Noname manuscript No.
(will be inserted by the editor)

Mesh Deformation-based Multi-tissue Mesh Generation for Brain Images

Yixun Liu[1,2] ·
Panagiotis Foteinos[1,2] ·
Andrey Chernikov[2] ·
Nikos Chrisochoides[2]

Received: date / Accepted: date

Abstract Multi-tissue meshing is necessary for the realistic building of a biomechanical model of the brain, which has been widely used in brain surgery simulation, brain shift, and non-rigid registration. A two step multi-tissue mesher is developed. First, a coarse multi-tissue mesh is generated by redistributing labels of a Body-Centered Cubic (BCC) mesh. Second, all the surfaces of the submeshes are deformed to their corresponding tissue boundaries.

To deform the mesh, two strategies are developed. One is based on a Point-based Registration (PBR) and the other is based on a Robust Point Matching (RPM). The PBR method explicitly calculates the correspondence, which takes both smoothing and quality into account, then resolves the displacement vector by minimizing an energy function. Unlike PBR method, RPM does not require the correspondence between the source points and the target points to be known in advance. To simultaneously resolve the displacement vector and the correspondence, the Expectation and Maximization optimization is employed to alternately estimate the correspondence and the displacement vector. To effectively cope with outliers, Least Trimmed Square, a robust regression technique, is employed to correct the regression bias induced by outliers. Both methods are effective in deforming the multi-tissue mesh. However, the PBR method favors quality and smoothing, and the RPM method favors fidelity.

The resulting mesh is characterized by its flexible control of four mesh properties: 1) tissue-dependent resolution, 2) fidelity to tissue boundaries, 3) smoothness of mesh surfaces, and 4) element quality. Each mesh property can be controlled on a tissue level.

Our experiments conducted on synthetic data, clinic MRI, visible human data, and brain atlas effectively demonstrate these features of this multi-tissue mesher.

This work is supported in part by NSF grants: CCF-1139864, CCF-1136538, and CSI-1136536 by the John Simon Guggenheim Foundation and the Richard T. Cheng Endowment.

[1]Department of Computer Science, The College of William and Mary, E-mail: yxliuwm@gmail.com
[2]Department of Computer Science, Old Dominion University, E-mail: {pafotein, andrey.n.chernikov, npchris}@gmail.com

1 Introduction

Multi-tissue mesh generation of medical images is a necessary procedure for building a heterogeneous biomechanical model, which has numerous applications, such as physical model-based non-rigid registration, segmentation and surgery simulations. However, there is little literature addressing this issue so far.

Several groups [3, 18, 14] presented multi-tissue mesh generation methods based on a Delaunay refinement. However, elements with small dihedral angles (aka, slivers) are likely to occur in Delaunay meshes because elements are removed only when the radius-edge ratio is large. Their dihedral angle quality is completely ignored. Meyer et al. [14] showed at least 0.6% slivers occurred in their experiments on frog data. Boltcheva et al. [3] and Pons et al. [18] employed a sliver exudation postprocessing technique [5] to remove slivers, demonstrating a very good quality mesh (minimal dihedral angle is larger than 4 degrees).

Unlike Delaunay-based methods, Zhang et al. [24] presented an octree-based method to generate a tetrahedral and hexahedral mesh. This method first identifies the interface between two or more different tissues and non-manifold nodes on the boundary. Then, all tissue regions are meshed with conforming boundaries simultaneously. Finally, edge-contraction and geometric flow schemes are used to improve the quality of the tetrahedral mesh.

Molino et al. [15] presented a crystalline, red-green strategy for mesh generation. This method starts from a Body-Centered Cubic (BCC) mesh, then deforms it to the object boundary. The geometry is represented by a signed distance function, and the refinement is performed by a red-green strategy. The BCC-based approach shows a very good quality mesh because the quality of the BCC mesh is high, and its regular refinement still leads to a BCC mesh. However, this approach is limited to a single tissue.

The contribution of this paper is a novel mesh generation method, which is characterized by *1) multi-*

tissue mesh, 2) tissue-dependent resolution, and 3) natural control of the trade-off among quality, fidelity, and smoothness on a tissue level, which, in fact, provides a mechanism to substantially improve the mesh quality by slightly deviating from the input boundary.

2 Method

Our approach requires a multi-label image as input, in which label 0 denotes the background, and positive integers indicate different tissues. The approach consists of two steps: coarse mesh generation (CMG) and mesh deformation, as shown in Fig. 1. Mesh deformation implements two strategies: Robust Point Matching (RPM) and Point-based Registration (PBR).

CMG includes two substeps:

1. BCC mesh:
   Use BCC mesh to subdivide the object space into connected tetrahedra. Note that this step does not distinguish different tissues. The resulting BCC mesh is homogeneous.
2. Coarse tissue dependent resolution multi-tissue mesh generation (CMesh):
   This step specifies which tissue each tetrahedron belongs to. Each tissue is capable of automatically adjusting its resolution based on its geometric complexity and the predefined subdivision criterion.

The resulting coarse multi-tissue mesh of step 1 includes different submeshes, and each submesh has its own resolution. The discrepancy between the surface of the submesh and its corresponding boundary in the multi-label image is corrected by RPM/PBR method. This step includes three substeps:

1. detect edges for each tissue in the multi-label image to obtain a target point set
2. extract surface nodes for each submesh to obtain a source point set
3. deform the surface of each submesh to its corresponding boundary based on RPM/PBR

The framework of the approach is shown in Fig. 1. Each step listed in this framework will be discussed in detail in the following sections.

2.1 Coarse mesh generation

The purpose of the coarse mesh generation is to obtain source points, which will be used in subsequent mesh deformation. The coarse mesh needs to take into account the following criteria: 1) a multi-tissue input, 2) a good conditioning for subsequent mesh deformation, and 3) fewer tetrahedra.

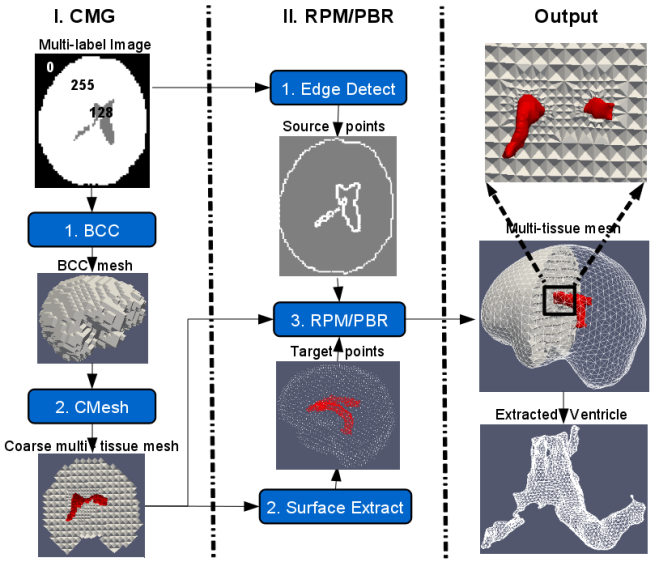


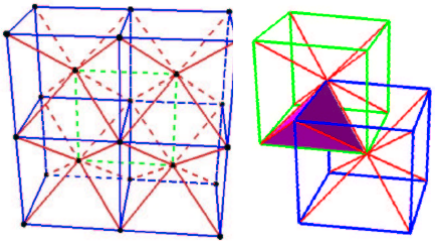
Fig. 1 Multi-tissue mesher framework.

This part includes two steps, as shown in Fig 1. Body-Centered Cubic provides an initial lattice, which has been well documented in [9,15]. For the completeness of this paper, we will briefly describe its properties and red-green subdivision, then focus on how CMesh generates and refines submeshes.

