

Research Article

Generation of Myocardial Wall Surface Meshes from Segmented MRI

Oskar Škrinjar¹ and Arnaud Bistoquet²

¹Department of Biomedical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

²Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Correspondence should be addressed to Oskar Škrinjar, oskar.skrinjar@bme.gatech.edu

Received 3 May 2009; Accepted 21 September 2009

Recommended by Lizhi Sun

This paper presents a novel method for the generation of myocardial wall surface meshes from segmented 3D MR images, which typically have strongly anisotropic voxels. The method maps a premeshed sphere to the surface of the segmented object. The mapping is defined by the gradient field of the solution of the Laplace equation between the sphere and the surface of the object. The same algorithm is independently used to generate the surface meshes of the epicardium and endocardium of the four cardiac chambers. The generated meshes are smooth despite the strong voxel anisotropy, which is not the case for the marching cubes and related methods. While the proposed method generates more regular mesh triangles than the marching cubes and allows for a complete control of the number of triangles, the generated meshes are still close to the ones obtained by the marching cubes. The method was tested on 3D short-axis cardiac MR images with strongly anisotropic voxels in the long-axis direction. For the five tested subjects, the average in-slice distance between the meshes generated by the proposed method and by the marching cubes was 0.4 mm.

Copyright © 2009 O. Škrinjar and A. Bistoquet. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Surface models of the epicardium and endocardium of the heart chambers are used in a number of biomedical applications for visualization [1], virtual reality [2], segmentation [3, 4], motion analysis [5, 6], shape analysis [7, 8], and modeling [9, 10] purposes. A typical approach to generate subject specific models is to apply a surface construction algorithm to segmented cardiac magnetic resonance (MR) or computed tomography (CT) images. Cardiac MRI is a widely used modality to image the heart. However, due to the tradeoff between image quality and temporal and spatial resolution, voxels in cardiac MR images are strongly anisotropic. Typically, the in-plane resolution is a few times higher than the out-of-plane resolution for clinically used cardiac MRI. While a number of mesh generation methods exist [11, 12], to the best of our knowledge, there is no surface mesh generation method designed for images with strongly anisotropic voxels. The most widely used method for surface mesh generation from images is the marching

cubes method [13]. If the marching cubes method is applied to an image with strongly anisotropic voxels without any additional processing, then the generated mesh has strongly irregular triangles, pronounced terracing artifacts and the number of triangles directly related to the number of voxels, none of which is desirable (Figure 1). Terracing artifacts are a consequence of marching cubes mesh closely following sharp voxel boundaries (instead of smoothly fitting them), they appear as “stairs” or “terraces” and in Figure 1 they are visible as dark and light red vertical stripes. Once the marching cubes mesh is generated, one can apply a number of techniques to improve the mesh quality, including mesh smoothing [14], simplification [15], and optimization [16]. A combination of these techniques was used in [17]. An alternative way to use the marching cubes is to first construct an implicit surface from the segmented image, and then apply the marching cubes to the scalar field (the implicit surface is the zero level set of the scalar field) on a uniformly sampled image domain [18]. This way one can control the size of the mesh triangles (determined by

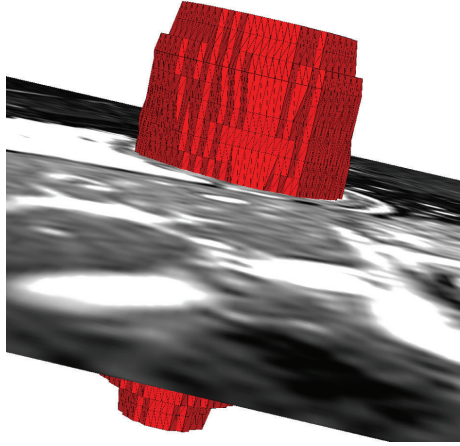


FIGURE 1: A left ventricular surface model generated by applying the marching cubes algorithm to a segmented cardiac MR image with 1.44 mm in-plane resolution and 8.0 mm slice thickness. The irregular triangles are a consequence of the voxel anisotropy. The surface mesh has pronounced terracing artifacts, and the number of triangles is directly related to the number of voxels in the image.

the sampling interval) as well as make them more regular (consequence of the uniform sampling), but they can still be badly shaped and the terracing artifacts remain. To alleviate these problems, Peiró et al. [18] used a number of mesh optimization techniques. Instead of using the discrete image, one can interpolate image intensities, for example by means of trilinear interpolation, before constructing a surface mesh [19]. However, in the case of strongly anisotropic voxels, this approach would still result in terracing artifacts, although somewhat smoothed. Lötjönen et al. [20] also used the marching cubes as the starting point of their mesh generation method, which, if the mesh is decimated enough, generates close-to-regular triangles. The method of Gibson [21], while significantly reducing terracing artifacts, shares with the marching cubes the problem of irregular triangles in the case of anisotropic voxels. Another group of methods constructs surface meshes from 2D contours of the segmented image structures [22–25]. They suffer from the same problem: if the voxels of the underlying image are anisotropic the resulting triangles are irregular. We also note related work in which researchers used a structured volumetric mesh of the left ventricle for myocardial motion recovery [26–29].

In this paper we present a method for generation of myocardial wall surface meshes from segmented MRI. The meshes are smooth and have prespecified number of triangles and close-to-regular triangles despite the highly anisotropic voxels. Since the marching cubes are the most widely used method, either as a stand-alone method or as a part of other methods, we compare the proposed method to the marching cubes.

2. Methods

2.1. Approach. The presented method is designed for surface mesh generation of the endocardium of the four cardiac

chambers and of the endocardium from segmented cardiac MRI. The four endocardial surfaces are, if the valves are ignored, topologically equivalent to a sphere. We also assume that the segmentation of the entire heart does not include other structures, which makes its outer surface topologically equivalent to a sphere. The main idea is to generate a triangulated mesh on a sphere and then map it independently to the five surfaces. For each segmented object we construct its surface in implicit form and then map the mesh from the sphere to the surface using the gradient field of the solution of the Laplace equation between the surface and the sphere. Each step of the method is explained in the following sections, and Figure 2 summarizes the method.

2.2. Sphere Triangulation. It can be shown that a sphere cannot be triangulated with an arbitrary number of equilateral triangles. In fact, there are only three configurations of a triangulated sphere with equilateral triangles: regular tetrahedron (4 equilateral triangles), regular octahedron (8 equilateral triangles), and regular icosahedron (20 equilateral triangles) [30]. A triangulation of a sphere with any other number of triangles cannot have all the triangles equilateral. There are a number of ways to approximately uniformly sample a sphere and construct the corresponding triangulation [31, 32]. Here we use the method of minimizing the electrostatic energy of equally charged particles on a sphere [33, 34]. Once the points are approximately uniformly distributed on a sphere, we construct a triangular mesh by using the Delaunay triangulation [35]. This method allows for the construction of a close-to-regular triangular mesh on a sphere with an arbitrary number of vertices V , which is related to the number of triangles T as $2V - T = 4$. This relationship follows from the Euler’s formula for polyhedra [30].

