Tetrahedral Mesh Generation for Non-Rigid Registration of Brain MRI: Analysis of the Requirements and Evaluation of Solutions

Andriy Fedorov and Nikos Chrisochoides

Center for Real-Time Computing, College of William and Mary, Williamsburg, VA
{fedorov,nikos}@cs.wm.edu

Summary. The application we target in this paper is the registration of pre-operative Magnetic Resonance Imaging with the data acquired intra-operatively during image-guided neurosurgery. The objective of this application is improved tracking of tumor boundaries and surrounding brain structures during open skull tumor resection. We focus on a validated, physics-based non-rigid registration approach, which has been used in clinical studies for the last three years. This approach requires tetrahedral tessellation of the brain volume for biomechanical model construction. The analysis of the requirements and available methods to construct such a discretization is the objective of our paper.

The paper presents a number of practical contributions. First, we survey the proposed approaches to tetrahedral mesh generation from medical image data. Second, we analyze the application-specific requirements to mesh generation. Third, we describe an end-to-end procedure of tetrahedral meshing for this application using off-the-shelf non-commercial software. Finally, we compare the performance of the considered mesh generation tools in the application context using generic and application-specific quantitative measures.

1 Introduction

Mesh generation for medical imaging applications has been attracting a lot of attention in the mesh generation community. Finite Element Method (FEM) is essential in modeling tissue deformation for these applications, therefore necessitating research and development of new mesh generation methods and tools. The existing imaging modalities, such as Magnetic Resonance Imaging (MRI), provide only limited knowledge about the internal organs. The lack of precise geometric models and the need to construct volume tessellations from the image data is a very practical limitation, which complicates the use of existing off-the-shelf meshing tools. Application-specific requirements make the problem even more challenging. In practice, there is no single widely accepted method to address the mesh generation needs of all applications.

In this paper we focus on tetrahedral mesh generation for physics-based Non-Rigid Registration (NRR) of brain MRI during Image-Guided Neurosurgery (IGNS). Specifically, we focus on the NRR approach developed by Clatz et al. [1], and subsequently validated in the clinical setting by Archip et al. [2]. The method has been studied at Brigham and Women's Hospital (Harvard Medical School, Boston) for the last three years.

The main goal in neurosurgery is maximum removal of the tumor tissue with the minimum damage to the healthy brain structures. It is very difficult to identify boundaries of certain tumors, e.g., gliomas, with the naked eye. Instead, pre-operative imaging is typically used to precisely locate the tumor, and the neighboring life-critical structures. Brain shift is often unavoidable during the tumor resection, making the pre-operative images invalid. Intra-operative images acquired periodically during the tumor resection have lower quality, and usually cannot substitute the pre-operative data. NRR is used to align the pre-operative MRI with the intra-operative scans. The specific NRR method we study uses a patient-specific bio-mechanical model to facilitate the brain shift estimation. Tetrahedral tessellation of the brain volume (more specifically, skull Intra-Cranial Cavity (ICC)) is essential for this application.

Our paper is the continuation of an earlier study [3]. We consider the complete process of mesh generation from the segmented image, and discuss specific application requirements derived from the established registration method. Our contributions are the summary of the state of the art methods for constructing tetrahedral meshes from images, and a practical evaluation of the existing off-the-shelf meshing tools. Our evaluation is based on the number of qualitative and quantitative metrics, which allow to compare the studied methods in the context of the FEM computations that facilitate IGNS.

2 Physics-based Non-Rigid Registration of Brain MRI

2.1 Formulation

The objective of image registration is to determine the transformation that aligns features in one (floating) image with the features in another (target) image. Image registration is a fundamental problem in medical image processing. The reader is referred to the survey by Hill et al. [4] for a thorough review on this topic. Non-rigid image registration is used when the imaged structure is subject to a non-affine transformation. Image registration methods are usually tailored to a specific clinical application. The specific clinical application we target is IGNS facilitated by the specific physics-based NRR approach [1].

The NRR method in [1] consists of the following steps. First, a sparse set of mathematical landmarks, which we call *registration points*, is identified within the pre-operative image of the brain volume. Once the intra-operative image is available, the time-critical part of the computation is initiated. The deformation is estimated at each registration point using block matching between

the pre- and intra-operative images (in the context of registration, these are the floating and target images, respectively). Block matching results inherently contain incorrect matches (outliers). Because of outliers, the deformation field cannot be derived by interpolating displacements at the registration points. Mechanical energy of the deforming mesh is used to regularize the displacements, i.e., smooth the impact of outliers on the solution:

$$W = (\mathbf{HU} - \mathbf{D})^T \mathbf{S}(\mathbf{HU} - \mathbf{D}) + \mathbf{U}^T \mathbf{KU}.$$

Here $\mathbf{U}$ is the vector of displacements at the mesh nodes, $\mathbf{K}$ is the stiffness matrix [5], $\mathbf{H}$ is the interpolation matrix from the tetrahedra vertices to the registration points inside a tetrahedron, $\mathbf{D}$ is the vector of displacements recovered at the registration points, and $\mathbf{S}$ is the block-diagonal positive semi-definite matrix which captures confidence in the corresponding entry of $\mathbf{D}$ [1].

The outliers are found by iteratively comparing the displacements at the registration points with those interpolated from the mesh vertices, and discarding the matches with the largest error magnitude. The displacements at the mesh vertices are estimated using the following formulation:

$$\mathbf{F}_0 = 0,\ \mathbf{U}_i = [\mathbf{H}^T \mathbf{S}_i \mathbf{H} + \mathbf{K}]^{-1} [\mathbf{H}^T \mathbf{S}_i \mathbf{D} + \mathbf{F}_{i-1}],\ \mathbf{F}_i = \mathbf{KU}_i. \tag{1}$$

At each iteration, the registration points are ordered by the absolute value of the error $\|\mathbf{D}_i - \mathbf{HU}_i\|$ weighted by the 3×3 image structure tensor $\mathbf{T}_i$ at the registration point. The weight is used because block matching can only recover displacements in the directions orthogonal to the edges in the image [1]. Outlier registration points are selected as a pre-defined, e.g., 5%, number of registration points with the largest error magnitude.

