Mitral Annulus Segmentation From 3D Ultrasound Using Graph Cuts

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1109/tmi.2010.2050595">http://dx.doi.org/10.1109/tmi.2010.2050595</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers (IEEE)</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Thu Dec 06 11:55:36 EST 2018</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/69906">http://hdl.handle.net/1721.1/69906</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td></td>
</tr>
</tbody>
</table>
Mitral Annulus Segmentation From 3D Ultrasound Using Graph Cuts

Robert J. Schneider*, Douglas P. Perrin, Nikolay V. Vasilyev, Gerald R. Marx, Pedro J. del Nido, and Robert D. Howe, Senior Member, IEEE

Abstract—The shape of the mitral valve annulus is used in diagnostic and modeling applications, yet methods to accurately and reproducibly delineate the annulus are limited. This paper presents a mitral annulus segmentation algorithm designed for closed mitral valves which locates the annulus in three-dimensional ultrasound using only a single user-specified point near the center of the valve. The algorithm first constructs a surface at the location of the thin leaflets, and then locates the annulus by finding where the thin leaflet tissue meets the thicker heart wall. The algorithm iterates until convergence metrics are satisfied, resulting in an operator-independent mitral annulus segmentation. The accuracy of the algorithm was assessed from both a diagnostic and surgical standpoint by comparing the algorithm’s results to delineations made by a group of experts on clinical ultrasound images of the mitral valve, and to delineations made by an expert with a surgical view of the mitral annulus on excised porcine hearts using an electromagnetically tracked pointer. In the former study, the algorithm was statistically indistinguishable from the best performing expert ($p = 0.85$) and had an average RMS difference of $1.81 \pm 0.78$ mm to the expert average. In the latter, the average RMS difference between the algorithm’s annulus and the electromagnetically tracked points across six hearts was $1.19 \pm 0.17$ mm.

Index Terms—Annulus, graph cuts, mitral valve, segmentation, ultrasound.

I. INTRODUCTION

The mitral valve annulus is an important cardiac structure that is defined as the fibrous saddle-shaped structure that anchors the mitral leaflets. Characterization of the geometry of the mitral annulus has proven valuable in an array of applications. Clinically, studies on annulus shape have shown that there is a correlation between annular geometry and pathology [1]–[5], [20]–[22], which in addition to being tedious and time consuming, is also prone to inaccuracies. One reason is that the usual practice is to pick points in 2D slices taken from a 3D volume, so the user only has access to a portion of the available information at any given step, forcing the user to mentally interpolate 3D information. The methods presented in [23] and [24] attempt to correct for the deficiencies in these manual slice-based segmentations, but do so by smoothing the segmentations without referring back to the original 3D data from which they were made.

Semi-automatic methods to delineate the 3D annulus from 3DUS are presented in [25] and [26]. The former method segments the annulus by compiling semi-automated annulus point delineations in 2D slices into a 3D annulus structure. As in the case of manual segmentation, this method does not take into account information in neighboring slices, so it suffers from inaccuracies and spatial inconsistencies. The latter method segments the annulus as a consequence of fitting an entire mitral valve model to 3DUS data using learning techniques, which consequently requires a training database of manually delineated points. The resulting accuracy of the method for the purpose of patient-specific annulus segmentation is unclear, as comparisons are only made to published population valve dimensions.

We present in this manuscript an accurate and robust segmentation algorithm (Fig. 1) designed to segment the mitral annulus in an ultrasound volume containing a closed valve. The algorithm begins by first finding a surface at the location of the thin valve closure [10]–[14].

The mitral annulus is most commonly visualized using ultrasound, given that ultrasound is a cheap, portable, and non-ionizing imaging modality capable of capturing the fast moving valve structures. Prior to three-dimensional ultrasound (3DUS), delineating the shape of the annulus was difficult, leading to the assumption that the annulus was planar, instead of saddle-shaped as is now apparent [6]. Even after the advancement from 2D to 3D ultrasound, visualizing the annulus remains a problem due to imaging noise and volume visualization limitations.

Despite widespread use of the annulus in both clinical and research applications, available methods to extract an accurate and reproducible geometry remain limited. One of the most common methods used in research applications is to segment and track implanted fiducials, such as tracking radiopaque markers with fluoroscopy [15]–[18] or tracking sonomicrometry transducers [19]. This is an invasive approach not feasible in a clinical setting. Another common method is manual segmentation of images [2]–[5], [20]–[22], which in addition to being tedious and time consuming, is also prone to inaccuracies. One reason is that the usual practice is to pick points in 2D slices taken from a 3D volume, so the user only has access to a portion of the available information at any given step, forcing the user to mentally interpolate 3D information. The methods presented in [23] and [24] attempt to correct for the deficiencies in these manual slice-based segmentations, but do so by smoothing the segmentations without referring back to the original 3D data from which they were made.

Semi-automatic methods to delineate the 3D annulus from 3DUS are presented in [25] and [26]. The former method segments the annulus by compiling semi-automated annulus point delineations in 2D slices into a 3D annulus structure. As in the case of manual segmentation, this method does not take into account information in neighboring slices, so it suffers from inaccuracies and spatial inconsistencies. The latter method segments the annulus as a consequence of fitting an entire mitral valve model to 3DUS data using learning techniques, which consequently requires a training database of manually delineated points. The resulting accuracy of the method for the purpose of patient-specific annulus segmentation is unclear, as comparisons are only made to published population valve dimensions.

We present in this manuscript an accurate and robust segmentation algorithm (Fig. 1) designed to segment the mitral annulus in an ultrasound volume containing a closed valve. The algorithm begins by first finding a surface at the location of the thin
leaflet tissue. The surface is constructed near a user-provided point using a thin tissue detector and the max-flow algorithm. The algorithm then locates the annulus by finding where at the surface the thin leaflet tissue meets the thicker tissue of the surrounding heart wall. This is done using the max-flow algorithm and active contour methods in projection images created from the surface and surroundings. A 3D mitral annulus contour is computed by projecting the annulus contour in the images back onto the mitral leaflet surface.

The details of the algorithm are described in Section II. An analysis of the algorithm is then presented in Section III, which includes studies which validate the algorithm's accuracy (Sections III-A and III-B), and an analysis of the sensitivity of the algorithm to the user-provided point (Section III-C).