2.3. Solution of the Laplace Equation. In order to construct the surface mesh, we define a homeomorphic mapping from the sphere to the surface and apply it to the mesh on the sphere. There are infinitely many ways to construct such a mapping, and here we define a scalar field u between the sphere and the surface (we make the sphere larger than the object and center it at the barycenter of the object), and then any point from the sphere is mapped to the surface by following ∇u , the gradient of u . In order for the mapping to be homeomorphic, the gradient flow has to be divergence free, that is, $\text{div } \nabla u = 0$, which leads to the familiar Laplace equation

$$\Delta u = 0. \quad (1)$$

We look for the solution of the Laplace equation that is equal to 0 at the sphere and 1 at the object surface. The Laplace operator is rotation invariant, and a solution to the Laplace equation has no local extrema, which makes it a suitable means to transport the mesh from the sphere to the surface of the object. If the sphere has a radius of R , then the external boundary condition can be specified as

$$u(\mathbf{r})|_{|\mathbf{r}|=R} = 0. \quad (2)$$

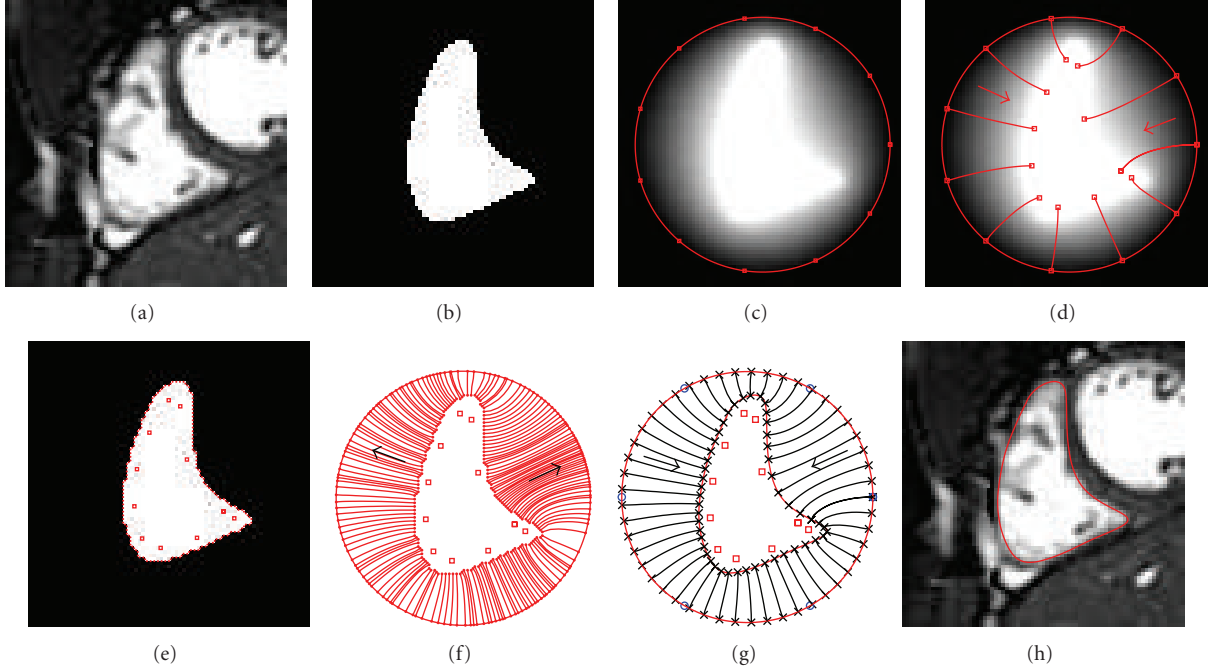


FIGURE 2: Mesh generation summary. The input image (a) is segmented into the object and background, resulting in a binary image (b). A sphere enclosing the object is centered at the object barycenter (c). The sphere is uniformly sampled with the number of points equal to the number of singularities. The binary image is resampled with isotropic voxels and the Laplace equation is numerically solved between the sphere (boundary condition of 0) and the object (boundary condition of 1). The solution of the Laplace equation is encoded in the gray levels in (c) and (d). The binary object is eroded, and the points are propagated from the sphere to the eroded object in the direction of the gradient of the Laplace equation solution to define the singularity locations, shown as red squares in (d) and (e). Boundary points, specified as midpoints for each pair of neighboring voxel, where one voxel is in the object and the other is in the background, are shown as red dots in (e). The singularity locations as well as the boundary points are used to specify the analytic solution of the Laplace equation. The boundary points are propagated in the negative gradient direction of the solution of the Laplace equation from the object boundary to the sphere (f). Their values of the underlying solution of the Laplace equation are interpolated at the sphere to define the stopping function. The number of degrees of freedom of the stopping function is defined by the number of control points, which are shown as blue circles in (g). An approximately uniform mesh is generated on the sphere. The vertices of the mesh on the sphere, shown as black crosses in (g), are propagated from the sphere in the direction of the gradient of the solution of the Laplace equation until the value of the underlying solution of the Laplace equation is equal to the corresponding value of the stopping function. The propagated mesh nodes define the final mesh, shown in (h). Figures (a)–(h) are two dimensional for illustration purposes, while the method is three dimensional.

The internal boundary condition is discussed in Section 2.4. We use the method of fundamental solutions to solve the Laplace equation. The solution is continuous (as opposed to discrete) and it is represented as a linear combination of functions, each satisfying the Laplace equation and the external boundary condition and each having a singularity within the sphere (see appendix)

$$u(\mathbf{r}) = \sum_{m=1}^M c_m f_{s_m}(\mathbf{r}), \quad (3)$$

where s_m , $m = 1, \dots, M$, are the locations of M singularities and c_m the corresponding coefficients. It is straightforward to show that u from (3) satisfies the Laplace equation (1) and the external boundary condition (2).