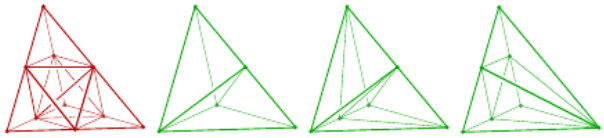
2.1.1 BCC mesh

BCC mesh is an actual crystal structure ubiquitous in nature. It is highly structured and easily refined initially or during the simulation [15]. The nodes of BCC are grid points of two interlaced grids like the blue grid and the green grid in Fig. 2(a). The edges of BCC consist of edges of the grid and additional edges between a node and its eight nearest neighbors in the other grid.

The refinement of the BCC mesh is performed by a red-green strategy. Initially, all BCC lattice tetrahedra are labeled with a red color. A red tetrahedron can be subdivided into eight children (1:8 refinement), and each child is labeled with a red color, as shown in Fig. 2(b). There are three choices for the internal edge of the tetrahedron. If the shortest one is selected, the resulting eight child tetrahedra are exactly the BCC tetrahedra except the size is one half of the original BCC. So, the quality of the refined mesh can be guaranteed using this red (regular) subdivision. This is the reason that we select BCC as the initial tetrahedral mesh although our method is general enough to start from any tetrahedral meshes. The red subdivision will lead to T-junctions at the newly-created edge midpoints where neighboring tetrahedra are not refined to the same level.



(a) A portion of the BCC lattice. The blue and the green connections depict the two interlaced grids, and the eight red connections at each node lace these two grids together.



(b) Red-greed subdivision

Fig. 2 BCC lattice and red-green subdivision (The two figures come from [15]).

To remove the T-junctions, green subdivision, including three cases, is performed. The three cases are:

1. there is one edge with T-junction
2. there are two opposite edges with T-junctions
3. there are three edges of a face with T-junctions

The green subdivision, corresponding to the three cases, is shown in Fig. 2(b). All the child tetrahedra of the green subdivision are labeled with a green color. The irregular green subdivision will reduce the quality of the tetrahedron. So, all the child tetrahedra will be removed, and the red subdivision is performed on their red parent when higher resolution is desired.

2.1.2 CMesh

CMesh is used to identify the submesh for each tissue in BCC mesh, and subdivide it if necessary. We define a label operation table, based on which label redistribution is performed to produce different submeshes. A predefined subdivision criterion is used to determine which submesh needs to be further subdivided. If a submesh needs to be subdivided, in order to reduce the number of the tetrahedra, only its boundary tetrahedra are further subdivided (multi-resolution).

In Fig. 3, we illustrate how CMesh identifies and subdivides submeshes. First, CMesh assigns each tetrahedron with a label of the tissue, to which most part of the tetrahedron belongs by simply counting the number of voxels within the tetrahedron (Fig. 3(a)). As a result, an initial multi-tissue mesh is produced. However, this multi-tissue mesh is not well conditioned for subsequent deformation because more than one face are probably on the interface. We term this kind of tetrahedron a bad conditioned tetrahedron. In this case, deforming four nodes easily crushes this tetrahedron. We prefer a submesh only including two kinds of tetrahedra: inner tetrahedra (no faces on the interface) and boundary tetrahedra (only one face on the interface). To reach this end, we redistribute the label of the bad conditioned tetrahedra according to the operations defined in Table 1 to generate a well conditioned multi-tissue mesh (Fig. 3(b)). The label redistribution operation is performed label by label, and, therefore, the labeling is unique. After label redistribution, we need to check if each submesh needs to be further subdivided. If it satisfies the criterion for the resolution, defined in Fig. 3(e), the algorithm stops, Otherwise, it subdivides (Fig. 3(c)) and redistributes labels (Fig. 3(d)). The above procedures repeat until the desired resolution is reached. The submesh, produced by the label redistribution, not only has good conditioning, but also reaches conformity with its neighboring submeshes.

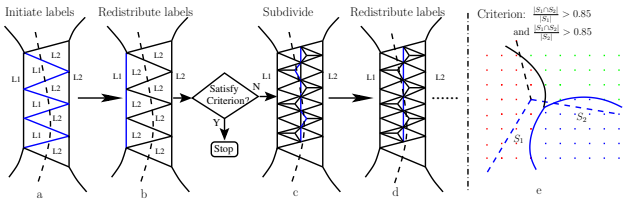


Fig. 3 Coarse multi-tissue mesh generation. (a) L1 and L2 are tissue labels, the dashed line is the real boundary, and the blue line is the submesh interface. (b) Redistribute labels according to operation table 1. (c) Subdivide if not satisfy the resolution criterion defined in (e). (d) Redistribute labels again. (e) Resolution criterion: 0.85 is the subdivision threshold, an experiment value evaluated on MRI ,visible human, and brain atlas. Points represent voxels and colors represent different tissues. S_1 is the voxel set within the blue submesh (blue dash lines), and S_2 is the voxel set within the blue tissue (blue curves).

Operation table The operation table decides how to redistribute the label of a tetrahedron based on its relation, termed as configuration, with face-adjacent tetrahedra. The purpose of the operations defined in Table 1 is to move the bad conditioned tetrahedra to its neighboring submeshes. If all the bad conditioned tetrahedra are removed from one submesh, this submesh and its neighboring submeshes will reach good conditioning at the same time. We clarify this point by taking case 5 defined in table 1 as an example. If the four face-adjacent tetrahedra of a given tetrahedron T have labels: $< L, L1, L1, L1 >$, denoted as $< L, 3L1 >$ for simplicity, the label of T will be reassigned with $L1$ because its three faces are on the interface between

submesh L and $L1$. Fig. 3 uses case 5 for redistribution. Because we use 2D triangles instead of 3D tetrahedra in Fig. 3, case 5 is degenerated from $< L, 3L1 >$ to $< L, 2L1 >$. In summary, the operations defined in Table 3 move a tetrahedron to its face-adjacent submesh if this tetrahedron is not an inner (case 1) or boundary tetrahedron (case 2). As a result, no tetrahedra with more than one face on the boundary exist, which leads to a well conditioned mesh for the subsequent deformation.

Table 1 Operation case table for a tetrahedron T with a label L.

Case	Configuration	Operation
1	4L	T=inner tetra
2	$< 3L, 1L1 >$	T=boundary tetra
3	$< 2L, 2L1 >$	T.label=L1
4	$< 2L, 1L1, 1L2 >$	T.label=L1
5	$< 1L, 3L1 >$	T.label=L1
6	$< 1L, L1, 2L2 >$	T.label=L2
7	$< 1L, 1L1, 1L2, 1L3 >$	T.label=L1

Criteria for subdivision In multi-label image, a tissue is defined by a set of voxels with the same intensity, say L. Heuristically, the closer the surface of a submesh is to the boundary of a tissue, the more voxels of the tissue are located in the submesh, and the more voxels with label L this submesh has. To quantitatively evaluate the similarity between the submesh and the tissue region, we define two voxel sets:

1. $S1$: all voxels in the submesh (the points within two dashed lines in Fig. 3 (e))
2. $S2$: all voxels in the tissue region (the points within the curve in Fig. 3 (e))

$S_1 \cap S_2$ defines the point set shared by the submesh and the tissue region. We expect the common region to be similar with the submesh and the tissue region. We use $\frac{|S_1 \cap S_2|}{|S_1|}$ to measure the similarity between the common region and the submesh, and $\frac{|S_1 \cap S_2|}{|S_2|}$ to measure the similarity between the common region and the tissue region. So, the subdivision criterion can be defined as:

$$\frac{|S_1 \cap S_2|}{|S_1|} < threshold \quad and \quad \frac{|S_1 \cap S_2|}{|S_2|} < threshold \tag{1}$$

where $threshold$ is an input parameter. $0 \leq \frac{|S_1 \cap S_2|}{|S_1|} \leq 1.0$ and $0 \leq \frac{|S_1 \cap S_2|}{|S_2|} \leq 1.0$, so $0 \leq threshold \leq 1.0$.