2.2 Finite element mesh generation

Tetrahedral mesh has a dual role in the formulation. First, it is used to find the stiffness matrix in the mechanical energy component. Second, it allows to regularize, or smooth, the displacements recovered by block matching locally within the mesh vertex neighborhood. The displacement recovered at each of the registration points within the mesh vertex cell complex affects the displacement at the corresponding mesh vertex. Therefore, it is important to maintain the empirically obtained ratio between the number of mesh vertices and the number of registration points under 0.1 (at least 10 registration points per mesh vertex) [1]. Ideally, this ratio should be maintained within each mesh vertex cell complex. We define the cell complex as a set of mesh tetrahedra incident on a mesh k-cell, e.g., mesh vertex is a 0-cell, and mesh edge 1-cell. The matrix $\mathbf{H}^T\mathbf{SH}$ has a non-zero 3×3 entry for each mesh vertex and edge with the cell complexes containing registration points. The corresponding submatrices can be expressed as the sum over the registration points in a cell. For example, the diagonal 3×3 sub-matrix that corresponds to the mesh vertex

$\mathbf{v}_i$ can be calculated as the following summation over the registration points in the cell complex $\mathcal{S}$ of $\mathbf{v}_i$:

$$[\mathbf{H}^T\mathbf{S}\mathbf{H}]_i = \sum_{\forall \mathcal{T} \in \mathcal{S}(\mathbf{v}_i)} \sum_{\forall k \in \mathcal{T}} h^{\mathcal{T}}_{\mathbf{v}_i}(k)^2 \frac{\mathrm{tr}(\mathbf{K})}{np} c(k)\mathbf{T}(k). \quad (2)$$

Here, $h^{\mathcal{T}}_{\mathbf{v}_i}(k)$ is the barycentric coordinate of the kth registration point with respect to $\mathbf{v}_i$ in the containing tetrahedron $\mathcal{T}$, n is the number of mesh vertices, p is the number of the registration points, $c(k)$ is the correlation coefficient from block matching, and $\mathbf{T}(k)$ is the image structure tensor at k.

The result of non-rigid registration is a deformation field, which defines deformation vector at each point in the image. That vector is computed by interpolating the displacements at the mesh vertices after the last iteration [5]:

$$\forall \mathbf{x} \in \mathcal{T} : \mathbf{u}(\mathbf{x}) = \sum_{j=0}^{3} h^{\mathcal{T}}_{\mathbf{v}_j}(\mathbf{x})\mathbf{u}(\mathbf{v}_i). \quad (3)$$

Note that the iterative procedure in Equation 1 requires solving the linear system of equations at each iteration within the time-critical part of the computation. The size of this system depends on the number of degrees of freedom in the mesh model. Therefore, it is important to minimize the size of the mesh as much as possible without sacrificing the accuracy of the solution.

2.3 Application-specific requirements to mesh generation

Based on the registration formulation, we can derive the following application-specific requirements to mesh generation:

R1 *Equi-distribution of the registration points w.r.t. mesh vertex cells*: small number of registration points (e.g., less than 10, but greater than 0) within the vertex cell complex makes the formulation more sensitive to outliers and introduces additional displacement error [1].

R2 *Minimization of the approximation error at registration points*: error of the displacements recovered at registration points can be reduced locally by using smaller mesh elements [6].

R3 *Prevention of tetrahedron inversion during mesh deformation*: while the interpolation error shown in Equation 3 does not depend on the tetrahedron shape, inversion or collapse of a tetrahedron will result in an unrealistic deformation field, e.g., points inside the different tetrahedra can map to the same image location. We can attempt to remedy this problem by adjusting theatetrahedra size locally according to the expected deformation.

Adaptive refinement of the mesh following the simulation, if necessary, is outside the scope of this paper.

3 Image-to-Mesh Conversion

3.1 Formulation and generic requirements to mesh generation

Following the notation of Hill et al. [4], we define the image domain Ω as the overlap between the bounded continuous set $\tilde{\Omega}$ (image field of view) and the infinite discrete sampling grid Γ_ς, characterized by the anisotropic sample spacing $\varsigma = (\varsigma^x, \varsigma^y, \varsigma^z)$, $\Omega = \tilde{\Omega} \cap \Gamma_\varsigma$. 3D image is usually acquired as a sequence of slice scans, where sampling spacing is square in the slice plane, with thicker dimension along the direction of slice acquisition. Voxel is an orthogonal parallelepiped-shaped region of the image field of view centered at the sampling grid point. Its dimensions are defined by the grid spacing ς. For the considered application, image A is a mapping of points in the image domain $\mathbb{R}^3$ to $\mathbb{R}$, $A : \mathbf{x} \in \Omega \mapsto A(\mathbf{x})$. In this paper we are mostly concerned with the medical applications that provide a *segmentation* of the object of interest. The output of segmentation is a binary image, i.e., $A(\mathbf{x}) \in \{0, 1\}$, with the subset $\{\mathbf{x} \in \Omega | A(\mathbf{x}) = 1\}$ corresponding to the voxels located inside the object. Let Σ be the surface that separates zero and non-zero voxels of this binary image. The surface Σ is defined implicitly, as we only know whether a given voxel is inside or outside the object.

The objective of mesh generation for FEM computations from the binary image data is to construct a conforming tetrahedral mesh $\mathcal{M} = (\mathcal{V}, \mathcal{T})$, which satisfies the following *generic requirements*:

R4 The mesh boundary (triangulation) should be close to Σ.
R5 Mesh size should be minimized to reduce the computational costs.
R6 Mesh elements should not have small angles [6].

These generic requirements have been in the focus of mesh generation community for decades. The application-specific requirements are usually addressed by constructing a customized mesh sizing function, or developing customized mesh generation methods. In this paper we explore the first approach. Therefore, the ability to accept a user-defined sizing function is an essential feature for a mesh generation method to be considered for our application.