2.4. Internal Boundary Condition. While expression (3) and the external boundary condition are in the continuous form, the internal boundary is discrete, defined by the object segmentation map. The strongly anisotropic voxels are the

main reason for irregular triangles and terracing effects present in the meshes generated by methods based on the marching cubes. For this reason, we represent the internal boundary in the continuous form as a level set of u from (3). First, we define the set of boundary points \mathbf{r}_n , $n = 1, \dots, N$. For each pair of neighboring voxels from the segmentation map that have different labels (i.e., one voxel belongs to the object and one does not) the midpoint between the two voxels is a boundary point. Then, we determine the parameters (singularity locations and coefficient values) of u such that its level set of 1 fits the boundary points in the least squares sense, that is, we minimize

$$O = \frac{1}{2} \sum_{n=1}^N [u(\mathbf{r}_n) - 1]^2. \quad (4)$$

We require all the singularities to be located within the object. Each singularity has the corresponding singularity outside the sphere (see appendix). Scalar field u is well defined in the domain (between the object surface and the

sphere) since there are no singularities. However, there is no closed form solution that involves the optimal locations of the singularities. To avoid numerical optimization which is prone to local extrema, we preset the singularity locations, and then find the optimal coefficients c_m that minimize (4). While a closed form solution for the optimal coefficients can easily be obtained, in the general case some of the optimal coefficients can have positive and some negative values. However, around each singularity with a negative coefficient there will be a region with negative values of u (u will tend to $-\infty$ at such singularities). These “islands” of negative values may be fully contained within the object but they also may protrude into the domain between the object surface and the sphere, affecting the 1 level set in an undesired way. To prevent this from happening, one can constrain the optimization to have only positive values for c_m . However, there is no closed form solution of this problem. To avoid numerical optimization which is prone to local extrema, we resort to an alternative approach. We approximately uniformly place the singularities inside the object relatively close to the object surface (see Section 2.6), and assume that all the coefficients have the same value, that is, $c_m = c$. The optimal value of c is obtained from

$$\frac{\partial O}{\partial c} = 0, \quad (5)$$

which has a closed form solution

$$c = \frac{\sum_{n=1}^N d_n}{\sum_{n=1}^N d_n^2}, \quad (6)$$

where $d_n = \sum_{m=1}^M f_m(\mathbf{r}_n)$. It can be shown that $c > 0$, that is, there will be no singularities with negative coefficients, which could lead to undesired mesh shapes. However, since the same coefficient is used for all the singularities, the fitting of the implicit surface (level set 1 of u) to the boundary points is not as accurate as in the case of singularities with nonequal coefficients. To increase the accuracy of the fitting, we use a stopping function (see Section 2.7).

2.5. Mesh Propagation from the Sphere to the Surface. To map the mesh from the sphere to the object surface, we propagate each mesh vertex along the gradient of u ; that is, the trajectory $\mathbf{r}(t)$ of a given vertex \mathbf{v} on the sphere is

$$\begin{aligned} \frac{d\mathbf{r}(t)}{dt} &= \nabla u(\mathbf{r}(t)), \\ \mathbf{r}(0) &= \mathbf{v}, \end{aligned} \quad (7)$$

where t is the parameter of the trajectory. We use the fourth-order Runge-Kutta method [36] to integrate the trajectory numerically. Since u is a continuous and exact solution of the Laplace equation, the only propagation error comes from the numerical integration error of the Runge-Kutta method. If the Laplace equation was solved approximately, the nonexactness of the solution would be translated into additional propagation error.

2.6. Placement of Singularities. To uniformly place the singularities inside the object relatively close to the object surface, we first approximately uniformly sample the sphere (as explained in Section 2.2) with the number of points equal to the number of singularities. Then, we resample the segmentation map to obtain isotropic voxels and erode the segmented object two times. In the next step, we numerically solve the Laplace equation between the sphere (with a boundary condition of 0) and the surface of the eroded object (with a boundary condition of 1) by using a relaxation method [36]. We utilize the isotropic voxels as the grid on which we solve the Laplace equation. Finally, we propagate the points from the sphere to the eroded object in the direction of the gradient of the solution of the Laplace equation to obtain the singularity locations.

2.7. Stopping Function. To increase the accuracy of the fitting of the object surface to the boundary points, instead of propagating the mesh from the sphere to the level set of 1, we define a “stopping” function on the sphere, which for every point on the sphere determines the value of u that point will be propagated to (that value would be 1 if the mesh was propagated to the level set of 1).

To represent the stopping function we use pseudospherical plate spline model on the sphere proposed by Wahba [37]

$$b(\hat{\mathbf{p}}) = \alpha_0 + \sum_{k=1}^K \alpha_k \psi(\hat{\mathbf{p}} \cdot \hat{\mathbf{q}}_k), \quad (8)$$

where $\hat{\mathbf{p}}$ is a unit vector representing a point on the sphere, K is the number of control points, $\hat{\mathbf{q}}_k$ are unit vectors defining the control points on the sphere, $\alpha_0, \dots, \alpha_K$ are scalar coefficients, and ψ is defined in [37] for $m = 4$. We set the control points $\hat{\mathbf{q}}_k$ by approximately uniformly sampling a sphere (as explained in Section 2.2) with K points.

At this point the singularity locations as well as coefficient c are set; that is, the scalar field u is completely defined. At each boundary point \mathbf{r}_n we record the value of the scalar field $u(\mathbf{r}_n)$. Let $\hat{\mathbf{p}}_n$ denote the unit vector of the point on the sphere obtained by propagating the boundary point \mathbf{r}_n to the sphere by following the negative gradient of u . We determine $\alpha_0, \dots, \alpha_K$ by requiring $b(\hat{\mathbf{p}})$ at points $\hat{\mathbf{p}}_n$ to approximate values $u(\mathbf{r}_n)$ in the least squares sense, that is, by minimizing

$$\sum_{n=1}^N [b(\hat{\mathbf{p}}_n) - u(\mathbf{r}_n)]^2. \quad (9)$$

The closed form solution is

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = (G^T G)^{-1} G^T \begin{bmatrix} u(\mathbf{r}_1) \\ u(\mathbf{r}_2) \\ \vdots \\ u(\mathbf{r}_N) \end{bmatrix}, \quad (10)$$

where

$$G = \begin{bmatrix} 1 & \psi(\hat{\mathbf{p}}_1 \cdot \hat{\mathbf{q}}_1) & \cdots & \psi(\hat{\mathbf{p}}_1 \cdot \hat{\mathbf{q}}_K) \\ 1 & \psi(\hat{\mathbf{p}}_2 \cdot \hat{\mathbf{q}}_1) & \cdots & \psi(\hat{\mathbf{p}}_2 \cdot \hat{\mathbf{q}}_K) \\ \vdots & \vdots & & \vdots \\ 1 & \psi(\hat{\mathbf{p}}_N \cdot \hat{\mathbf{q}}_1) & \cdots & \psi(\hat{\mathbf{p}}_N \cdot \hat{\mathbf{q}}_K) \end{bmatrix}. \quad (11)$$

Coefficient c is computed such that (4) is minimized, which means that the values $u(\mathbf{r}_n)$ are close to 1. Coefficients $\alpha_0, \dots, \alpha_K$ are computed such that (9) is minimized, which means that $b(\hat{\mathbf{p}}_n)$ is close to $u(\mathbf{r}_n)$, which in turn is close to 1. Since typically points $\hat{\mathbf{p}}_n$ relatively densely cover the sphere, the stopping function $b(\hat{\mathbf{p}})$ is close to 1 everywhere on the sphere.