The reason that we simultaneously use two values as the criterion is to avoid case a and case b in Fig. 4. Moreover, in order to avoid case c in Fig 4, we do not simply use $\frac{|S_1|}{|S_2|}$.

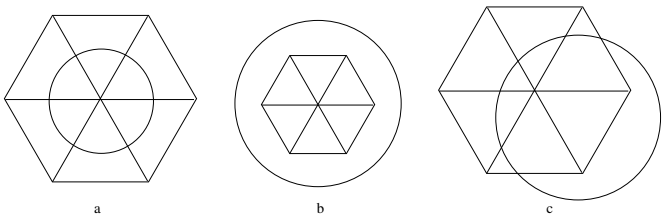


Fig. 4 Three special cases. The circle represents the tissue region, and the polygon represents the submesh. For simplicity, the voxels are not shown. All these three cases show a big discrepancy between the tissue boundary and the submesh boundary. However, for case (a), because the tissue is totally covered by the submesh, $\frac{|S_1 \cap S_2|}{|S_2|}$ has the highest value 1.0. For case (b), because the submesh is totally covered by the tissue region, $\frac{|S_1 \cap S_2|}{|S_1|}$ has the highest value 1.0. For case (c), $\frac{|S_1|}{|S_2|}$ can be 1.0 if the submesh and tissue region have the same number of voxels.

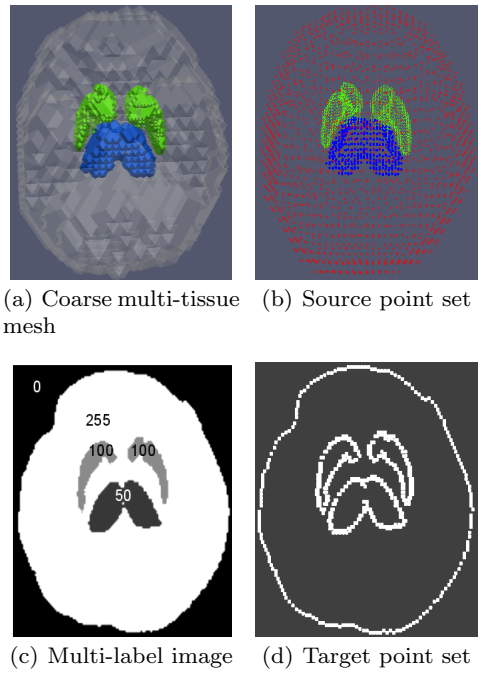
The criterion relies on the number of the voxels, and, therefore, it is susceptible to the resolution of the multi-label image. For instance, if the resolution is very low, we cannot find any voxels in a tetrahedron. To overcome this difficulty, up-sampling is performed automatically if no voxels are detected in a tetrahedron. To improve the performance, we do not perform up-sampling in the whole image, but restrict it to the bounding box of the tetrahedron.

2.2 Mesh Deformation

This step is used to 1) deform the coarse mesh close to the boundary, 2) maintain the quality of the coarse mesh, and 3) generate a smooth mesh. The coarse mesh needs to be deformed to the boundary. Unlike the interpolation method used in [15], we develop two mesh deformation strategies based on RPM and PBR. PBR method needs to explicitly specify the correspondence (displacement) between two point sets, and, therefore, it provides a mechanism to control the smoothing and the quality of the mesh. RPM method does not require explicit correspondence calculation, but handles it in an Expectation and Maximization framework. Compared to PBR method, this method lacks flexible control on the smoothing and quality, but is capable to reach a higher fidelity.

2.2.1 Source point set and target point set

Two point sets are needed in the mesh deformation: the source and target point sets. The source points are the surface nodes of the mesh, and the target points are the edge points in the multi-label image. The source point set is obtained by extracting the surface nodes of each



(a) Coarse multi-tissue mesh (b) Source point set (c) Multi-label image (d) Target point set

Fig. 5 Point sets. The source point set (b) includes all the surface nodes of the coarse mesh (a), and the target point set (d) are the edge points in the multi-label image (c).

submesh. The target point set is obtained by canny edge detection, which is facilitated by ITK implementation [12]. For each source point, its potential correspondence point is located in the neighborhood of the source point. It is computationally intensive to search for the correspondence point in all target points. We associate each source/target point with a label to denote which tissue it belongs to, and, therefore, the search is only restricted to the target points, which have the same label with the source point.

Figure 5 shows the source point set and the target point set produced by visible human data. These intermediate results for other data will not be shown in Section 3.

2.2.2 Extended Robust Point Matching for Mesh Deformation

RPM is a flexible framework for non-rigid point matching, in which the thin-plate spline (TPS) is employed as the parameterization of the non-rigid spatial mapping, and the softassign is used for the correspondence [10].

Our extension are twofold: First, the TPS smoothing term in RPM is replaced with the stress energy of a biomechanical model used in [6], and, therefore, the underlying deformation can be estimated more realistically. Second, we combine RPM with a powerful robust regression technique: Least Trimmed Square [20], to effectively deal with outliers.

To solve the mapping function and correspondence, the point matching problem is formulated as a functional minimization decomposed into a regularization energy and a similarity energy.

A Fuzzy Linear Assignment Energy Functional
Suppose there are two point sets S (Source point set) and T (Target point set) in $\Re^3$ consisting of points $s_i, i = 1, 2, ...p$ and $t_i, i = 1, 2, ...l$, respectively. The functional is constructed as follows:

$$W(u, C) = \int_{\Omega} \sigma(u)^t \epsilon(u) + \lambda \sum_{i=1}^{p} \| s_i + u(s_i) - \sum_{t_j \in \Omega_R} c_{ij} t_j \|^2 \tag{2}$$

The first term is the regularization energy defined by the stress energy of a linear elastic model, and the second term is the similarity energy. A physical model is capable of realistically describing the movement of the soft tissue, allowing accurately estimating the inner deformation given the boundary condition. This physical model is widely used in the image registration field [13, 6]. λ is used to control the trade-off between these two energies. Using the stress energy as the regularization term will make the estimation of the mapping function more realistic than other work [10, 16, 23], which use the smoothing measure of TPS or CSRBF as the regularization term.

In the similarity energy, Ω_R defines the search range, which is a sphere centered at the source point with radius R. c_{ij} is the probability with which the point s_i corresponds with t_j located in Ω_R. u is the unknown displacement field, and C is the unknown correspondence matrix with entry c_{ij}. The correspondence matrix C is similar with that in [10], but we define a range Ω_R, and only take into account the target points located in Ω_R. The search range basically makes RPM act as a multi-resolution matching. As the range reduces, the matching will go from the coarse level to the fine level. c_{ij} is calculated as equation (3). For each source point s_i, assume its potential correspondences are subject to the Gaussian distribution:

$$c_{ij} = \frac{c'_{ij}}{\sum_{k=1}^{k=m} c'_{ik}}, \quad c'_{ij} = \frac{1}{R\sqrt{2\pi}} e^{\frac{-(t_j - s_i)^2}{2R^2}}, \quad \forall t_j \in \Omega_R, j = 1 \dots m \tag{3}$$

Combining the search range with Least trimmed squares (LTS) [20], we can effectively detect outliers existing in both point sets.

