Tetrahedral Image-To-Mesh Conversion Approaches For Surgery Simulation and Navigation

Andrey N. Chernikov, Panagiotis A. Foteinos, Yixun Liu, Michel Audette, Andinet Enquobahrie, and Nikos P. Chrisochoides

Abstract In this paper we evaluate three different mesh generation approaches with respect to their fitness for use in a surgery simulation and navigation system. The behavior of such a system can be thought of as a trade-off between material fidelity and computation time. We focus on one critical component of this system, namely non-rigid registration, and conduct an experimental study of the selected mesh generation approaches with respect to material fidelity of the resulting meshes, shape of mesh elements, condition number of the resulting stiffness matrix, and the registration error. We concluded that meshes with very bad fidelity do not affect the accuracy drastically. On the contrary, meshes with very good fidelity hurt the speed of the mesher due to the poor quality they exhibit. We also observed that the speed of the solver is very sensitive to mesh quality rather than to fidelity. For these reasons, we think that mesh generation should first try to produce high quality meshes, possibly sacrificing fidelity.

Andrey N. Chernikov
Department of Computer Science, Old Dominion University, Norfolk, VA, 23529, e-mail: achernik@cs.odu.edu

Panagiotis A. Foteinos
Department of Computer Science, College of William and Mary, Williamsburg, VA; Department of Computer Science, Old Dominion University, Norfolk, VA, 23187, e-mail: pfot@cs.wm.edu

Yixun Liu
Department of Computer Science, Old Dominion University, Norfolk, VA, 23529, e-mail: yxliuwm@gmail.com

Michel Audette
Department of Modeling, Simulation and Visualization Engineering, Old Dominion University, Norfolk VA, 23529, e-mail: maudette@odu.edu

Andinet Enquobahrie
Kitware Inc., Carrboro NC, 27510, e-mail: andinetenqu@kitware.com

Nikos P. Chrisochoides
Department of Computer Science, Old Dominion University, Norfolk VA, 23529, e-mail: nikos@cs.odu.edu

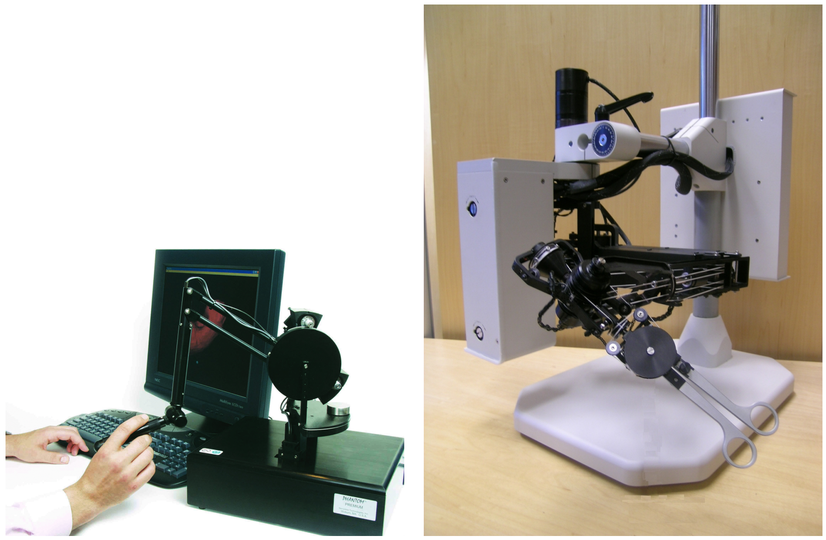
1 Introduction

Surgical simulation is the application of computers to synthesizing an anatomical response to a simulated therapy. This is achieved through a software program that synthesizes tissue response to virtual surgical tools, typically a mechanical response to cutting or manipulation. This behavior can be thought of as a trade-off between material fidelity and computation time, whose weighted emphasis on one or the other can be characterized as a spectrum. At one end of the spectrum we have *predictive* simulation, which consists of highly faithful off-line computations used by expert surgeons to predict the outcome of, and optimize, an intervention, on the basis of an anatomical model of the patient derived from that individuals preoperative image dataset. At the other end of the spectrum, the objective of *interactive* simulation is to offer a means of training surgical residents in order to improve their skill without risk to a real patient, by way of a haptic device manipulated by the user to position a virtual surgical tool, while producing a force feedback that simulates tissue resistance and a real-time graphical rendering of an anatomical model at that point in simulated time. Figure 1 illustrates some commonly used haptic devices.