We propagate the mesh vertices from the sphere to the object surface according to (7). For a given mesh vertex \mathbf{v} on the sphere, the corresponding unit vector is $\mathbf{v}/|\mathbf{v}|$ and the value of the stopping function for that vertex is $b(\mathbf{v}/|\mathbf{v}|)$. Over the course of propagation the underlying value of the scalar field $u(\mathbf{r}(t))$ grows and when it reaches $b(\mathbf{v}/|\mathbf{v}|)$ the propagation stops.

Instead of using the value of 1 to stop all the vertices, we increase the accuracy of the fitting of the boundary points by using the stopping function. The more control points the more accurate the fitting, and the fewer control points the smoother the final surface.

2.8. In-Slice Distance Calculation. To quantify the closeness of the meshes generated by the marching cubes and the proposed method we compute the in-slice distance between the respective mesh cross-sections in a given slice (results given in Section 3.4). To measure the in-slice distance we densely sample the two cross-sections. For a given point (\mathbf{p}_1) in the first set of samples we find the closest point (\mathbf{p}_2) in the second set of samples and then for \mathbf{p}_2 we find the closest point (\mathbf{p}_3) in the first set of samples. If \mathbf{p}_3 is the same as \mathbf{p}_1 then we say that \mathbf{p}_1 and \mathbf{p}_2 form a pair of corresponding points in the two cross-sections. We find all the pairs of corresponding points in the two cross-sections and compute their distances, from which we compute the average distance and standard deviation.

3. Results

3.1. Test Images. The method was tested using anatomical cardiac MRI scans of five healthy volunteers. The scans were acquired using steady-state free-precession short axis cine imaging with a 1.5 T clinical MRI scanner (Intera, Philips Medical Systems, Best, The Netherlands). The scans had 12-17 contiguous short axis slices with 256×256 pixels, 8 mm slice thickness, 1.44 mm in-plane resolution and a 20 cm field of view. A flip angle of 65° , TR of 3.4 milliseconds, and TE of 1.7 milliseconds were used. The four cardiac chambers as well as the entire heart were manually segmented in all five scans.

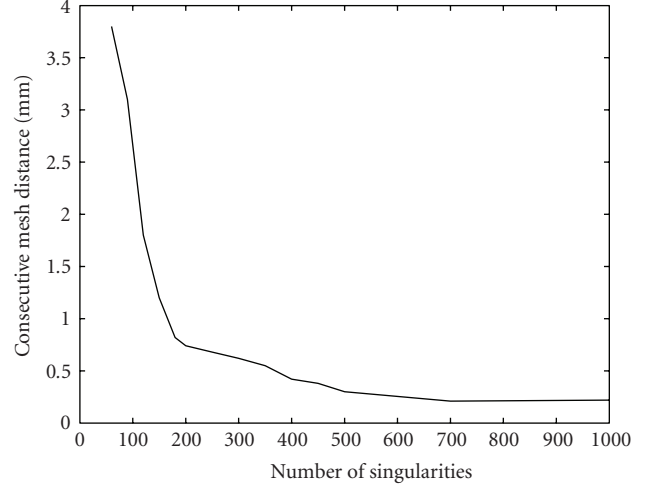


FIGURE 3: Average distance between consecutive meshes as a function of the number of singularities.

3.2. Method Parameters. The presented method has three parameters: the number of singularities (M), the number of control points (K), and the number of mesh vertices (V). Note that M and K affect the shape of the continuous implicit surface, while V affects the triangulation of the continuous implicit surface.

In this section we analyze the effect of the parameter values on the resulting mesh. The studies were done on the right ventricle of one of the subjects, since the right ventricle is more curved than the other three chambers and is the only chamber that has both convex and concave regions.

In the first study we generated a sequence of surface meshes by increasing the number of singularities. Then we computed the average distance between consecutive meshes in the sequence to quantify the change the mesh undergoes as M is increased (Figure 3).

In the second study we generated a sequence of surface meshes by increasing the number of control points. Then we computed the average distance between consecutive meshes in the sequence to quantify the change the mesh undergoes as K is increased (Figure 4).

In the third study we generated a sequence of surface meshes by increasing the number of mesh vertices. Then we computed the average distance between consecutive meshes in the sequence to quantify the change the mesh undergoes as V is increased (Figure 5).

3.3. Mesh Quality. To measure the mesh quality, we use a triangle quality index suggested in [18], which can be evaluated as

$$Q = \frac{8(p-a)(p-b)(p-c)}{abc}, \quad p = \frac{a+b+c}{3}, \quad (12)$$

where a , b , and c are the lengths of the three sides of the triangle. It can be shown that $0 \leq Q \leq 1$ for any triangle, $Q = 1$ for an equilateral triangle, Q is close to zero for irregular triangles, and $Q = 0$ for a zero-area triangle. Triangles with $Q > .5$ are considered to be of reasonably good quality.

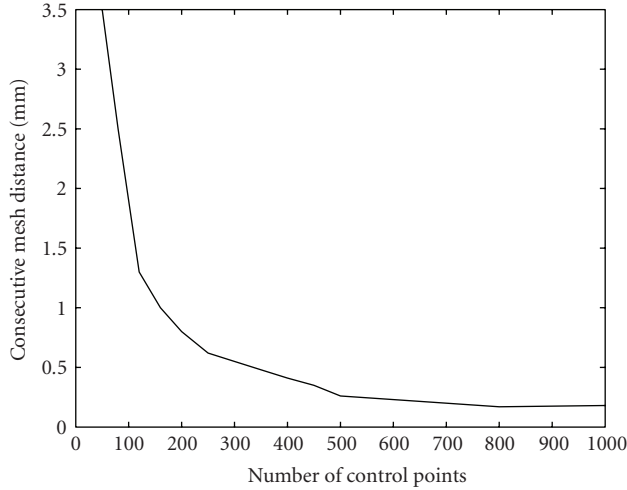


FIGURE 4: Average distance between consecutive meshes as a function of the number of control points.

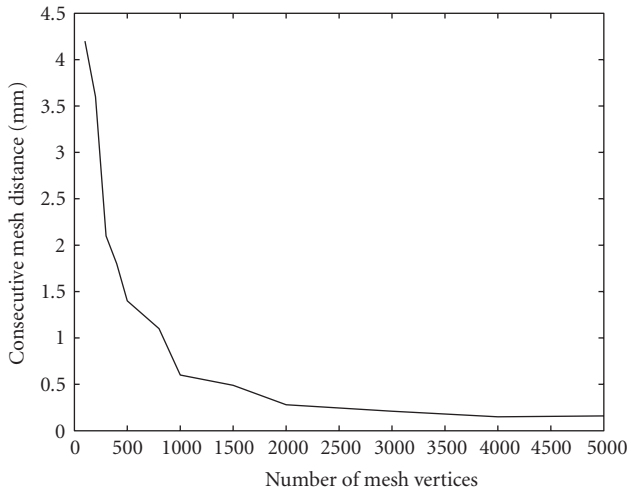


FIGURE 5: Average distance between consecutive meshes as a function of the number of mesh vertices.