**Algorithm Design**

**A. Ultrasound Data**

The acquired data for this study was in the form of full volume reconstructions, allowing for the visualization of the entire mitral valve in a single 3DUS volume (iE33 Echocardiography System with transesophageal, X7–2t, and transthoracic, X7–2, probes, Philips Healthcare, Andover, MA). The condition of the valves ranged from normal to varying types and degrees of pathology. While the position and orientation of a valve within a volume varied, the segmentation algorithm requires that the thin tissue of the leaflets can be seen across the entire valve.

From each 4D data set (3D + time), a single ultrasound frame containing a closed valve was selected. The frame was chosen at or near peak systole, as this is a time in the cardiac cycle when the valve is closed and moving slowly. Restricting the selection to closed valves enabled accurate computation of a surface at the location of the leaflets. The dimensions of the volume were roughly 200 × 200 × 200 voxels, with a resolution on the order of 0.5–0.75 mm per voxel. In this manuscript, the selected 3DUS volume is referred to as Ω, and individual voxel locations as \( \mathbf{v}_i \), where each \( \mathbf{v}_i \) has corresponding \((x_i, y_i, z_i)\) coordinates. For \( \Omega \) with \( N \) voxels, \( i \in \{1, \ldots, N\} \).

**B. Thin Tissue Detector**

Delineating the location of the annulus directly from 3DUS is difficult, so the algorithm first segments the location of the leaflets using a thin tissue detector. The thin tissue detector (TTD) highlights thin structures at a particular scale in the 3DUS volume. The TTD is computed using characteristics of the gradient field of both \( \Omega \) and \( \Omega_F \), where \( \Omega_F \) is a filtered version of \( \Omega \), reducing the effects of speckle in the volume. \( \Omega_F \) is computed by convolving \( \Omega \) with a Gaussian kernel, \( G(\sigma) \), with standard deviation, \( \sigma \), approximately equal to half the maximum expected thickness of the mitral leaflets, taking into account any abnormal thickness due to disease state. The kernel dimension is \( 4\sigma + 1 \) to include two standard deviations.

The TTD is computed using the results of three computations. The first of these

\[
\Theta(\mathbf{v}_i) = \frac{\sum_{a=1}^{J-1} \sum_{b=a+1}^{J} \arccos \left( \frac{\nabla \Omega_F(\mathbf{v}_a) \cdot \nabla \Omega_F(\mathbf{v}_b)}{\|\nabla \Omega_F(\mathbf{v}_a)\| \|\nabla \Omega_F(\mathbf{v}_b)\|} \right)}{\sum_{j=1}^{J-1} \sum_{j=1}^{J} \arccos \left( \frac{\nabla \Omega_F(\mathbf{v}_j) \cdot \nabla \Omega_F(\mathbf{v}_j)}{\|\nabla \Omega_F(\mathbf{v}_j)\| \|\nabla \Omega_F(\mathbf{v}_j)\|} \right)}
\]

is a measure of the average angle between gradient vectors of \( \Omega_F \) within a neighborhood, where \( \nabla \) denotes the gradient operator. A neighborhood at \( \mathbf{v}_j \) is defined as the set of all neighboring voxels, \( \mathbf{v}_p \), which are contained within the cube of side length \( p \) centered at voxel \( \mathbf{v}_j \), making \( j \in \{1, \ldots, J\} \), with \( J = p^3 \) for a cubical neighborhood. At the location of the leaflets, there will be opposing gradient fields as a result of the dark blood pools above and below the thin, bright leaflets. Therefore, \( \Theta \) will have a higher value at the leaflets in comparison to more homogeneous regions, such as in the middle of the blood pool or thick tissue regions. The gradient vectors at the leaflets should be pointing inward, so

\[
\Phi(\mathbf{v}_i) = \sum_{q=1}^{6} \sum_{k=1}^{K} \left( \frac{\nabla \Omega_F(\mathbf{f}_k)}{\|\nabla \Omega_F(\mathbf{f}_k)\|} \cdot \mathbf{f}_q \right)_{\text{norm}}
\]

measures the flux of the vectors across the neighborhood boundary faces. In computing \( \Phi \), the inward directed unit normal of each boundary face is denoted \( \mathbf{f}_{q,\text{norm}} \), where \( q \in \{1, \ldots, 6\} \) for a cubical neighborhood. Neighbors with a face adjacent to face \( q \) are denoted \( \mathbf{f}_k \), where \( k \in \{1, \ldots, K\} \) and \( K = p^2 \) for a cubical neighborhood.

\( \Theta \) and \( \Phi \) are computed using directions of gradient vectors but not their magnitude, meaning they contain no information
about edges. However, the mitral leaflets are near two strong edges as a consequence of the thin tissue residing between two blood pools. Therefore, to quantify the number and proximity of strong edges to a voxel

\[ \Pi = G(\sigma) \ast \nabla \Omega. \]  

(3)

The complete TTD is then \( (\Theta \Phi[0\rightarrow1] \Pi)[0\rightarrow1] \), where \( (\cdot)[0\rightarrow1] \) indicates that the values for that term are normalized to the range of 0 to 1. High TTD values indicate that the voxel is likely part of a thin structure and potentially at the location of the mitral leaflets. The TTD computed for a 3DUS image of a prolapsed mitral valve is shown in Fig. 2, where the TTD is inverted to better show the highlighted thin tissue regions.

C. Valve Position and Orientation

To accurately construct a surface at the location of the mitral leaflets, an initial estimation of the position and orientation of the mitral valve relative to the volume is needed. In the first iteration of the algorithm, the position is determined by the user, who is asked to provide a point, \( \mathbf{x}_c \), somewhere near the center of the valve but not necessarily on the valve. In subsequent iterations, assuming they are needed, \( \mathbf{x}_c \) is defined as the center of the computed annulus from the previous iteration.

Given \( \mathbf{x}_c \) near the center of the valve, the orientation of the mitral valve is then estimated by computing a corresponding best-fit mitral valve plane. The idea is to threshold to separate leaflet tissue and blood, then fit a plane to the tissue. This is done by first clustering the values of the TTD from voxels residing in a spherical region of interest of radius \( r_{\text{pca}} \) centered at \( \mathbf{x}_c \) into two clusters using a \( k \)-means algorithm. The sample mean and standard deviation of the TTD at voxels for the cluster likely containing the leaflets, which is the cluster with the higher TTD average, are denoted \( \mu_{\text{high}} \) and \( \sigma_{\text{high}} \), respectively. A principle-component analysis (PCA) on the set of voxels \( \mathbf{v}_{\text{pca}} : \{ \mathbf{v} | \text{TTD}(\mathbf{v}) > (\mu_{\text{high}} - \sigma_{\text{high}}) \cap |\mathbf{v} - \mathbf{x}_c| \leq r_{\text{pca}} \} \) then defines the orientation of the cluster. The mitral valve plane, \( M\mathbf{V}_{\text{plane}} \), is then the plane passing through \( \mathbf{x}_c \) with normal direction \( \mathbf{n}_c \) equal to the direction of least variance as determined in the PCA.