Typically, the biomechanics engine used to achieve a response at near-haptic rates (some interpolation is feasible for haptic rates of 500 Hz or more), in the context of interactive simulation, is less constitutively faithful than that of predictive simulation, although much recent work is devoted to reconciling the conflicting requirements of interactivity and material faithfulness.

Irrespective of whether a medical simulator emphasizes interactivity or predictive computation, the simulation requires an anatomical model on which to carry out its synthesized therapy. For most clinical applications, such a model is not drawn with 3D CAD software, but rather extracted by image analysis from a patient dataset. As a result, the starting point for this model is one or more MR or CT volumes, which in the multi-modal case can be co-registered and resampled, which leads to a volumetric scalar or vector image, typically of several hundred voxels along each axis. For example, a 1mm isotropic MR image of the head is usually at least $256 \times 256 \times 256$, which equates with more than 16 million voxels, which in turn precludes efficient computation directly based on raw or segmented image data. In addition, many biomechanical engines require the decomposition of a geometrically complex body into simple shapes, e.g.: elements, given that the computation itself is typically a matrix equation based on simple, well understood elemental expressions.

These requirements, computational efficiency and geometric decomposition, motivate the need for a representation of the anatomy in terms of simple shapes, such as triangles and tetrahedra. It is worth noting that in the mesh generation community, the generation of tetrahedra corresponds to unstructured mesh generation as contrasted from structured meshes which are typically comprised of hexahedra. The latter elements are not generally used in medical simulation, because this meshing approach requires a significant amount of user interaction (in contrast with tetrahedral meshing, which can be automated). The reason is that hexahedral meshes are more rigid structures and cannot be always automatically constructed for complex geometries [24]. Moreover, the subdivision of a hexahedron does not reduce to



(a) (b)

Fig. 1 Commercial haptic devices: (a) Sensable's 6 degree-of-freedom (d.o.f.) Phantom 6S/1.5; (b) MPB Technologies' 7 d.o.f. Freedom 7S.

more hexahedra, which limits their applicability to interactive simulation, whereas a tetrahedron ultimately is divisible into more tetrahedra.

Finally, recent surgery simulation research emphasizes so-called meshless methods [5], which involve a system of equations derived from point-centered shape functions. Meshless methods discretize partial differential equations, including continuum mechanics expressions, through shape functions with compact support defined on a local cloud of points (or nodes), rather than on non-overlapping elements. Despite the name that implies that no mesh is involved, the latter approach requires a preliminary meshing that establishes neighboring vertices in the point cloud used in the discretization.

2 Background

2.1 Non-Rigid Registration

We used the non-rigid registration method described by Clatz *et al.* [7] which is shown to be robust enough to be usable to clinical studies. Below, we outline the main aspects of this NRR method.

The method consists of three steps, namely, *feature points selection*, *block matching*, and *system solution*. See Figure 2 for an illustration. During feature points selec-

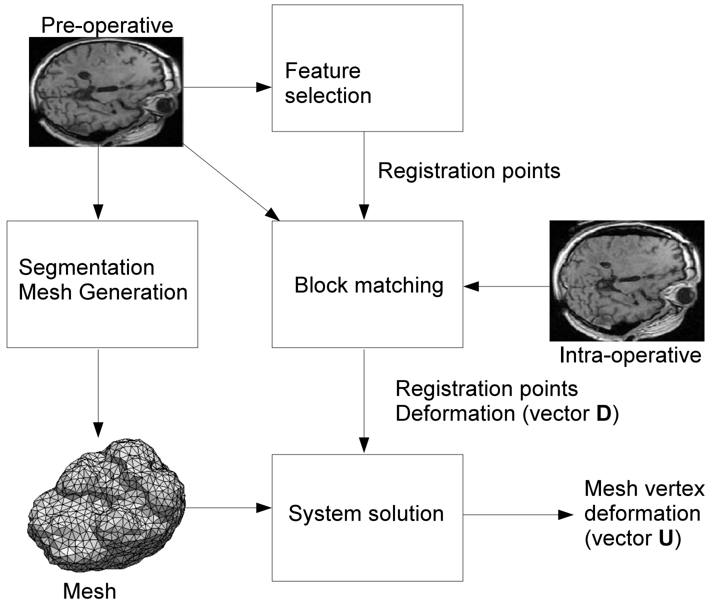


Fig. 2 The non-rigid registration procedure.

tion, a sparse set of points is chosen from the pre-operative image. These points are called *registration* points. Then, the correspondence of these points into the intra-operative image is found via a block matching scheme. Specifically, for a given registration point r, a small window around it in the intra-operative image is searched; the corresponding point r' reported is the one that maximizes the correlation coefficient between r' and r.

