to fit the contours. The mesh after the i -th iteration is denoted as $M^i = (P^i, C)$, where P^i represents the updated position of vertices. Note that the topological connectivity of the mesh, C , remains the same throughout Step 2. The template mesh will be progressively morphed into different meshes, $M^0 \rightarrow M^1 \rightarrow \dots \rightarrow M^i \rightarrow \dots \rightarrow M^t$, until it reaches the final mesh, $M^t = (P^t, C)$.

The rest of Section 2 presents further details of Steps 1 and 2.

2.1 Step 1: Manual specification of partial contours

With the GUI application developed for Step 1, the user can: (1) navigate through the 3D TRUS image freely by viewing cross-sectional images at arbitrary positions and orientations; (2) adjust the image intensity and contrast at any time of the navigation; and (3) specify partial or full contours on cross-sectional images. The system is built so that all the operations can be performed intuitively and easily using a standard PC mouse or digitizer.

As shown in Figure 4(a), the system uses two graphics windows to display simultaneously: (1) a 3D view of across-sectional image and 3D contours; and, (2) a 2D view of a cross-sectional image and 2D contours. All the image processing calculations are performed by in-house software built with the VTK and ITK Toolkit [7, 8].

It should be noted that the proposed method would not work well if most of the partial contours are extracted from only a portion of the shape. This problem is avoided by giving the user real-time visual feedback as he/she adds each contour. The 3D view can be rotated freely so the user can avoid specifying contours only for a portion of the shape.

2.1.1 Real-time cross-sectional image display

Figure 4(b) shows examples of a cross-sectional image.

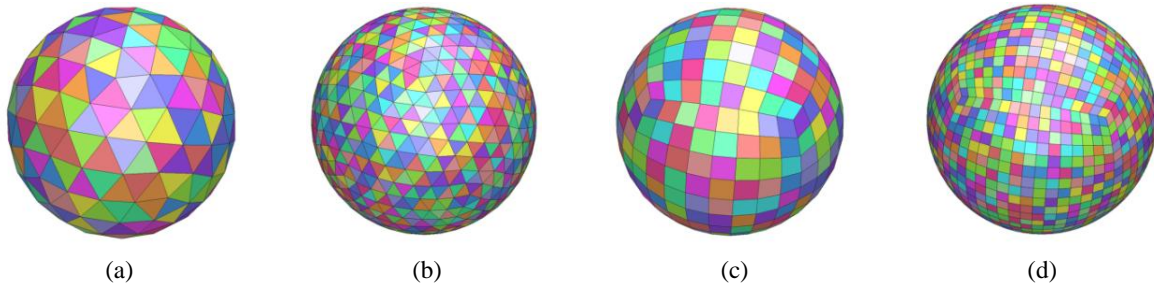


Figure 5. Template polygonal models of the prostate: (a) triangular template with 320 elements, (b) triangular template with 1,280 elements, (c) quadrilateral template with 384 elements, (d) quadrilateral template with 1,536 elements.

The user controls the position and orientation of the cross-sectional plane by moving the mouse in both horizontal and vertical directions. The cross-sectional image is updated in real-time, with a refreshing rate of 5-10 frames per second – the actual refreshing rate depends upon the performance of the graphics card installed on the PC. This range of refreshing rate of a cross sectional image is sufficient for a stress-free interaction with a 3D TRUS image.

2.1.2 Image intensity and contrast adjustment

The current GUI system allows the user to adjust the intensity and contrast of a cross-sectional image in real-time, by a combination of a keystroke and mouse movement (see Figure 4(b)). The intensity and contrast adjustment is similar to the function that is available in commercial/non-commercial image processing software packages, such as Adobe Photoshop [9] and IrfanView [10]. It is our future plan to enhance the image processing capability of the system to include fairing and de-noising operations [11].

2.1.3 Specification of partial / full contours

Figure 4(d) shows a set of partial contours specified by the user on 10-15 cross-sectional images. The users can display a 3D view of the partial contours anytime during Step 1 in order to decide if a sufficient number of contours has been specified or not.

2.2 Step 2: Automated smooth interpolation of partial contours

While the GUI and functionality design of Step 1 is relatively straightforward, the selection of the best mesh deformation method for Step 2 requires extensive benchmarking and comparison of multiple candidates.

The key requirements to consider in designing the computational method for Step 2 are:

- (1) the method must interpolate a limited number of partial contours to form a natural prostate shape;
- (2) the surface of the generated 3D prostate shape must be smooth and realistic;
- (3) the computational cost for converging to the final shape must be low enough to complete in about a minute with a standard PC; and,
- (4) the method must create a water-tight polygonal mesh

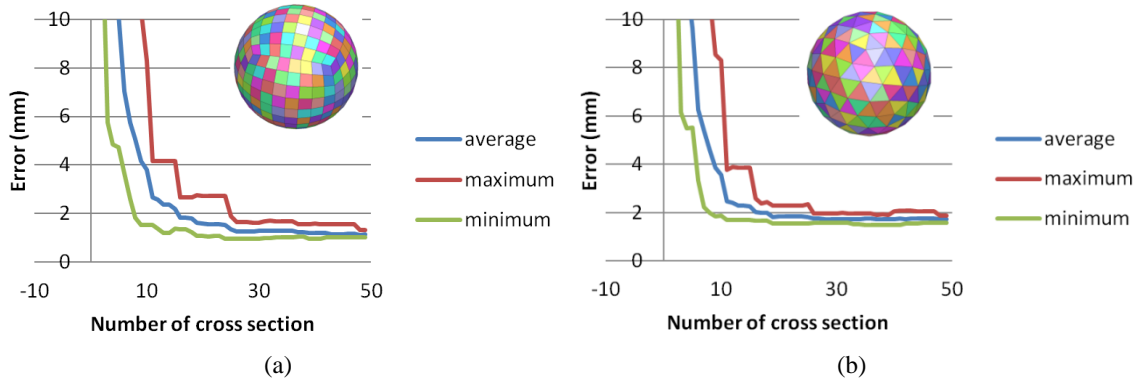


Figure 6. Error convergence with tri-mesh and quad-mesh templates. A quad-mesh template yields 30-40% smaller error compared with a tri-mesh template having a similar number of mesh elements: (a) error convergence with a tri-mesh template with 320 elements, (b) error convergence with a quad-mesh template with 384 elements.

with no self-intersections.

2.2.1 Template polygonal mesh

In order to select the type and appropriate level of detail of the initial template mesh, M^0 , we conducted computational experiments using two types of polygonal meshes – tri mesh and quad mesh – of different mesh sizes. Several meshes were created for benchmarking with the number of mesh elements ranging from 200 to 2,000, examples of which are shown in Figure 5. Figure 6 compares how a triangular-mesh template with 320 elements and a quadrilateral-mesh template with 384 elements each minimize the error between the final deformed mesh, M^1 , and the correct 3D geometry of the prostate. In order to compare the performance of template polygonal meshes of different numbers of nodes and elements, we

used one of the reconstructed 3D shapes as the true 3D shape. Note that this 3D shape can be any 3D prostate shape for the purpose of the comparison.

The result of the computational experiment determined that:

- (1) M^0 of 300-400 elements would have the fewest possible number of mesh elements that yields 1.5 mm or less error (Figure 6);
- (2) A quad mesh yields a 30-40% more accurate result compared with a tri mesh having approximately the same number of elements.

Based on these observations, we chose to use the quad-mesh shown in Figure 5(c) as the initial template mesh.