3.2 Related work

An intrinsic difficulty of generating meshes from the binary image data is the processing and recovery of the object geometry. General-purpose mesh generators (for solid and geometric modeling applications) expect that the object boundary is parametrized, i.e., it is defined by means of constructive solid geometry primitives, or explicitly (e.g., through the boundary discretization, as a collection of patches). Therefore, in order to convert the binary image into a tetrahedral mesh, one can either (1) recover the parametrized object surface followed by a conventional mesh generation technique, or (2) use a mesh generation method, which operates directly on the binary image. The

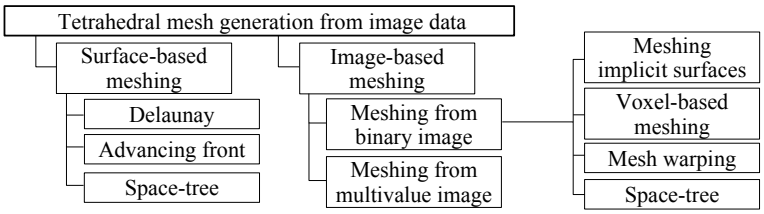

Fig. 1. A hierarchy of the studied approaches to mesh generation.

reader is referred to the survey by Owen [7] for a discussion of the classical approaches to volume meshing. Next, we overview some of the methods designed to construct tetrahedral discretizations directly from the image data.

A step which precedes geometry processing is segmentation of the structure of interest from the multivalued image. Segmentation is a fundamental problem in medical image processing [8], and is outside the scope of this paper. We assume that the segmentated ICC is provided by the application.

The objective of the surface recovery step is to construct an explicit representation of Σ. The methods which recover piecewise-linear surface approximation and provide surface triangulation are most practical, as this is the input most mesh generation methods expect. It is desired for the triangulated surface (1) to have the same topology as Σ, (2) to be sufficiently close to Σ, and (3) to have guaranteed quality of the triangles in the surface discretization, as defined by the triangle aspect ratio, while (4) minimizing the number of triangles. The first two requirements are important for the accurate representation of the object, while the other two are essential in order to satisfy the generic requirements to volume mesh generation.

A straightforward approach to recover iso-surface is by means of the Marching Cubes (MC) algorithm [9]. However, the original version of this algorithm may produce a triangulation with topological problems. Another practical problem is the inability of the MC algorithm to generate adaptive surface triangulations. The surface produced by MC has "jagged" artifacts because of voxel sampling, which may create subsequent problems with the simulation [10]. Surface mesh simplification, or remeshing, is often a necessary post-processing step when MC is used [11, 12].

Parametrized surface representation can be recovered by more advanced techniques, which provide theoretical guarantees about the recovered surface. Since the seminal work of Amenta and Bern [13], a number of algorithms have been proposed [14, 15, 16]. These methods differ in their performance, robustness to noise and theoretical guarantees.

A number of approaches to volume mesh construction were proposed, that are capable to operate on image date directly. We separate such methods into the following two categories.

The methods from the first category do not require segmentation and create meshes from the multi-value image data [17, 18]. The assumption is

usually made that the pixels which correspond to the same tissue have similar intensity, and the object boundary can be defined by some isosurface value. Therefore, these methods attempt to minimize the error of approximating this isosurface, while maintaining good quality of the mesh tetrahedra. Such approach to mesh generation is very practical for volume rendering and certain FEM applications. However, brain segmentation is one of the very challenging problems in image segmentation, which cannot be solved by thresholding only [19, 8]. Direct isosurface-based meshing of the brain volume from the multivalued image may lead to large errors in the surface recovery.

The second category of the image-based mesh generation methods operate on binary images, produced by specialized image segmentation algorithms. Among the methods in this category we separate four groups.

The methods in the first group attempt to recover the surface of the object at the level of voxel resolution [20, 21, 22]. Therefore, we call those techniques *voxel-based meshing*. The sizes of surface triangles in the meshes constructed using this approach are comparable with the size of voxels, i.e., the surface triangulation is not adaptive. The use of such techniques is problematic for our application, because the control over the element size is limited due to the fixed high resolution of the surface discretization.

A large number of methods that are based on space-tree decompositions have been proposed recently for meshing binary images. Conceptually, this approach has a long history in classical mesh generation. Yerry and Shephard [23] were some of the first to present an octree-based approach to 3-d mesh generation. Mitchell and Vavasis [24] describe a quadtree-based algorithm with theoretical bounds on the mesh size. These ideas have been adopted to construct tetrahedral meshes from binary images [25, 26, 27]. Such methods recover the surface by finding the points of intersection of the adaptive space-tree with the surface of the object defined as a binary image. Mesh quality near the surface can be compromised, as the newly inserted mesh nodes can be arbitrarily close to the existing nodes. Mesh optimization [28] is commonly used as a post-processing step. In practice, the methods from this group are well-suited for meshing binary images, and were shown to be quite effective for a number of medical applications. Some of these methods were designed and evaluated on the segmented brain MRI data [25, 26, 27], which is the geometry used in our application. However, the control over the customized element sizing is usually very limited, and has not been evaluated previously.

The methods based on surface matching use a template volume mesh, which is warped to match the surface of the modeled object [29]. While the advantages of this approach are good surface fidelity, control over the mesh size and high speed, the quality of the elements undergoing deformation during warping can be compromised. Mesh optimization is a commonly used post-processing step for the methods in this group. This concept is most suitable for meshing objects that have very similar geometries. The geometry of ICC is quite similar between different subjects. However, the mesh element sizing depends on criteria R1-3, which are case-specific. Therefore, it is not feasible to

construct a single template mesh for our application to satisfy case-dependent distributions.

Representation of the object as a binary image conveniently lends itself to the construction of an implicit function describing the object surface. Implicit function is a mapping $\phi : \mathbb{R}^3 \mapsto \mathbb{R}$, and the object surface is defined as the kth level set of this function, $\phi(\mathbf{x}) = k$. An approximation of the implicit function representing the object surface can be easily obtained by computing the distance transform on the binary image [27], and using the zero level set as a surface definition. A number of volume mesh generation methods have been introduced recently to mesh implicitly defined surfaces [30, 31].

