# Image-Based Variational Meshing

Orcun Goksel, *Member, IEEE,* and Septimiu E. Salcudean, *Fellow, IEEE*

***Abstract*—In medical simulations involving tissue deformation, the finite element method (FEM) is a widely used technique, where the size, shape, and placement of the elements in a model are important factors that affect the interpolation and numerical errors of a solution. Conventional model generation schemes for FEM consist of a segmentation step delineating the anatomy followed by a meshing step generating elements conforming to this segmentation. In this paper, a single-step model generation technique is proposed based on optimization. Starting from an initial mesh covering the domain of interest, the mesh nodes are adjusted to minimize an objective function which penalizes intra-element intensity variations and poor element geometry for the entire mesh. Trade-offs between mesh geometry quality and intra-element variance are achieved by adjusting the relative weights of the geometric and intensity variation components of the cost function. This meshing approach enables a more accurate rendering of shapes with fewer elements and provides more accurate models for deformation simulation, especially when the image intensities represent a mechanical feature of the tissue such as the elastic modulus. The use of the proposed mesh optimization is demonstrated in 2D and 3D on synthetic phantoms, MR images of the brain, and CT images of the kidney. A comparison with previous meshing techniques that do not account for image intensity is also provided demonstrating the benefits of our approach.**

***Index Terms*—Mesh generation, mesh optimization, biomedical image segmentation, deformable patient models.**

## I. Introduction

THE finite element method (FEM) is a common technique in medical simulations. Its speed and accuracy depend on the number of nodes/elements used and their shape and placement in the domain. In this paper, a variational modeling approach is presented to produce high-quality FEM meshes automatically for given tissue domains in both 2D and 3D. The method aligns FEM elements to group similar image intensities, as a way of clustering (segmenting) the domain, while maintaining *good* element quality for FEM. The use of such a method becomes particularly important when the input image represents a mechanical feature distribution of the tissue such as elastic modulus. This is because the elasticity parameters of each element is represented with a single value in FEM and the proposed method minimizes an objective function defined by the error between this single-value discretization and the measured modulus distribution within each element. Nevertheless, as presented in this paper, the method is applicable to the meshing of most medical imaging modalities without the need for an intermediate segmentation.

Manuscript received May 20, 2010. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

O. Goksel and S.E. Salcudean are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada {orcung,tims}@ece.ubc.ca.



In the conventional modeling methods for the FEM simulation of tissue deformation, a discrete representation of the anatomy of interest is obtained from an intensity image by employing two steps, *segmentation* and *meshing*. Segmentation, which consists of *recognition* and *delineation* of anatomy, has been studied in several medical contexts using numerous different approaches [1]. Although automatic segmentation techniques do exist, recognition is not a computationally well-defined problem and thus is usually achieved with manual intervention, leading to semi-automatic implementations. In contrast, delineation, which in many cases can be stated roughly as *grouping similar pixels*, allows for algorithmic approaches. Segmentation overall often requires *a priori* information about both the anatomy and the medical imaging modality, therefore does not have a one-size-fits-all solution.

The result of segmentation is a representation of the organ boundary, which is often in an explicit form such as a surface mesh, although implicit representations are also viable. This anatomical boundary is then supplied to a meshing scheme to tile the space with elements. The final mesh is then used for simulating tissue deformation for procedures such as laparoscopic surgery [2], brain surgery [3], breast biopsy [4], and brachytherapy [5]. In most scenarios, the time to complete a simulation is limited. For instance, simulation rates of hundreds of Hertz are required for haptic displays. This imposes a stringent node budget on the deformable model.

It is possible to generate anatomical models using structured meshes such as converting image voxels into hexahedral elements [4]. However, unstructured meshing is a common choice since the same or better surface/volume representations can be achieved using fewer elements [6], [7]. Popular unstructured meshing techniques include octree-based methods, Delaunay approaches, and advancing fronts, most of which were originally developed and tuned for the modeling of mechanical structures. The meshing efforts for medical applications have often focused on generating meshes inside given segmentation contours [8] or 3D surface triangulations [3], [9]–[11]. Common approaches first tile the bounding box of the anatomy with elements, and then cull the elements outside the anatomy, followed by a final stage of clipping [3], [8], [9] or compressing [10], [11] the elements that cut the anatomical boundaries. Graded meshes with smaller elements closer to the surfaces were employed in applications where the surface mesh resolution is the primary concern [9], [11], [12]. Typical implementations of meshing techniques from different conceptual categories are compared by Fedorov *et al.* [13] in the context of deformable registration of brain MR images.

Image segmentation is a special case of the general data *clustering* techniques [14], where not only similarity of intensities but also spatial distribution such as connectivity of voxels (so that they form a single object) is often desired. Such additional constraints are enforced by the inherent functioning of the particular segmentation method such as watershed and active contours [15] and define the overall performance of that method. Indeed, segmentation as part of a meshing process can be seen as an even more-specific clustering problem, where the clustered regions must have concrete geometrical shapes such as tetrahedra. Gevers [16] introduces a 2D method for segmenting photographic images using triangle splitting and merging based on intensity distributions in elements. A similar technique is presented in [17] for tetrahedra. However, these methods aiming computer graphics applications do not generate meshes suitable for FEM. A recent method by Reid *et al.* [18] also considers shape quality measures for segmenting micro-structures in 2D medical images. Such methods involve several local mesh operations (e.g. annealing, smoothing, fixing in [18]) and their extension to 3D is not trivial. In contrast, our proposed penalty-based mesh evolution scheme to minimal image partitioning problem has multiple advantages.

Unstructured anatomical meshing in several medical simulation applications have been carried out using algorithms that were originally developed for mechanical structures, such as TetGen [19] and GHS3D [20]. These methods require the anatomy to be segmented as a piece-wise linear complex, such as a surface mesh. More recent techniques that target meshing of medical images use a voxel-based segmentation of the domain, often called a *labeled image*, where each voxel is marked to be inside or outside of a given anatomical structure [8], [21]–[23]. In contrast, the variational image meshing (VIM) method introduced in this paper uses original medical images without requiring any preceding segmentation.

Most segmentation methods require some sort of manual intervention, not only demanding the scarcely available time of health professionals but also preventing an automated modeling for FEM. Furthermore, there are certain drawbacks with modeling approaches consisting of two separate steps of segmentation and meshing, since these two processes run successively with different goals. In such a two-step approach, meshing strictly follows the surfaces given by segmentation. But the segmentation may be unreliable, or may simply not delineate every visible structure in the image. For instance, in the case in which anatomy regions are not distinguishable in the original image, an arbitrary surface is still generated by the segmentation algorithm. This constrains the mesh to an arbitrary boundary, which also may not be FEM-friendly, especially if the meshing method leaves the initial surface triangulation unchanged, e.g. [19]. Also, structures that have not been segmented because they are not the focus of a particular segmentation algorithm, still affect the overall tissue deformation. Including such structures in the model at little or no cost using an image-aware meshing approach is clearly beneficial. As suggested in this paper, if the segmentation is postponed by integrating its objective into the meshing process, then element/node placement choices during model generation can be made by considering the goals of both processes, allowing a trade-off between them.

As a motivating example, consider the problem of simulating needle insertions into the prostate using deformable models [5], possibly produced using elastography [24]. For a fast simulation, only a limited number of nodes can be accommodated, and for accurate FEM, certain mesh geometric quality has to be met. Furthermore, a model of the prostate region should follow anatomical boundaries, where visible. Although the prostate is not distinguishable in ultrasound images from the peri-prostatic tissue at the base and apex, which complicates its segmentation, a finite element model based on a naturally registered elastography image can simulate the prostate region deformation even without necessarily knowing where the prostate boundary is. To the best of our knowledge, no previously published meshing approach addresses this issue, for which our proposed method VIM offers a solution.

## II. Previous Work on Optimal Tessellations

During finite element modeling of deformation, the two main sources of error are *interpolation errors* of the approximation to the function and its gradient (which is strain for deformation), and *numerical errors* during the solution of the approximation. Numerical errors depend mainly on the conditioning of the finite element matrices involved and also on the numerical precision and type/order of mathematical operations applied in obtaining the solution. Interpolation errors are affected by various characteristics of the mesh and the approximant function, such as the polynomial order of the function approximation. Among these error contributors are the shape and size of mesh elements [25]. Consequently, it has been an active field of research to obtain meshes that will introduce minimal amount of such errors in FEM simulations.