2.2.2 Comparison of mesh-smoothing methods

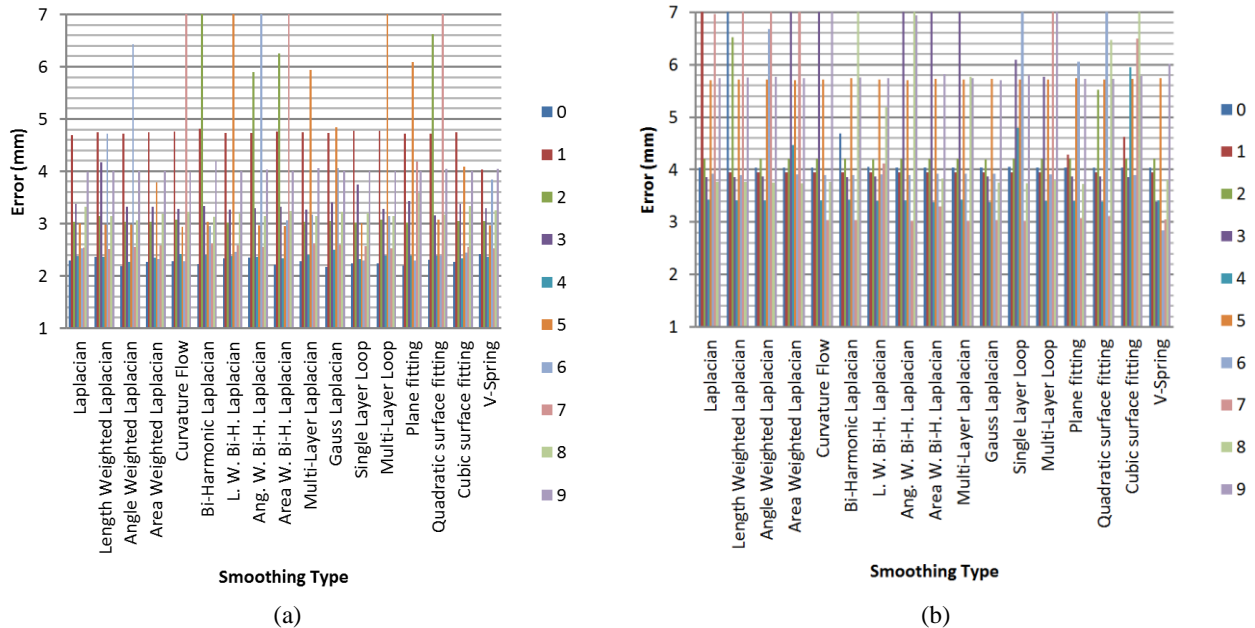


Figure 7. 3D error comparison with various smoothing schemes: (a) error comparison with a 320-element tri-mesh template, (b) error comparison with a 1,280-element tri-mesh template.

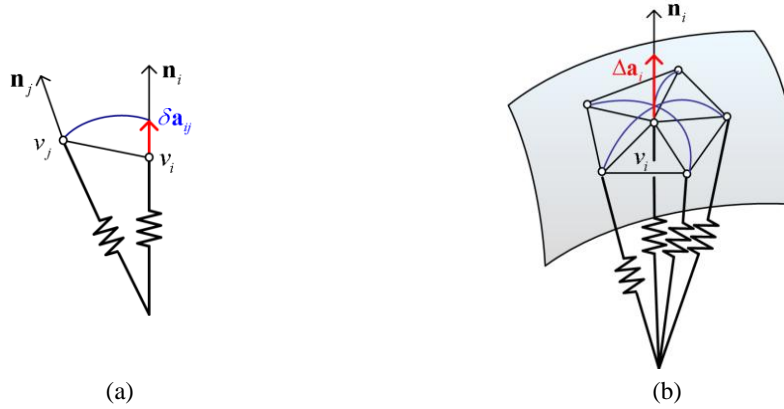


Figure 8. V-Spring uses a spring force that acts in the normal direction to minimize curvature variation: (a) displacement of v_i due to its neighbor, v_j , (b) net displacement due to all neighbors.

To select the smoothing method that best satisfies the four requirements of Step 2, discussed at the beginning of Section 2.2, we implemented 17 mesh-smoothing schemes and compared the accuracy of each method. These smoothing schemes were: (1) Laplacian, (2) length-weighted Laplacian, (3) angle-weighted Laplacian, (4) area-weighted Laplacian, (5) curvature flow, (6) bi-harmonic Laplacian, (7) length-weighted bi-harmonic Laplacian, (8) angle-weighted bi-harmonic Laplacian, (9) area-weighted bi-harmonic Laplacian, (10) multi-layer Laplacian, (11) Gauss Laplacian, (12) single-layer Loop, (13) multi-layer Loop, (14) plane fitting, (15) quadratic surface fitting, (16) cubic surface fitting, and (17) V-spring. These methods are described in literatures [12–20].

The comparison of the mesh-smoothing methods was performed as follows. A pre-defined polygonal model of a prostate was designated as the ground-truth shape. We then created different sets of 3D contours by taking cross sections of different orientations including the directions parallel to the xy -plane, yz -plane and zx -plane of the 3D TRUS image coordinate frame, O . We used two sets of 3D contours, one with 13 cross-sectional images and the other with 25 cross-sectional images. For each of the 17 mesh-smoothing methods, therefore, we had six error numbers, each of which measured the average error between the ground-truth shape and the shape generated by deforming the template mesh to fit the input 3D contours. Although there are slight variations between two different input 3D contour sets, we could still

see which method generally yields the smallest error. According to the results displayed in Figure 7, we concluded that the V-Spring method [18, 19] performs the best, and hence it is our method of choice for mesh smoothing. It is not very surprising that V-Spring performs well for our application, as the method is designed to minimize curvature variation, and our target organ, the prostate, is a smooth round shape with no acute curvature change. Details of the V-Spring method are presented in the next section.

2.2.3 Proposed mesh smoothing method–V-Spring

As concluded from the comparison study presented in Section 2.2.2, we chose to use the V-Spring method, originally proposed by Yamada et al. [18, 19] for generating a smooth polygonal mesh of the prostate. This section presents an overview of the V-Spring method and explains how it was adapted to generate a 3D prostate shape for the cryosurgical training application.

V-Spring is a computational method for generating a smooth polygonal surface by iteratively updating a mesh shape with two types of internal spring forces applied to vertices: (1) a spring force that acts in the normal direction to minimize the curvature variation, and (2) a spring force that acts in the tangential direction to optimize vertex distribution. For the first type of internal force, a spring is attached to each of the vertices, $V = \{v_1, \dots, v_{n_v}\}$, of the initial mesh M^0 . For a convex shape, these neighboring springs form a “V” shape, thus giving the name to the method. The spring length ap-

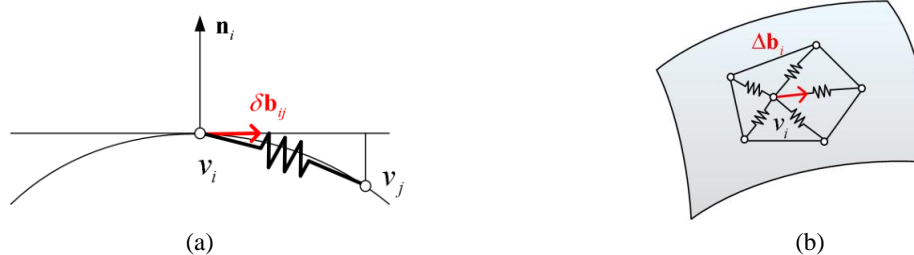


Figure 9. V-Spring uses a spring force that acts in the tangential direction to optimize vertex distribution: (a) displacement of v_i due to its neighbor v_j , (b) net displacement due to all neighbors.