Overall, we observe that a great variety of methods for tetrahedral meshing of binary images have been developed. Most of these methods were proposed and evaluated in the context of their fitness to a specific application. Little or no attention is usually paid to the comparison of the newly proposed techniques with the existing methods, and few implementations are available to conduct such an evaluation by external groups. Also, most of the effort is usually directed to developing a method that delivers good practical results, and not on establishing theoretical guarantees about the produced meshes. The lack of such guarantees makes it even more difficult to select the most appropriate algorithm from the range of seemingly similar methods.

With the specific application we target, our goal is first to derive the precise requirements to mesh generation. These requirements can then be used to customize and evaluate readily available, established methods to address the problem of mesh generation for the NRR application. The results of such evaluation can be used next to identify problems within the existing approaches and justify the development of new mesh generation methods for this application. However, before such necessity is justified, we believe the possibility of using existing off-the-shelf tools must be carefully examined.

4 Methodology

4.1 Mesh generation tools

We evaluate three conceptually different approaches to constructing tetrahedral meshes from binary images. Common to all these methods is their ability to adjust the mesh element size locally according to the value of sizing function, or local refinement rules.

Tetgen is an implementation of the Delaunay mesh generation and refinement algorithm by Si [32]. This method is accompanied by the proof of termination and bounds on the circumradius-to-shortest edge ratio. The implementation has a number of very practical features: it accepts user-defined sizing function, the exterior boundaries are refined simultaneously with the volume and "... are never over-refined" [32]. The size of the mesh can be

controlled by the *alpha* parameters, which are not part of the basic Delaunay refinement [32]. We used *Tetgen* version 1.4.2. The implementation works with the input surface defined as a piecewise linear complex (PLC).

NETGEN is an advancing front mesh generator developed by Shöberl [33]. We used version 4.5rc2 of the code. The implementation can be used to construct both surface triangulations of the parametrized surfaces, and adaptive volume tetrahedralizations. It gives the user some control over the mesh grading, its implementation is accompanied by a GUI environment, and the acceptable inputs include triangulated surface of the domain.

RGM is a space-tree based mesher we presented earlier [27] that works directly on binary image data. Our implementation is based on the algorithm of Molino et al. [30], which builds a mesh from the implicit definition of the domain. *RGM* is designed to work directly with the binary image, and we use the *Mesquite* [34] mesh optimization library to improve the mesh quality following the surface recovery. We used slightly modified version compared to the code available online. Specifically, we use *Mesquite* instead of *GRUMMP* for mesh optimization, and implement a custom subdivision rule, as described in the next section.

Both *Tetgen* and *NETGEN* require parametrized surface representation. We use implicit surface meshing method by Boissonnat and Oudot [16] implemented in the Surface mesh generation package of CGAL [35]. The implementation generates adaptive triangular surface directly from the binary image. The guarantees of approximation accuracy and surface quality are provided. All the interaction with the image data is done via the Insight Toolkit (ITK) [36][1].

4.2 Adaptive mesh generation

A sizing function $H(p) > 0$ specifies the desired length of edges at point p [32]. It can be defined analytically, or, more often, its values are prescribed at the vertices of a background mesh. The value of sizing at the non-vertex locations can then be derived by interpolation. Both *Tetgen* and *NETGEN* accept background mesh to control local mesh size. We use the sizing function defined at the vertices of a background mesh to address the application-specific requirements to mesh generation.

We use the same CGAL-recovered surface mesh for both background mesh and the mesh used in FEM calculations. The background mesh is built using *Tetgen*, with the small uniform bound on the tetrahedron volume. The sizing value at each mesh vertex was initialized with the distance to the kth

[1] Off-the-shelf software tools we used (links valid as of July 15, 2008): *Tetgen*: `http://tetgen.berlios.de/`, *NETGEN*: `http://www.hpfem.jku.at/netgen/`, *RGM*: `http://www.na-mic.org/svn/NAMICSandBox/trunk/TetrahedralMeshGeneration/`, CGAL: `http://cgal.org/`, ITK: `http://itk.org/`, VTK: `http://vtk.org/`, Paraview: `http://paraview.org`.

registration point closest to it, to reflect the density of the registration point distribution. The idea here is that the shape of a perfect mesh vertex cell complex is close to a ball with the radius prescribed by the background mesh. We use CGAL [35] k-neighbor search to find the k closest registration points and the distance to the furthest point for each background mesh vertex. The process of background mesh initialization is parametrized by the number of closest registration points k. Our goal was to have around 30 registration points in the cell complex of each vertex. However, based on the experimental results, the mean value in distribution of the registration points both for *Tetgen* and *NETGEN* was not approaching the desired bound when we set $k = 30$. Experimentally, we arrived at a result that the best distribution is obtained by using larger values for k (we used $k = 100$), and adjusting the *Tetgen* mesh by reducing the *alpha* parameters of the implementation, see [32]. The *NETGEN* mesh was constructed using the same background mesh, but the sizing values were scaled down by constant to have similar number of nodes compared to the adaptive *Tetgen* mesh.

The sizing of the adaptive mesh constructed with *RGM* was controlled by a custom subdivision function. This function is called for each tetrahedron during the refinement at each mesh resolution, and returns *true* if the tetrahedron requires subdivision. We calculate the number of registration points inside the cells of the four tetrahedron vertices. The tetrahedron is refined if the number of the registration points in each vertex cell exceeds parameter n. We used two subdivision resolutions in all cases. The process of adaptive mesh construction with *RGM* is parametrized by the value of constant n, and the spacing of the initial lattice, see [27]. These parameters were experimentally chosen so that the size of the adaptive mesh (the number of mesh vertices) is approximately the same as the size of the adaptive *Tetgen* mesh.