D. Graph Construction and the Max-Flow Algorithm

The annulus segmentation algorithm makes extensive use of the max-flow algorithm [27]. Generically, a graph, \( G = (V, E) \), is a set of nodes, \( V \), connected by edges, \( E \), which have either a directed or undirected capacity. In the case of the max-flow algorithm, there are also two special nodes called the source and the sink. The max-flow algorithm finds the maximum flow which can originate from the source, flow through the edges of predefined capacity, and enter the sink. In doing so, according to the min-cut/max-flow theorem [28], the max-flow algorithm generates a set of saturated edges called the min-cut which separates the graph into two regions—one containing the source and another containing the sink.

A generalization of the graph construction used in the algorithm is shown in Fig. 3, where the 2D and 3D graphs are used to find contours and surfaces, respectively. The graph construction, in particular attaching the source and sink to opposite ends of the graph, allows us to enforce prior knowledge that the min-cut should reside between the opposite ends.

E. Mitral Leaflet Surface via Max-Flow

With an estimation of the valve position and orientation, a graph can be constructed on which we implement the max-flow algorithm to find a surface at the location of the mitral leaflets. The graph, \( G_{\text{surf}} = (V_s, E_{\text{surf}}) \), resides within a cylinder of radius \( r_{\text{grani}} \) and height \( 2r_{\text{grani}} \) which is centered at \( \mathbf{x}_c \) with an axis directed along \( \mathbf{n}_c \). The graph consists of nodes \( V_s \) located on a rectilinear grid directed along \( \mathbf{n}_c \) with a voxel spacing, and undirected edges \( E_{\text{surf}} \) which connect the nodes, making \( G_{\text{surf}} \) 6-connected except on the edges of the graph. The source connects to all nodes on one face of the cylinder, while the sink connects to all nodes on the opposite face (Fig. 4).

To find a surface at the location of the leaflets, we define the edge capacities, \( E_{s,p} \) and \( E_{s,o} \), in (4) and (5), respectively, such that between connected nodes \( V_{s,i} \) and \( V_{s,j} \)

\[
E_{s,p} = \frac{\omega_p}{1 + \alpha_s (\text{TTD}(V_{s,i}) + \text{TTD}(V_{s,j}))^2}
\]

\[
E_{s,o} = \frac{\omega_o}{1 + \alpha_s (\text{TTD}(V_{s,i}) + \text{TTD}(V_{s,j}))^2}
\]

where \( \omega_p, \omega_o \), and \( \alpha_s \) are scalar weights. This form lowers the edge capacity around the leaflets, encouraging the min-cut to
be located at the leaflets. $E_{sp}$ defines the capacity for edges parallel to $n_c$, and $E_{sd}$ the capacity for all other edges. This is an anisotropic edge capacity assignment [29] used to control the curvature of the surface: $\omega_p < \omega_0$ encourages a flat surface, while $\omega_p > \omega_0$ allows for a high curvature surface. For those edges where one of the nodes is either the source or the sink, the edge capacity is set to infinity.

The min-cut is found using the max-flow implementation by Kolmogorov [30]. The resulting min-cut defines the mitral valve leaflet surface, $MV_{surf}$. Assuming $x_c$ is located near the center of the valve and $r_{graph}$ is sufficiently large, $MV_{surf}$ extends beyond the leaflets and contains the mitral annulus.

F. Projection Image Formation

The annulus is where the thin leaflets connect to the thicker cardiac walls. We therefore search at and around $MV_{surf}$, which contains the annulus, for this location. Because the curvature of $MV_{surf}$ was controlled using $\omega_p$ and $\omega_0$ in the max-flow implementation, $MV_{surf}$ is a regular surface in $\mathbb{R}^3$ that can be described as the graph of a function, $Z_{surf}$, described in the $x'y'z'$-plane

$$MV_{surf} (x'; y') = (x', y', Z_{surf} (x', y')).$$

The $x'y'z'$-coordinate system is centered at $x_c$ with the $z'$ axis parallel to $n_c$ (Fig. 5).

The annulus, a closed curve on $MV_{surf}$, therefore projects to a closed planar curve in the $x'y'$-plane. To locate this planar curve, two images, $P_{int}$ and $P_{ttl}$, are used, where

$$P_{int} (x', y') = \sum_{\zeta=\sigma}^{2\pi} \Omega (x', y', Z_{surf} (x', y') \pm \zeta)$$

$$P_{ttl} (x', y') = TTD (x', y', Z_{surf} (x', y')).$$

Examples of the images are shown in Fig. 5. $P_{int}$ is the value of $\Omega$ in regions above and below $MV_{surf}$, excluding a band of thickness $2r$ where the leaflets are located, and provides information about where the tissue surrounding the leaflets resides. $P_{ttl}$ is the value of the $TTD$ at $MV_{surf}$, and provides information about where the thin tissue resides at the surface. It is clear from the example images of $P_{int}$ and $P_{ttl}$ in Fig. 5 that there exists a centrally located dark region in $P_{int}$ and a bright region in $P_{ttl}$. This region corresponds to the location of the mitral valve in the projection space, making the annulus the border of this region.