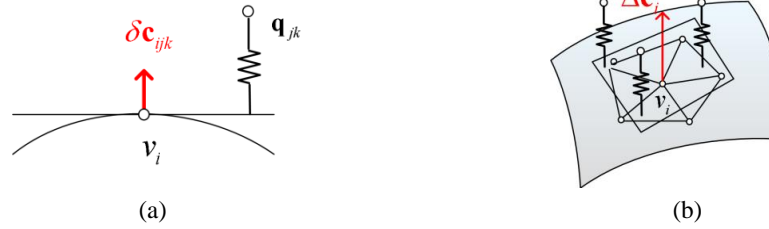


Figure 10. V-Spring uses spring forces for fitting, or constraining, the final surface to approximate the set of partial contours: (a) displacement of v_i due to its neighbor v_j , (b) net displacement due to all neighbors.

proximately represents the local curvature.

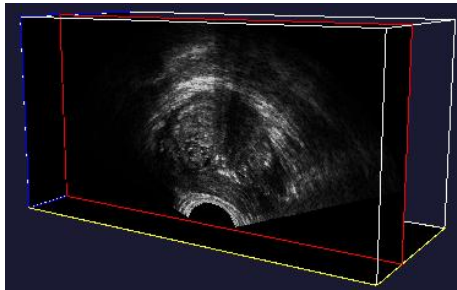
During the iterative mesh deformation process, the springs work together to equalize their lengths to minimize the curvature variation of the surface, as illustrated in Figure 8. Based on the spring model, in each deformation iteration the displacement of Vertex v_i due to its neighboring vertex, v_j , is calculated as followed (see [18, 19] for details):

$$\delta \mathbf{a}_{ij} = \frac{1}{\|\mathbf{p}_j - \mathbf{p}_i\|} \left[\frac{(\mathbf{p}_j - \mathbf{p}_i) \cdot (\mathbf{n}_i + \mathbf{n}_j)}{1 + (\mathbf{n}_i \cdot \mathbf{n}_j)} \right] \mathbf{n}_i, \quad (1)$$

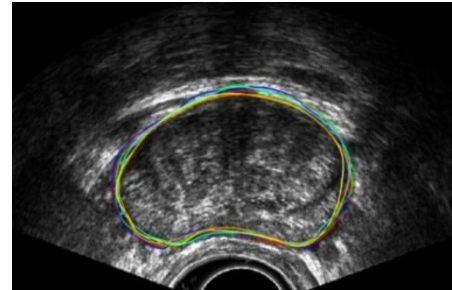
where \mathbf{p}_i and \mathbf{p}_j are the positions of Vertices v_i and v_j , respectively, and \mathbf{n}_i and \mathbf{n}_j are the unit normal vectors at Vertices v_i and v_j , respectively. The total normal displacement, $\Delta \mathbf{a}_i$, of v_i due to all of its neighboring vertices is then calculated as the vector sum of all the displacements:

$$\Delta \mathbf{a}_i = \sum_{j=1}^m \delta \mathbf{a}_{ij}, \quad (2)$$

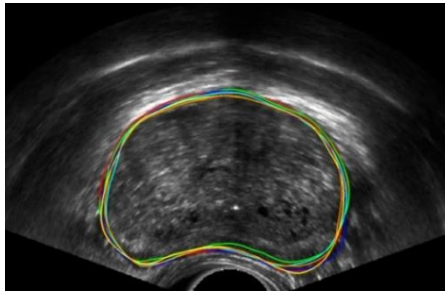
here m is the number of neighboring vertices of Vertex v_i .



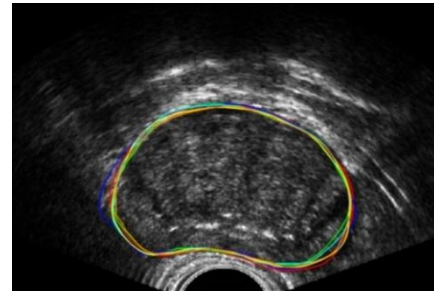
(a)



(b)



(c)



(d)

	Prostate 1 (Healthy)	Prostate 2 (Cancerous)	Prostate 3 (Cancerous)
Average difference	1.1 mm	1.1 mm	1.0 mm
Maximum difference	2.7 mm	2.8 mm	3.0 mm

(e)

Figure 11. Manually specified contours deviate approximately 1 mm on average with a maximum deviation of 2.5-3.0 mm, if the user is asked to plot a complete contour: (a) 3D view provided with cross-sectional images, (b) selected contours – Prostate 1 (healthy), (c) Selected contours – Prostate 2 (Cancerous), (d) selected contours – Prostate 3 (Cancerous), (e) consistency of cross-sectional contours manually specified by five operators.

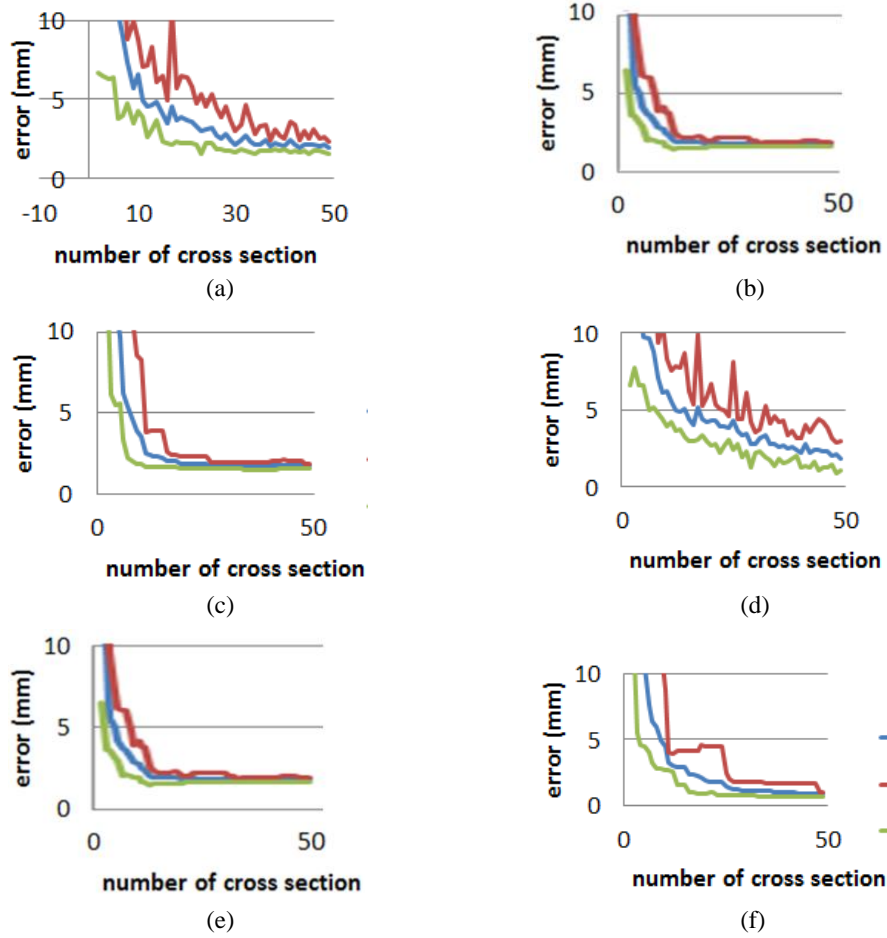


Figure 12. Error convergence with different types of input contours: (a) full contours, yz-planes, 320 tri elements, (b) full contours, random planes, 320 tri elements, (c) partial contours, random planes, 320 tri elements, (d) full contours, yz-planes, 1,280 tri elements, (e) full contours, random planes, 1,280 tri elements, (f) partial contours, random planes, 1,280 tri elements.