Having computed the deformation vector $\mathbf{D}$ on the registration points (as a result of the block matching step), the deformation vector on the mesh vertices $\mathbf{U}$ (the unknowns) is calculated so that the following energy is minimized:

$$W = \underbrace{(\mathbf{HU} - \mathbf{D})^{\top} (\mathbf{HU} - \mathbf{D})}_{\text{Error energy}} + \underbrace{\mathbf{U}^{\top}\mathbf{KU}}_{\text{Mechanical energy}} \tag{1}$$

In the above equation, $\mathbf{K}$ is the $|U| \times |U|$ mechanical stiffness matrix. $\mathbf{H}$ is the linear interpolating matrix of size $|D| \times |U|$; this matrix contains the measurements of the linear shape functions on every registration point. The contributing shape functions for each registration point r_i are those defined over the mesh nodes whose forming mesh element includes r_i.

The block matching deformation d_i of a registration point r_i affects the deformation of a mesh node v_j, only if v_j is incident upon a mesh element e that contains r_j. In fact, if the minimization of the error energy (also known as matching energy) was perfect (i.e., if it vanished), then the linear interpolation (of the solution of the mesh nodes of e) on r_i would give the value d_i. As Clatz *et al.* show [7], this method tries to minimize this exact error energy E:

$$E = \sqrt{(\mathbf{HU} - \mathbf{D})^\top (\mathbf{HU} - \mathbf{D})} = |\mathbf{HU} - \mathbf{D}| \tag{2}$$

which is the interpolation error on the registration points $r_1, r_2, \ldots, r_{|D|}$.

Since the minimization of only the error energy is under-constrained, the mechanical energy in Equation (1) is used to model the deformation of the brain as a physical body based on FEM. This, in turn, is used to discover and discard the outlier registration points, i.e., points whose deformation estimation from block matching contradicts the physical properties of the brain. For information about the construction of the mechanical stiffness matrix $\mathbf{K}$, see Delingette and Ayache [8].

The deformation vector $\mathbf{U}$, over which energy W is minimized, is computed through the following iterative equations:

$$\begin{aligned} \mathbf{F}_0 &= 0\,, \\ \left(\mathbf{K} + \mathbf{H}^\top\mathbf{H}\right)\mathbf{U}_i &= \mathbf{H}^\top\mathbf{D} + \mathbf{F}_{i-1}, i = 1, 2, \ldots\,, \\ \mathbf{F}_i &= \mathbf{KU}_i, i = 1, 2, \ldots \end{aligned}$$

Clatz *et al.* [7] showed that the system above converges. Also, observe that $\mathbf{K} + \mathbf{H}^\top\mathbf{H}$ is the matrix responsible for the robustness of NRR; its condition number affects both the accuracy and the speed of the solution.

2.2 Image-To-Mesh Conversion

The problem of unstructured Image-To-Mesh conversion (I2M) is the following. Given an image as a collection of voxels, such that each voxel is assigned a label of a single tissue or of the background, construct a tetrahedral mesh that overlays the tissues and conforms to their boundaries. In this paper we study three I2M algorithms with respect to their suitability for real-time finite element analysis, based on the following requirements:

- The mesh offers a reasonably close representation (fidelity) of the underlying tissues. Our approach is to expose parameters that allow for a trade-off between the fidelity and the final number of elements with the goal of improving the end-to-end execution time of the FE analysis codes.
- The number of tetrahedra in the mesh is as small as possible provided the two requirements above are satisfied. This requirement is based on the cost of assembling and solving a sparse system of linear equations in the finite element method, which directly depends on the number of tetrahedra.
- Elements do not have very small angles which lead to poor conditioning of the stiffness matrix in Finite Element (FE) Analysis for biomechanics applications.

There is a large body of work on constructing guaranteed quality meshes for Computer Aided Design (CAD) models. The specificity of CAD-oriented approaches is that the meshes have to match exactly to the boundaries of the models. In contrast, the I2M problem allows for a certain distance between the mesh boundary and the image boundary, usually specified as a fidelity tolerance.