G. Projected Mitral Annulus Contour

The border that we wish to delineate in the projection space, which corresponds to the projected location of the mitral annulus, is a single closed contour which can be delineated with such methods as snakes [31] or level sets [32]. These methods, however, are sensitive to initial contour position and generally require a manually initialized contour. To avoid the variability inherent in user input, we developed an automated method which initializes contours by computing min-cuts on 2D graphs using the max-flow algorithm, with edge capacities derived from both $P_{int}$ and $P_{ttl}$ for added robustness. This method is preferred over our preliminary method which initialized generic contours in the projection images [33]. To overcome noise and anatomic variability in $P_{int}$ and $P_{ttl}$, multiple spatial scales of the projection images are used to construct multiple resolution-specific contours. Treating these contours as snakes and forcing them to a common location in high resolution versions of $P_{int}$ and $P_{ttl}$ produces a single contour at the desired location. A summary of the approach is shown in Fig. 6. The advantage of using the method of snakes versus level sets is it allows for a simple integration of an attractive energy between contours and does not allow a contour to split during its evolution.
1) Contour Initialization via Max-Flow: The graph, $\Gamma_{\text{cont}} = \langle V_c, E_c \rangle$, used in the max-flow implementation to initialize a contour in the projection space is shown in Fig. 7. The nodes, $V_c$, are positioned along rays emanating from a center point, $x_c$. The rays span the defined projection space and are evenly spaced along an angular offset of $\Delta \gamma$, while nodes are evenly spaced along the rays at an offset of $\Delta r$. The nodes nearest to the center connect to the source, while nodes farthest from the center connect to the sink.

To initialize a contour at a specific resolution of the projection images, $P_{\text{int}}$ and $P_{\text{tdl}}$ are first filtered using a Gaussian, $G(\sigma_n)$, to generate $F_{\text{int},n}$ and $F_{\text{tdl},n}$, respectively. For the case where $m$ different contours will be initialized at $n$ different resolutions, $n \in \{1, \ldots, m\}$. The values of $F_{\text{int},n}$ and $F_{\text{tdl},n}$ at the node locations are found using bilinear interpolation and are normalized across all nodes to the range $[0 \ldots 1]$.

It is assumed that the border of the centrally located regions surrounds the center of the rays, $x_c$. To find this border, we first compute at the node locations

$$F_{\text{drv},n} = \left( \lambda_{\text{int}} \frac{\partial F_{\text{int},n}}{\partial R} - \lambda_{\text{tdl}} \frac{\partial F_{\text{tdl},n}}{\partial R} \right) (1 - F_{\text{int},n})^2$$

where $\lambda_{\text{int}}$ and $\lambda_{\text{tdl}}$ are scalar weights and $R$ is the distance along a ray. This is a drive image that will help to define the graph edge capacities, with high values at locations that are most likely the desired border. Partial derivatives of $F_{\text{int},n}$ and $F_{\text{tdl},n}$ are computed along the rays because we would like to find the location where $F_{\text{int},n}$ changes from dark to light as we travel out from $x_c$ along a ray, rather than light to dark. Similarly, we want to find the location where $F_{\text{tdl},n}$ changes from light to dark. $F_{\text{drv},n}$ is scaled by the inverse of $F_{\text{int},n}$ to encourage the contour to be found near the darker regions of $F_{\text{int},n}$.

The edge capacities, $E_{c,s,n}$ and $E_{c,d,n}$, of the graph are undirected and defined using $F_{\text{drv},n}$ such that between connected nodes $V_{ci}$ and $V_{cj}$

$$E_{c,s,n} = \frac{\omega_s}{1 + \alpha_c (F_{\text{drv},n}(V_{ci}) + F_{\text{drv},n}(V_{cj}))^2}$$

$$E_{c,d,n} = \frac{\omega_d}{1 + \alpha_c (F_{\text{drv},n}(V_{ci}) + F_{\text{drv},n}(V_{cj}))^2}$$

where $\omega_s$, $\omega_d$, and $\alpha_c$ are scalar weights. $E_{c,s,n}$ defines the capacity for edges between nodes on the same ray, and $E_{c,d,n}$ defines the capacity for edges between nodes on different rays. This is done to control the deviation of a contour from a circle: $\omega_s < \omega_d$ encourages a more circular min-cut, while $\omega_s > \omega_d$ allows for a less circular min-cut.

Due to the construction of the graph, nodal resolution varies with distance from the center. Minimizing these variations along the min-cut requires iterating the contour initialization to force the ray center to the centroid of the area contained within the min-cut. For the first iteration, $x_c$ is the projection of the point $x_c$ onto the projection space. The final location of the ray center is $x_{f,n}$, and the final min-cut is contour $C_n$, where $n$ is specific to the resolution of the projection images.

2) Contour Convergence via Active Contours: The multiple contours initialized using $m$ different resolutions of $P_{\text{int}}$ and $P_{\text{tdl}}$ are forced to converge as snakes. This allows us to find a balance between the expected shape of the contour and the expected location of the contour determined from the images.

We restrict the nodes of the snake to reside on rays emanating from a center point, $x_s$, which is computed as the centroid of the ray centers, $x_{f,n}$, where $n \in \{1, \ldots, m\}$. The rays are equally spaced at an angular offset of $\Delta \theta$ (Fig. 8). For snake $n$, the radial location of the snake node on ray $\phi_{\tau}$ is referred to as $R_n(\phi_{\tau})$, where $\tau \in \{1, \ldots, T\}$ and $T$ is the number of rays, and is initially determined from the location of $C_n$.

Snakes evolve simultaneously such that they are forced to converge to a single snake. The energy used in the snake evolution is

$$E_{\text{snake},n} = \omega_{\text{drv}} F_{\text{drv},n} + \omega_{\text{xy}} F_{\text{xy},n} + \omega_{z} E_{z,n} + \omega_{\text{att}} E_{\text{att}}$$

where $\omega_{\text{drv}}$, $\omega_{\text{xy}}$, $\omega_{z}$, and $\omega_{\text{att}}$ are scalar weights. $E_{\text{drv},n}$ is an image energy derived from the projection images, $E_{\text{xy},n}$ is the curvature energy of the snake in the projection plane, $E_{z,n}$ is the curvature energy of $M/V_{\text{surf}}$ at the snake location, and $E_{\text{att}}$ is an attractive energy which forces the multiple snakes to converge to a single snake.

H. Image Energy

The image energy, $E_{\text{drv},n}$, is derived in much the same way that $F_{\text{drv},n}$ is derived for the graph. $P_{\text{int}}$ and $P_{\text{tdl}}$ are first fil-
tered using a Gaussian, \( G(\sigma_s) \), and normalized to the range of \([0-1]\) to generate \( F_{\text{int},s} \) and \( F_{\text{ttd},s} \), respectively. The image energy is then

\[
E_{\text{drv},n} = (1 - F_{\text{int},s})^2 \left( \lambda_{\text{ttd}} \frac{\partial F_{\text{ttd},s}}{\partial R} - \lambda_{\text{int}} \frac{\partial F_{\text{int},s}}{\partial R} \right) R_n
\]

(13)

where \( \lambda_{\text{int}} \) and \( \lambda_{\text{ttd}} \) are the same scalar weights used in (9), and \( E_{\text{drv},n} \) is normalized so that its magnitude is less than one. Imparting force \( U_{\text{drv},n} \) defined as

\[
U_{\text{drv},n} = -\frac{\partial E_{\text{drv},n}}{\partial R}
\]

(14)
on the snake nodes subsequently minimizes the image energy at the location of the snake.