We generated a sequence of surface meshes by increasing the number of mesh vertices. Then we computed the average quality index for each mesh in the sequence (Figure 6). One can see that the quality index is practically independent of the number of mesh vertices and that it is relatively high ($Q \approx .85$).

It should be noted that the average Q cannot be 1. This is true even for the sphere, since it cannot be triangulated with an arbitrary number of equilateral triangles. Figure 7 shows the mesh on the sphere and the corresponding right ventricular mesh for four different numbers of mesh vertices and reports the corresponding Q values.

3.4. Comparison to the Marching Cubes. The method was tested on the endocardial surfaces of the four cardiac chambers as well as on the epicardial surface of the entire heart for five subjects. The numbers of singularities, control

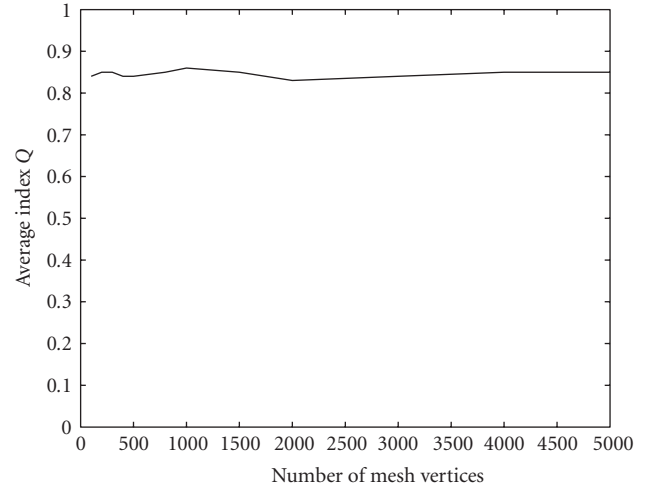


FIGURE 6: Average triangle quality index as a function of the number of mesh vertices.

TABLE 1: The average in-slice distance (D) between the surface meshes generated by the marching cubes and the proposed method is given for each mesh. The in-plane resolution was $1.44 \text{ mm} \times 1.44 \text{ mm}$. The numbers of singularities (M), control points (K), and mesh vertices (V) used to generate the meshes with the proposed method are also reported.

	M	K	V	D [mm]
LV endocardium	227	204	1180	$0.4 \pm .07$
RV endocardium	246	225	1220	$0.3 \pm .05$
LA endocardium	62	66	748	$0.5 \pm .06$
RA endocardium	57	58	684	$0.4 \pm .05$
Epicardium	422	406	2472	$0.3 \pm .05$

points, and mesh vertices used for the test are reported in Table 1. Figures 8 and 9 show the endocardial and epicardial surface meshes for one of the subjects.

Since the marching cubes are the most widely used method, either as a stand-alone method or as a part of other methods, we compared the endocardial and epicardial surface meshes of the five subjects generated by the proposed method to the corresponding surface meshes generated by the marching cubes. The comparison was done in the short-axis slices since the in-plane resolution was 5 times higher than the out-of-plane resolution. Figure 10 shows cross-sections of segmented blood pools and the corresponding contours from endocardial meshes obtained by the marching cubes and the proposed method. Table 1 provides the average in-slice distance between the surface meshes generated by the marching cubes and the proposed method. The averages were computed over all the short-axis slices of each cardiac chamber for the five subjects.

4. Discussion

The presented method can be used for the surface mesh generation of any object that is topologically equivalent to a sphere. While the method can be extended to control

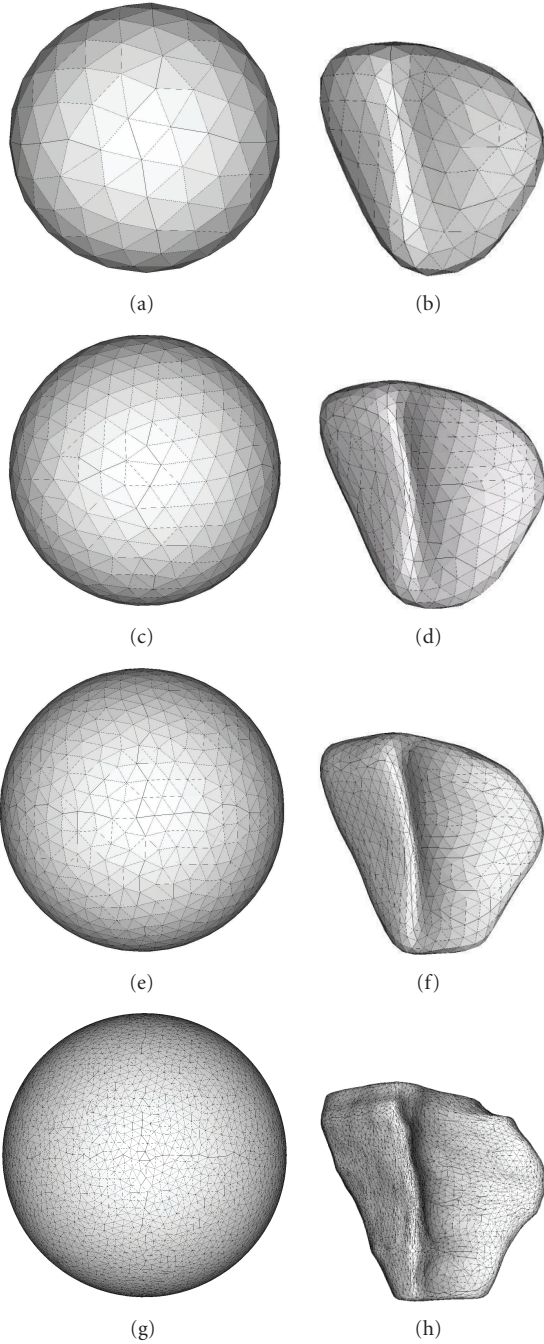


FIGURE 7: Each row shows a mesh on the sphere and the corresponding right ventricular mesh obtained by propagating the mesh from the sphere to the right ventricular surface. The numbers of mesh vertices for the four rows are 200, 500, 1000, and 5000. The corresponding mean \pm std (min, max) Q values for the mesh on the sphere are $0.93 \pm 0.07(0.75, 1)$, $0.94 \pm 0.06(0.78, 1)$, $0.93 \pm 0.07(0.77, 1)$, and $0.95 \pm 0.05(0.76, 1)$, and for the right ventricular mesh are $0.85 \pm 0.07(0.68, 0.99)$, $0.84 \pm 0.06(0.62, 0.99)$, $0.86 \pm 0.06(0.65, 0.99)$, and $0.85 \pm 0.07(0.63, 0.99)$.