Labelle and Shewchuk [15] described an Isosurface Stuffing method for guaranteed quality tetrahedral meshing of domains defined by general surfaces. They offer a one-sided fidelity guarantee (from the mesh to the model) in terms of Hausdorff distance, and, provided the surface is sufficiently smooth, also in the other direction (from the model to the mesh). Their algorithm first constructs a body-centered cubic (BCC) lattice that covers the model, then fills the BCC with high quality template elements, and warps the mesh vertices onto the model surface, or inserts vertices on the surface, and modifies the mesh. Using interval arithmetic, they prove that new elements have dihedral angles above a certain threshold. However, images are not smooth surfaces, and to the best of our knowledge, this technique has not been extended to mesh images. One approach could be to interpolate or approximate the boundary pixels by a smooth surface, for example using the *m-reps* segmentation technique [21], but it would be complicated by the need to control the maximum approximation (interpolation) error. On the other hand, an I2M solution can benefit from the fact that images provide more information on their structure than general surfaces. For example, the tasks of finding the local feature size [12] and all connected components can be done relatively easily on images since they already provide the finest known sampling of the space.

There are also heuristic solutions to the I2M problem, some of them developed in our group [10, 16], that fall into two categories: (1) first coarsen the boundary of the image, and then apply CAD-based algorithms to construct the final mesh, (2) construct the mesh which covers the image, and then warp some of the mesh vertices onto the image surface. The first approach tries to address the fidelity and then the quality requirements, while the second approach does it in reverse order. Unfortunately, neither of these approaches can guarantee the quality of elements in terms of dihedral angles. Both of them face the same underlying difficulty which consists in separating the steps that attempt to satisfy the quality and the fidelity requirements. As a result, the output of one step does not produce an optimal input for the other step. An approach based on filling in brick elements with quality tetrahedra was developed by Hartmann and Kruggel [10], however, it keeps an over-refined mesh near the boundaries. Another method by Dogan et al. [9] produces a mesh as a by-product of an iterative segmentation procedure, by an application of a CAD-oriented mesh generator Triangle [71] to the segmented boundaries.

Zhang et al. [29] described an algorithm to construct adaptive and quality 3D meshes from imaging data. Similar to our approach, they create an initial octree-based mesh, and then improve its quality using iterative edge contraction. Specifically, their approach removes tetrahedra with the worst ratio of the longest to shortest edge length by contracting their shortest edges; however, when it is detected that a requested ratio threshold cannot be reached the strategy is reversed to point insertion through longest edge bisection. Another approach proposed by Reid et al. [22]

and Goksel et al. [13] is to iteratively deform an initial mesh by vertex movement and other operations to conform to the boundaries in the image.

In Computer Aided Surgery (CAS) and specifically in image guided neurosurgery, Magnetic Resonance Images (MRI) obtained before the procedure (pre-operative) provide extensive information which can help surgeons to plan a resection path. Careful planning is important to achieve the maximal removal of malignant tissue from a patient's brain, while incurring the minimal damage to healthy structures and regions of the brain. However, current practices of neurosurgical resection involve the opening of the scull and the dura. This results in a deformation of the brain (known as the brain shift problem) which creates discrepancies between the pre-operative imaging data and the reality during the operation. A correction is possible using non-rigid registration (NRR) of intra-operative MRI with pre-operative data.

In this paper, we target Finite Element (FE) based approaches for the non-rigid registration [7]. These methods use real-time landmark tracking across the entire image volume which makes the non-rigid registration more accurate but computationally expensive, as compared to similar methods that use surface tracking [11]. The non-rigid registration problem should be solved fast enough, so that it can be usable in clinical studies [2, 3].

Image-to-Mesh (I2M) conversion is a critical component of real-time FE-based non-rigid registration of brain images. In this paper one of the I2M evaluation criteria is the wall-clock time to construct the mesh. While in the current formulation the NRR approach makes use of a single mesh constructed before the surgery, we aim to address a general scenario, i.e., when the changes in the object geometry caused by the surgical intervention cannot be accommodated by a pre-existing mesh.

3 Evaluation Of Mesh Generation Techniques

3.1 Mesh Fitness Criteria

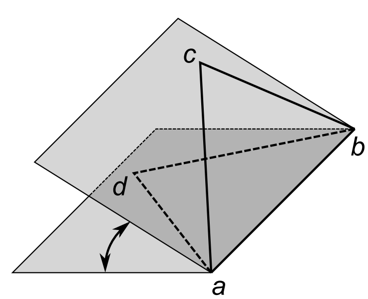


Fig. 3 The dihedral angle between two triangular faces *abc* and *abd* is the angle between two planes containing each of these faces.

A mesh is characterized by its *fidelity* and *quality*. Fidelity measures how well the mesh boundary resembles the surface of the biological object. Quality assesses the shape of mesh elements; the higher the minimum dihedral angle of the mesh elements is, the higher the quality. See Figure 3 for an illustration of a dihedral angle.