I. Contour Curvature Energy

To account for contour curvature but avoid undesirable evolution of the snake in the absence of strong image forces, as is common with typical methods [34], we use the method presented in [35]. This method, instead of minimizing the curvature of the snake, minimizes the change in curvature. The energy \( E_{\text{xy},n} \) is then

\[
E_{\text{xy},n} = \left| \frac{\partial \kappa_n}{\partial s} \right|
\]

(15)

where \( \kappa \) is the curvature of the contour and \( s \) is arc length.

Whereas the method in [35] minimizes (15) by moving snake nodes to the perpendicular bisector of neighboring nodes, we restrict the nodes to move along the rays. In doing so, we find \( R_{\text{nc},\text{des}}(\phi_r) \), which is the desired location of the node from snake \( n \) along ray \( \phi_r \) which will minimize (15). Snake nodes are forced toward the location of \( R_{\text{nc},\text{des}}(\phi_r) \) using \( U_{\text{xy},n}(\phi_r) = R_{\text{nc},\text{des}}(\phi_r) - R_n(\phi_r) \).

J. Surface Height Curvature Energy

The shape of the surface at a snake’s location is easily obtained, as it is known that \( MV_{\text{surf}} \) can be described as the graph of a function, \( Z_{\text{surf}} \), which is defined in the projection space. Therefore, we can define \( Z_n(\phi_r) \) to be the height of \( MV_{\text{surf}} \) above the node on snake \( n \) residing on ray \( \phi_r \). Since the annulus is not flat, but rather saddle-shaped, \( E_{z,n} \) is defined as

\[
E_{z,n} = \left| \frac{\partial^2 Z_n}{\partial \phi_r^2} \right|
\]

(16)

This energy is minimized by imparting a force

\[
U_{z,n} = -\frac{\partial E_{z,n}}{\partial R}
\]

(17)
on the nodes, which encourages a constant local change in elevation of the annulus contour with respect to changes in \( \phi \).

K. Attractive Energy

As a means to allow the snakes to interact—so that a node of one snake which is at the global minimum can draw nodes of other snakes away from local minimums—and consequently

force the snakes to converge to a common location, we designed the attractive energy \( E_{\text{att}} \)

\[
E_{\text{att}}(\phi_r) = R_{\text{att}}(\phi_r) - R_{\text{att}}(\phi_r)
\]

\[n_{\text{min}}(\phi_r) = \arg\min_n (R_n(\phi_r))\]

\[n_{\text{max}}(\phi_r) = \arg\max_n (R_n(\phi_r))\]

(18)

\( E_{\text{att}} \) is minimized in each ray by driving the nodes from the different snakes on that ray towards the center of the node range, \( R_{\text{att}}(\phi_r) = (1/2)(R_{n_{\text{min}}}(\phi_r) + R_{n_{\text{max}}}(\phi_r)) \), using \( U_{\text{att},n} = R_{\text{att}} - R_n \).

L. Snake Update

Given the described energies which evolve the individual snakes, and the methods designed to minimize the energies, the snakes update according to

\[
R_{n,t+1} = R_{n,t} + \Delta t \omega_{\text{drv}} U_{\text{drv},n} + \omega_{\text{xy}} U_{\text{xy},n} + \omega_z U_{z,n} + \omega_{\text{att}} U_{\text{att},n}
\]

(19)

where \( R_{n,t+1} \) is the updated snake, \( R_{n,t} \) is the current snake, and \( \Delta t \) is a time step. The snakes are updated until they stabilize to within 0.1 pixels, which occurs in roughly 100 iterations using the parameter values shown in Table I.

M. 3D Mitral Annulus Contour

The 3D annulus is constructed from the final snake by computing the \( x' \) and \( y' \) coordinates using the angle of the rays and radial location of the nodes, and the \( z' \) coordinates using \( Z_n \). We compute a continuous annulus contour in \( \mathbb{R}^3 \) using a cubic interpolation of the defined points.

Algorithm Convergence

The point \( \mathbf{x}_c \) which initially positions and orients the graph used to find \( MV_{\text{surf}} \) is provided initially by a user who is asked to provide a point near the center of the valve. Therefore, the location of the user-specified point could have an effect on the shape of the 3D annulus contour. However, this dependency is resolved by iterating the algorithm until convergence metrics are satisfied (Fig. 9).

After the first execution of the algorithm, and in subsequent iterations, the center point of the resulting 3D annulus contour, \( \mathbf{x}_{c,\text{raw}} \), is compared to the location of \( \mathbf{x}_c \). If \( |\mathbf{x}_{c,\text{raw}} - \mathbf{x}_c| > \epsilon \) where \( \epsilon \) is a predefined scalar distance, \( \mathbf{x}_c \) is set equal to \( \mathbf{x}_{c,\text{raw}} \) and the algorithm iterates. Otherwise, it is assumed the annulus
TABLE I
MITRAL ANNULUS SEGMENTATION ALGORITHM PARAMETERS
USED FOR ALL STUDIED VALVES

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>σ</td>
<td>1</td>
<td>mm</td>
<td>αc</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>3</td>
<td>voxels</td>
<td>m</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>r_{pca}</td>
<td>12.5</td>
<td>mm</td>
<td>σ_{n=1}</td>
<td>2.5</td>
<td>mm</td>
</tr>
<tr>
<td>r_{graph}</td>
<td>25</td>
<td>mm</td>
<td>σ_{n=2}</td>
<td>1</td>
<td>mm</td>
</tr>
<tr>
<td>ω_p</td>
<td>1.5</td>
<td></td>
<td>Δφ</td>
<td>10</td>
<td>degrees</td>
</tr>
<tr>
<td>ω_o</td>
<td>1</td>
<td></td>
<td>σ_s</td>
<td>1</td>
<td>mm</td>
</tr>
<tr>
<td>Δγ</td>
<td>25</td>
<td>degrees</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Δ r</td>
<td>0.5</td>
<td>mm</td>
<td>ω_{x,y,z}</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>λ_int</td>
<td>5</td>
<td></td>
<td>ω_{int}</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>λ_{t,t,d}</td>
<td>25</td>
<td></td>
<td>e</td>
<td>0.5</td>
<td>mm</td>
</tr>
</tbody>
</table>

contour has been accurately segmented. Enforcing this convergence means the resulting annulus segmentation is independent of the initial user-provided point, making the algorithm’s results operator-independent. Section III-C shows that a relatively large region of convergence exists.