Given a set of vertices, their Delaunay tessellation offers several favorable features compared to other possible tessellations. In particular, when a function $f$ has bounded second derivatives, Delaunay tessellation minimizes the worst interpolation error between $f$ and its piecewise linear approximant defined by this tessellation (mesh). Delaunay tessellation received significant attention in the mesh generation literature. The major problem with Delaunay refinement approaches to mesh generation in 3D has been the existence of *slivers* in the final meshes. Slivers are a class of degenerate tetrahedra that are not necessarily removed by the Delaunay refinement. The degeneracy of such tetrahedra is not correctly captured by many commonly-used mesh quality measures such as the edge-radius ratio, which is the ratio between the shortest edge length and the circumsphere radius. Edge ratios, area-volume ratios, and minimum/maximum dihedral angles are among various other quality measures suggested in the literature. Reviews of these measures with their relation to function approximation and FEM were given by Shewchuk [25] and Field [26].

To achieve better-shaped elements, modifications to Delaunay refinement and other local optimization schemes were proposed in the literature [27]–[29]. However, these methods require significant implementation effort and result in non-convex functionals that are difficult to analyze and derive theoretical guarantees for. More recent approaches to mesh smoothing such as the Centroidal Voronoi-Delaunay Tessellation (CVDT) [30] and Optimal Delaunay Triangulations (ODT) [31], [32] consider minimizing a quadratic energy through updates of vertex positions and their connectivity. Advantages of these methods are (i) the simplicity of each update and (ii) the implication of eventual convergence due to the monotonically-decreasing energy definitions. Building on these and other previous work, Alliez *et al.* introduces a 3D mesh generation technique producing superior meshes compared to similar methods in the literature [33]. This method is adapted in this work.

For anisotropic meshing, Chen [32] notes that ODT equidistributes the edge lengths of a mesh under a metric related to the Hessian of the approximated function $f(\mathbf{x})$. Alliez *et al.* adjust the edge length distribution in space using a mesh density function, e.g. to have finer elements modeling higher curvature boundary surfaces of a 3D object [33]. However, it is not clear from these and other work, how the information about a feature distribution through the tissue (e.g. distribution of elastic modulus) can be incorporated into such variational meshing techniques, assuming this distribution is known in the continuum *a priori*. Our proposed technique VIM addresses this aspect of meshing.

The methods for image-compliant meshing are introduced in Section III and a numerical approach for their use is presented in Section IV. The results from various imaging modalities and comparisons with other selected meshing techniques are demonstrated in Section V, which is followed by discussion and conclusions in sections VI and VII, respectively.

## III. Methods

The method involves combining an element geometric quality metric with an image-based metric into an objective function to be minimized. First, the use of the former metric alone is described. Next, an optimal parameter discretization of the elements of a tessellation is presented. Based on this, an image-based measure is defined for an optimal image partitioning and, finally, its combination with the geometric definition above is introduced.

### *A. Geometric Energy for Mesh Optimization*

Let $g_{\mathcal{T}}(\mathbf{x})$ be a piece-wise linear approximation of a function $f(\mathbf{x})$ on a given tessellation $\mathcal{T}$. It is known that the integral approximation error $E_G = \|g_{\mathcal{T}} - f\|_{\mathcal{L}^1}$ is minimized by the Delaunay tessellation of the domain $\mathcal{M} \subset \mathbb{R}^N$ when $(i)$ the function $f$ is a paraboloid, i.e. $f(\mathbf{x}) = \|\mathbf{x}\|^2$, and $(ii)$ $g_{\mathcal{T}}$ is its *overlaid* circumscribing piece-wise linear approximant [31]. This was used in iterative optimization schemes in the literature to obtain geometrically high-quality meshes [32], [33] and is also adopted as the *geometrical* mesh quality measure in our method in the form given below:

$$E_G = \sum_{\tau \in \mathcal{T}} \int_{\tau} g_{\tau}(\mathbf{x})\, d\mathbf{x} - \int_{\mathcal{M}} f(\mathbf{x})\, d\mathbf{x} \quad (1)$$

$$= \frac{1}{N+1} \sum_i f(\mathbf{x}_i) |\Omega_i| - \int_{\mathcal{M}} f(\mathbf{x})\, d\mathbf{x} \quad (2)$$

$$= \frac{1}{N+1} \sum_i \mathbf{x}_i^2 |\Omega_i| - \int_{\mathcal{M}} \mathbf{x}^2\, d\mathbf{x} \quad (3)$$

where $\Omega_i$ is the *1-ring neighbourhood* of node $\mathbf{x}_i$, that is the set of elements having $\mathbf{x}_i$ as a corner as in Fig. 1(a). This neighbourhood will be referred as *1-ring* throughout this paper.

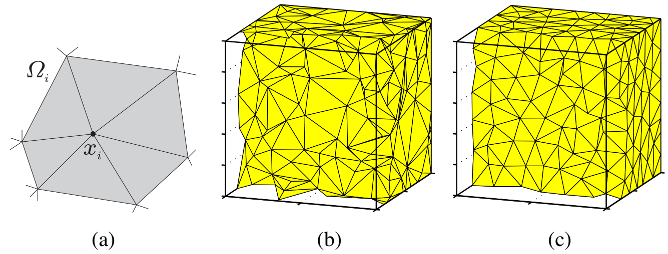



Fig. 1. (a) *1-ring neighbourhood* of a node, (b) a random 3D mesh initialization, and (c) its *geometrically* optimized configuration.

An example of optimizing this geometric energy $E_G$ alone is shown in Fig. 1, where the 3D mesh was initialized using a random distribution of vertices as in Fig. 1(b). This optimization involves *Lloyd's relaxation*, which is performed as alternate updates: a global Delaunay tessellation of given node locations $\mathbf{x}_i$ and node relocations so that the above-given cost function is minimized. Figure 1(c) shows the optimized meshes after having converged.

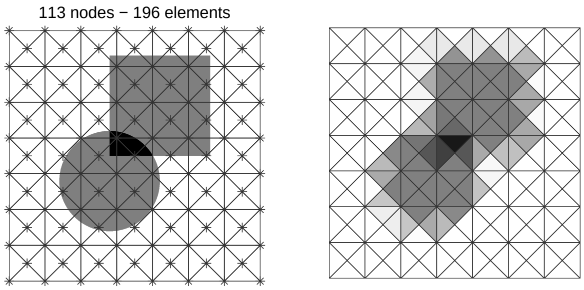
### *B. Element Discretization*

In this section, a penalty-function for image discretization is defined based on a mesh element and the image pixels/voxels covered by that element. Since an element uses a discretized value to represent any spatial location within itself, an $\mathcal{L}_2$-norm difference of the discretized element value and the voxels represented by that value is used as the discretization error metric. This definition is in a way similar to the common Mumford-Shah functional used in various image segmentation methods to partition pixels based on their similarity [34]. However, in contrast to such methods, where often there is a term constraining the length/curvature of segmenting curve for regularization purpose, the planar faces of mesh elements in our method already constrains the way that image pixels can be partitioned in space.

Let $h(\mathbf{x})$ denote the distribution of a feature in $\mathcal{M}$, the domain of interest. In the FEM, such a feature is discretized on a mesh so that each mesh element $\tau_j$ has a single assigned value, $\tilde{h}_j$, modeling this feature within that element. In this paper, the discretization error associated with this single value approximation $\tilde{h}_j$ and the values it represents, $\{h(\mathbf{x}), \mathbf{x} \in \tau_j\}$, is defined as an $\mathcal{L}_2$-norm:

$$E_D^j = \int_{\tau_j} \left| h(\mathbf{x}) - \tilde{h}_j \right|^2 d\mathbf{x}\ . \quad (4)$$

It is evident that for a constant $h$ within an element, the best $\tilde{h}_j$ is that constant value itself. In general, this error is minimized for $\tilde{h}_j$ being the average mean value of the given distribution. Therefore, for a given element and given (background) feature distribution, a discretized feature value $\tilde{h}_j$ is assigned to that element $\tau^j$ as the average of $\{h(\mathbf{x}), \mathbf{x} \in \tau_j\}$. This average mean discretization is further discussed in the context of elastic strain energy formulation for FEM in Section VI-C.




(a) (b)

Fig. 2. An initial synthetic phantom (a) with its discretization (b).