It is well known that the quality of the mesh affects both the accuracy and the speed of the solver [26], because the angles of the elements influence the condition number of the stiffness matrix. In the literature, a good deal of effort has been put towards high-quality mesh generation [6, 12, 15, 19, 28].

It is not clear, however, what the impact of fidelity on the accuracy and speed of the solver is. The reason is because there is a complicated trade-off between quality and fidelity. The need for a better surface approximation always implies a deterioration of mesh quality, simply because well-shaped elements cannot fill the space formed by sharp surface creases or by surface parts of high curvature. Also, higher fidelity usually results in an increase of the number of mesh elements which in turn affects both the mesher's and the solver's speed.

In this paper, we evaluated the impact of three public mesh generators [12, 13, 27] on the accuracy and speed of NRR. The meshers were chosen carefully to cover a wide range of mesh generation approaches. The Delaunay mesh algorithm in [12] offers simultaneous meshing of the surface and the volume of the object. The algorithm in [27] is Delaunay but requires the surface of the object as input. Finally, the algorithm in [13] is an optimization-based technique which compresses an initial body-centered cubic lattice (BCC) to the surface. (See Section 4 for more details.) For each mesher, we conducted an extensive series of experiments controlling the fidelity of the output mesh used for the subsequent NRR [7].

3.2 Mesh Generation Libraries

In this paper, we tested the influence of three meshers on NRR, namely, *High Quality Delaunay* mesher (HQD) [12], *Tetgen* [27], and *Point Based Matching* mesher (PBM) [18]. Below, we briefly describe each of them.

HQD meshes both the surface and the volume of the object at the same time without an initial dense sampling of the object surface, as is the case in other Delaunay volume techniques [20, 23]. As a result, the number of elements of the output mesh is small.

Tetgen is a Delaunay mesh generator as well. However, it assumes that the surface of the object is already meshed and represented as a polyhedron. This polyhedron is also known as a *Piecewise Linear Complex* (PLC). Tetgen requires a PLC of the object surface as its input. We used the algorithm in [4] for the PLC generation, implemented in the *Computational Geometry Algorithms Library* (CGAL) [1].

PBM is an optimization-based approach. It starts with a triangulation of a regular grid, i.e., a body-centered cubic lattice (BCC), and then it compresses the outer nodes closer to the object surface as a result of energy minimization. In fact, the smaller the energy achieved, the better the fidelity of the output mesh. This method is able to recover the surface of multi-tissue objects. In this paper, only the single-tissue version of PBM is considered.

3.3 Evaluation Methodology

As mentioned in Section 2.1, registration computes the deformation on the mesh nodes, so that the error energy $E = |\mathbf{HU} - \mathbf{D}|$ is minimized. Mesh generation affects how accurately the error energy is minimized. Therefore, we assess the accuracy of registration by keeping track of this error E. There are two nested loops in the registration algorithm. The outer loop, which is run 10 times following the original paper by Clatz *et al.* [7], is used to discard outlier registration points. The inner loop runs the FE solution and has a fixed threshold on the convergence of the linear solver. This threshold, however, does not translate to a fixed error in the registration result due to the influence of mesh fidelity. Below we describe and report the results of two types of experiments. In the experiments of the first type we vary mesh fidelity and measure the registration error for the selected meshing algorithms. In the experiments of the second type we fix both the fidelity and the registration error, and measure the wall-clock time for the mesher and the solver.

Observe, however, that the outcome of the registration depends on the accuracy of the block matching step (vector $\mathbf{D}$). Also, notice that the mesh does not affect the result of block matching (see Figure 2). Since we are interested in evaluating the impact of mesh generation on registration, we wanted to make registration independent of block matching. For this reason, we synthetically deformed the pre-operative image according to the bio-mechanical properties of the brain. More specifically, we initially ran the registration procedure to register the pre-operative with the intra-operative image as shown in Figure 2, but that time we did not focus on the behavior of the mesh. We just wanted the solution on the mesh nodes. Then, by (linearly) interpolating the solution of the mesh nodes on any point of the image, we obtained a synthetically deformed (intra-operative) image. After this initial registration, all the other registrations (aiming at evaluating mesh generation) are performed between the pre-operative and the synthetically deformed image; that is, the real intra-operative image is replaced by the deformed one. In this way, we achieve two things:

- we know the "true" deformation on any point, and therefore we know the "true" block matching result on any set of registration points, and
- we do not simulate an arbitrary deformation, but rather a realistic one, because the deformed image was obtained taking into account the elasticity properties of the brain through the stiffness matrix $\mathbf{K}$ of Equation 1.