4.3 Evaluation

We compare two sets of meshes constructed with each of the evaluated methods, i.e., with and without using custom mesh sizing. The meshes are used to register real MRI subject to synthetic deformations. We use synthetically deformed images, because the true deformation field (the ground truth deformation) cannot be recovered in the real IGNS cases, and the registration accuracy cannot be evaluated at an arbitrary image location. The ability to assess the registration error is essential for our evaluation. We provide the details on the generation of the synthetic deformation field in [37].

We try to create meshes with the similar number of vertices within each group (uniform and adaptive), and compare them using a set of quantitative metrics. There are two groups of metrics we use. The first group includes the mesh properties, which can be directly optimized during the process of mesh construction. These include element shape and surface approximation accuracy. We assess the element shape by the minimum dihedral angle for each tetrahedron of the mesh. Surface approximation accuracy is evaluated

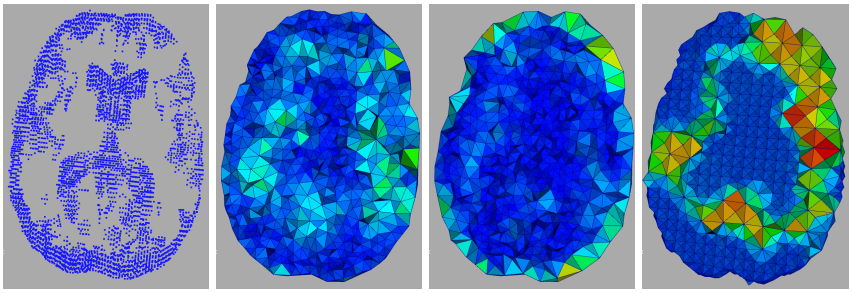

Fig. 2. Left to right: selected registration points, and adaptive tetrahedral mesh cuts (same slice) generated with *Tetgen*, *NETGEN*, and *RGM*. Tetrahedra are colored according to their volume, from blue (smallest) to red (largest).

as the percentage of the registration points covered by the mesh, which is a practical measure for the NRR application.

The second group includes quantitative metrics, which cannot be directly optimized by the existing mesh generation methods. Let $\bar{\mathbf{D}}_i$ be the ground truth displacement at the registration point i, which aligns given point in the floating image with the corresponding point in the target image. This value is known to us, because the true deformation field is synthetic. We define the following application-specific quantitative metrics:

1. *Approximation error* at a registration point is defined as $\|\mathbf{D}_i - \mathbf{H}\mathbf{U}_i\|$. We assess the accuracy of approximation by the percentage of the registration points, where the magnitude of this error exceeds 1.0. We call those registration points "error points", while reporting results (errors below this threshold are in the sub-voxel range).
2. *Outlier detection sensitivity*, defined as the ratio of the true outliers within the discarded registration points to the total number of the discarded registration points. The true outlier is defined as a registration point, where $\|\mathbf{T}_i\mathbf{D}_i - \bar{\mathbf{D}}_i\| > 1.0$.
3. *RMS of the absolute error* at the registration points, absolute error being defined as $\|\mathbf{H}\mathbf{U}_i - \bar{\mathbf{D}}_i\|$.
4. *Distribution of the registration points* with respect to mesh vertices.

Note that the goal of the study was not to tune the NRR parameters to get the optimal registration results. We attempt to perform a controlled study of the impact of mesh generation on the NRR performance, while keeping fixed the other parameters that can influence registration accuracy.

5 Results

We constructed synthetic deformation fields and performed NRR on the MRI scans 1 through 3 in the set of 18 images available from the Internet Brain

Table 1. Registration points, true outliers, and their coverage by the mesh surfaces.

case id	reg. points	outliers,%	reg. points inside,%		outliers inside,%	
			CGAL	RGM	CGAL	RGM
IBSR01	56447	7.8%	95.6%	91.1%	6.9%	5.5%
IBSR02	57526	16.2%	94.9%	90.8%	14.3%	12.4%
IBSR03	46525	18.8%	95.3%	90.5%	16.0%	13.1%

Table 2. Min/average/max of registration points per mesh vertex cell complex.

case id	uniform-graded meshes			adaptive meshes		
	Tetgen	NETGEN	RGM	Tetgen	NETGEN	RGM
IBSR01	0/135/460	0/137/715	0/130/390	0/36/111	0/36/432	0/33/277
IBSR02	0/131/479	0/136/667	0/124/374	0/35/118	0/35/444	0/30/168
IBSR03	0/127/408	0/126/611	0/121/345	0/27/87	0/27/320	0/28/254

Segmentation Repository (IBSR)[2]. The synthetic deformations were generated with 20 "knots" in the deformation grid on average [37]. The deformation magnitude at each "knot" was under 5% of the brain size to maintain the validity of the linear elastic physical model [5, 1].

The parameters used for CGAL surface mesh generation [35] were: angular bound 30°, surface radius and distance bounds 10.0, surface precision bound 0.001. Each method was used to construct two meshes for each registration case. The prescribed element size was uniform throughout the volume of the first mesh. The second mesh was constructed to adapt the element size according to the sizing function designed in Section 4.2. We adjusted the implementation-specific parameters to have the uniform and adaptive meshes with approximately 1.5k and 6k vertices, respectively.

Figure 2 shows cross-sections of the adaptive meshes. The adaptive meshes generated with *NETGEN* have a layer of relatively large elements near the surface of the mesh. This is explained by the nature of the Advancing Front algorithm, which does not insert new points on the triangulated surface. We explored the option to use the CGAL triangulation as the support surface, and instructed *NETGEN* to construct a new triangulation to respect the prescribed element sizing. However, the re-triangulated surfaces contained small triangles, which did not obey the prescribed edge sizing.

None of the meshes contained sliver elements. We observed, that the minimum dihedral angle was the largest, 14°, in the *NETGEN*-generated meshes. The values of this metric for *Tetgen* and *RGM* were 8° and 5° respectively.