It is difficult to find the analytical solution from equation (2). We use finite element method to discretize

the problem by approximating:

$$u = \sum_{i=0}^{i=n} N_i U_i, \tag{4}$$

where n is the number of the vertices of the finite element mesh, N is the shape function, and U is node displacement vector. For simplicity, we define a vector D with entry:

$$D_i(c_{ij}) = s_i - \sum_{t_j \in \Omega_R} c_{ij} t_j. \tag{5}$$

The homogeneous biomechanical model used in [6] is generalized with a more flexible tissue-aware model. As a result, equation (2) can be discretized as:

$$W(U,C) = \sum_{i=1}^{n} (U^T K_i U + \lambda_i (H_i U - D_i(C))^T (H_i U - D_i(C))), \tag{6}$$

where n is the number of the tissues, and K_i is the global stiffness matrix assembled by the tetrahedra within i-th tissue. K_i is related with two biomechanical attributes of the i-th tissue: Young's modulus and Possion's ratio. The building of K_i has been well documented in [1]. H_i is the global linear interpolation matrix assembled by matching points.

Each matching point o_k with number k contained in tetrahedron with vertex number $c_i, i \in [0:3]$ contributes to four 3×3 submatrices: $[H]_{kc_0}$, $[H]_{kc_1}$, $[H]_{kc_2}$, and $[H]_{kc_3}$. Readers are referred to [6,13,1] for details. $[H]_{kc_i}$ is defined as: $[H]_{kc_i} = diag(h_i, h_i, h_i)$. The linear interpolation factor h_i is calculated as:

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} v_{c_0}^x & v_{c_1}^x & v_{c_2}^x & v_{c_3}^x \\ v_{c_0}^y & v_{c_1}^y & v_{c_2}^y & v_{c_3}^y \\ v_{c_0}^z & v_{c_1}^z & v_{c_2}^z & v_{c_3}^z \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} o_k^x \\ o_k^y \\ o_k^z \\ 1 \end{bmatrix} \tag{7}$$

where v_{c_i} is the vertex with number c_i. Because we use the node as the matching point, which means o_k is same with one of the four nodes, equation (7) is reduced to:

$$h_i = \begin{cases} 1 & \text{for } o_k = v_{c_i} \\ 0 & \text{for } o_k \neq v_{c_i} \end{cases} \tag{8}$$

We term energy function (6) as a tissue-aware model because it is able to use λ_i to balance the quality and fidelity for the i-th tissue whether this model is homogeneous (same Young's modulus and Possion's ratio for all tissues) or not.

The displacement vector U and correspondence matrix C are resolved in an Expectation and Maximization framework, in which C is estimated using equation (3) in E step, and U is calculated by minimizing $W(U)$ in M step.

$$\frac{\partial W}{\partial U} = 0 \Rightarrow \sum_{i=1}^{i=n} (K_i + \lambda_i H_i^T H_i) U = \sum_{i=1}^{i=n} \lambda_i H_i^T D_i(C) \tag{9}$$

$\sum_{i=1}^{i=n} (K_i + \lambda_i H_i^T H_i)$ is semi-positive definite matrix, therefore we can use Conjugate Gradient (CG) [22] to resolve the linear system of equations. This component is computed in parallel, facilitated by PETSc implementation [17].

Expectation and Maximization The Expectation and Maximization (EM) algorithm [8] is a general optimization technique for maximum-likelihood [11] estimation of the unknown model parameter in the presence of missing or hidden data.

$$L(\theta) = lnP(X|\theta) = ln(\sum_z P(X|z,\theta)P(z|\theta)), \tag{10}$$

where X is measurement data, θ are unknown model parameters, and z are hidden variables.

To estimate the model parameter, EM proceeds iteratively, and each iteration of the EM algorithm consists of two steps: The E step and the M step. In the E step, the missing data are estimated given the observed data and current estimate of the model parameters. In the M step, the likelihood function is maximized under the assumption that the missing data are known. The estimate of the missing data from the E step are used in lieu of the actual missing data. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration.

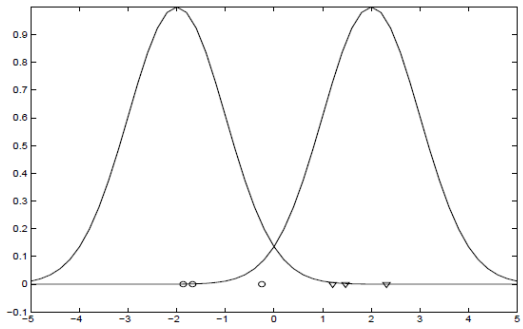


Fig. 6 An EM example: Mixture components and data. The data consists of three samples drawn from each mixture component, shown above as circles and triangles. The means of the mixture components are -2 and 2, respectively, which need to be estimated from six samples.

Fig. 6 from [7] shows two Gausian mixtures and six samples drawn from the mixtures, in which the mean of

each mixture is unknown. The purpose is to estimate the two means without knowing from which mixture each sample is drawn.

Since there are two mixtures and six samples, all the possible data associations can be represented by a 2×6 table.

In E step, a soft assignment, a posterior probability the sample associating with the mixture, is estimated for each sample. Then, a lower bound to the true likelihood function is constructed as [4]:

$$l(\theta|\theta_n) = L(\theta_n) + \Delta(\theta|\theta_n), \tag{11}$$

where θ_n is the estimation of θ at n-th iteration, and $\Delta(\theta|\theta_n)$ is the difference of the log likelihood defined as:

$$\Delta(\theta|\theta_n) = \sum_z P(z|X,\theta_n) ln(\frac{P(X|z,\theta)P(z|\theta)}{P(z|X,\theta_n)P(X|\theta_n)}) \tag{12}$$

As shown in Fig. 7 from [4], the lower bound function $l(\theta|\theta_n)$ is equal to the true likelihood function at $\theta = \theta_n$. The new updated θ_{n+1} is the value, which maximizes the lower bound function in M step.

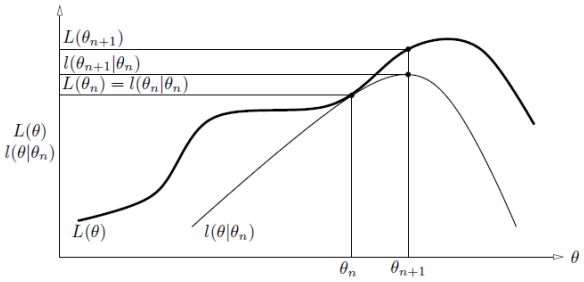


Fig. 7 Lower bound function $l(\theta|\theta_n)$ of the likelihood function $L(\theta)$

Each next bound is an increasingly better approximation to the real likelihood, and, therefore, EM is capable of guaranteeing the convergence.

The intuition behind EM is: alternate between estimating the unknowns and the missing data. The point matching problem can be stated as: find the mapping function (unknown) between the source point set and target point set in the absence of the correspondence (missing data). The EM proceeds as follows:

- E-step: estimate correspondence given current estimate of the mapping function according to equation (3)
- M Step: calculate mapping function given correspondence according to equation (9)

Outlier rejection We present an outlier detection technique by combining the search range with Least Trimmed Square (LTS) estimator [20]. LTS estimator is a robust regression technique tolerant to outliers. Considering a linear regression model for sample $(\boldsymbol{x}_i, y_i)$ with a response variable y_i and a vector of p explanatory variables $\boldsymbol{x}_i$:

$$y_i = \beta \mathbf{x}_i + \varepsilon_i, i = 1, \ldots, n. \tag{13}$$

where $\boldsymbol{\beta}$ is the coefficient vector, and ε is a random error term.