Since we want to measure the influence of mesh generation, only the mesh changes in every experiment. That is, for all the various meshes, the pre-operative image and the set of registration points (together with their deformation $\mathbf{D}$ of course) remain fixed.

As mentioned above, we wish to have control over the fidelity of the output mesh produced by the different meshers. In this paper, we use the *two-sided Hausdorff distance* D_H to measure fidelity.

In our case, metric D_H is defined upon two finite sets A, B as follows:

$$D_H(A,B) = \max\{h(A,B), h(B,A)\}, \text{ where}$$
$$h(A,B) = \max_{a\in A}\min_{b\in B}|a-b|$$

The lower the value of $D_H(A,B)$, the more similar sets A, B are. In fact, $D_H(A,B)$ is equal to 0 if and only if sets A, B are identical.

Fidelity of a mesh is measured as the 2-sided Hausdorff distance D_H of the following sets:

- set A: a densely sampled point set on the surface of the biological object, and
- set B: a densely sampled point set on the boundary facets of the mesh.

Notice that the mesh boundary point set B does not consist of only boundary mesh vertices. The reason is because otherwise, the Hausdorff distance of the meshes produced by HQD would always be 0 (or very close to 0), since this method guarantees that the boundary mesh vertices lie precisely on the object surface.

Having defined fidelity, we proceed by explaining how we control fidelity for each mesher.

For HQD, this is possible through the parameter δ (see [12] for a more detailed explanation). Low values of δ increase the sampling on the object surface which yields better fidelity. High values of δ produce meshes whose boundary crudely approximates the real surface.

For Tetgen, we had to change the fidelity of the PLC given by CGAL. We, therefore, had to adjust two parameters responsible for the PLC's fidelity. The first imposes an upper bound on the circumradius of the *Delaunay balls* and the second forces an upper bound on the distance between the circumcenter of the boundary facets and the corresponding center of their Delaunay balls. More information can be found in [4].

For PBM, control of fidelity is accomplished by adjusting the parameter λ. This parameter defines the trade-off between quality and fidelity: high values of λ make the optimization more sensitive to good fidelity, while low values do not change a lot the position of the initial (high-quality) BCC. However, we observed that λ does not offer a very flexible control over flexibility. Therefore, to get meshes of substantially different fidelity, we had to change not only λ but also the density of the initial BCC.

An important indicator of solution accuracy is the numerical conditioning of the linear system measured by the condition number. The condition number measures the extent by which a relative perturbation of the input affects the relative perturbation of the output. In the experimental evaluation below, we used Matlab's `cond(A)` function which computes the condition number as the ratio of the largest singular value of `A` to the smallest.