II. VALIDATION

To determine the appropriateness of the algorithm for diagnostic applications, we assessed the algorithm’s abilities to accurately locate the mitral annulus in 3DUS compared to a group of experts (Section III-A). For surgical planning purposes, we compared the algorithm’s results to the appearance of the mitral annulus as seen in a surgical view (Section III-B). We also explored the sensitivity of the algorithm to the initial user-specified point which serves as the algorithm’s input (Section III-C).

It is important to note that in these studies, the same parameter values (Table I) were used for every valve and every study. These parameters were found by testing the algorithm on 11 clinical images, none of which were included in the validation studies. The clinical images contained both normal and diseased mitral valves (3 normal, 4 with mitral prolapse, and 4 with mitral regurgitation). Of the 11 images, 10 were acquired with a transesophageal approach (X7–2t probe), and one with a transthoracic approach (X7–2 probe).

The algorithm was coded mostly in MATLAB (The MathWorks, Natick, MA), but used C++ for both computing Φ in the TTD, and for the implementation of the max-flow algorithm. Approximate running times for various portions of the algorithm (64-bit PC, 3.0 GHz Intel Core 2 Duo processor with 4 GB of RAM) were 30 s to compute the TTD, 30 s to compute the mitral leaflet surface, and 15 s to subsequently initialize and converge the contours, meaning each iteration took roughly 45 s. Typically two iterations were needed for convergence.

A. Validation Study Using Manual Image Delineations

Noise in 3DUS and volume visualization limitations make delineating the annulus from 3DUS difficult. As a result, a ground truth segmentation of the annulus cannot be defined from images. Therefore, to validate the annulus segmentation algorithm, we compared the algorithmic annulus to manual segmentations of the annulus performed by a group (n = 10) of cardiologists and trained echocardiography technicians, who will be collectively referred to as experts. We performed the analysis by first comparing the segmentations made by each expert to the collection of segmentations made by the rest of the experts for each valve. In this way, a measure of the performance for each expert could be quantified. We then compared the algorithm segmentations to the collection of segmentations made by the entire group of experts for each valve.

Experts were provided with 3DUS images of 10 different closed valves. The state of the valve and the nature of the 3DUS acquisition for each valve is summarized in Table II. The images were slices at 10° increments about a mitral valve center point and axis. We asked the experts to delineate the two annulus points in each image of the valve, for a total of 36 annulus points per valve. While the segmentations were performed in a single frame, temporal information in each image was available to the experts to allow them to accurately delineate the location of the annulus. To reduce the effects of training and fatigue in the analysis, we presented the valves to the experts in random order.

The comparison of an expert to other experts, or the algorithm to the group of experts, was done on a point-by-point basis. As some annulus points were better defined than others, it was appropriate to penalize more for deviations of a point from a well-defined annulus location than from a poorly-defined location. The measure of a well-defined and poorly-defined annulus location was based on the collection of segmentations to which a particular segmentation was being compared. Additionally, as the experts could only provide points in the space of the image, points did not deviate out of the image plane. Therefore, the analysis of points was done in ℜ² space.

When comparing an expert’s segmentation to the collection of segmentations made by the remaining experts, we refer to the i-th point of the expert’s segmentation as A_i, where i ∈ {1, ..., 36}. The mean and covariance of the i-th point as determined from the segmentations from the rest of the group are referred to as \overline{A}_i and \Sigma_i, respectively. A normalized (Mahalanobis) distance was then computed for each A_i, and the performance of an expert quantified as

\[ MND = \frac{1}{36} \sum_{i=1}^{36} \sqrt{(A_i - \overline{A}_i)^T \Sigma_i^{-1} (A_i - \overline{A}_i)} \]  

which is the mean normalized distance (MND) across all points for a valve.

We then analyzed the algorithm results in much the same way we analyzed the segmentations from each expert, where the i-th point of the algorithm’s segmentation was \overline{A}_i, and \overline{A}_i and \Sigma_i were determined from the segmentations from the remaining experts.
TABLE III
MEAN NORMALIZED DISTANCES AND DISTRIBUTION PARAMETERS FOR THE EXPERTS AND ALGORITHM

<table>
<thead>
<tr>
<th>Expert</th>
<th>Valves</th>
<th>Mean ± Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.23 ± 0.19</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.19 ± 0.19</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2.68 ± 0.19</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2.19 ± 0.19</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1.50 ± 0.19</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1.03 ± 0.19</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1.03 ± 0.19</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>1.15 ± 0.19</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>1.44 ± 0.19</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>2.70 ± 0.19</td>
</tr>
<tr>
<td>Algorithm</td>
<td>11</td>
<td>1.19 ± 0.19</td>
</tr>
</tbody>
</table>

TABLE IV
RMS DIFFERENCE BETWEEN THE ALGORITHMIC ANNULUS AND MANUAL DELINEATIONS MADE BY EXPERTS ON CLINICAL IMAGES

<table>
<thead>
<tr>
<th>Valve</th>
<th>RMS Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.95</td>
</tr>
<tr>
<td>2</td>
<td>1.37</td>
</tr>
<tr>
<td>3</td>
<td>2.75</td>
</tr>
<tr>
<td>4</td>
<td>0.96</td>
</tr>
<tr>
<td>5</td>
<td>1.19</td>
</tr>
<tr>
<td>6</td>
<td>2.50</td>
</tr>
<tr>
<td>7</td>
<td>3.22</td>
</tr>
<tr>
<td>8</td>
<td>1.82</td>
</tr>
<tr>
<td>9</td>
<td>1.27</td>
</tr>
<tr>
<td>10</td>
<td>1.09</td>
</tr>
<tr>
<td>Mean±Std. Dev.</td>
<td>1.81±0.78</td>
</tr>
</tbody>
</table>

B. Validation Study Using Surgical View Delineations

Given the poor image quality of 3DUS, validating the algorithm’s annulus to manual delineations made in 3DUS only answered the question of whether the algorithm was interpreting the images in the same way as a group of experts. The conclusion could not be made, however, that the annulus was understood to be the same shape as seen in a surgical view. Therefore, we further validated the algorithm by comparing the algorithm results to a fully visible mitral annulus delineated by a pointing device tracked using an electromagnetic (EM) tracker (miniBIRD–Model 800, Ascension Technology Corporation, Burlington, VT).