the triangle size based on the surface curvature, there is no need for such an approach in the case of myocardial wall surfaces since they are not highly curved. The method, unlike

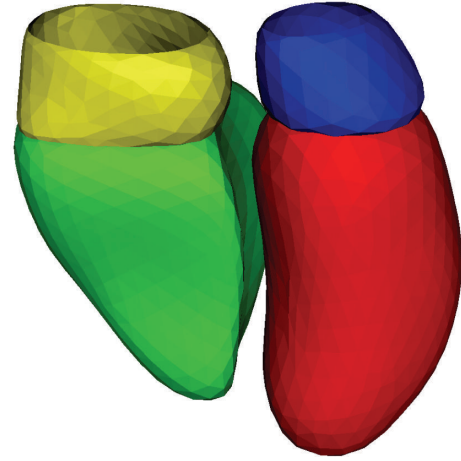


FIGURE 8: Endocardium surface meshes generated by the proposed method for the left ventricle (red), right ventricle (green), left atrium (blue), and right atrium (yellow).



FIGURE 9: Epicardium surface mesh generated by the proposed method for the entire myocardium.

other mesh generation methods, allows for a direct control of the number of triangles and vertices in the mesh, which is particularly useful in modeling (e.g., FEM) applications.

The method can be used for the generation of surfaces that are not closed. For example acquired cardiac MRI might not contain slices going through the apex and the base, in such case the corresponding endocardial and epicardial surfaces are not closed. In such cases one can segment the acquired slices, generate the mesh using the proposed method and then clip the bottom and top part of the mesh. This was done for the two atria in Figure 8 and for the epicardium in Figure 9.

In the examples presented in this paper we generated triangulated surface meshes. However, the method is independent of the type of the mesh; that is, it can be used with any mesh elements as long as the sphere can be meshed with such elements. Once the sphere is meshed, the vertices of the mesh are propagated from the sphere to the surface of the segmented object in the way explained in Section 2.5.

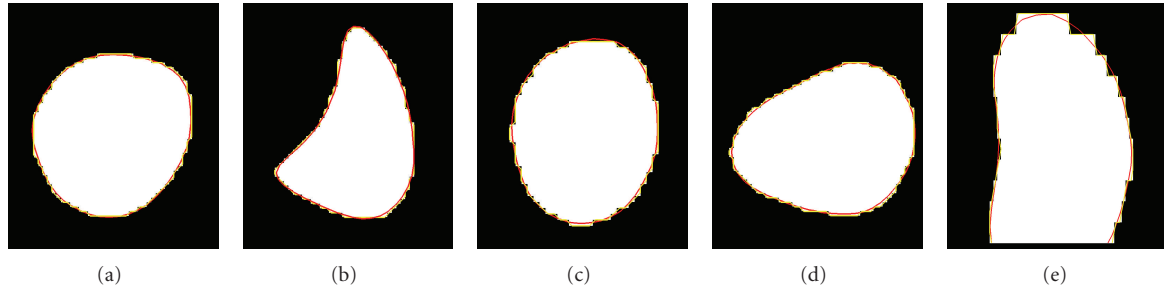


FIGURE 10: Contours of endocardial meshes generated by the marching cubes (yellow) and the proposed method (red) in short-axis sections for (a) left ventricle, (b) right ventricle, (c) left atrium, and (d) right atrium, and in a long-axis section for (e) left ventricle. The endocardial boundaries are defined by the blood pool segmentation shown in the binary images.

Figures 3 and 4 show that the method converges as the number of singularities or the number of control points is increased, which is a desirable behaviour. It means that beyond certain number of singularities and control points the method behaves the same. By comparing Figures 3 and 4 one can see that the two parameters have very similar behaviour, and this is why we used similar values for the two parameters (Table 1) when we compared our method to the marching cubes (Section 3.4). The two parameters have the same effect: they control the smoothness of the underlying continuous surface. The higher their values the smoother the surface, and the lower their values the better the surface fits the boundary points. For these reasons, one can use the same value for the two parameters and treat them as one parameter.

While the number of singularities and the number of control points control the smoothness of the underlying implicit continuous surface, the number of mesh vertices affects the triangulation of the continuous surface. From Figure 5 one can see that the method converges as the number of vertices is increased, which is a desirable behaviour.

The reason why the graphs in Figures 3, 4, and 5 do not exactly go to zero is the numerical errors in the implementation of the method resulting in submillimeter differences in the final location of the mesh vertices. We use a continuous and exact solution of the Laplace equation, since an approximate solution would increase the numerical errors.

The constant and relatively high value of the triangle quality index in Figure 6 shows that the mesh has close-to-regular triangles for a range of numbers of vertices. The same conclusion can be made from Figure 7.

The proposed method generates meshes that are very close to the ones obtained by the marching cubes (Figure 10). While the differences in the short axis planes between the meshes generated by the two methods were submillimeter (Table 1), the meshes generated by the proposed method had about five times fewer triangles than the corresponding meshes generated by the marching cubes. However, unlike the meshes generated by the marching cubes (Figure 1), the meshes generated by the proposed method (Figures 8 and 9) are smooth and have close-to-regular triangles.

The proposed method can be used with segmented images that have anisotropic voxels. The segmentation boundary points are not strictly interpolated. Instead, they are approximated with an implicit surface that fits them in the least square sense. The surface smoothness versus the goodness of fit is controlled by the number of singularities and number of control points, which can take the same value. If the implicit surface is smooth then the resulting mesh is also smooth. Thus, there is no need for artificial smoothing of the mesh that may shrink or affect the mesh in some other undesired way.

The entire method has been designed to completely avoid numerical optimization and consequently the problem of local extrema.

In the proposed method the segmentation boundary points are approximated with an implicit surface, which is then triangulated by propagating a regular mesh from a sphere to the surface. There are other ways to construct continuous surfaces that interpolate or approximate a given set of points (e.g., [38]). However, our representation allows for an exact continuous solution of the Laplace equation, while other surface models would require a numerical solution, which in turn would increase the mesh propagation error. Instead of propagating a regular mesh from the sphere to the surface, one can use a method for direct triangulation of implicit surfaces (e.g., [39–41]). These methods are more general than our method since they can deal with an arbitrary topology. They march triangles over the surface and use heuristics to close the triangulation. Unlike these methods, the proposed method, while restricted only to spherical topology, does not need a heuristics to close the triangulation and it can generate meshes with a prespecified number of triangles.