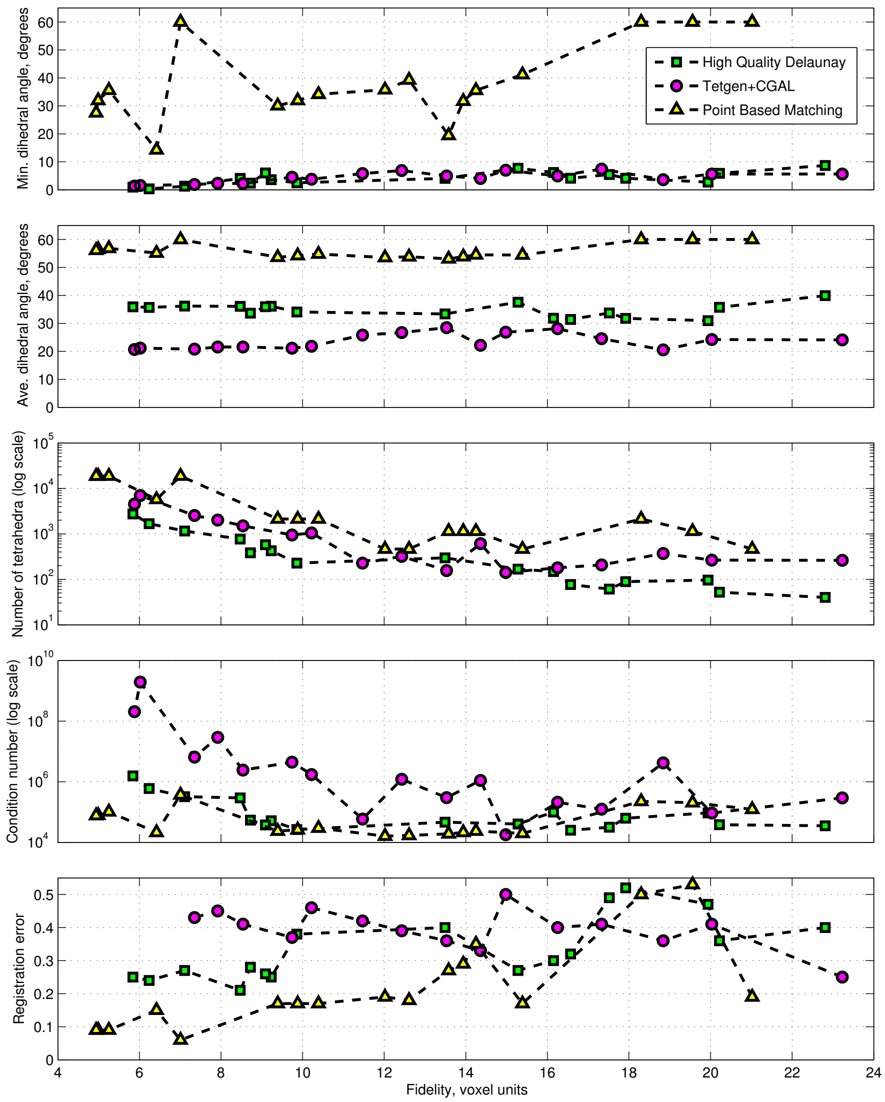


Fig. 4 Mesh properties and the resulting solution characteristics, depending on mesh fidelity measured in terms of symmetric Hausdorff distance. The *x*-axis measures the same fidelity values for all plots, and therefore is annotated only once.

3.4 Results

Figure 4 presents the results obtained by various meshes produced by High Quality Delaunay (HQD), Tetgen+CGAL, and Point Based Matching (PBM) approaches. On all plots, the *x*-axis measures mesh fidelity in terms of the Hausdorff distance D_H

Table 1 Timings (in seconds) for various meshes obtained by different methods. Both the mesh and the solver execution times are reported.

D_H	HQD			Tetgen+CGAL			PBM		
	Mesher	Solver	Total	Mesher	Solver	Total	Mesher	Solver	Total
15-16.5	6.89	0.04	6.93	0.01	0.06	0.07	132.34	0.05	132.39
14-15.5	6.4	0.05	6.45	0.01	0.17	0.18	165.02	0.06	165.08
13-14.5	10.23	0.06	10.29	0.02	0.16	0.18	164.93	0.06	164.99
8.5-9.5	21.57	0.08	21.65	0.09	4.88	4.97	189.19	0.09	189.28
7-8	17.62	0.46	18.08	0.13	**45**	**45.13**	263.39	0.19	263.58

between the mesh and the object surface. All distances are shown with respect to the unit voxel width, and each voxel has physical dimensions 1 mm $\times$ 1 mm $\times$ 1 mm. The condition number depicted is of the matrix $\mathbf{K}+\mathbf{H}^\top\mathbf{H}$ which is responsible for the accuracy and speed of the NRR solver (see Section 2.1). The registration error —as defined in Equation (2) —obtained after the end of the registration process. Figure 5 illustrates the meshes obtained by the three meshing approaches for the best and the worst fidelity.

For HQD, we observe that the error does not fluctuate considerably. All the errors are about less than half the size of a voxel, even when the D_H distance is very large. For Tetgen+CGAL, similarly, fidelity does not seem to affect the error considerably. Also, although the minimum dihedral angles are larger than those in HQD, the average minimum dihedral angles are 10 to 15 degrees less than those in HQD. This results in generally higher error than the error in HQD, but still the differences in accuracy are not very obvious. However, the much larger condition numbers affect the speed of the FEM solver a lot. The FEM solver we use relies on the `bicgstab` linear solver of the GMM library. Actually, for the two runs corresponding to the meshes with the two best fidelity values and with the two higher condition numbers, the solver could not even converge. For the PBM mesh we observe that the quality is very good: the minimum and the average minimum dihedral angles reach perfection. This results in much lower condition numbers and generally lower error than HQD and Tetgen. Again, we observe that fidelity does not play that important role in the accuracy of the NRR. Even meshes with very bad fidelity yield an error less than half the size of the voxel.

Also, see that for the two runs when the solver using the Tetgen meshes did not converge, the condition number is extremely large. We wanted to look into the timings of both the meshers and the solver in more depth, and to determine what the influence of fidelity on speed is. We selected 5 meshes from each method of approximately the same fidelity respectively and measured the time for meshing and the time for solving the registration problem. For each case, the solver has been running until the error becomes less than 0.5 (half the size of the voxel). Table 1 summarizes the results.