The synthetic registration cases differ in the number of registration points, and in the number of true outliers. Moreover, because of the differences in

[2]The MR brain data sets were provided by the Center for Morphometric Analysis at Massachusetts General Hospital and are available at `http://www.cma.mgh.harvard.edu/ibsr/`.

mesh surfaces recovered by CGAL and *RGM* respectively, different percentage of those points are located inside the mesh domain, as summarized in Table 1. This is an important observation, because the percentage of outliers impacts the registration error, which is also used in the evaluation. Also, due to the lower precision of surface approximation by *RGM*, about 4% fewer registration points and 1% to 3% fewer outliers are located within the *RGM*-generated meshes. The distribution of outliers is non-uniform, and many of them are located close to the surface.

Based on the results presented in Table 2 and Figure 3, the use of custom mesh sizing can significantly improve the distribution of registration points compared to uniform-sized meshes. In conjunction with the sizing function, *Tetgen* achieves the best distribution results overall. Although the average values for the distributions are similar for all meshes, *Tetgen* meshes have lower maximum values and better distribution: the distribution curve approaches normal distribution with the mean close to the desired number ($k = 30$). The advantage of *Tetgen* over *NETGEN* is that mesh points can be inserted at arbitrary locations on the surface during refinement. *RGM* is limited even more than *NETGEN*, as new points can be inserted only at the periodic predefined locations, based on the initial lattice structure. Nevertheless, the distributions in *RGM* meshes are consistently better compared with *NETGEN*. This might be caused by large elements near the surface of the *NETGEN* meshes.

Note that empty vertex cells do not pose a problem. The corresponding mesh nodes will move following the neighboring vertices during registration. Problems can be caused by few registration points (the contribution of outliers is not smoothed by the correctly recovered displacements), or by very large number of registration points in the cell (increased approximation error).

The non-rigid registration was performed with the default parameters suggested by Clatz et al. [1]. The quantitative metrics that are not directly optimized by mesh generation are summarized in Table 3. The approximation accuracy is consistently improved for all mesh generation methods when the refined meshes are used. However, this is the only metric that is clearly con-

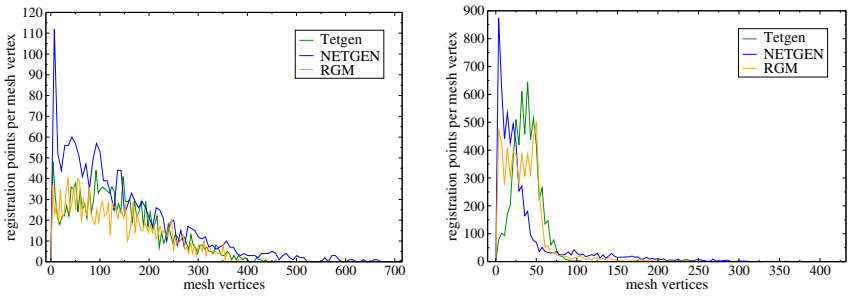

Fig. 3. Distribution of the number of registration points per mesh vertex: uniform-graded meshes (left) and adaptive refined meshes (right); IBSR01.

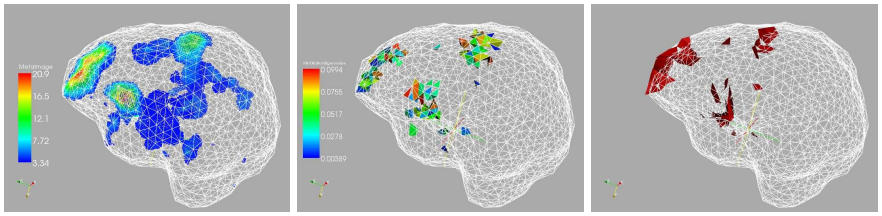

Fig. 4. Left: Image voxels with the largest values of the error with respect to ground truth. Center: mesh elements with the minimum eigenvalue of the dilation matrix below 0.1. Right: inverted mesh elements after NRR. IBSR02, *Tetgen* adaptive mesh, wireframe shows the undeformed mesh.

Table 3. Application-specific metrics for the evaluated meshes.

case id	metric	uniform size meshes			adaptive meshes		
		Tetgen	NETGEN	RGM	Tetgen	NETGEN	RGM
IBSR01	mesh points	1617	1596	1607	6044	6020	6209
	error points	2.4%	2.6%	1.9%	1.6%	1.8%	1.1%
	sensitivity	53.5%	52.6%	43.8%	52.3%	52.5%	42.8%
	RMS error	1.60	1.62	1.61	1.69	1.65	1.60
IBSR02	mesh points	1682	1617	1696	6166	6255	6993
	error points	6.7%	7.1%	5.9%	4.9%	5.8%	4.3%
	sensitivity	64.7%	64.4%	59.4%	63%	64.7%	60.5%
	RMS error	1.98	1.92	1.82	2.55	2.27	2.23
IBSR03	mesh points	1410	1413	1404	6631	6503	6033
	error points	7.7%	8.1%	6.3%	4.8%	6.4%	4.5%
	sensitivity	72.9%	71.2%	65.7%	70.7%	74.6%	69.3%
	RMS error	2.61	2.53	2.25	3.52	3.08	3.05

nected with the size of the elements of the mesh. We do not observe improvement in either outlier detection sensitivity or RMS of the registration error. On the contrary, RMS error is increasing in the refined meshes. We suggest that there are two major reasons why this may be the case.

First, the adaptively refined meshes are more susceptible to the element inversion during NRR. The areas of the high error in the deformation field recovered by NRR are spatially co-located with the areas of the mesh, where tetrahedra invert or become highly skewed. We use the tetrahedron measure proposed by Baker [38] and identify highly skewed mesh elements by the minimum eigenvalue of the element dilation matrix. Figure 4 shows spatial correlation of the locations of skewed and inverted elements with the areas of image with the largest registration error.

Second, refined meshes decrease the error of approximating displacements recovered by block matching both at the outlier and non-outlier registration points. Ideally, the mesh should be constructed in such a way that in each mesh vertex cell the number of outliers is less than the number of correct dis-

placements. In general outliers are distributed non-uniformly. As we decrease the size of the mesh elements, it becomes more likely that the registration points inside some cells will be dominated by the outliers, causing higher error with respect to the true deformation.