We note that harmonic functions have already been used to represent shapes [42] and that there are other ways to map a sphere to the surface or vice versa (e.g., [43]).

In the myocardial motion analysis community researchers used a structured volumetric mesh of the left ventricle [26–29]. The model was defined in the prolate spheroidal coordinate system, and its epicardial and endocardial surfaces were fit to left ventricular wall boundary points extracted from cardiac MRI. The obtained continuous

smooth model was meshed by mapping a premeshed ellipsoid to the model. A disadvantage of this approach is that the size and regularity of the mesh elements is not uniform; rather, it depends on the distance of the mesh elements from the pole of the prolate spheroidal coordinate system. Our proposed approach is similar to theirs in that we also fit a continuous smooth model to boundary points extracted from images and then map a premeshed sphere to the model. However, our model does not have poles or any other special points and all the mesh elements are uniform in size and regularity.

5. Conclusion

We have developed a novel method for the construction of endocardial and epicardial surface meshes from 3D segmented cardiac MR images with a prespecified number of vertices and triangles. Even when the voxels are strongly anisotropic, the resulting meshes are smooth and have close-to-regular triangles while closely following the segmentation.

Appendix

Harmonic Function with a Spherical Isolevel and Single Singularity

This section describes the solution to the Laplace equation over a spherical domain that has a single singularity somewhere within the domain and that is equal to a constant on the boundary of the domain. Let the sphere center be the coordinate system origin, R denote the radius of the sphere, \mathbf{s} the location of the singularity, and \mathbf{r} the independent variable. The solution $f_s(\mathbf{r})$ needs to satisfy the following:

$$\lim_{\mathbf{r} \rightarrow \mathbf{s}} f_s(\mathbf{r}) = \infty, \quad (\text{A.1})$$

$$f_s(\mathbf{r})|_{|\mathbf{r}|=R} = 0, \quad (\text{A.2})$$

$$\Delta f_s = 0. \quad (\text{A.3})$$

The fundamental solution of the Laplace equation in 3D is $1/|\mathbf{r}|$ and it represents a singularity at the origin. The fundamental solution centered at \mathbf{s} , that is, function $1/|\mathbf{r} - \mathbf{s}|$, satisfies (A.1) and (A.3), but it does not satisfy the boundary condition (A.2). It turns out that the sum of two shifted fundamental solutions, one centered at \mathbf{s} and one centered at $\mathbf{s}R^2/|\mathbf{s}|^2$ and multiplied by $-R/|\mathbf{s}|$, satisfies (A.1), (A.2), and (A.3). Note that the second singularity is outside the sphere, that is, there is only one singularity within the spherical domain. The solution is

$$f_s(\mathbf{r}) = \frac{1}{|\mathbf{r} - \mathbf{s}|} - \frac{R}{|\mathbf{s}|} \frac{1}{|\mathbf{r} - \mathbf{s}R^2/|\mathbf{s}|^2|}, \quad (\text{A.4})$$

or, alternatively

$$f_s(\mathbf{r}) = \frac{1}{\sqrt{|\mathbf{r}|^2 - 2\mathbf{r}\mathbf{s} + |\mathbf{s}|^2}} - \frac{R}{\sqrt{|\mathbf{s}|^2|\mathbf{r}|^2 - 2R^2\mathbf{r}\mathbf{s} + R^4}}. \quad (\text{A.5})$$

It is straightforward to show that (A.5) satisfies (A.1), (A.2), and (A.3). Also, $f_s(\mathbf{r}) > 0$ when $|\mathbf{r}| < R$, $f_s(\mathbf{r}) = 0$ when $|\mathbf{r}| = R$, and $f_s(\mathbf{r}) < 0$ when $|\mathbf{r}| > R$. In (A.2) it is assumed that f_s is zero at the domain boundary. The zero can be replaced by a constant C by simply adding C to f_s . The gradient of f_s is

$$\begin{aligned} \nabla f_s &= \frac{\partial f_s}{\partial \mathbf{r}} \\ &= \frac{\mathbf{s} - \mathbf{r}}{\left(|\mathbf{r}|^2 - 2\mathbf{r}\mathbf{s} + |\mathbf{s}|^2\right)^{3/2}} + \frac{R\left(|\mathbf{s}|^2\mathbf{r} - R^2\mathbf{s}\right)}{\left(|\mathbf{s}|^2|\mathbf{r}|^2 - 2R^2\mathbf{r}\mathbf{s} + R^4\right)^{3/2}}. \end{aligned} \quad (\text{A.6})$$

Expression (A.6) is used in the evaluation of ∇u in (7).

Acknowledgments

The authors would like to thank Dr. John Oshinski from the Department of Radiology, Emory University, USA, for providing the images used in this work. This research was supported by the American Heart Association under Grant 0855345E.

References

- [1] M. Sermesant, K. Rhode, and G. I. Sanchez-Ortiz, "Simulation of cardiac pathologies using an electromechanical biventricular model and XMR interventional imaging," *Medical Image Analysis*, vol. 9, no. 5, pp. 467–480, 2005.
- [2] M. Wierzbicki, M. Drangova, G. Guiraudon, and T. Peters, "Validation of dynamic heart models obtained using nonlinear registration for virtual reality training, planning, and guidance of minimally invasive cardiac surgeries," *Medical Image Analysis*, vol. 8, no. 3, pp. 387–401, 2004.
- [3] H. C. van Assen, M. G. Danilouchkine, A. F. Frangi, et al., "SPASM: a 3D-ASM for segmentation of sparse and arbitrarily oriented cardiac MRI data," *Medical Image Analysis*, vol. 10, no. 2, pp. 286–303, 2006.
- [4] M. R. Kaus, J. von Berg, J. Weese, W. Niessen, and V. Pekar, "Automated segmentation of the left ventricle in cardiac MRI," *Medical Image Analysis*, vol. 8, no. 3, pp. 245–254, 2004.
- [5] E. Bardinet, L. D. Cohen, and N. Ayache, "Tracking and motion analysis of the left ventricle with deformable superquadrics," *Medical Image Analysis*, vol. 1, no. 2, pp. 129–149, 1996.
- [6] X. Papademetris, A. J. Sinusas, D. P. Dione, R. T. Constable, and J. S. Duncan, "Estimation of 3-D left ventricular deformation from medical images using biomechanical models," *IEEE Transactions on Medical Imaging*, vol. 21, no. 7, pp. 786–800, 2002.
- [7] C. Lorenz and J. von Berg, "A comprehensive shape model of the heart," *Medical Image Analysis*, vol. 10, no. 4, pp. 657–670, 2006.
- [8] J. Lötjönen, S. Kivistö, J. Koikkalainen, D. Smutek, and K. Lauerma, "Statistical shape model of atria, ventricles and epicardium from short- and long-axis MR images," *Medical Image Analysis*, vol. 8, no. 3, pp. 371–386, 2004.
- [9] M. Sermesant, C. Forest, X. Pennec, H. Delingette, and N. Ayache, "Deformable biomechanical models: application to 4D cardiac image analysis," *Medical Image Analysis*, vol. 7, no. 4, pp. 475–488, 2003.