### C. Objective Function for Image Compliance

Assigning a single-value equivalent of a known feature distribution (an image) within an element as the average mean value is demonstrated in Fig. 2 for a structured mesh overlaid on a synthetic phantom. The element shading in Fig. 2(b) denotes the average intensity of the underlying image pixels.

The element discretization error defined in (4) is integrated over the mesh to yield the following cost function describing the fitness of a mesh to an image:

$$E_D = \sum_j \int_{\tau_j} \left| h(\mathbf{x}) - \tilde{h}_j \right|^2 d\mathbf{x} \; . \tag{5}$$

Note that with the earlier assumption of $\tilde{h}_j$ being the average mean value in the element, (5) can be rewritten as:

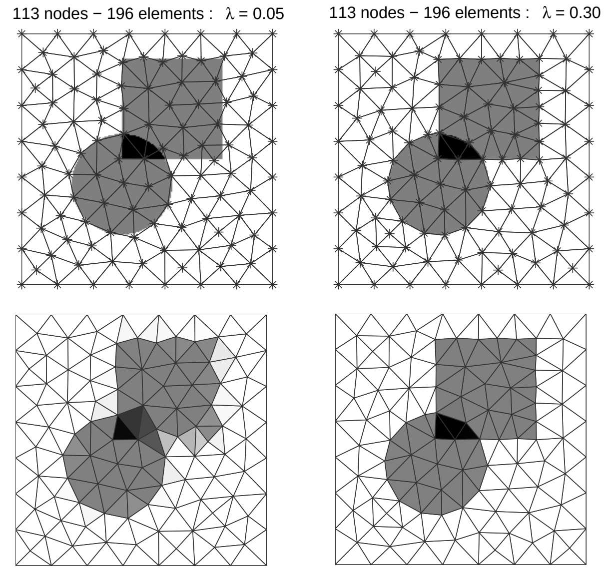
$$E_D = \sum_j v^j \, \mathbf{var} \left( h(\mathbf{x}) : \mathbf{x} \in \tau_j \right) \tag{6}$$

where $v^j$ is the volume of element $j$ and $\mathbf{var}$ is the second-moment of a distribution around its mean, namely the *variance*. Thus, this definition penalizes elements with larger image intensity variations. *However, it does not enforce suitability of element size and shape for the FEM.* As a result, in order to derive a variational scheme trading off between element geometry and image representation, the two error metrics $E_G$ and $E_D$ above are combined as follows:

$$E = (1 - \lambda) E_G + \lambda E_D \tag{7}$$

where $\lambda \in [0, 1)$ is the weighting factor of discretization.

A visualization of this combined error is presented in Fig. 3 for a simple 2D mesh, overlaid on an image with a diamond-shaped feature as in Fig. 3(a). For the purpose of this illustration, the triangulation and the four corner node position are kept fixed. As the position of the center node is changed, (7) is calculated for this mesh using (3) and (6). The combined cost $E$ is seen in Figs. 3(b)–(e) as a function of the center node position for different weighting factors $\lambda$. Note that with the two competing measures in this example: ($i$) the best triangle aspect ratios are achieved by minimizing $E_G$ when the mid-node is placed at the center, and ($ii$) the variance of some triangles reduce to zero minimizing $E_D$ when the mid-node is placed at the corners of the diamond shape. Consequently, for small $\lambda$ values, $E$ is minimized at the center disregarding the underlying image, whereas the shape corners are captured better by the node as $\lambda$ increases. The preliminary results of this variational approach were presented in [35].




(a) (b)

Fig. 4. The initial phantom mesh in Fig. 2 optimized using (a) $\lambda = 0.05$ and (b) $\lambda = 0.30$ are shown as meshes overlaid on the image (top) and the corresponding image approximations (bottom).

## IV. Implementation

The objective $E$ is a function of node location $x$ and mesh connectivity (tessellation) $\mathcal{T}$. In this paper, this cost function is minimized in a numerical optimization scheme, where node locations and element connectivity are updated alternately minimizing the cost based on Lloyd's relaxation. Figure 4 demonstrates two example mesh models generated for the 2D synthetic phantom in Fig. 2 after optimization using VIM with two different weighting constants $\lambda$.

### A. Mesh Initialization

Our optimization procedure is initialized by a geometrically optimal mesh, which can be found by minimizing $E_G$ alone starting from, for instance, a random distribution of nodes. Alternatively, a regular mesh structure, which is naturally a minimizer for $E_G$, can be used for initialization. In this paper, this latter approach is followed by using regular-grid tessellations as the *initial* meshes for VIM.

### B. Node and Connectivity Updates

Formally, the optimal location of a node $\mathbf{x}_i$ minimizing $E$ can be found using a constrained optimization approach, where the 1-ring polygonal/polyhedral region $\Omega_i$ seen in Fig. 1(a) is the feasible domain. The perimeter of $\Omega_i$ (the outer edges/faces of neighbouring triangles/tetrahedra) can thus be defined as a set of inequality constraints $\mathbf{A}_i \mathbf{x}_i < \mathbf{b}_i$ where an optimal node location is sought as:

$$\begin{aligned} \mathbf{x}_i' &= \underset{\mathbf{x}_i \in \Omega_i}{\arg\min} \; E(\mathbf{x}_i) \\ &= \underset{\mathbf{x}_i}{\arg\min} \; E(\mathbf{x}_i) \quad s.t. \quad \mathbf{A}_i \mathbf{x}_i < \mathbf{b}_i \; . \end{aligned} \tag{8}$$

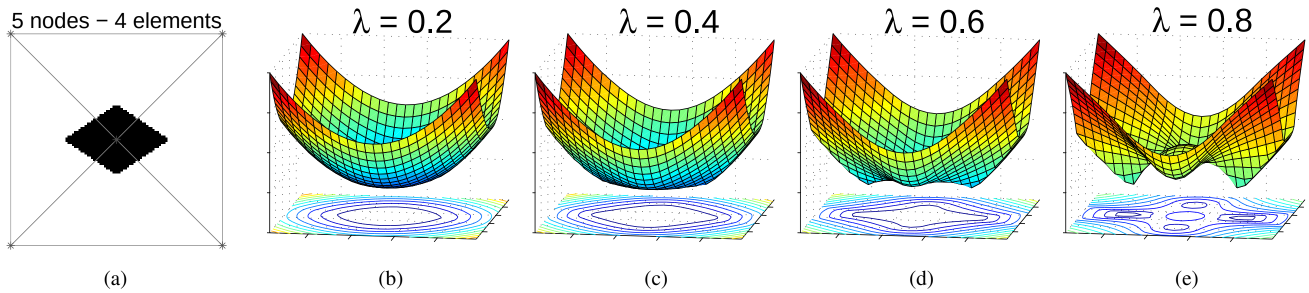



Fig. 3. A simple four-element mesh (a) and the combined cost $E$ as a function of mid-node position for $\lambda$ of (b) 20%, (c) 40%, (d) 60%, and (e) 80%.

Due to the lack of a closed-form definition for either the critical-point or the gradients of $E$, a numerical optimization of (8) above within each 1-ring is a costly operation. Furthermore, considering the outer loop of the Lloyd's relaxation which also updates the positions of all other nodes including the 1-ring neighbours of $\mathbf{x}_i$, finding an exact optimum location turns out to be less effective than expected. Consequently, we follow an approach similar to [36], where an optimal location is sought in the directions toward 1-ring centroids. Starting from each centroid, the search step length is reduced using the *golden search method* until it becomes small with respect to the image resolution. Accordingly, $\mathbf{x}_i$ is moved to the position that yields the smallest cost $E$.

For updating the connectivity, the edge/face swapping strategy from GRUMMP project [37] is implemented. This method considers the alternative local mesh changes where edges/faces are replaced and reconnected in different configurations for improving a given metric [27]. For our purposes, each edge/face is swapped if and only if the alternative swapped connectivity will reduce our objective function.

During both the node and the connectivity updates, in order to eliminate configurations that cause inverted or very flat elements (such as slivers in 3D), an additional quality threshold is enforced on elements. Any configuration that results in elements worse than this threshold limit is rejected. In the examples of this paper, a maximum dihedral angle bound of 150° is used for this purpose. This generates sliver-free high-quality meshes. Minimum dihedral angles, radius ratios, or other element quality measures can also be bounded.