We performed this validation study using six freshly excised porcine hearts. We removed the left atrium to fully expose the mitral annulus, and to inhibit motion, secured the heart to a plastic frame using suture. After securing the heart in a water tank, to roughly simulate a peak systolic state, we artificially distended the left ventricle by connecting the aorta to a tube from a container of water elevated about two meters above the heart, and by closing the coronary arteries using suture (Fig. 11). An expert acquired a 3DUS volume of the loaded mitral valve with a transthoracic probe, and shortly after, used an EM tracked pointing device to delineate the fully visible mitral annulus with about 30 evenly placed points. Taking into account the sensor resolution and calibration procedure, point locations were computed to within a measured accuracy of 0.21 mm.

We segmented the annulus from the acquired 3DUS volume using the semi-automatic algorithm, and registered the EM delineated points to the algorithmic annulus by first manually...
aligning the two segmentations and then using an iterative-closest-point algorithm to refine the alignment. The RMS differences between the two can be seen in Table V, with Fig. 12 showing a typical comparison. The average RMS distance of the EM delineated annulus from the algorithmic annulus was $1.19 \pm 0.17$ mm, where we computed the standard deviation of the RMS distance from three different EM delineations of the annulus made by the expert on each mitral valve.

Table V

<table>
<thead>
<tr>
<th>Valve</th>
<th>RMS Difference (mm) (average ± std. dev.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.15 ± 0.04</td>
</tr>
<tr>
<td>2</td>
<td>1.35 ± 0.18</td>
</tr>
<tr>
<td>3</td>
<td>1.42 ± 0.01</td>
</tr>
<tr>
<td>4</td>
<td>1.38 ± 0.12</td>
</tr>
<tr>
<td>5</td>
<td>0.95 ± 0.42</td>
</tr>
<tr>
<td>6</td>
<td>0.91 ± 0.26</td>
</tr>
<tr>
<td>Mean</td>
<td>1.19 ± 0.17</td>
</tr>
</tbody>
</table>

C. Sensitivity Study

Just as important as the algorithm’s accuracy is the accuracy needed for the algorithm’s input, which is the user-specified point. For this study, we first established a baseline segmentation for each valve, i.e., the algorithm’s segmentation resulting from a carefully placed user-specified point near the center of the valve. The algorithm was then run on test points that were placed in the region around the center point, where we determined the center point from the baseline segmentation. If we refer to a segmentation resulting from a test point as a test annulus, then a test annulus was said to converge to the baseline if all points from the test annulus were less than 0.5 mm away from the baseline. The 3DUS images used in this study were the same as those used in the imaging validation study described in Section III-A.

Based on the test points that resulted in the algorithm converging to the baseline, we could then establish a region of convergence. To describe the region of convergence in general terms, we refer to two directions—an axial direction, which is an estimation of the mitral valve axis from the baseline segmentation, and a radial direction, which is any direction perpendicular to the axis extending out from the center point towards the annulus. We represent the location of all test points relative to these axes, and normalize the coordinates by the radius of the baseline annulus in the plane defined by the axis and the test point. We then approximate the region of convergence as the ellipsoidal region shown in Fig. 13. Table VI shows the results of the sensitivity study, where it can be seen that the region of convergence extends to an average of 85% of the annulus radius in the axial direction and 106% of the annulus radius in the radial direction.

Table VI

<table>
<thead>
<tr>
<th>Valve</th>
<th>Axial</th>
<th>Radial</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.96</td>
<td>1.34</td>
</tr>
<tr>
<td>2</td>
<td>0.44</td>
<td>0.65</td>
</tr>
<tr>
<td>3</td>
<td>0.76</td>
<td>1.13</td>
</tr>
<tr>
<td>4</td>
<td>1.24</td>
<td>1.08</td>
</tr>
<tr>
<td>5</td>
<td>0.72</td>
<td>1.11</td>
</tr>
<tr>
<td>6</td>
<td>0.71</td>
<td>0.98</td>
</tr>
<tr>
<td>7</td>
<td>0.96</td>
<td>1.16</td>
</tr>
<tr>
<td>8</td>
<td>0.78</td>
<td>0.68</td>
</tr>
<tr>
<td>9</td>
<td>1.26</td>
<td>1.17</td>
</tr>
<tr>
<td>10</td>
<td>0.67</td>
<td>1.26</td>
</tr>
</tbody>
</table>

Mean ± Std. Dev. 0.85 ± 0.26 1.06 ± 0.23
III. DISCUSSION

A. Performance and Validation

In both validation studies, the mitral annulus segmentation algorithm accurately and robustly segmented the 3D mitral annulus contour from 3DUS. When comparing the algorithm to a group of experts, results showed that the algorithm consistently segmented the annulus near the location selected by the experts (Table III). The consistency of the algorithm was better than the group of experts, i.e., while the algorithm might not have the lowest MND values for each valve, across all valves, the algorithm had a lower mean MND than the group of clinicians \((p < 0.001)\). Additionally, the algorithm and the best scoring expert were statistically indistinguishable \((p = 0.85)\). We also analyzed the expert segmentations by computing the RMS difference between the algorithm and the expert average, in which we found an average RMS difference of \(1.81 \pm 0.78\) mm. Lastly, we compared the computed annulus shape to a fully exposed porcine annulus delineated using an EM tracked pointer. Results showed an average RMS difference of \(1.19 \pm 0.17\) mm between the algorithm and the delineated points. This smaller RMS difference, compared to the validation study using the group of experts, is due to the fact that the true annulus could be seen. In contrast, the group of experts could only use 3DUS images to interpret the annulus location. Consequently, when image quality was poor, there was typically a lack of consensus as to the annulus location, resulting in a larger RMS difference between the expert average and the algorithm.

The two validation studies were complimentary. The first study used actual clinical images, which display the full range of noise and distortion encountered in medical practice. Here, the performance of the algorithm cannot be compared to a “gold standard” determination of the annulus location; instead, the results were compared to the annulus location determined by a group of practicing clinicians. This approach is most immediately relevant to diagnostic applications, where determination of the annulus shape provides insight into various pathological states [1]–[5]; current clinical practice is based on the expert average and the algorithm.