In addition to the normal displacement, the V-Spring method also defines the second type of springs and internal forces to optimize the vertex distribution, as illustrated in Figure 9. The process is a modified Laplacian smoothing process in which each vertex is connected to its neighboring vertices by a linear spring of zero neutral length. A vertex is moved toward the bary center of its neighbors to maintain a uniform vertex distribution. The modification from the original Laplacian smoothing is to limit the motion of a vertex to the tangential direction. The tangential displacement due to one of the neighboring vertices is calculated as

$$\delta \mathbf{b}_{ij} = \frac{1}{m} [\mathbf{p}_j - \mathbf{p}_i - [(\mathbf{p}_j - \mathbf{p}_i) \cdot \mathbf{n}_i]] \mathbf{n}_i, \quad (3)$$

where m is the number of neighboring vertices. The total tangential displacement, $\Delta \mathbf{b}_i$, is then calculated as the vector sum of all the displacements:

$$\Delta \mathbf{b}_i = \sum_{j=1}^m \delta \mathbf{b}_{ij}.$$

While the above-mentioned two types of springs and internal forces make the final surface shape smooth, a third type of springs and external forces is used for fitting, or constraining, the final surface to approximate the set of partial contours, $S = \{s_1, \dots, s_{n_m}\}$. Recall that each contour consists of a sequence of n_{mi} 3D points,

$$s_i = (\mathbf{q}_{i1}, \dots, \mathbf{q}_{in_{mi}}), i = 1, \dots, n_m \quad (4)$$

Although the V-Spring method allows direct constraints by which the final surface interpolates exactly the set of contour points, this is not appropriate for our application because the user's input of the partial contours in Step 1 is not exact and may be slightly contradictory. We thus adopt indirect constraints by which the final surface approximates the contour points but does not interpolate them. In this way, if the input contour points are contradictory, an average solution is created. The surface is fitted to the partial contours by defining springs that move the surface in the normal direction. The displacement of Vertex v_i due to a spring that connects the vertex and one of the contour points, \mathbf{q}_{jk} , is calculated as:

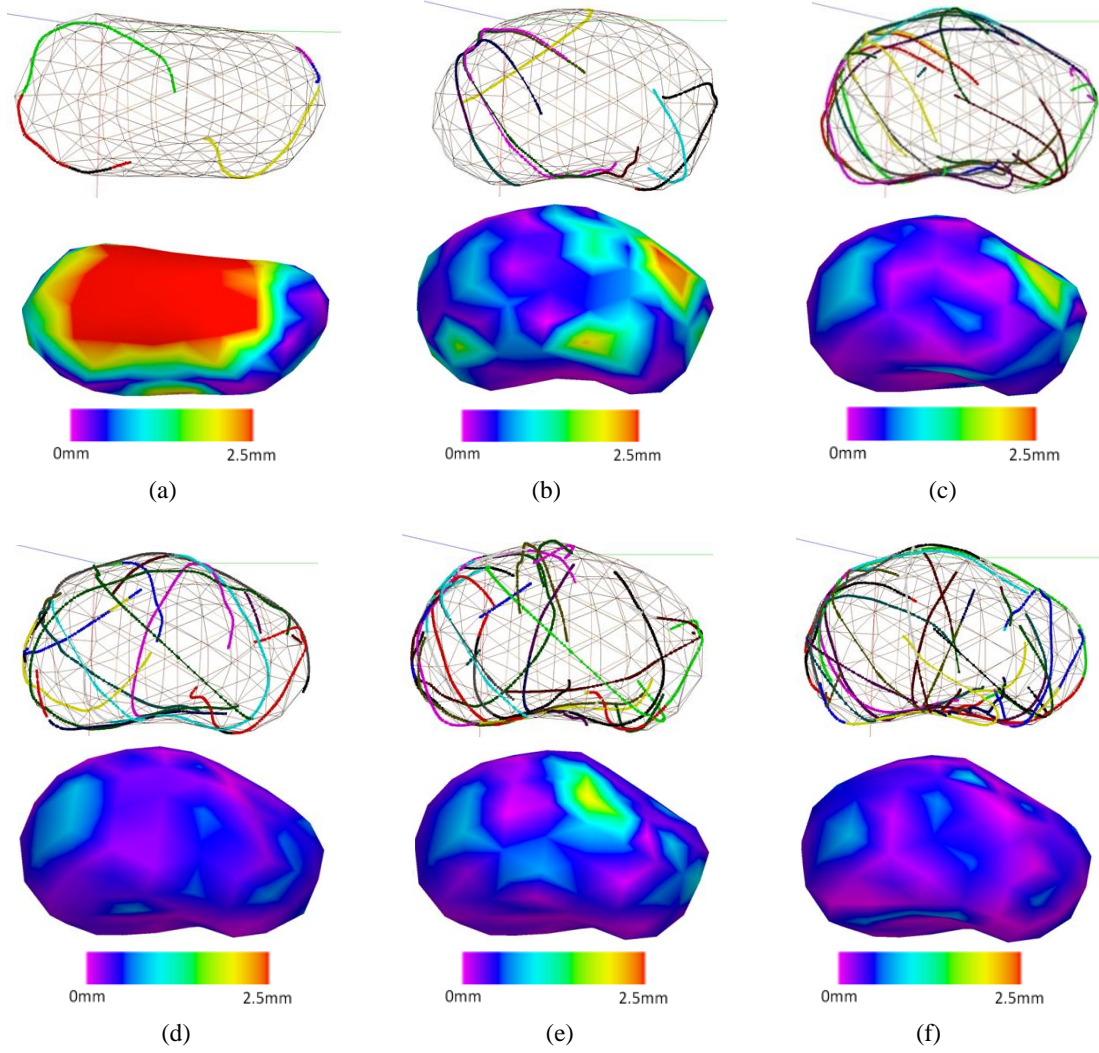


Figure 13. Shape convergence with multiple, randomly selected partial contours using Scheme 3: (a) 5 partial contours, (b) 10 partial contours, (c) 15 partial contours, (d) 20 partial contours, (e) 25 partial contours, (f) 30 partial contours.

$$\delta \mathbf{c}_{ijk} = k_{jk} [(\mathbf{q}_{jk} - \mathbf{p}_i) \cdot \mathbf{n}_i] \mathbf{n}_i \quad (5)$$

where k_{jk} denotes a weight assigned to \mathbf{q}_{jk} and represents a weighted component of vector $(\mathbf{q}_{jk} - \mathbf{p}_i)$ along the normal vector, \mathbf{n}_i . In each deformation iteration, these two displacements are combined to update the positions and normals of each vertex, morphing the initial mesh, M^0 , iteratively to the final mesh, M^t . The total displacement in the normal direction due to the external forces from all the contour points is then calculated as a weighted average:

$$\Delta \mathbf{c}_i = \frac{\sum_{j=1}^n \sum_{k=1}^{n_{mi}} w_{jk} \delta \mathbf{c}_{ijk}}{\sum_{j=1}^n \sum_{k=1}^{n_{mi}} w_{jk}} \quad (6)$$

where w_{jk} are weights for averaging. In the current implemen-

tation, the weights are calculated as the inverse of the distance from Vertex v_i to the foot of the perpendicular of contour point \mathbf{q}_{jk} onto the tangent plane (see Figure 10):

$$w_{jk} = [(\mathbf{q}_{jk} - \mathbf{p}_i) - [(\mathbf{q}_{jk} - \mathbf{p}_i) \cdot \mathbf{n}_i] \mathbf{n}_i]^{-1} \quad (7)$$

With the two internal spring forces combined with the external force, the total displacement of Vertex v_i in each deformation iteration is defined as the sum of three displacements:

$$\Delta \mathbf{c}_i = \Delta \mathbf{a}_i + \Delta \mathbf{b}_i + \Delta \mathbf{c}_i \quad (8)$$

The iterative deformation morphs the initial template mesh, M^0 , into a sequence of meshes, M^1, M^2, M^3, \dots , until the change of the mesh's shape becomes less than a given threshold. The behavior of the V-Spring model changes depending on the balance among the three types of springs. For more details, see [18, 19].