### C. Normalization of Cost Weighting Factor

Our design parameter $\lambda$ sets the trade-off between geometry and image compliance. However, its actual effect on the result depends on many factors such as the domain size, the number of elements, and the intensity variation of the given image. Depending on such parameters, $E_G$ and $E_D$ may have very different scales separated by orders of magnitude. Consequently, in order to have control over the range of practical $\lambda$ values, an additional step of $E_G/E_D$ normalization is employed. In the first iteration where $E_G$ is optimum, a scaling factor between the values of these two error definitions is calculated so that the given $\lambda$ will indeed be interpreted as the percentage error contribution of discretization rather than an absolute quantity. This scaling factor is then fixed and used for the rest of the iterations. This treatment normalizes $\lambda$ allowing us to report consistent *effective* values (percentages), which are easier to associate. For instance, a $\lambda$ of 30% means that the optimization process targets minimizing the cost combined as 30% discretization and 70% geometry components with respect to their initial error contributions.

### D. Convergence Measure

In this paper, a convergence measure for the optimization is set as the root-mean-square nodal position update at iteration $t$ normalized by the mesh bounding-box size $b$ as follows:

$$\upsilon = \frac{1}{b}\sqrt{\frac{1}{n}\sum_{i}^{n}\left(\mathbf{x}_i^{t+1} - \mathbf{x}_i^{t}\right)^2} \tag{9}$$

where $n$ is the number of mesh nodes. A mesh is considered to have converged when $\upsilon < 10^{-3}$.

## V. Results

A 2D slice from MR image data of the brain is shown in Fig. 5(a). Figures 5(b)-(e) present image discretizations by the initial and final meshes for two sample optimizations at different mesh resolutions. The converged mesh of the latter case is seen overlaid on the initial image in Fig. 5(f). Similarly, the image and the discretizations from two optimizations for a 2D CT slice of the kidney are presented in Fig. 6. Note that, once a mesh is optimized using VIM, a superior image representation is achieved even using a relatively coarse mesh. Furthermore, not only the kidney itself but also the tumor in it and the vertebra on the right are followed by the mesh, without requiring a separate segmentation for each.

VIM is next demonstrated in 3D. The optimized mesh of a synthetic phantom with a spherical inclusion is seen in Fig. 7. For the presentation of 3D results, along with cutaway views of meshes, segmentations of corresponding anatomy of interest using a simple operation of element thresholding are also presented. This thresholding method is detailed in Sec. VI-B.

Slices from a 3D MR image volume of brain ventricles are seen in Fig. 8(a). The optimized mesh is shown in figures 8(b)-(d) with cutaway and thresholded views of ventricles, where the shading of each element face indicates the discretized $\tilde{h}_j$ value within that element.

The evolution of the combined cost values $E$ during the optimization of some of the examples presented in this paper

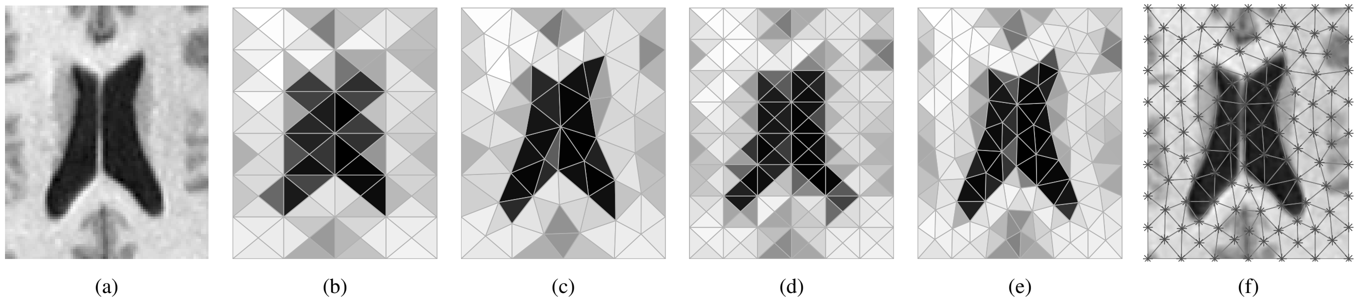



Fig. 5. Mesh optimization on a 2D MR image slice (a) of brain ventricles. Initial (b) and optimized (c) discretizations with 59 nodes; initial (d) and optimized (e) discretizations with 111 nodes. The finer optimized mesh is seen as overlaid on image (f).

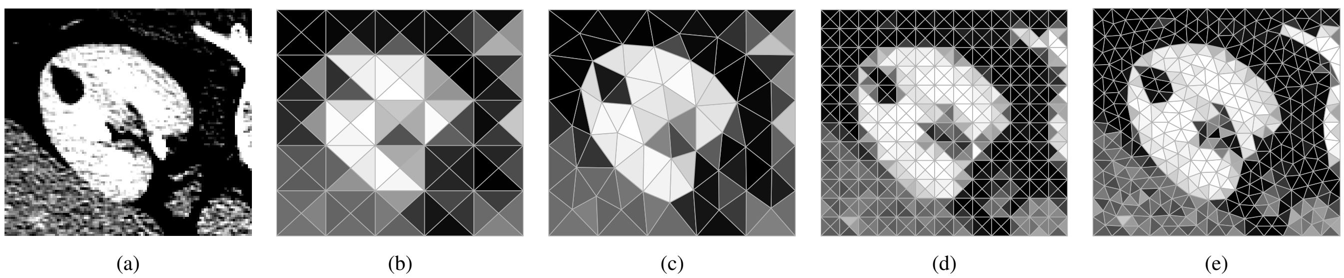



Fig. 6. Mesh optimization on a 2D CT image slice (a) of the kidney. Initial (b) and optimized (c) discretizations with 61 nodes; and initial (d) and optimized (e) discretizations with 338 nodes.

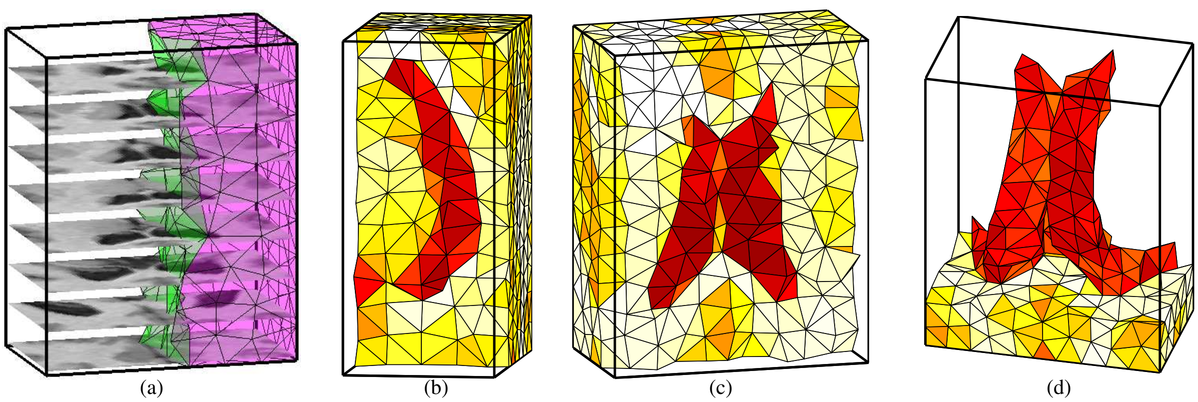


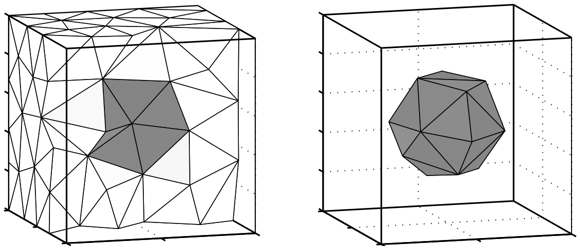

Fig. 8. Mesh optimization for 3D MR image volume of the brain. (a) shows part of a mesh with some image slices from the voxel volume. A 858 node mesh converged after 9 iterations is presented with (b)-(c) two cutaway views showing discretized element values on faces and (d) the thresholded elements showing the ventricles.

are plotted in Fig. 9. These were normalized to their initial mesh quantities in order to present their change in percentage.

In order to evaluate element quality and volume approximation performance, the meshes generated by VIM were compared with three popular meshing software: GHS3D [20], TetGen [19], and CGAL [23]. GHS3D is a tetrahedral meshing engine used in several commercial engineering packages. We used its implementation in Ansys$^{TM}$. TetGen generates constrained Delaunay tessellations, the input of which is a piece-wise linear complex (e.g., a surface mesh) delineating the boundaries of different regions to be meshed. In the Computer Graphics Algorithms (CGAL) package, the labeled voxel-volume meshing technique based on [22] is used.