Algorithm Design

The algorithm uses a thin tissue detector which can be likened to a simplified three-dimensional ridge detector [36] for determining the valve orientation, computing the mitral leaflet surface, and finding the location of the projected mitral annulus contour. This is a key reason the algorithm is successful: the only extensive regions of thin tissue in the image volume are the mitral valve leaflets, so the algorithm focuses on the mitral apparatus and ignores the rest of the volume. This generates accurate results despite poor placement of the user input point. Characterizing the leaflet location using the thin tissue detector, however, does place restrictions on the ultrasound imaging, in that the thin leaflet tissue needs to be visible [Fig. 14(a)]. This is not the case when the ultrasound wave propagation is parallel to the leaflet surface orientation [Fig. 14(b) and (c)], as this produces little acoustic return. The algorithm will fail on images such as these. However, similar restrictions of pathology of the mitral valves, the probe orientation relative to the valve, and the ultrasound machine settings, the algorithm accurately segmented the annulus using the same algorithm parameter values for every image, reinforcing the robustness of the method.

The measured accuracy of this algorithm is difficult to compare to prior work, as previously published semi-automatic mitral annulus segmentation algorithms include minimal validation [25], [26]. In [26], indirect quantitative analysis of the method was performed by comparing measurements from a model to published population valve dimensions. As an example, for average annulus diameter dimensions, the method reported errors of roughly 15%. We measured the annuli from our algorithm and the EM delineations discussed in Section III-B and found an average diameter difference under 4%.
would also exist for the previous work in annulus segmentation presented in [25] and [26], and is to be expected for algorithms wishing to delineate the mitral annulus from a single ultrasound frame and with no use of temporal information.

With ultrasound imaging, anatomical structures become more or less defined depending on their orientation to the acoustic propagation (Fig. 14). This helps to explain why the presented method cannot be used to find the annulus throughout the cardiac cycle, but rather was designed for just closed valves. For instance, if the probe orientation such as that shown in Fig. 14(a) was used to view the mitral valve throughout the cardiac cycle, when the valve opens [Fig. 14(b)], the leaflets become oriented such that they become poorly defined. Additionally, when the valve opens, the leaflets can become difficult to distinguish from the left ventricle wall that they are pushed against.

The graph cut method (max-flow algorithm) is used in the annulus segmentation algorithm first in a 3D graph to find a surface at the location of the mitral leaflets, and then in a 2D graph to find a contour at the location of the projected mitral annulus in the projection images. There are several reasons why we use the graph cut method. One of the main reasons is the fact that it finds an optimal solution to an energy minimization problem, meaning we will not fall victim to the local minima that many methods are subject to which attempt to find an optimal surface or contour (i.e., active contours [31], level sets [32], etc.). This is especially important in cardiac ultrasound, where anatomic variation along with the noise, artifacts, and inhomogeneities inherent in ultrasound imaging have the potential to create several local minima. Other reasons to use the graph cut method are that it does not require an initialization, and a limited amount of prior knowledge can be enforced through mindful construction of the graph. Therefore, the need for additional user input, for either initialization or to encourage a specific geometry, is eliminated without sacrificing accuracy, and we avoid the variation inherent in any additional user inputs. The combination of these characteristics of the graph cut method helps to make the annulus segmentation algorithm accurate and also helps to explain why the algorithm has such a large region of convergence relative to the user-provided point.

The annulus segmentation algorithm can be simplified to two main steps: find the leaflets, then search along the leaflets to find the annulus. This was consistent with the approach used by the experts (cardiologists and technicians) to delineate the annulus. Fig. 15 shows an example of the distribution of the human selected annular points which are aligned along the leaflets. The main difference was that after finding the leaflets, the human observers had access to temporal information to delineate the annulus location, whereas the algorithm used only a single 3DUS volume.

B. Future Work

One of the most significant improvements to the algorithm would be to integrate temporal information to more accurately and robustly locate the annulus. Currently, the algorithm finds the annulus as the location at the leaflets where the tissue thickness changes abruptly along the computed leaflet surface. A drawback of this approach, however, is that if the TTD is improperly tuned or if the leaflets are severely calcified, the annulus might not be accurately located, which might be ameliorated when temporal information is used.

The large region of convergence for the algorithm makes it an attractive candidate for complete automation. Because the only input to the algorithm is the user-specified point, if this point can be delineated by some automated process, the annulus segmentation algorithm could be made fully automatic. Special care would need to be taken, however, to account for the different appearances of valves due to varying pathologies, ultrasound probe orientations, and ultrasound machine settings. An example of a method that could be used for this automatic selection is the work presented in [37], which automatically segments the mitral valve plane and left ventricular long-axis. The intersection of the plane and the axis could be used to define the user-specified point.

The annulus segmentation algorithm presents many opportunities in clinical studies. The algorithm can help to analyze large clinical databases to explore correlations between pathologies and annular geometries. It can also be used to monitor patient anatomy over time to determine if geometry at an earlier age has any indication as to pathologies that may develop at later ages.

To build on this algorithm, future work will also include segmentation and/or tracking of the annulus over the duration of the cardiac cycle. However, given that the algorithm can segment the mitral annulus whenever the leaflets are coapted, the shape and motion of the annulus throughout most of the cardiac cycle can already be obtained from the present method.

IV. Conclusion

The annulus segmentation algorithm accurately and robustly segments the mitral annulus from a 3DUS frame using only a single user specified point. The approach first uses a thin tissue detector to localize the mitral leaflets, and then the graph cut algorithm to define the leaflet surface. The annulus is detected as the curve where the thin leaflet surface meets the thick heart wall using another graph cut implementation and multiscale active contours. Validation using clinical 3DUS images showed that estimated annulus positions were comparable to expert manual
segmentation. A second validation study showed that the algorithm matched annulus delineations made by an expert using an EM tracked pointing device, with an RMS error of 1.19 ± 0.17 mm across six valves. The algorithm is insensitive to the placement of the user-specified point within about half of the valve radius, and a single set of parameters was used for all valve images, with no fine-tuning required. The algorithm can be used to increase speed and consistency in determining annulus location for diagnostic, surgical planning, and database applications.

ACKNOWLEDGMENT

The authors would like to thank T. Zickler and R. Zabih for helpful discussions on the max-flow algorithm, and H. Loyola for discussions on the statistical analysis.

REFERENCES