We tried to resolve the first problem by modifying the sizing function to reflect the deformation magnitude averaged over the k closest registration points. The parameters for CGAL surface mesh recovery were chosen according to the maximum of the averaged deformation magnitude near the object surface. None of the evaluated meshers was able to follow the prescribed sizing distribution closely. Both *Tetgen* and *NETGEN* created large tetrahedra near the mesh surface. We cannot attempt to improve the fitness of the mesh to the sizing function by reducing the default values of the *alpha* parameters, as we have done for the meshes evaluated previously. The *alpha* parameters control the bound on the shortest edge length at a mesh point, see Lemma 1 by Si [32]. Their reduction introduces small volume elements. The construction of meshes that adapt to the degree of deformation requires further study.

6 Discussion

In this section we summarize our quantitative and qualitative analysis for each of the six major requirements identified for the NRR application.

R1: Equi-distribution of the registration points. We were not able to achieve equi-distribution, which may be an NP-hard problem. However, in all cases, the use of custom sizing function significantly improved that distribution. The mean values for the number of registration points were close for all meshes. In all test cases, meshes constructed with *Tetgen* had lower values of the maximum number of registration points and better distribution overall.

R2: Reduction of the approximation error at registration points. In the general case, approximation error can be reduced by refining the mesh. This is observed in Table 3 when comparing uniform and adaptive meshes. Further refinement is problematic, as it would violate the requirement R1. When comparing the meshes with respect to the number of error points, we observe that this number is consistently lower for *Tetgen* vs. *NETGEN*. Because *Tetgen* better follows the values of sizing, it creates smaller elements in the areas with high registration point density. Therefore, the approximation error will also be reduced for the points in those areas. *RGM* has consistently the lowest number of error points. This can be explained, because there are fewer registration points located inside the mesh, while the mesh sizes are comparable with the *Tetgen* and *NETGEN* counterpart meshes.

R3: Prevention of tetrahedra inversion. Tetrahedra inversion can be prevented by increasing the size of the mesh elements. This goes contrary to what is required by the requirements R1 and R2. Moreover, it is not clear what should be the optimum element size to avoid inversion. The balancing of this requirements with R1 and R2 is the subject of future work.

R4: Object surface approximation accuracy. CGAL implicit surface recovery procedure was used to construct surface triangulations for *Tetgen* and *NETGEN*. This allows to control over both the surface approximation accuracy and the angles in surface triangulation. The distribution of registration points is highly non-uniform, and is more dense in the areas of the prominent features of the brain, cortex being one of those areas. Poor surface approximation accuracy discards registration points located in the cortex area.

R5: Mesh size optimality. The number of vertices in the meshes we constructed were adjusted using the custom parameters for mesh generation in each particular case. However, given the similar number of mesh vertices, the evaluated meshes perform differently with respect to the quantitative metrics we compared. An important practical concern is the ability to control the total size of the mesh. Based on our experience, *Tetgen* is the most flexible in this respect. In order to control the size of the mesh with *NETGEN*, the sizing values at the background mesh vertices should be scaled. Moreover, the size of the elements near the surface is very hard to control. The size of the *RGM* mesh can be changed by varying the size of the initial lattice, and by parametrizing the refinement rules, which is not straightforward in practice (e.g., changing the number of the maximum registration points for the tetrahedron vertex cells, which can only be done in integer increments).

R6: Control over minimum dihedral angle. Based on our experimental data, the meshes produced by all three methods are very similar when judged by the distribution of minimum dihedral angle. *NETGEN* produces meshes with the relatively largest values of the minimum dihedral angle. However, all of the compared meshes had minimum angle larger than 5°, which is an acceptable value for the stiffness matrix calculations.

In summary, we were able to address all of the application-specific requirements except R3 using the off-the-shelf mesh generation tools. Custom sizing function was essential to meet requirements R1 and R2. Among the evaluated meshing tools, the Delaunay-based approach provides best theoretical guarantees, best practical results, and is the most flexible in the mesh size control. At the same time, the RMS error values, which are the metrics of the most practical relevance, are very similar for all the evaluated methods.

The open questions related to this study are the following: (1) construction of the sizing function, which balances the conflicting requirements, and gives the ability to assign weights according to their importance; (2) improved fitness of the generated meshes to the desired mesh sizing, and studying the guarantees of such fitness; (3) further evaluation of the impact of the mesh on the error of registration with respect to the ground truth.

Acknowledgement. We thank Olivier Clatz for insightful discussions and for contributing the NRR code for this study, Simon K. Warfield and Ron Kikinis for providing the opportunity of evaluating our work in the clinical setting. We thank Hang Si and Joachim Shöberl for the development and continuing support of extraordinary mesh generation software. This research was supported in part by NSF

grants CSI-0719929 and CNS-0312980, and by John Simon Guggenheim Memorial Foundation.