A sphere of radius 1 embedded at the center of a $3{\times}3{\times}3$ cube was used as the synthetic test domain. Each software above requires the domain geometry to be input in a different format due to the different nature of their individual algorithms. For Ansys, the geometry was defined implicitly in analytical form and a tetrahedral mesh was generated. As the input to TetGen, the sphere and cube *surface* meshes extracted from the Ansys tetrahedral mesh results were used. For CGAL and VIM, a $100^3$ voxel 3D image of the cube was generated with the sphere having a different color (label).

Three different mesh sizes, 1, 0.5, and 0.2, were used for comparison as seen in Fig. 10. However, each method has a different interpretation of this desired mesh size. For instance, in Ansys, this value defines the initial subdivision length of lines (e.g., cube edges), whereas in TetGen it is the target element edge length that is used to terminate the subdivision. As a result, these methods resulted in somewhat different



(a) (b)

Fig. 7. The optimized 3D mesh of a synthetic phantom image with a spherical inclusion. In the cutaway view in (a), element shading represents the discretized values of cut elements. The inclusion extracted using element thresholding is presented in (b).

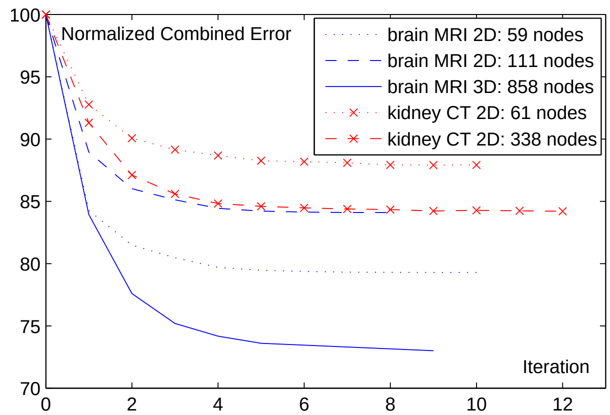
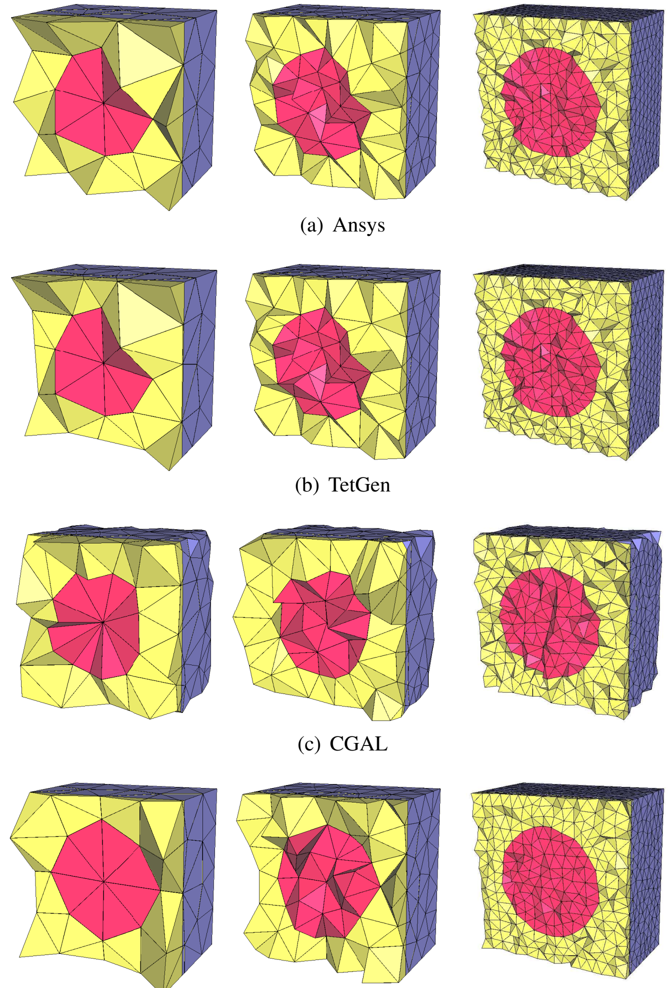



Fig. 9. Combined cost $E$ during the optimization of some examples in this paper.



(a) Ansys

(b) TetGen

(c) CGAL

(d) VIM

Fig. 10. Meshing of a sphere-embedded 3x3x3 cubic domain using Ansys, TetGen, CGAL, and our proposed VIM method. Three mesh resolutions, 1, 0.5, and 0.2, are presented in the left, center, and right columns, respectively.

number of nodes/elements. In order to initialize VIM, regular node grids of sizes comparable to the meshes generated by the three other methods were chosen.

CGAL requires a parameter to control the maximum distance of generated surfaces to input geometry boundaries, which was empirically set to its best value 0.1 . A low setting floods the mesh with many elements, whereas for higher values the geometry is not respected and/or inconsistent meshes are generated. The effect of this parameter on following the boundaries is seen on the outer surfaces of the cubes in Fig. 10(c). Other quality-related parameters of TetGen and CGAL were set to their best possible theoretical limits.

In Table I, a qualitative mesh comparison is presented. Mesh element quality is reported using the worst normalized radius-ratio (NRR), and minimum and maximum dihedral angles (DA) in meshes. NRR is the inradius-to-circumradius ratio normalized by the space dimension to a scale of $(0, 1]$. DA is the angle in degrees between the two faces of a tetrahedron ranging between $(0°, 180°)$. The worst values of these metrics define a lower bound for the accuracy of FEM using that mesh [25], where larger NRR, larger $min\{DA\}$, and smaller $max\{DA\}$ indicate better meshes for numerical simulations.

As a surface approximation measure, three other metrics are also reported, quantifying how well the given sphere geometry is reproduced by each mesh. These metrics are the numbers of voxels that are categorized erroneously for being inside/outside the sphere, namely: false negative (FN) voxels lie outside the spherical mesh despite being inside the actual (analytical) sphere, false positive (FP) voxels lie inside the spherical mesh but outside the actual sphere, and TF stands for the total number of such false-categorized voxels. TF is also given in percentage normalized to the number of voxels ($\approx$volume) within the sphere. For extracting the sphere from VIM-generated meshes, the thresholding method described in Section VI-B is used.

## VI. Discussion

### A. Comparison with Selected Methods

The first three meshing techniques in Table I, despite using different algorithms, all aim at placing mesh nodes at interfaces of given domain boundaries. Therefore, for the given convex spherical region, these techniques generate underlaid approximations with a large number of FN voxels (and with FP$\approx$0). Note that the volume of the sphere is under-approximated by these meshes. In contrast, using element variances, VIM aligns the faces of the elements to cut the spherical

TABLE I
MESH QUALITY AND SURFACE APPROXIMATION COMPARISON FOR THE MESHES IN FIGURE 10

| Method | Mesh Size | | Quality Metrics | | | Approximation Metrics | | |
|---|---|---|---|---|---|---|---|---|
| | #vertices | #elements | min{NRR} | min{DA} | max{DA} | FN | FP | TF |
| Ansys (GHS3D) | 100 | 310 | 0.47682 | 27.9440 | 134.702 | 41390 | 0 | 41390 (26.70%) |
| | 452 | 1828 | 0.43801 | 20.8189 | 141.445 | 12026 | 0 | 12026 ( 7.76%) |
| | 4636 | 23256 | 0.31221 | 16.2556 | 151.801 | 2684 | 0 | 2684 ( 1.73%) |
| TetGen | 107 | 340 | 0.20114 | 9.7820 | 163.681 | 41390 | 0 | 41390 (26.70%) |
| | 461 | 1925 | 0.14289 | 9.0751 | 164.751 | 12026 | 0 | 12026 ( 7.76%) |
| | 4746 | 24712 | 0.15244 | 7.2965 | 164.848 | 2684 | 0 | 2684 ( 1.73%) |
| CGAL | 263 | 813 | 0.19305 | 10.7146 | 162.705 | 18440 | 0 | 18440 (11.89%) |
| | 489 | 1977 | 0.14145 | 10.0799 | 165.105 | 13757 | 10 | 13767 ( 8.88%) |
| | 2910 | 14507 | 0.05755 | 2.9525 | 175.366 | 3565 | 46 | 3611 ( 2.33%) |
| VIM | 125 | 387 | 0.40437 | 19.4780 | 146.035 | 2921 | 5643 | 8564 ( 5.52%) |
| | 343 | 1334 | 0.14745 | 11.8827 | 149.986 | 1279 | 2818 | 4097 ( 2.64%) |
| | 4096 | 20790 | 0.20894 | 10.6295 | 149.986 | 520 | 684 | 1204 ( 0.78%) |

boundary optimally, therefore, leading to less than half the total false-categorized voxels at each resolution compared to the other methods.