- [10] P. Shi, A. J. Sinusas, R. T. Constable, and J. S. Duncan, "Volumetric deformation analysis using mechanics-based data fusion: applications in cardiac motion recovery," *International Journal of Computer Vision*, vol. 35, no. 1, pp. 87–107, 1999.
- [11] J. F. Thompson, B. K. Soni, and N. P. Weatherill, *Handbook of Grid Generation*, CRC Press, Boca Raton, Fla, USA, 1998.
- [12] V. D. Liseikin, *Grid Generation Methods*, Springer, Berlin, Germany, 1st edition, 1999.
- [13] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.
- [14] G. Taubin, "Curve and surface smoothing without shrinkage," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '95)*, pp. 852–857, Boston, Mass, USA, June 1995.
- [15] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," *Computer Graphics*, vol. 26, no. 2, pp. 65–70, 1992.
- [16] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," *Computer Graphics*, vol. 27, pp. 19–25, 1993.
- [17] E. J. Bekkers and C. A. Taylor, "Multiscale vascular surface model generation from medical imaging data using hierarchical features," *IEEE Transactions on Medical Imaging*, vol. 27, no. 3, pp. 331–341, 2008.
- [18] J. Peiró, L. Formaggia, M. Gazzola, A. Radaelli, and V. Rigamonti, "Shape reconstruction from medical images and quality mesh generation via implicit surfaces," *International Journal for Numerical Methods in Fluids*, vol. 53, no. 8, pp. 1339–1360, 2007.
- [19] J.-O. Lachaud and A. Montanvert, "Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction," *Medical Image Analysis*, vol. 3, no. 2, pp. 187–207, 1999.
- [20] J. Lötjönen, P.-J. Reissman, I. E. Magnin, J. Nenonen, and T. Katila, "A triangulation method of an arbitrary point set for biomagnetic problems," *IEEE Transactions on Magnetics*, vol. 34, no. 4, pp. 2228–2233, 1998.
- [21] S. F. F. Gibson, "Constrained elastic surface nets: generating smooth surfaces from binary segmented data," in *Proceedings of the Medical Image Computing and Computer Aided Intervention (MICCAI '98)*, pp. 888–898, Cambridge, Mass, USA, October 1998.
- [22] A. B. Ekoule, F. C. Peyrin, and C. L. Odet, "Triangulation algorithm from arbitrary shaped multiple planar contours," *ACM Transactions on Graphics*, vol. 10, no. 2, pp. 182–199, 1991.
- [23] D. Meyers, S. Skinner, and K. Sloan, "Surfaces from contours," *ACM Transactions on Graphics*, vol. 11, no. 3, pp. 228–258, 1992.
- [24] M. Ziolkowski, "Methods of mesh generation for biomagnetic problems," *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 1345–1348, 1996.
- [25] J.-Y. Lai, J.-L. Doong, and C.-Y. Yao, "Three-dimensional CAD model reconstruction from image data of computer tomography," *International Journal of Imaging Systems and Technology*, vol. 10, no. 4, pp. 328–338, 1999.
- [26] A. A. Young and L. Axel, "Three-dimensional motion and deformation of the heart wall: estimation with spatial modulation of magnetization—a model-based approach," *Radiology*, vol. 185, no. 1, pp. 241–247, 1992.
- [27] G. Luo and P. A. Heng, "LV shape and motion: B-spline-based deformable model and sequential motion decomposition," *IEEE Transactions on Information Technology in Biomedicine*, vol. 9, no. 3, pp. 430–446, 2005.
- [28] K. D. Costa, P. J. Hunter, J. S. Wayne, L. K. Waldman, J. M. Guccione, and A. D. McCulloch, "A three-dimensional finite element method for large elastic deformations of ventricular myocardium: II—prolate spheroidal coordinates," *Journal of Biomechanical Engineering*, vol. 118, no. 4, pp. 464–472, 1996.
- [29] J. Li and T. S. Denney Jr., "Left ventricular motion reconstruction with a prolate spheroidal B-spline model," *Physics in Medicine and Biology*, vol. 51, no. 3, pp. 517–537, 2006.
- [30] I. N. Bronshtein and K. A. Semendyayev, *Handbook of Mathematics*, Springer, Berlin, Germany, 3rd edition, 1998.
- [31] E. B. Saff and A. B. J. Kuijlaars, "Distributing many points on a sphere," *Mathematical Intelligencer*, vol. 19, no. 1, pp. 5–11, 1997.
- [32] A. Katanforoush and M. Shahshahani, "Distributing points on the sphere, i," *Experimental Mathematics*, vol. 12, no. 2, pp. 199–209, 2003.
- [33] T. Erber and G. M. Hockney, "Equilibrium configurations of N equal charges on a sphere," *Journal of Physics A*, vol. 24, no. 23, pp. L1369–L1377, 1991.
- [34] L. Glasser and A. G. Every, "Energies and spacings of point charges on a sphere," *Journal of Physics A*, vol. 25, no. 9, pp. 2473–2482, 1992.
- [35] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, Cambridge, UK, 2nd edition, 2001.
- [36] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C—The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 2nd edition, 1992.
- [37] G. Wahba, "Spline interpolation and smoothing on the sphere," *SIAM Journal on Scientific and Statistical Computing*, vol. 2, pp. 1–15, 1981.
- [38] G. Turk and J. F. O'Brien, "Modelling with implicit surfaces that interpolate," *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 855–873, 2002.
- [39] E. Hartmann, "A marching method for the triangulation of surfaces," *The Visual Computer*, vol. 14, no. 3, pp. 95–108, 1998.
- [40] T. Karkanis and A. J. Stewart, "Curvature-dependent triangulation of implicit surfaces," *IEEE Computer Graphics and Applications*, vol. 21, no. 2, pp. 60–69, 2001.
- [41] S. Akkouche and E. Galin, "Adaptive implicit surface polygonization using marching triangles," *Computer Graphics Forum*, vol. 20, no. 2, pp. 67–80, 2001.
- [42] A. Duci, A. Yezzi, S. Soatto, and K. Rocha, "Harmonic embeddings for linear shape analysis," *Journal of Mathematical Imaging and Vision*, vol. 25, no. 3, pp. 341–352, 2006.
- [43] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis, "On the laplace-beltrami operator and brain surface flattening," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 700–711, 1999.