References

1. Clatz, O., Delingette, H., Talos, I.F., Golby, A.J., Kikinis, R., Jolesz, F.A., Ayache, N., Warfield, S.K.: Robust non-rigid registration to capture brain shift from intra-operative MRI. IEEE Trans. Med. Imag. **24** (2005) 1417–1427
2. Archip, N., Clatz, O., Whalen, S., Kacher, D., Fedorov, A., Kot, A., Chrisochoides, N., Jolesz, F., Golby, A., Black, P.M., Warfield, S.K.: Non-rigid alignment of pre-operative MRI, fMRI, and DT-MRI with intra-operative MRI for enhanced visualization and navigation in image-guided neurosurgery. NeuroImage, **35** (2007) 609–624
3. Fedorov, A., Chrisochoides, N., Kikinis, R., Warfield, S.K.: An evaluation of three approaches to tetrahedral mesh generation for deformable registration of MR images. In: Proc. of IEEE ISBI'06. (2006) 658–661
4. Hill, D.L.G., Batchelor, P.G., Holden, M., Hawkes, D.J.: Medical image registration. Physics in Medicine and Biology **46** (2001) R1–R45
5. Delingette, H., Ayache, N. In: Soft tissue modeling for surgery simulation. 1 edn. Volume XII of Handbook of Numerical Analysis: Special volume: Computational models for the human body. Elsevier, Netherlands (2004) 453–550
6. Shewchuk, J.R.: What is a good linear element? Interpolation, conditioning, and quality measures. In: Proc. of 11th IMR. (2002) 115–126
7. Owen, S.J.: A survey of unstructured mesh generation technology. In: Proc. of 7th IMR. (1998) 239–267
8. Yoo, T.S.: Insight into images. A K Peters (2004)
9. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Proc. of ACM SIGGRAPH. (1987) 163–169
10. Boyd, S.K., Müller, R.: Smooth surface meshing for automated finite element model generation from 3d image data. J. of Biomechanics **39** (2006) 1287–1295
11. Dey, T.K., Li, G., Ray, T.: Polygonal surface remeshing with Delaunay refinement. In: Proc. of 14th IMR. (2005) 343–361
12. Cebral, J., Castro, M.A., Löhner, R., Burgess, J.E., Pergolizzi, R., Putman, C.M.: Recent developments in patient-specific image-based modeling of hemodynamics. Mecanica Computacional **XXIII** (2004) 1471–1482
13. Amenta, N., Bern, M.: Surface reconstruction by Voronoi filtering. In: Proc. of 14th SCG, ACM Press (1998) 39–48
14. Kolluri, R., Shewchuk, J.R., O'Brien, J.F.: Spectral surface reconstruction from noisy point clouds. In: Proc. of SGP'04. (2004) 11–21
15. Dey, T.K., Levine, J.A.: Delaunay meshing of isosurfaces. In: Proc. of IEEE Int. Conf. on Shape Modeling and Applications (SMI '07). (2007) 241–250
16. Boissonnat, J.D., Oudot, S.: Provably good sampling and meshing of surfaces. Graphical Models **67** (2005) 405–451
17. Frey, P., Sarter, B., Gautherie, M.: Fully automatic mesh generation for 3-d domains based upon voxel sets. Int. J. for Num. Meth. in Engng. **37** (1994) 2735–2753
18. Zhang, Y., Bajaj, C., Sohn, B.S.: 3d finite element meshing from imaging data. Comp. Meth. in App. Mech. and Engng. **194** (2005) 5083–5106

19. Cline, H.E., Lorensen, W.E., Kikinis, R., Jolesz, F.: Three-dimensional segmentation of MR images of the head using probability and connectivity. J. of Comp. Assisted Tomography **14** (1990) 1037–1045
20. Cebral, J.R., Löhner, R.: From medical images to CFD meshes. In: Proc. of 8th IMR. (1999) 321–331
21. Hartmann, U., Kruggel, F.: A fast algorithm for generating large tetrahedral 3d finite element meshes from magnetic resonance tomograms. In: IEEE Workshop on Biomedical Image Analysis. (1998) 184
22. Udeshi, T.: Tetrahedral mesh generation from segmented voxel data. In: Proc. of 12th IMR. (2003) 425–436
23. Yerry, M.A., Shephard, M.S.: Automatic three-dimensional mesh generation by the modified-octree technique. Int. J. Num. Meth. Engng **20** (1984) 1965–1990
24. Mitchell, S.A., Vavasis, S.A.: Quality mesh generation in higher dimensions. SIAM J. Comp **29** (2000) 1334–1370
25. Ferrant, M.: Physics-based Deformable Modeling of Volumes and Surfaces for Medical Image Registration, Segmentation and Visualization. PhD thesis, Universite' Catholique de Louvain (2001)
26. Mohamed, A., Davatzikos, C.: Finite element mesh generation and remeshing from segmented medical images. In: Proc. of IEEE ISBI. (2004) 420–423
27. Fedorov, A., Chrisochoides, N., Kikinis, R., Warfield, S.K.: Tetrahedral mesh generation for medical imaging. ISC Insight Journal, 2005 MICCAI Open Source Workshop (2005) `http://hdl.handle.net/1926/35`.
28. Freitag, L.A., Knupp, P.M., Munson, T.S., Shontz, S.M.: A comparison of optimization software for mesh shape-quality improvement problems. In: Proc. of 11th IMR. (2002) 29–40
29. Castellano-Smith, A.D., Hartkens, T., Schnabel, J., Hose, D.R., Liu, H., Hall, W.A., Truwit, C.L., Hawkes, D.J., Hill, D.L.G.: Constructing patient specific models for correcting intraoperative brain deformation. In: Proc. of MICCAI'01. Volume 2208. (2001) 1091–1098
30. Molino, N.P., Bridson, R., Teran, J., Fedkiw, R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In: Proc. of 12th IMR. (2003) 103–114
31. Persson, P.O., Strang, G.: A simple mesh generator in MATLAB. SIAM Review **46** (2004) 329–345
32. Si, H.: On refinement of constrained Delaunay tetrahedralizations. In: Proc. of 15th IMR. (2006) 509–528
33. Schöberl, J.: NETGEN: An advancing front 2d/3d-mesh generator based on abstract rules. Computing and Visualization in Science **1** (1997) 41–52
34. Brewer, M., Diachin, L.F., Knupp, P.M., Leurent, T., Melander, D.J.: The Mesquite mesh quality improvement toolkit. In: Proc. of 12th IMR. (2003) 239–250
35. CGAL: Cgal, Computational Geometry Algorithms Library (2008) `http://www.cgal.org`.
36. Ibanez, L., Schroeder, W.J.: The ITK Software Guide. Kitware Inc (2003)
37. Billet, E., Fedorov, A., Chrisochoides, N.: The use of robust local Hausdorff distances in accuracy assessment for image alignment of brain MRI. ISC Insight Journal, January-June (2008) `http://hdl.handle.net/1926/1354`.
38. Baker, T.J.: Mesh movement and metamorphosis. Engng with Comp. **18** (2002) 188–198