In terms of element quality, VIM is seen to surpass TetGen and CGAL and approach the Ansys results. Recall that Ansys has access to the actual analytical representation of the domain, giving it a distinct advantage and flexibility in node placement. In fact, *in contrast to the three other methods that mesh already segmented images, only VIM is capable of meshing given intensity images with arbitrary values*. This synthetic example here was chosen such that compatible input representations exist for all four methods compared.

### B. Element Thresholding for Segmentation

At any optimization iteration, once the average mean $\tilde{h}$ of the image voxels within every element are found, the elements below/above a given threshold can be culled from the visualization. Subsequently, the remaining elements are grouped into bins (*objects of interest*), so that any element can be reached from any other element in the same object by only traversing through neighbouring element faces. Partitioning a discretized mesh into sets of such connected regions presents a way of segmenting this mesh. The largest such connected set is presented in Figs. 7 and 8 as the mesh representation of the anatomy of interest using our method. Note that extracting a surface from these volumetric meshes is not the focus of this paper and more sophisticated methods can indeed be developed for this. The thresholded mesh figures in this paper are mainly presented for visualization purposes.

Recall that our method balances two metrics in the volume, and does not focus only on the image compliance of the surface. Therefore it is not fair to compare such surfaces extracted from meshes with other surface segmentation methods such as active contours. Nevertheless, it is valuable to show that, once an image is meshed, if a (rough) surface/anatomy segmentation is also needed, the presented thresholding method can be used to extract the anatomical structures. Furthermore, instead of this simple thresholding, other sophisticated segmentation techniques from the literature can also be applied on these meshes. Since these generated meshes are (optimal) representations of images at a lower-resolution than their original voxel volumes, mesh segmentation approaches should run faster than the conventional voxel-based segmentations. Moreover, note in the images that the optimal discretization of voxels into mesh elements introduces a *smoothing* effect, while preserving the edges but removing the noise, which is beneficial for further processing.

### C. Discretization of Known Elastic Modulus Distribution

In this section, the element discretization as the average mean in Section III-B is shown to be consistent with deformation modeling using FEM. Therefore, using the proposed discretization cost function indeed results in optimal FEM meshes minimizing the error due to parameter discretization.

For a linear stress-strain relationship, the elastic strain energy of a 4-node tetrahedral element can be written in terms of the four corner displacements $\mathbf{u}^j$ as:

$$E^j_{strain}(\mathbf{u}^j) = \frac{1}{2}\int_{\tau^j} {\mathbf{u}^j}^T {B^j}^T C(\mathbf{x}) B^j \mathbf{u}^j d\mathbf{x} \qquad (10)$$

where $C(\mathbf{x})$ is the element material stiffness matrix and $B^j$ is the constant partial-derivative matrix, which is derived using the integration of barycentric coordinates within the element and hence is fixed for given tetrahedron corner positions [38].

In the conventional derivation of element strain energy, the material stiffness matrix is constant within each element, i.e. $C(\mathbf{x}) = \mathrm{C}_j$, since the material properties, Young's modulus and Poisson's ratio, are discretized as constants in each element. Then, (10) leads to:

$$\begin{aligned} E^j_{strain}(\mathbf{u}^j) &= \frac{1}{2}{\mathbf{u}^j}^T {B^j}^T \mathrm{C}_j B^j \mathbf{u}^j \int_{\tau^j} d\mathbf{x} && (11) \\ &= \frac{1}{2}{\mathbf{u}^j}^T {B^j}^T \mathrm{C}_j B^j \mathbf{u}^j v^j \ . && (12) \end{aligned}$$

C can indeed be written as a linear function of Young's modulus, i.e. $\mathrm{C}_j = \tilde{\mathcal{E}}_j\, \mathrm{C}'_j$, where $\tilde{\mathcal{E}}_j$ is the Young's modulus discretization in this element. Consequently, (12) yields:

$$E^j_{strain}(\mathbf{u}^j) = \frac{1}{2}{\mathbf{u}^j}^T {B^j}^T \mathrm{C}'_j B^j \mathbf{u}^j\, \tilde{\mathcal{E}}_j v^j \ . \qquad (13)$$

It is a common assumption to take the Poisson's ratio constant in a soft tissue domain. This is acceptable considering the nearly incompressible characteristic of soft tissues. However, the Young's modulus does often change substantially between different tissue structures. Assume that this Young's modulus distribution, $\mathcal{E}(\mathbf{x})$, within the domain $\mathcal{M}$ is known *a priori*. There exist several methods in the literature for the acquisition and derivation of tissue elasticity, see [39] for a review. Consequently, the material stiffness matrix in the element can be formulated as $C(\mathbf{x}) = \mathcal{E}(\mathbf{x})\, \mathrm{C}'_j$. Substituting this in (10) yields:

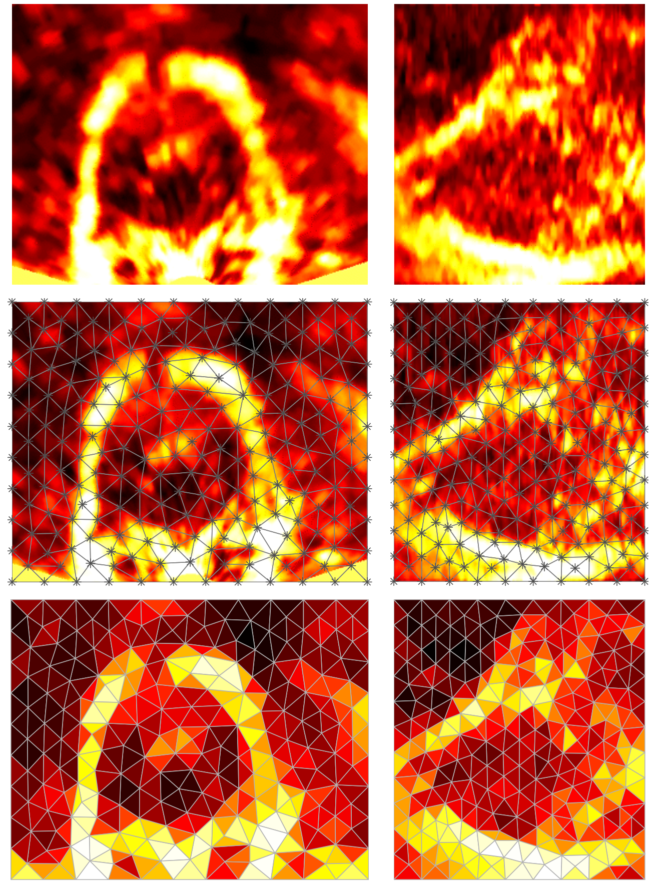
$$E^j_{strain}(\mathbf{u}^j) \quad = \quad \frac{1}{2}{\mathbf{u}^j}^T {B^j}^T \mathrm{C}'_j B^j \mathbf{u}^j \int_{\tau^j} \mathcal{E}(\mathbf{x}) d\mathbf{x}\ . \quad (14)$$

For the discretization within each element to be optimal, these two energy formulations in (13) and (14) should be equal, leading to:

$$\tilde{\mathcal{E}}_j v^j \quad = \quad \int_{\tau^j} \mathcal{E}(\mathbf{x}) d\mathbf{x} \quad (15)$$

which is satisfied when $\tilde{\mathcal{E}}_j$ is the mean of the distribution within element $\tau^j$.

To demonstrate mesh optimization from mechanical tissue features, the method was applied to prostate elastography images acquired using the *vibro-elastography* technique of Salcudean *et al.* [24]. Elastography is the technique in which tracked localized displacements in response to a mechanical excitation allow for the identification of mechanical tissue properties [39], [40]. For the purpose of this paper, a 2D sagittal transfer function image of the prostate is meshed. The prostate, which is typically stiffer that its surrounding, is seen in Fig. 11. The optimized meshes and their corresponding discretizations are also shown in this figure.