3. Validation of the proposed shape-reconstruction method

Three case studies were conducted to validate the proposed shape-reconstruction method. The goal of each case study was as follows:

Case Study 1: to confirm the effectiveness of the interactive GUI system developed for Step 1 and the consistency of contour plotting across different users.

Case Study 2: to measure the accuracy, convergence, and computational cost of the V-Spring-based deformable surface fitting process of Step 2.

Case Study 3: to demonstrate the performance of the entire 3D shape reconstruction process consisting of Step 1 and

Step 2.

3.1 Case study 1: Interactive contour specification in Step 1

In Step 1, using the interactive GUI system, the user adjusts, navigates through, and visualizes the input TRUS image to specify partial contours. The goal of Case Study 1 was two-fold: (1) to confirm that the developed GUI system is easy-to-use and efficient enough to specify a set of 10~15 cross-sectional planes and 20~30 partial contours within several minutes; and (2) to measure the consistency of contour plotting across different users and confirm that contours can be specified consistently within 2 mm for mid to high quality image regions.

Experiments were conducted as follows. For each of four

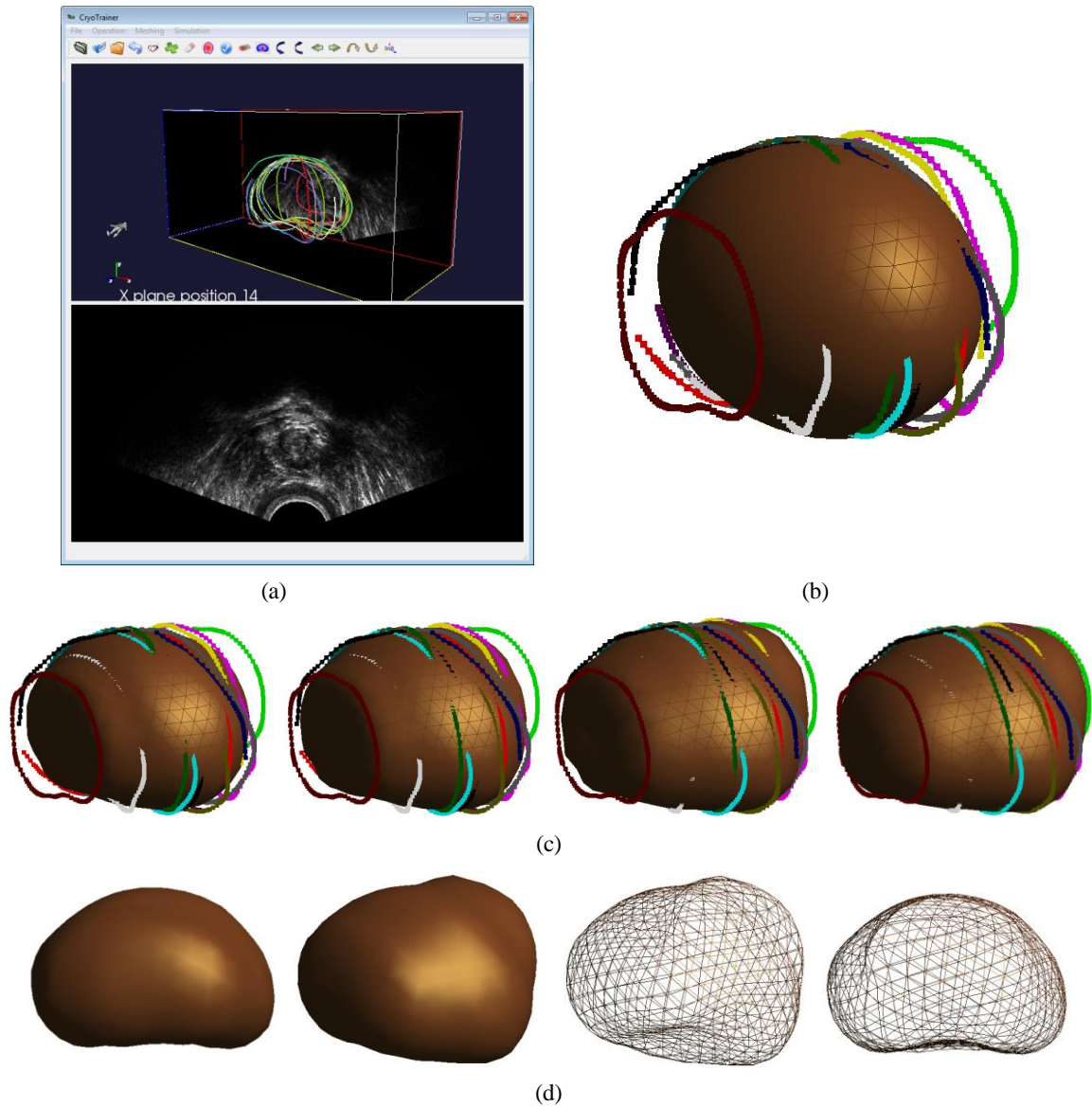


Figure 14. Case Study 3: the entire shape reconstruction process yields a smooth, realistic, and water-tight 3D shape of the prostate: (a) step 1: manual specification of contours, (b) low quality TRUS image, (c) step 2: deformable surface fitting, $M^0 \rightarrow M^1 \rightarrow \dots \rightarrow M^i \rightarrow \dots \rightarrow M^l$, (d) generated 3D shape of the prostate.