The LTS estimator is defined as:

$$\overline{\beta} = \underset{\beta \in \Re^P}{\operatorname{argmin}} \sum_{i=1}^{h} r_i^2(\beta) \tag{14}$$

where $r_i^2 \leq \ldots \leq r_n^2$ are the ordered squared residuals. Equation (14) is very similar to the traditional least square with the only difference that only h observations with the smallest squared residuals are used in the summation, thereby allowing the fit to stay away from the outliers. The best robustness properties are achieved when h, termed as trimming constant, is approximately $n/2$, in which case the breakdown point attains 50% [20]. It is computationally intensive to determine the LTS estimator by examining the total of $\binom{n}{h}$ subsamples when n is large [19].

In this paper, combining with a search range R, we present an approximation method. This method contains two steps: a trial step and an outlier rejection step.

Trial step: Using EM algorithm to find the mapping function corresponding to the search range R, and then transform the source points. The purpose of this step is not the mapping function due to its bias induced by the outliers, but the detection of outliers in the next step.

Outlier rejection step: based on the transformed source point set, for each source point, find target points within the search range $R = R \times a$, where a is the annealing parameter, and is equal to 0.93 as suggested in [10]. If there are no target points within the search range of the source point, this source point is marked as an outlier. Replace the original source point set with this marked source point set, and estimate the mapping function again. The difference between this modified LTS and the traditional LTS is that we use the search range instead of h to perform the outlier rejection, and, therefore, there is no need for the ordering of the residuals. For the target points, only the points in the range are involved in the computation, and the other points will be marked as outliers. So, this method can be used to deal with the outliers in both point sets.

The complete pseudo codes are presented in Algorithm 1, in which **Coarse Mesh Generation** generates a coarse multi-tissue mesh and **RPM Deformation** deforms the coarse mesh to the tissue boundary.

Algorithm 1 MULTI-TISSUE MESH GENERATION

M=MultiTissueMesher($MultiLabelImage, tolerance$)

Require: $MultiLabelImage, tolerance$
Ensure: M: tissue dependent resolution multi-tissue mesh
1. **Coarse Mesh Generation:**
2. Generate BCC mesh M
3. Assign label for each tetrahedron in M
4. **repeat**
5. Label redistribution according to Table 1 to yield multi-tissue mesh M
6. **for** each subMesh **do**
7. **if** satisfy the subdivision criterion (equation (1)) **then**
8. Subdivide M along the boundary using red green strategy
9. **end if**
10. **end for**
11. **until** no subdivision
12. **RPM Deformation:**
13. Generate source point set S by surface extraction from M *(out of deformation loop, different from PBR)*
14. Generate target point set T by edge detection from $MultiLabelImage$
15. Assemble K_i
16. Assemble H_i using equation (7)
17. **repeat**
18. LTS trial step:
19. E step: Estimate correspondence C according to equation (3)
20. Calculate D_i using equation (5)
21. M step: Solve U according to equation (9)
22. Transform S based on U: $S \Leftarrow U(S)$
23. LTS outlier rejection step:
24. $S \Leftarrow S - s_i$ if there are no target points in $\Omega_{R\times a}$
25. recalculate U based on outliers rejected S
26. Deform M using $M \Leftarrow M + U$
27. $error \Leftarrow \|U_i - U_{i-1}\|$ between successive iterations
28. $R \Leftarrow R \times a$
29. **until** $error < \epsilon$
30. Remove the tetrahedra with label 0 from M

2.2.3 Point-based Registration for Mesh Deformation

The classic PBR [6] is used to register two images: floating image and reference image. The PBR is based on the concept of energy minimization. A sparse set of registration points within the floating image are identified. The displacement between the floating and the reference images is estimated using Block Matching [2] at each registration point. These displacements are applied as a boundary condition on a biomechanical model to derive the entire brain deformation.

In our work, we extend this PBR method, and use it in the mesh generation field. In the mesh generation, the registration points will be fixed to the nodes of the mesh instead of the feature points. The displacements of these registration points are estimated by taking fidelity, smoothing, and quality into account. The displacement is known in the energy function instead of a variable relying on the correspondence C in RPM, i.e. $D(C)$.

The energy function is constructed similarly with equation (6), except that D is assumed to be known.

$$W(U) = \sum_{i=1}^{n} (U^T K_i U + \lambda_i (H_i U - D_i)^T (H_i U - D_i)), \tag{15}$$

To incorporate smoothing into the registration framework, we calculate D according to the relaxed target position using the classic Laplacian smoothing. Generally, mesh smoothing is performed as a postprocessing after the mesh generation. However, this will lead to the smoothing out of control of the biomechanical model. So, we reflect the smoothing as we calculate D by naturally incorporating it into energy function (6). The i-th entry d_i of distance vector D is calculated as follows:

Let the source point corresponding to d_i be s, its normal be n, and the set of its neighboring nodes be S. $\mid S \mid$ denotes the set's cardinality or size. The normal n is calculated by averaging the normals of the surface faces, which share the source point s. For each point $p_i \in S$, calculate its closest target point $t_i, i = 1 \dots m$. For s, calculate its closest target point q. The relaxed (smoothed) position of s is $s' = \frac{\sum_{k=1}^{k=m} t_i + q}{|S|+1}$. Projecting $s' - s$ onto the normal of s leads to:

$$d_i = (\frac{\sum_{k=1}^{k=m} t_k + q}{\mid S \mid +1} - s) \cdot n \tag{16}$$

We illustrate the calculation of d_i in Fig. 8.

Once D is known, U can be resolved using equation (9). After we obtain U, we can update the positions of the nodes of the mesh. This procedure will be repeated until the average error between source points and target points is below a predefined tolerance or the iteration reaches maximum number. The average error is evaluated by:

$$\bar{d} = \frac{\sum \|s_i - t_i\|}{\mid S \mid}, \tag{17}$$

where s_i is a source point, t_i is the closest target point of s_i, and S is a source point set. The average error is also used to evaluate the fidelity in Section 3.

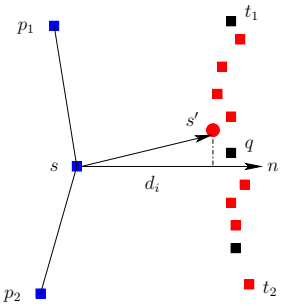


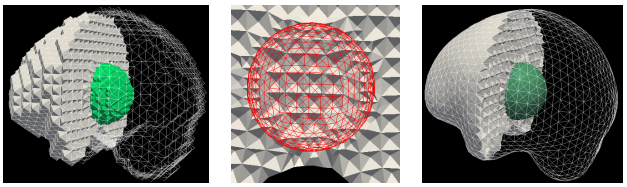
Fig. 8 The calculation of d_i of node s. p_1 and p_2 are two neighboring nodes of s. t_1, t_2, and q are the closets points corresponding to p_1, p_2, and s, respectively. Their average position is s'. Project $s' - s$ on unit normal n of the node s to produce d_i.

The whole method, including the coarse mesh generation and the PBR based deformation, is presented in Algorithm 2.