(a) (b)

Fig. 11. Meshing of a transversal (a) and a para-sagittal (b) slice from prostate vibro-elastography. The prostate is the darker oval structure in the center.

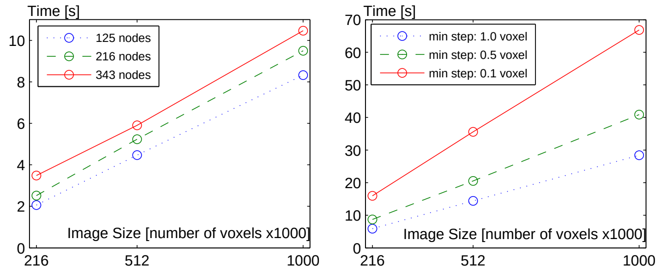
### D. Connectivity and Node Updates

For the case where the objective function is purely geometric, i.e. $\lambda = 0$, $E_G$ has a simple algebraic (quadratic) definition as in (3). This can also be observed for relatively small values of $\lambda$ as in Fig. 3(b). For $E_G$ alone, a geometrical closed-form expression of its critical-point within each 1-ring exists to be used for node updates [33]. Furthermore, a Delaunay tessellation is guaranteed to minimize this cost component. However, our objective function $E$ is also a function of the image and thus is not algebraically defined in a simple form preventing us from using such closed-form node or connectivity updates. Instead, we resort to a numerical optimization scheme for minimizing this objective function.

For the optimization of the objective function $E$, node update directions toward the centroids of the 1-ring neighbours are considered. Using multiple step-lengths in these directions provide sufficient sampling within the feasible 1-ring region. Alternatively, a random walk in 1-ring also proved to be effective for finding alternative node locations minimizing $E$. We have attempted other popular optimization methods using numerical gradients as well. However, they have not performed as well as the approaches above due to the image-dependent non-convex nature of $E$. Furthermore, a fixed number of sample points assures a predictable processing time for each node update.

The cost component $E_G$ can be rapidly computed algebraically using (3), where the second term (volume integral) has a closed-form definition for the rectangular/quadrilateral shaped image domains in this paper. The component $E_D$ is found using numerical integration over the voxels, during which the enclosure of voxels by elements is determined using the barycentric coordinates of voxels within the bounding box of this element. This process of voxel mapping to their enclosing elements is the computational bottle-neck in the current VIM implementation. This operation can be accelerated using fast grid-point location approaches such as [41], [42].

The execution time of connectivity updates scales linearly with the number of voxels in the volume and the mesh size (number of faces/edges, in particular). This is seen in Fig. 12(a), where the connectivity update time per one iteration is plotted for three different mesh sizes at $60^3$, $80^3$, and $100^3$ voxel images of the same spherical inclusion domain. Node update time is also affected by the minimum step size considered toward the neighbour centroids and is given accordingly in Fig. 12(b). In fact, increasing mesh size decreases the average distance to neighbour centroids and hence reduces the number of steps the given optimization implementation considers. Consequently, for given constant minimum step size, changing mesh size does not significantly affect the execution time of node updates. The largest VIM-produced mesh presented in this paper in Fig. 10(b)(right) has more than 20 thousand elements and took 15 minutes to generate on a 2.33 GHz processor using a C implementation having no particular optimization.




(a) (b)

Fig. 12. Given as a function of number of voxels ($60^3$, $80^3$, and $100^3$), (a) connectivity update time per iteration for meshes with 125, 216, and 343 nodes; and (b) node update time per iteration for minimum optimization step-sizes of 1, 0.5, and 0.1 voxels.

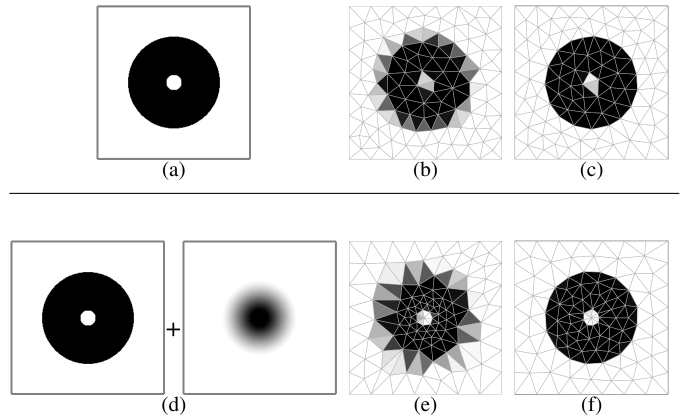


Fig. 13. Demonstration of using a sizing field for variable element sizes throughout the domain: (a) synthetic image, (b) initial and (c) optimized meshes with uniform element sizing; (d) the image and the element sizing field to be imposed; (e) initial and (f) optimized meshes with the application of this sizing field.

In the 3D implementation, the corner nodes of the given rectangular prism shaped domain are kept fixed during the optimization process, while the nodes on faces and edges are allowed to move tangentially. The position of the internal nodes are optimized as described. These constraints can be relaxed and non-prism domains can be accommodated using (outer) domain-boundary complying techniques as in [33]. Note that, if desired, nodes can also be added to the mesh during optimization, such as near elements contributing a higher cost $E$.

### E. Mesh Sizing

A desired local mesh-size was achieved in [33] by defining a weighted volume from a background density distribution. This was used to generate smaller elements to model higher curvature surfaces. A similar approach can be used to effectively refine FEM meshes near smaller anatomical features or higher strain regions. For instance, consider the synthetic image in Fig. 13(a) representing two circular *anatomical features* of different sizes. A standard VIM optimization starts from an initial uniform mesh distribution such as in Fig. 13(b), which subsequently converges to Fig. 13(c). Note that the small feature in the center cannot be represented well at this mesh size. Alternatively, a mesh sizing field (*density function*) shown in Fig. 13(d) can be imposed leading to an increased element resolution near the center as demonstrated in Fig. 13(e).

Our geometric cost $E_G$ can be modified using a sizing field in order to obtain denser meshes near desired features. For the purpose of this example, an alternative approach was followed by using the sizing field only in the initialization process with $E_G$ alone. Subsequently, our regular optimization scheme VIM was initialized with this nonuniform mesh seen in Fig. 13(e). This converges to the local minimum configuration in Fig. 13(f), where the small feature is seen to be successfully modeled by this mesh. Note that although the overall number of nodes/elements are the same in the uniform and nonuniform meshes presented in Figs. 13(c)&(f), the nodes/elements in the latter one are concentrated where they are actually needed.

In order to take advantage of varying mesh resolution for medical images, a desired sizing function has to be determined such as by extracting the features of interest. Alternatively, such sizing can be integrated into the optimization process using an adaptive method, e.g., such that the mesh density around elements with higher intensity variation is increased.

### F. Relationship to deformation and FEM

There exist methods as part of FEM *post-processing* that can refine or modify a mesh based on a computed simulation output such as element strains during deformation. This requires running the simulation first, which in turn necessitates *a priori* knowledge of the boundary conditions. These may not be known prior to meshing, or their location and nature may change substantially from simulation to simulation, e.g. as medical tools interact with the anatomy. Moreover, although post-process refinement techniques adjust node/element density locally, they do not formulate an intrinsically optimal placement for an element. Unlike in mechanical/civil engineering, larger deformations are involved in medical simulations and simulation accuracy around important anatomical features are often preferred over accuracy at high-strain regions (which are mostly the medical tool contact points). Our proposed method optimizes meshes assuming that there is no prior knowledge of the boundary constraints.

Advances in ultrasound and MR-based elastography offer significant potential for our method, since elastography-recovered tissue values can be assigned to mesh elements for subsequent FEM use with minimal loss of information during discretization. In this paper, a vibro-elastography image meshing example is provided in Fig. 11. There is substantial work in the literature on the identification of mechanical tissue features such as the tissue elastic modulus from such elastography data [39].

## VII. Conclusions

In this paper, a penalty function based on FEM interpolation error is combined with a proposed image-representation cost and the combined objective function is minimized to produce high-quality FEM elements that also discretize a given image in a desirable manner. With the emerging fields of elastography imaging and tissue parameter identification, this method becomes a powerful tool allowing the unsupervised single-step meshing of images with a low, and if desired *fixed*, number of vertices. This enables the fast generation of patient-specific models for deformation simulation. Note that such an optimized discretization can be used further for a fast approximate segmentation since the optimized elements represent an image using far fewer degrees-of-freedom than the underlying pixels.