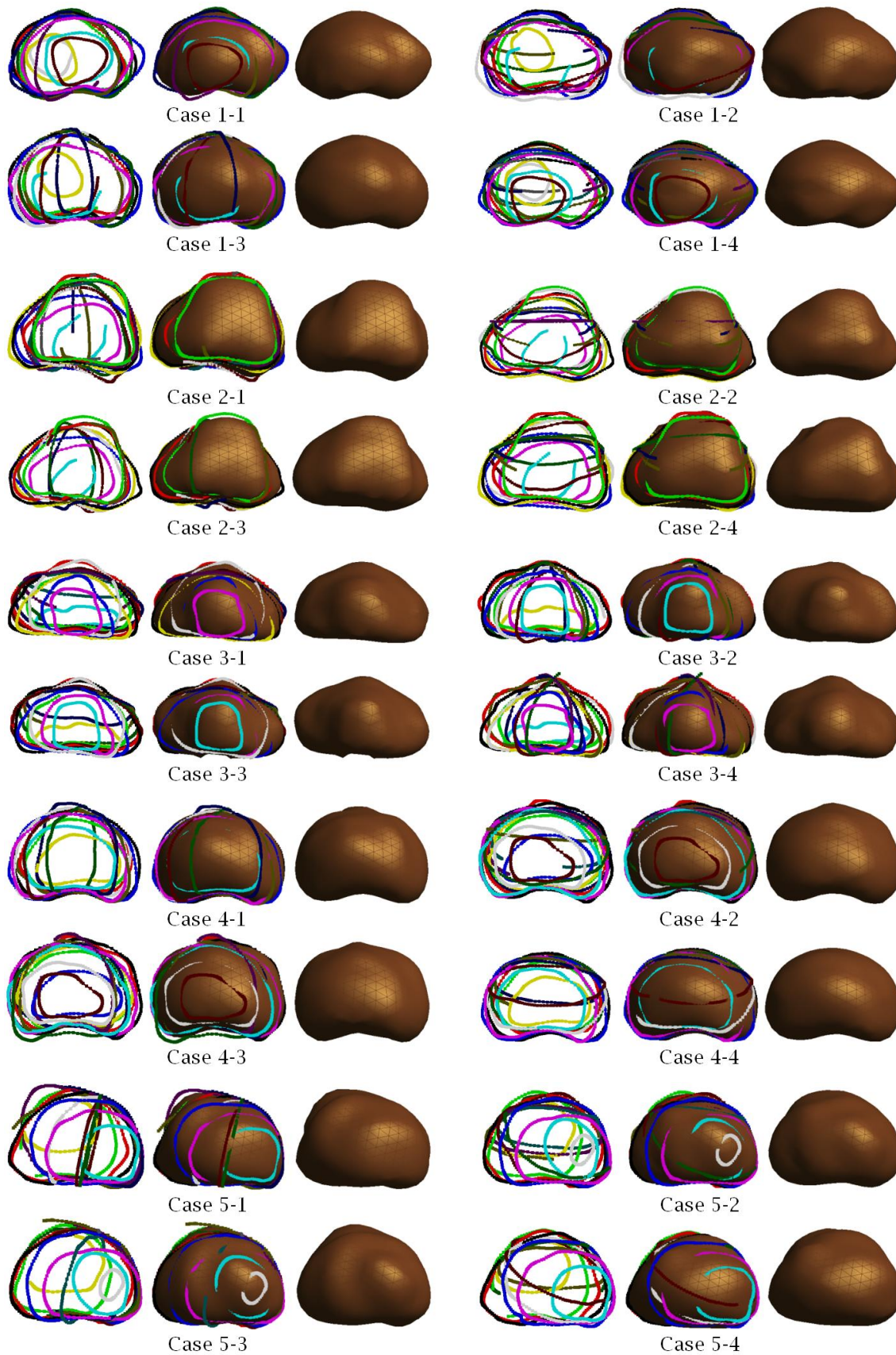


Figure 15. Examples of 3D prostate shapes created by the proposed shape reconstruction method.

randomly selected cross-sectional images, five users were asked to specify a closed contour. As the input TRUS image was fairly noisy with many speckles and shadows, the users were forced to guess about certain regions in drawing contours. Figure 11 shows the color-coded contours specified by the five different users. It is observed that for some regions

where the image is clearer, the user-specified contours are more consistent and the average deviation is within 1.5 mm. For other regions where the image is less clear, the user-specified contours are less consistent, and the average deviation exceeds 2 mm.

The results of Case Study 1 suggest that the contour selec-

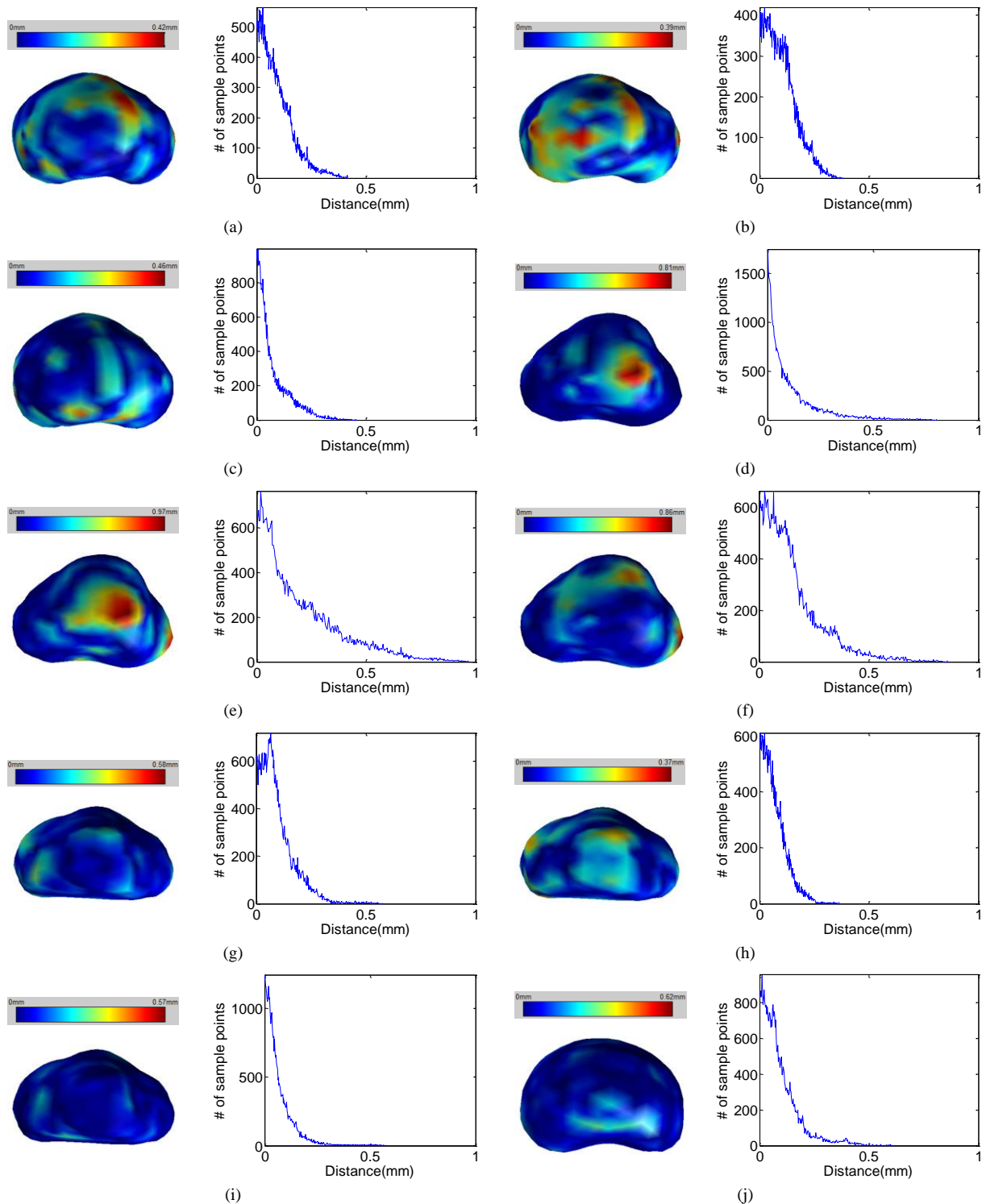


Figure 16. Continued.