Algorithm 2 MULTI-TISSUE MESH GENERATION

M=MultiTissueMesher($MultiLabelImage, tolerance$)

Require: $MultiLabelImage, tolerance$
Ensure: M: tissue dependent resolution multi-tissue mesh
1. **Coarse Mesh Generation:**
2. Generate BCC mesh M
3. Assign label for each tetrahedron in M
4. **repeat**
5. Label redistribution according to Table 1 to yield multi-tissue mesh M
6. **for** each subMesh **do**
7. **if** satisfy the subdivision criterion (equation (1)) **then**
8. Subdivide M along the boundary using red green strategy
9. **end if**
10. **end for**
11. **until** no subdivision
12. **PBR Deformation:**
13. Generate target point set T by edge detection from $MultiLabelImage$
14. **repeat**
15. Generate source point set S by surface extraction from M *(in deformation loop, different from RPM)*
16. Calculate D_i using equation (16)
17. Assemble K_i
18. Assemble H_i using equation (7)
19. Solve U using equation (9)
20. Deform M using $M \Leftarrow M + U$
21. Calculate error $\bar{d}$ using equation (17)
22. **until** reach maximum iteration or $\bar{d} < tolerance$
23. Remove the tetrahedra with label 0 from M



(a) Coarse brain and sphere (b) Closeup of the cut through brain and sphere view (c) Compressed brain and sphere

Fig. 9 Multi-tissues mesh generation for synthetic data. The coarse multi-material mesh (a) is compressed into (c), then cut through and zoomed in as (b)

3 Results

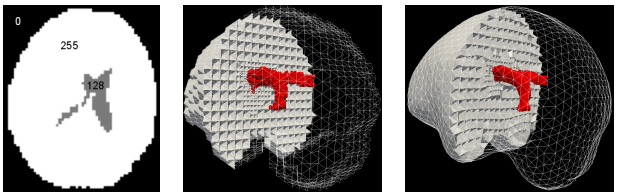
To completely evaluate the method, we first conduct experiments on synthetic data with an artificial sphere inserted into the brain to show the smoothing of the multi-tissue mesh. Then, a clinic MRI, which includes two tissues: brain and ventricle, is used to generate a two tissue mesh. We use this data for the comparison between RPM method and PBR method and the conformity evaluation. Furthermore, we use two nerves in the visible human data to evaluate the tissue-aware quality control. Finally, we qualitatively and quantitatively evaluate the method on a non-manifold data, i.e. a brain atlas. Note that except the results presented in the comparison between PBR and RPM, the other results are generated using PBR method due to its advantages regarding surface smoothing and mesh quality.

3.1 Synthetic data

We construct a synthetic data by inserting a sphere into the brain. The results of the synthetic data are shown in Figure 9. Figure 9(a) is generated using Algorithm 2 with BCC parameter $8mm$ and subdivision threshold 0.8. The outer boundary of the brain is not further subdivide, but its inner interface with the sphere is further subdivided, as shown in Figure 9(b). Figure 9(c) shows the smoothing of the brain and the sphere from an outer view.

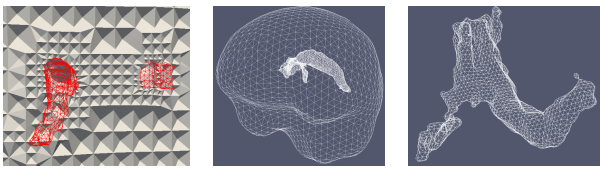
3.2 Real MRI

The ventricle has different biomechanical attributes from other tissues in the brain, and, therefore, it is often used to build a heterogeneous biomechanical model [21]. We evaluate our method on this simple heterogeneous model: the ventricle and the rest of the brain, in which the Young's modulus $E = 10Pa$, Poisson's ratio $\nu = 0.1$ for ventricle, and $E = 3000Pa$, $\nu = 0.45$ for the rest



(a) Multi-label image (b) Coarse mesh (c) Final mesh

Fig. 10 Multi-tissue mesh generation for MRI data. (a) is the multi-label image. The coarse multi-tissue mesh (b) is generated with subdivision threshold 0.85. (c) is the deformed multi-tissue mesh. The numbers of source points and target points are 4497 and 31241, respectively.



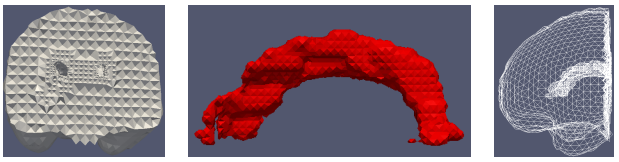
(a) Closeup (b) Wireframe view (c) Extracted ventricle

Fig. 11 (a) is the closeup of the inner ventricle, (b) is the wireframe view of the two submeshes, and (c) is the extracted ventricle.

of the brain [21]. The results are shown in Fig. 10. Fig. 10(a) is the multi-label image, in which labels 128 and 255 denote the ventricle and the brain, respectively. Fig. 10(b) is the coarse multi-tissue mesh, and Fig. 10(c) is the final (deformed) multi-tissue mesh. The deformed mesh is cut through and zoomed in as Fig. 11(a). Fig. 11(b) is the wireframe view of two submeshes, and Fig. 11(c) is the extracted ventricle. The subdivision threshold we used to produce Fig. 10(b) is 0.85. With this parameter, the outer boundary of the brain is not further subdivided, but its inner interface with the ventricle is subdivided twice. Fig. 11(b) clearly shows that the ventricle has higher resolution than the brain.

From Fig. 10(a), we can see that the segmented brain and ventricle are not smooth, but the brain submesh (Figure 10(c)) and the extracted ventricle submesh (Figure 11(c)) are very smooth. It demonstrates that this method has a low requirement for the segmentation due to the incorporation of the smoothing into the PBR framework.

To show the conformity of the interfaces, we first extract two submeshes: the brain and the ventricle. The extracted brain is shown in Fig. 12(a), in which the hole is induced by the extracted ventricle. The extracted ventricle is shown in Fig. 12(b). We want to insert the ventricle into the hole to show the conformity on the interface between the ventricle surface and the hole surface. The ventricle surface should not be



(a) Ventricle hole (b) Ventricle Surface (c) Hole wireframe

Fig. 12 (a) is the brain with a ventricle hole. (b) is the extracted ventricle surface. (c) is the wireframe view of the hole. The front surfaces of the brain are culled to show the hole.

too smooth to distinguish surface triangles. Otherwise, the conformity is not easily to be observed.

To show the conformity, we need to visualize the two surfaces on the interface simultaneously. So, the hole should be visualized in a different way from the ventricle. We use wireframe to show the hole as Fig. 12(c). Note that the front surface of the brain in Fig. 12(c) is culled to clearly show the hole. Fig. 13 is the result of inserting the ventricle into the hole. Part of the interface located in the bounding box is zoomed in to show the conformity. We conducted this experiment on Dell PowerEdge (2 x dual-core Opteron 2218, 2.6 GHz CPU) with a runtime of about 5 minutes.

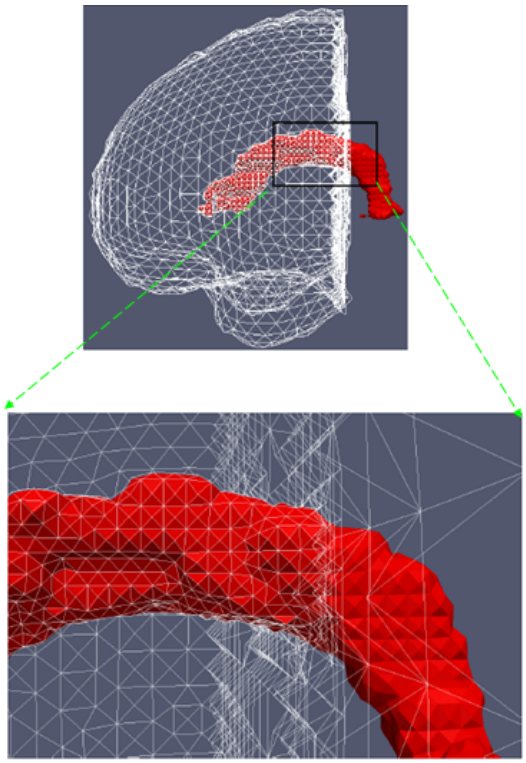


Fig. 13 Multi-tissue mesh conformity.