The method was presented both in 2D and 3D using synthetic data, MR images of brain, CT images of the kidney, and elastography imaging of the prostate. These sample medical image results along with comparisons with selected meshing techniques in terms of mesh quality and surface approximation demonstrate that the sound theoretical promise of our proposed VIM method generates successful meshes suitable for numerical simulations.

## Acknowledgment

The authors would like to thank Kirill Sapchuk for his help with the implementation.

## References

[1] M. Sonka and J. M. Fitzpatrick, Eds., *Handbook of medical imaging*. SPIE Press, 2000.

[2] S. Cotin, H. Delingette, and N. Ayache, "Real-Time elastic deformations of soft tissues for surgery simulation," *IEEE Trans on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999.

[3] M. Ferrant, A. Nabavi, B. Macq, F. A. Jolesz, R. Kikinis, and S. K. Warfield, "Registration of 3-d intraoperative MR images of the brain using a finite-element biomechanical model," *IEEE Trans Med Imag*, vol. 20, no. 12, p. 1384–1397, 2001.

[4] H. M. Yin, L. Z. Sun, G. Wang, T. Yamada, J. Wang, and M. W. Vannier, "ImageParser: a tool for finite element generation from three-dimensional medical images," *BioMedical Engineering OnLine*, vol. 3, p. 31, 2004.

[5] O. Goksel, S. E. Salcudean, and S. P. DiMaio, "3D simulation of needle-tissue interaction with application to prostate brachytherapy," *Computer Aided Surgery*, vol. 11, no. 2006, pp. 279–288, 2006.

[6] S. J. Owen, "A survey of unstructured mesh generation technology," in *Int Meshing Roundtable*, vol. 3, 1998.

[7] R. Löhner, "Automatic unstructured grid generators," *Finite Elements in Analysis and Design*, vol. 25, no. 1-2, pp. 111–134, 1997.

[8] N. Archip, R. Rohling, V. Dessenne, P. Erard, and L. P. Nolte, "Anatomical structure modeling from medical images," *Computer Methods and Programs in Biomedicine*, vol. 82, no. 3, pp. 203–215, 2006.

[9] W. J. Schroeder, B. Geveci, and M. Malaterre, "Compatible triangulations of spatial decompositions," in *IEEE Visualization*, Austin, Texas, USA, Oct 2004, pp. 211–217.

[10] M. Müller and M. Teschner, "Volumetric meshes for real-time medical simulations," in *Workshop on Bildverarbeitung für die Medizin*, Erlangen, Germany, 2003, pp. 279–283.

[11] J. Teran, N. Molino, R. Fedkiw, and R. Bridson, "Adaptive physics based tetrahedral mesh generation using level sets," *Engineering with Computers*, vol. 21, no. 1, pp. 2–18, 2005.

[12] F. Labelle and J. R. Shewchuk, "Isosurface stuffing: fast tetrahedral meshes with good dihedral angles," *ACM Trans Graph*, vol. 26, no. 3, pp. 57–66, 2007.

[13] A. Fedorov, N. Chrisochoides, R. Kikinis, and S. Warfield, "An evaluation of three approaches to tetrahedral mesh generation for deformable registration of brain MR images," in *IEEE Int Symp Biomedical Imaging: Nano to Macro*, 2006, pp. 658–661.

[14] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[15] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[16] T. Gevers, "Adaptive image segmentation by combining photometric invariant region and edge information," *IEEE Trans Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 848–852, 2002.

[17] A. J. Cuadros-Vargas, L. G. Nonato, E. Tejada, and T. Ertl, "Generating segmented tetrahedral meshes from regular volume data for simulation and visualization applications," in *Procs of Computational Modeling of Objects Presented in Images (CompIMAGE)*, 2006.

[18] A. C. Reid, S. A. Langer, R. C. Lua, V. R. Coffman, S. Haan, and R. E. García, "Image-based finite element mesh construction for material microstructures," *Computational Materials Science*, vol. 43, no. 4, pp. 989–999, 2008.

[19] H. Si, "TetGen: A quality tetrahedral mesh generator and a 3D Delaunay triangulator," http://tetgen.berlios.de/.

[20] "GHS3D: Tetrahedral mesh generator," http://www.distene.com/build/meshing.html.

[21] J. Dardenne, S. Valette, N. Siauve, N. Burais, and R. Prost, "Variational tetrahedral mesh generation from discrete volume data," *The Visual Computer*, vol. 25, no. 5, pp. 401–410, 2009.

[22] J.-D. Boissonnat and S. Oudot, "Provably good sampling and meshing of surfaces," *Graphical Models*, vol. 67, pp. 405–451, 2005.

[23] "CGAL, Computational Geometry Algorithms Library," http://www.cgal.org.

[24] S. Salcudean, D. French, S. Bachmann, R. Zahiri-Azar, X. Wen, and W. Morris, "Viscoelasticity modeling of the prostate region using vibro-elastography," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2006, pp. 389–396.

[25] J. R. Shewchuk, "What is a good linear element? interpolation, conditioning, and quality measures," in *Int Meshing Roundtable*, 2002.

[26] D. A. Field, "Qualitative measures for initial meshes," *Int J for Numerical Methods in Eng*, vol. 47, no. 4, pp. 887–906, 2000.

[27] L. A. Freitag and C. Ollivier-Gooch, "Tetrahedral mesh improvement using swapping and smoothing," *Int J of Numerical Methods in Engineering*, vol. 40, pp. 3979–4002, 1997.

[28] S. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, and S. Teng, "Sliver exudation," in *Annual Symp on Computational Geometry*, Miami Beach, Florida, United States, 1999, pp. 1–13.

[29] B. Cutler, J. Dorsey, and L. McMillan, "Simplification and improvement of tetrahedral models for simulation," in *Eurographics/ACM SIGGRAPH Symp on Geometry Processing*, Nice, France, 2004, pp. 93–102.

[30] Q. Du and D. Wang, "Tetrahedral mesh generation and optimization based on centroidal voronoi tessellations," *Int J for Numerical Methods in Eng*, vol. 56, pp. 1355–1373, 2003.

[31] L. Chen and J. Xu, "Optimal Delaunay triangulation," *J Computational Mathematics*, vol. 22, no. 2, pp. 299–308, 2004.

[32] L. Chen, "Mesh smoothing schemes based on optimal Delaunay triangulations," in *Int Meshing Roundtable*, 2004, pp. 109–120.

[33] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun, "Variational tetrahedral meshing," *ACM Trans on Graphics (SIGGRAPH)*, vol. 24, no. 3, pp. 617–625, 2005.

[34] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577–685, 1989.

[35] O. Goksel and S. E. Salcudean, "High-quality model generation for finite element simulation of tissue deformation," in *Medical Image Computation and Computer Assisted Intervention (MICCAI)*, London, UK, Sep 2009, pp. 248–256.

[36] L. A. Freitag and P. M. Knupp, "Tetrahedral element shape optimization via the jacobian determinant and condition number," in *Int Meshing RoundTable*, 1999, pp. 247–258.

[37] C. Ollivier-Gooch, "GRUMMP: Generation and refinement of unstructured, mixed-element meshes in parallel," http://tetra.mech.ubc.ca/GRUMMP/.

[38] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*. Butterworth-Heinemann, 2000.

[39] J. F. Greenleaf, M. Fatemi, and M. Insana, "Selected Methods for Imaging Elastic Properties of Biological Tissues," *Annual Review of Biomed Eng*, vol. 5, pp. 57–78, 2000.

[40] J. Ophir, I. Céspedes, H. Ponnekanti, Y. Yazdi, and X. Li, "Elastography: a method for imaging the elasticity of biological tissues," *Ultrasonic Img*, vol. 13, no. 2, pp. 111–134, 1991.

[41] A. Fousse, E. Andres, J. Françon, Y. Bertrand, and D. Rodrigues, "Fast point locations with discrete geometry," in *SPIE Conf Vision Geometry*, Denver, Colorado, USA, Jul 1999, pp. 33–44.

[42] O. Goksel and S. E. Salcudean, "B-mode ultrasound image simulation in deformable 3-D medium," *IEEE Trans Med Imag*, vol. 28, no. 11, pp. 1657–1669, Nov 2009.