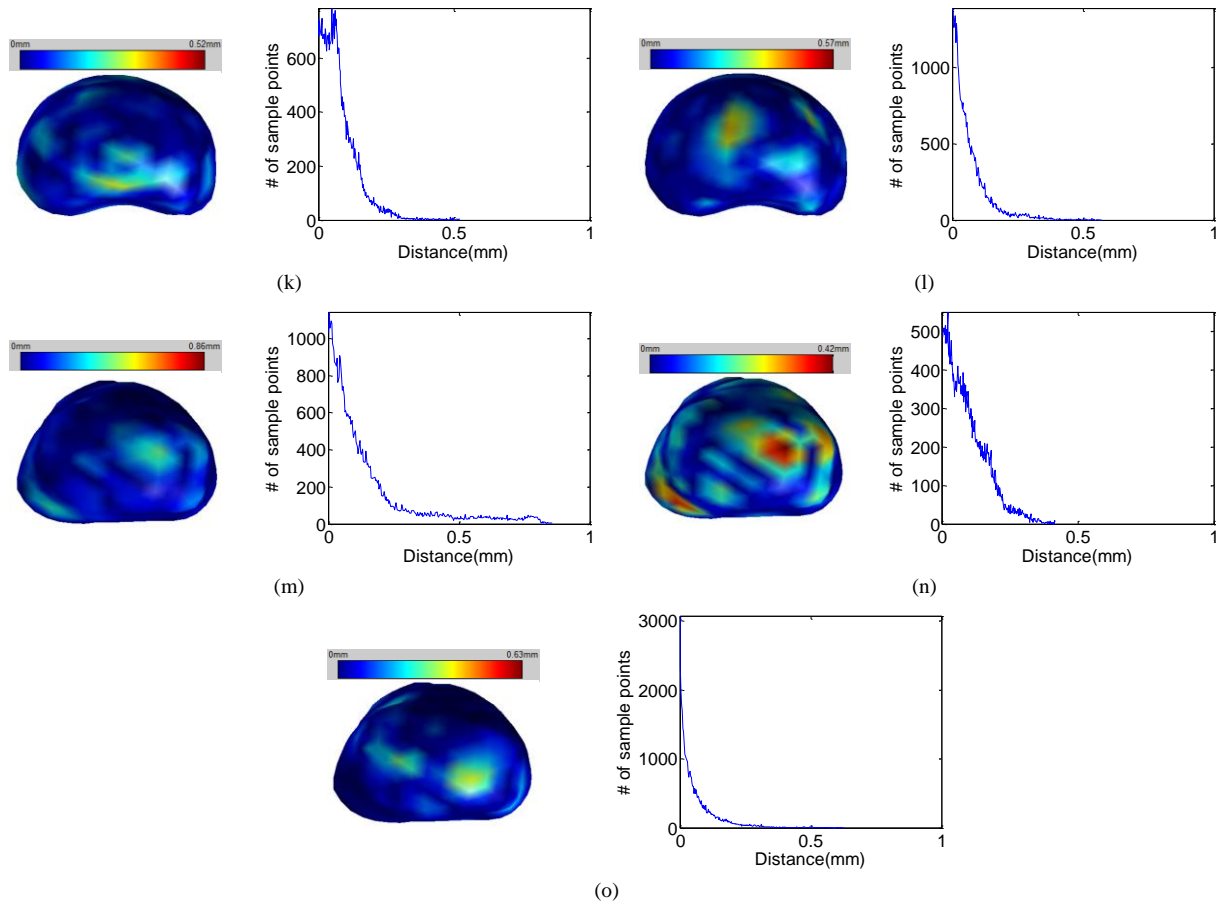


Figure 16. The error distances map and histogram of a process to the other process for each case: (a) Error distance of case 1-3 to case 1-1, (b) Error distance of case 1-3 to case 1-2, (c) Error distance of case 1-3 to case 1-4, (d) Error distance of case 2-2 to case 2-1, (e) Error distance of case 2-2 to case 2-3, (f) Error distance of case 2-2 to case 2-4, (g) Error distance of case 3-4 to case 3-1, (h) Error distance of case 3-4 to case 3-2, (i) Error distance of case 3-4 to case 3-3, (j) Error distance of case 4-4 to case 4-1, (k) Error distance of case 4-4 to case 4-2, (l) Error distance of case 4-4 to case 4-3, (m) Error distance of case 5-4 to case 5-1, (n) Error distance of case 5-4 to case 5-2, (o) Error distance of case 5-4 to case 5-3.

tion can be sufficiently accurate, (i.e., within 1.5 mm) in order to be used in a computer-assisted cryosurgery trainer, when the user specifies partial contours in mid- to high-quality image regions only. The developed GUI system allows the user to do so with greater ease than the conventional image segmentation system. Specifying partial contours on each cross section takes 5~10 seconds including the selection of the cross-sectional plane. Specifying contours on a sufficient number (i.e., 10~15) of cross-sectional planes would take several minutes, but this time can be further reduced as the user becomes more proficient in using the GUI system.

3.2 Case study 2: Validation of surface fitting in Step 2

The goal of Case Study 2 was to confirm that the accuracy, convergence, and computational cost of the proposed V-Spring-based deformable surface fitting meet the requirements.

In order to assess the accuracy of the shape reconstruction, a pre-defined polygonal model of the prostate was selected as

the “ground truth” model. A pre-defined 3D shape was used here in order to measure the accuracy of the Step 2 process, even though the true shape is unknown in a real application scenario. A set of contours was generated by three schemes:

- (1) Scheme 1: Full contours on multiple cross sections parallel to the yz -plane.
- (2) Scheme 2: Full contours on cross-sectional planes with randomly generated positions and orientations.
- (3) Scheme 3: Partial contours on cross-sectional planes with randomly generated positions and orientations.

The accuracy was measured by the error, or the average distance, between the ground-truth model and the deformed surface model. Figure 12 displays how the error diminishes as the number of contours increases for each of the three methods. For the input contours created by Schemes 1 and 2, the error converges after 15~25 contours, while the input contours created by Scheme 3 require 25~30 contours for a convergence. Note that the nature of user-specified contours in a real application is between that of Scheme 2 and Scheme

Table 1. Mean and standard deviation of the Hausdorff distance, Dice's coefficient, 25 percentile, 50 percentile, 75 percentile and 95 percentile of the error distance.

Statistics of error distance		Case 1	Case 2	Case 3	Case 4	Case 5
Hausdorff	Mean (mm)	0.4484	0.9058	0.4719	0.5283	0.7665
	Standard deviation (mm)	0.0486	0.1152	0.0850	0.0600	0.2085
Dice's coefficient	Mean (mm)	0.9544	0.9428	0.9493	0.9655	0.9441
	Standard deviation (mm)	0.0085	0.0114	0.0108	0.0066	0.0171
25 percentile	Mean (mm)	0.0325	0.0491	0.0308	0.0264	0.0292
	Standard deviation (mm)	0.0111	0.0145	0.0106	0.0092	0.0129
50 percentile	Mean (mm)	0.0720	0.1163	0.0638	0.0572	0.0693
	Standard deviation (mm)	0.0205	0.0288	0.0177	0.0155	0.0258
75 percentile	Mean (mm)	0.1325	0.2340	0.1109	0.1061	0.1422
	Standard deviation (mm)	0.0249	0.0514	0.0232	0.0219	0.0439
95 percentile	Mean (mm)	0.2425	0.4942	0.2142	0.2333	0.4001
	Standard deviation (mm)	0.0279	0.1028	0.0386	0.0334	0.1447

3. In Scheme 3, the error does not converge monotonically and exhibits a significant amount of fluctuation. This is because the random selection of partial contours on each cross sectional plane yields shapes with larger variations. It should also be noted that a set of partial contours generated by Scheme 3 yields less accurate results than a set of evenly distributed partial contours specified carefully by the user. Thus, the error resulting from Scheme 3 leads to the most conservative estimation of the accuracy of the proposed surface-fitting method. Figure 13 shows how the error is reduced as the number of partial contours created by Scheme 3 is increased. The result of this convergence study indicates that 20~30 partial contours are sufficient to bring the average error down to 1mm and the maximum error down to 2 mm. This corresponds to 10~15 cross-sectional planes assuming that each cross-sectional plane contains two partial contours on average.

The computation time for the V-Spring-based surface fitting for 30 partial contours was approximately 2 seconds for a template with 320 tri elements and 5 seconds for a template with 1,280 tri elements. The computational time was measured on a standard Windows workstation with an Intel Core i5 2.44 GHz CPU with 8 GB of memory.

The result of Case Study 2 suggests that the accuracy, convergence characteristics, and computational cost of the proposed V-Spring-based method are suitable for the purpose of the Step 2 surface fitting.

3.3 Case study 3: Validation of the complete process

The entire reconstruction process of Step 1 and Step 2 was performed to confirm that: (1) the surface of the generated 3D shape of a prostate was smooth and realistic; (2) the computational cost for converging to the final shape was low enough to finish within five minutes using a standard PC; and, (3) the method created a water-tight polygonal mesh with no self-intersections.