Table 2 shows the comparison between the RMP method and the PBR method. It clearly shows that

RPM method is capable of dramatically improving the fidelity, but at the same time deteriorates the mesh quality. On the contrary, the PBR method can effectively maintain the quality of the mesh at the cost of sacrificing fidelity. The meshing-research community has been taking the input-model boundary as a hard constraint, which has been making any types of meshing unnecessarily difficult. However, in reality, only fraction of the input-model boundary is really important, and other part of the boundary can be slightly perturbed without making a substantial impact on the finite element simulation. This PBR method provides a mechanism to naturally control the balance between the fidelity and the quality of the mesh.

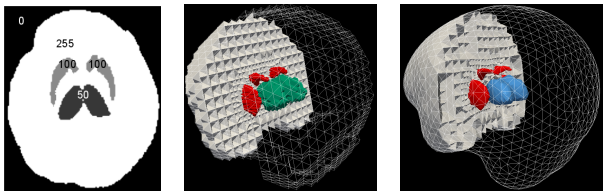
Table 2 Comparison between RPM and PBR methods in terms of minimum dihedral angle and average distance with λ_i fixed.

	PBR		RPM	
Iteration Num.	Dihedral angle	Distance	Dihedral angle	Distance
0	[39.00,39.00]	4.71	[39.00,39.00]	4.71
1	[24.31,50.22]	2.03	[8.59,71.62]	1.21
2	[15,67,61.58]	1.25	[3.70,80.58]	0.33
3	[9.02,77.48]	0.87	[1.05,86.54]	0.17
4	[5.35,80.97]	0.66	[0.31,110.76]	0.03
5	[4.62,81.05]	0.61	[0.11,115.41]	0.01

3.3 Visible human

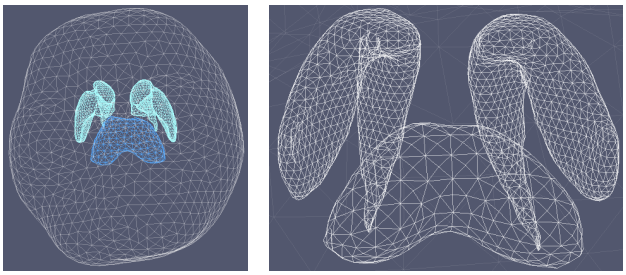
We also evaluate the method using visible human data[1]. Its multi-label image is shown in Fig. 14(a). This data includes three tissues: two nerves (dorsal thalamus (DT) with label 50 and caudata nucleus (CN) with label 100) and the brain with label 255. Fig. 14 and Fig. 15 show the results of this data. We use the same subdivision threshold 0.85 with that in MRI data. Fig. 15(a) and Fig. 15(b) clearly demonstrate the tissue-dependent resolution: nerve CN with resolution 1 (subdivided once), nerve DT with resolution 2, and the brain with resolution 0.

We use this data for the evaluation of the tissue-aware control of the quality. The results are shown in Fig. 16. The top three figures are the closeup of DT and CNP ($\lambda_{DT} = \lambda_{CNP} = 1.0$), the dihedral angle distribution of the tissue DT, and the dihedral distribution of the tissue CNP. The bottom three figures are the results as we fix λ_{CNP}, but reduce λ_{DT} to 0.25. The left two figures do not show a big difference, but the two middle figures clearly show the quality of DT improves from [13.6,76.1] to [15.1,80.6] because we pay more attention to the quality of DT. The two right figures do not show any big differences because we do not change



(a) Multi-label image (b) Coarse mesh (c) Final mesh

Fig. 14 Multi-tissue mesh generation for visible human data. (a) is the multi-label image. The coarse multi-tissue mesh (b) is generated with subdivision threshold 0.85. (c) is the deformed multi-tissue mesh. The numbers of source points and target points are 5828 and 26060, respectively.



(a) Wireframe view (b) Extracted two nerves

Fig. 15 (a) is the wireframe view of the three submeshes, and (b) is the extracted two nerves.

λ_{CNP}. Compared to MRI experiments, more time is needed (9 minutes) because more tissues are involved.

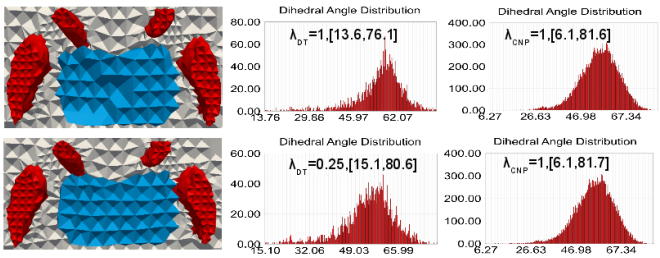


Fig. 16 Tissue-aware quality control. The two values in the bracket are minimum and maximum dihedral angles.

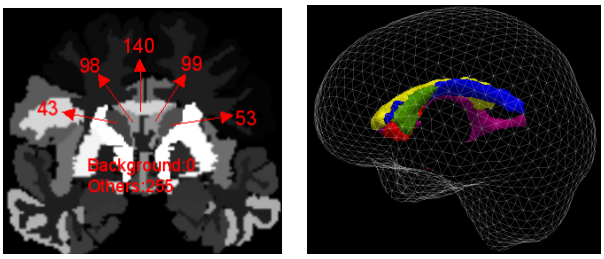
3.4 Brain atlas

We use the brain atlas[2] to evaluate the method on non-manifold surfaces. The multi-label image is shown in Fig. 17(a), and the final multi-tissue mesh, produced with the same trade-off parameters ($\lambda_1 = \lambda_2 = ...\lambda_6 = 1.0$), is shown in Fig. 17(b).

We zoom in the interfaces of these tissues to show the conformity in Fig. 18 in a different point of view from Fig. 13. Fig. 19 has three subfigures, showing the

[1] http://www.nlm.nih.gov/

[2] http://www.spl.Harvard.edu/publications/item/view/1265



(a) Brain atlas (b) Final multi-tissue mesh

Fig. 17 Multi-tissue mesh for brain atlas. Five tissues along with the rest of the brain (a) are discretized. 43: right caudata nucleus (RCN), 53: left caudata nucleus (LCN), 98: right anterior horn of lateral ventricle (RAHLV), 99: left anterior horn of lateral ventricle (LAHLV), 140: corpus callosum (CC). (b) is the final mesh. The numbers of source points and target points are 6225 and 39136, respectively.

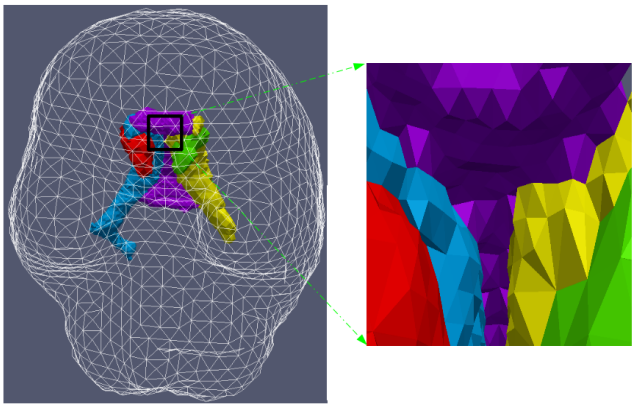


Fig. 18 Conformity of interfaces.

fidelity, tissue-dependent resolution, and quality, respectively. The fidelity part shows the comparison of the fidelity before PBR (left) and after PBR (right). The figure is generated by cutting through the mesh, and overlapping it with the same slice of the multi-label image. The black arrows point to the places where bigger improvement of the fidelity occurs. Compared to the inner structures, the brain shows bigger improvement of the fidelity. The reason is, compared to the inner structures, the brain has lower resolution and, therefore, lower fidelity. Since we do not pay more attention to the inner structures (the same λ_i for all tissues), the tissue with lower fidelity improves its fidelity more. The fidelity is evaluated using equation (17), and the measurements are listed in Table 3. In the resolution part, the mesh is cut through to show the tissue-dependent resolution. In the quality part, we present the distribution of the dihedral angle and aspect ratio under different trade-off parameters λ ($\lambda_1 = \lambda_2 = ...\lambda_6 = \lambda$). The values in brackets are the minimum and maximum values for the whole mesh. The values for each submesh are listed in Table 3. As we increase λ from 1.0 to 1.5, i.e paying less attention to the quality, the minimum dihedral angle reduces from 4.57 to 3.96, and the maximum aspect ratio increases from 8.80 to 15.83. It takes about 14 minutes to generate the final multi-tissue mesh.