Figure 14 illustrates the entire Step 1 and Step 2 processes

of the proposed shape reconstruction method. Figure 14(a) shows the Step 1 process of manual contour specification, and Figure 14(b) shows a set of specified partial contours, which then become part of the input to the Step 2 deformable surface fitting. Figure 14(c) illustrates the convergence process of the V-Spring method. Figure 14(d) shows the final 3D shape of the prostate. Note that the user can choose the resolution of the final output model, and the system generates a finer mesh using a standard subdivision surface scheme. The final shape is a smooth, realistic, and watertight shape suitable for a computer-assisted cryosurgery training system. The same shape reconstruction process has been applied four times by different users to each of the five 3D TRUS images yielding the 20 3D prostate shapes shown in Figure 15. The figure shows for each trial: (1) specified 3D contours (left); (2) the generated 3D prostate shape superimposed on the specified 3D contours (middle); and (3) the generated 3D prostate shape (right).

To confirm the robustness and repeatability of the proposed 3D prostate shape reconstruction method, we measure seven error statistics including the Hausdorff distance, Dice's coefficient, 25/50/75/95-percentile of the distance for the four reconstructed D prostate shapes for each TRUS image. Since the distance is directional, for each group of four 3D shapes, 12 sets of distance statistics are generated. The mean and the standard deviation of each set are summarized in Table 1. To further investigate the deviation between two shapes created from the same TRUS image, we select randomly one of the four reconstructed 3D shapes and measure the distances from the selected shape to the other three shapes. Figure 16 illustrates the distance distribution color-mapped on the 3D shape along with the distance histogram. The error statistics confirms the high level of robustness and repeatability of the proposed 3D shape reconstruction method.

4. Conclusions

HA 3D TRUS image-based shape reconstruction method

for the prostate is proposed. Generated shapes are smooth, realistic, and water-tight, making the method suitable for a computer-assisted trainer for prostate cryosurgery. Because the system lets the user adjust, navigate through, and visualize the TRUS images freely, the user can choose a set of cross-sectional images and specify partial contours of the prostate with a high level of confidence. The system then creates a complete prostate shape using deformable mesh smoothing. The entire 3D shape reconstruction process takes only several minutes, and the resulting shape is more realistic and accurate than a 3D shape created by a conventional method that uses an automated image segmentation technique.

As future work, we plan to automate the specification of partial contours in Step 1. Although the complete contour identification is challenging for low- to medium-quality TRUS images, we do not need to generate complete contours if we use the fully automated deformable mesh fitting scheme proposed in this paper – this will alleviate the difficulty of the automated contour specification. After adjusting the intensity and contrast and applying various noise-reduction filters automatically, the image processing method should be able to specify automatically the partial contours that can be identified with a high level of confidence, mimicking the Step 1 process performed by the user.

Acknowledgements

The authors would like to thank Dr. Aaron Fenster and his colleagues at the Robarts Research Institute of The University of Western Ontario, London, Ontario, Canada, for providing all the 3D TRUS images used in our research project.

This study was supported by Award Number R01CA134261 from the National Cancer Institute. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Cancer Institute or the National Institutes of Health. The development of the V-Spring method was supported by a CAREER Award from the National Science Foundation, a research grant from IBM, and a research grant from Honda R&D.

References

- [1] Rossi M, Tanaka D, Shimada K, Rabin Y. Computerized planning of prostate cryosurgery using variable cryoprobe insertion depth. *Cryobiology*. 2010; 60(1): 71-79.
- [2] Tanaka D, Shimada J, Rossi M, Rabin Y. Cryosurgery planning using bubble packing in 3D. *Computer Methods in Biomechanical and Biomedical Engineering*. 2008; 11(2): 113-121.
- [3] Developing a computerized training tool for cryosurgery [Internet]. c2014 [cited 2014 Jul 1]. Available from: http://projectreporter.nih.gov/project_info_description.cfm?projectnumber=1R01CA134261-01A2
- [4] Fenster A, Downey D, Cardinal N. Three dimensional ultrasound imaging. *Physics in Medicine and Biology*. 2001; 46(5): 67-99.
- [5] Tong S, Downey D, Cardinal N, Fenster A. A three-dimensional ultrasound prostate imaging system. *Ultrasound in Medicine and Biology*. 1996; 22(6): 735-746.
- [6] 3D Doctor [Internet]. c2014 [cited 2014 Jul 1]. Available from: <http://www.ablesw.com/3d-doctor/>
- [7] The Visualization Toolkit [Internet]. c2014 [cited 2014 Jul 1]. Available from: <http://www.vtk.org>
- [8] ITK– Segmentation & Registration Toolkit [Internet]. c2014 [cited 2014 Jul 1]. Available from: <http://www.itk.org>
- [9] Adobe Photoshop [Internet]. c2014 [cited 2014 Jul 1]. Available from: <http://www.adobe.com/PhotoshopCS5>
- [10] IrfanView [Internet]. c2014 [cited 2014 Jul 1]. Available from: <http://www.irfanview.com>
- [11] Image Processing Toolbox for MATLAB & Simulink [Internet]. c2014 [cited 2014 Jul 1]. Available from: <http://www.mathworks.com/products/image>
- [12] Desbrun M, Meyer M, Schröder P, Barr AH. Implicit fairing of irregular meshes using diffusion and curvature flow. In: *SIGGRAPH 99; Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*; 1999 Aug 8-13; Los Angeles, CA; p. 317-324.
- [13] Helenbrook BT. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*. 2003; 56: 1007-1021.
- [14] Guskov I, Sweldens W, Schröder P. Multi-resolution signal processing for meshes. In: *SIGGRAPH 99; Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*; 1999 Aug 8-13; Los Angeles, CA; p. 325-334.
- [15] Vieira M, Shimada K, Furuhashi T. Smoothing of noisy laser scanner generated meshes using polynomial fitting and neighborhood erosion. *Journal of Mechanical Design*. 2004; 126(3): 495-503.
- [16] Li Z, Ma L, Jin X, Zheng Z. A new feature-preserving mesh-smoothing algorithm. *The Visual Computer*. 2009; 25(2): 139-148.
- [17] Vieira M, Shimada K. Surface mesh segmentation and smooth surface extraction through region growing. *Computer-Aided Geometric Design*. 2005; 22(8): 771-792.
- [18] Yamada A, Furuhashi T, Shimada K, Hou K. A discrete spring model for generating fair curves and surfaces. In: *Proceedings of the 7th Pacific Conference on Computer Graphics and Applications*; 1999 Oct 5-7; Seoul, Korea, p. 270-279.
- [19] Shimada K, Yamada A, Tomotake T, Hou KH. US Patent US6, 226,405: Method and apparatus for updating node position. Issued 2001.
- [20] Ghanei A, Soltanian-Zadeh H, Ratkiewicz A, Yin FF. A three-dimensional deformable model for segmentation of human prostate from ultrasound images. *Medical Physics*. 2001; 28(10): 2147-2153.
- [21] Hu N, Downey DB, Fenster A, Ladak HM. Prostate surface segmentation from 3D ultrasound images. In: *Proceedings of the IEEE International Symposium on Biomedical Imaging*; IEEE Computer Society Press; 2002 Jul 7-10; Washington, DC, USA, p. 613-616.
- [22] Mahdavi S, Chng N, Spadinger I, Morris WJ, Salcudeana SE.

Semi-automatic segmentation for prostate interventions. *Medical Image Analysis*. 2011; 15(2): 226-237.

- [23] Garnier C, Bellanger JJ, Wu K, Shu H, Costet N, Mathieu R, et al. Prostate segmentation in HIFU therapy. *IEEE Trans Medical Imaging* 2011; 30(3): 792-803.