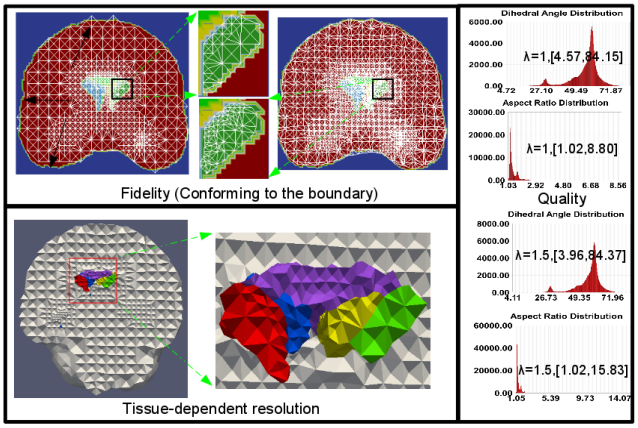


Fig. 19 The evaluation of the fidelity, tissue dependent resolution, and quality on the brain atlas.

A good quality mesh is characterized by the absence of slivers, i.e. tetrahedra with a very small dihedral angle, or aspect ratio close to 1. One observation from the quality part is the number of the tetrahedra with ratio around 1 increases from 20000 to 40000 even when we pay less attention to the quality (increase λ from 1 to 1.5). This can be explained by the fact that lots of tetrahedra happen to improve their quality as they are deformed to the boundary.

Table 3 Quantitative evaluation for the multi-tissue mesh on the brain atlas. The atlas is regularized using a spacing: $1mm \times 1mm \times 1mm$ and a size: $240 \times 240 \times 259$. The parameters are subdivision threshold=0.85 and $\lambda = 1.0$.

Nerve	Aspect ratio	Dihedral angle	Distance	#Tetras	#Nodes
RCN	[**1.03**,3.75]	[**13.36**,79.80]	0.80	2944	814
LCN	[**1.07**,3.01]	[**24.7**,72.60]	0.91	612	220
RAHLV	[**1.02**,6.84]	[**10.06**,79.12]	0.79	9480	2589
LAHLV	[**1.03**,4.07]	[**17.74**,78.40]	0.82	3849	1136
CC	[**1.03**,3.96]	[**13.56**,78.14]	0.82	14937	3766
Other	[**1.02**,8.80]	[**4.57**,84.15]	0.99	109466	21407

4 Conclusion

This paper presents a BCC-based multi-tissue mesh generation approach. This method inherits the advantages of BCC lattice mesh, and extends it to a multi-tissue mesher. To make submesh interfaces well posed for deformation and reach conformity, we design a label redistribution algorithm based on a predefined operation table. The proposed multi-tissue mesh generation

method can reach tissue-dependent resolution by using a red-green subdivision under the guide of a subdivision criterion. Two mesh deformation strategies are developed and compared with each other. The RPM method demonstrates its benefit on mesh fidelity, and the PBR methods demonstrates its benefit on mesh quality. The experiments on the synthetic data, clinic MRI, visible human, and brain atlas demonstrate the effectiveness of this method. Although the proposed multi-tissue mesher is only evaluated on brain images, it can be easily applied on other soft tissues by specifying suitable biomechanical parameters.

References

1. Bathe, K.: Finite Element Procedure. Prentice-Hall (1996)
2. Bierling, M.: Displacement estimation by hierarchical block matching. Proc. SPIE Vis. Comm. and Image Proc. **1001**, 942–951 (1988)
3. Boltcheva, D., Yvinec, M., Boissonnat, J.D.: Mesh generation from 3d multi-material images. In: MICCAI, vol. 5762, pp. 283–290 (2009)
4. Borman, S.: The expectation maximization algorithm: A short tutorial. unpublished paper available at http://www.seanborman.com/publications. Tech. rep. (2004)
5. Cheng, S.W., Dey, T.K., Edelsbrunner, H., Facello, M.A., Teng, S.H.: Silver exudation. J. ACM **47**(5), 883–904 (2000)
6. Clatz, O., Delingette, H., Talos, I.F., Golby, A., Kikinis, R., Jolesz, F., Ayache, N., Warfield, S.: Robust non-rigid registration to capture brain shift from intra-operative MRI. IEEE Trans. Med. Imag. **24**(11), 1417–1427 (2005)
7. College, F.D., Dellaert, F.: The expectation maximization algorithm. Tech. rep. (2002)
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, Series B **39**, 1–38 (1977)
9. Glazer, A.M., Burns, G.: Space Groups for Solid State Scientists. Academic Press, Inc., Philadelphia, PA, USA (1990). 2nd ed.
10. Haili, C., Rangarajan, A.: A new point matching algorithm for non-rigid registration. Comput. Vis. Image Underst. **89**(2-3), 114–141 (2003)
11. John, A.: R. a. fisher and the making of maximum likelihood 1912-1922. Statistical Science **12**(3), 162–176 (1997)
12. Kitware: http://www.itk.org/
13. Liu, Y., Yao, C., Zhou, L., Chrisochoides, N.: A point based non-rigid registration for tumor resection using IMRI. In: 2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 1217–1220 (2010)
14. Meyer, M., Whitaker, R., Kirby, R.M., Ledergerber, C., Pfister, H.: Particle-based sampling and meshing of surfaces in multimaterial volumes. IEEE Transactions on Visualization and Computer Graphics **14(6)**, 1372–1379 (2008)
15. Molino, N., Bridson, R., Teran, J., Fedkiw, R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In: International Meshing Roundtable, pp. 103–114 (2003)
16. Papademetris, X., Jackowski, A.P., Schultz, R.T., Staib, L.H., Duncan, J.S.: Computing 3d non-rigid brain registration using extended robust point matching for composite multisubject fmri analysis. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003, vol. 2879/2003, pp. 788–795 (2003)
17. PETSc: http://www.mcs.anl.gov/petsc/petsc-as/
18. Pons, J.P., Ségonne, F., Boissonnat, J.D., Rineau, L., Yvinec, M., Keriven, R.: High-quality consistent meshing of multi-label datasets. In: Information Processing in Medical Imaging, vol. 4584, pp. 198–210 (2007)
19. Rousseeuw, J.P., Driessen, K.: Computing lts regression for large data sets. Data Min. Knowl. Discov. **12**(1), 29–45 (2006)
20. Rousseeuw, P.J., Leroy, A.M.: Robust regression and outlier detection. John Wiley & Sons, Inc., New York, NY, USA (1987)
21. Wittek, A., Miller, K., Kikinis, R., Warfield, S.K.: Patient-specific model of brain deformation: Application to medical image registration. Journal of biomechanics **40**(4), 919–929 (2007)
22. Y., S.: Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2003)
23. Yang, X., Zhang, Z., Zhou, P.: Local elastic registration of multimodal medical image using robust point matching and compact support rbf. In: 2008 International Conference on BioMedical Engineering and Informatics, vol. 2, pp. 113–117 (2008)
24. Zhang, Y., Hughes, T.J., Bajaj, C.L.: An automatic 3d mesh generation method for domains with multiple materials. Computer Methods in Applied Mechanics and Engineering **199**(5-8), 405 – 415 (2010)