We observe that the meshing time of PBM is extremely large: more than 2 minutes in all cases. Actually, most of this time is spent for the initial BCC creation.

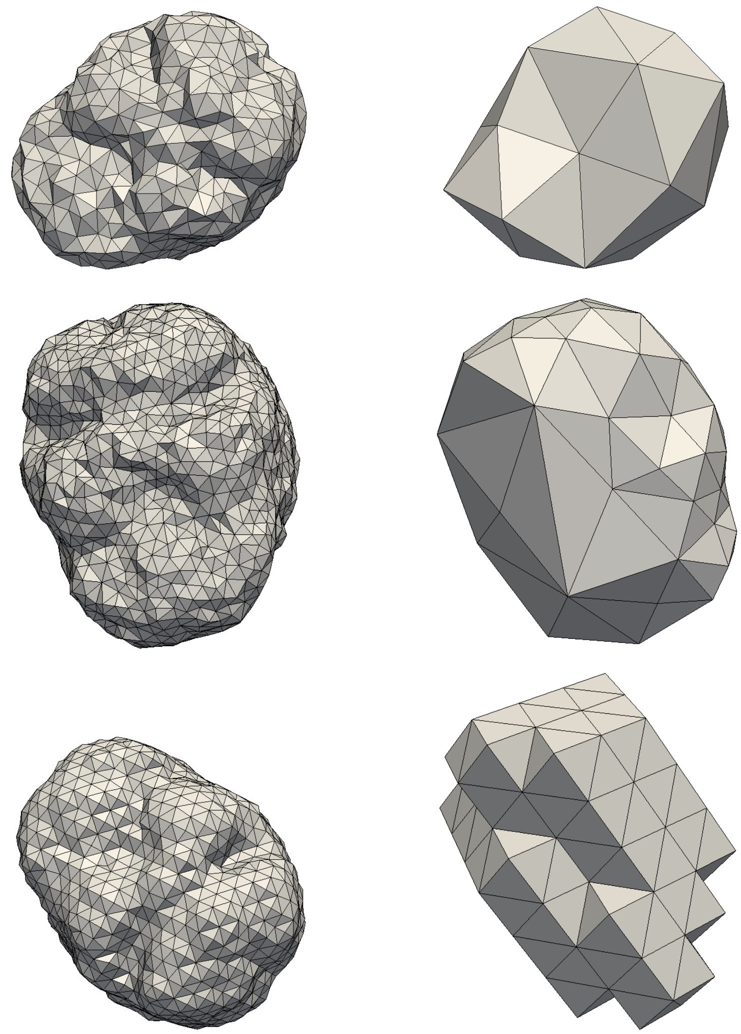


Fig. 5 The rows show the meshes obtained with the three studied approaches, from top to bottom: HQD, Tetgen+CGAL, and PBM. In each row the left image shows the mesh with the lowest D_H value, and the right image shows the mesh with the highest D_H value.

On the other hand, the Tetgen+CGAL scheme is very fast: less than 2 seconds in all cases, even for the bottom mesh which consists of 2,539 elements.

As far as the solver's time is concerned, PBM yields the best meshes. Overall, however, the registration process is much slower than the other methods due to the time consuming mesh generation time. For Tetgen, the solver took much time, when the Hausdorff distance dropped below 8.5 (see bold entries). The minimum dihedral angle for this fidelity is more than 1°, but the very low average minimum dihedral angle (the lowest among all the methods) seems to affect the condition number a lot and consequently the speed of the solver. Although the HQD meshes have elements with very small angles, the average minimum angle is much better than Tetgen (10 to 15 degrees larger). This is why when the solver ran on HQD's meshes, its execution time was less than 2 seconds in all cases, yielding a good overall execution time, even when the D_H distance drops below 8.5.

4 Discussion

In this section, we summarize our findings.

The two Delaunay meshes (i.e., HQD and Tetgen) exhibit low quality when the fidelity increases substantially (when the Hausdorff distance drops below 8 units approximately, in our case studies). This quality deterioration yields a very large condition number which affects the execution time of the solver (see Table 1). We also observe that not only the minimum but also the average minimum dihedral angle plays an important role to the solver's speed. To see it, compare the solver's speed of HQD to the solver's speed of Tetgen when the Hausdorff distance of the meshes is between 7 and 8 units. When Tetgen's mesh was used, the solver was 45 times slower. For these values of fidelity, Tetgen meshes have better minimum dihedral angles than HQD meshes, but they also have much lower average minimum dihedral angles (15 degrees smaller), which is likely to be the reason for a much worse condition number and the consequent large solver's speed.

The accuracy of the solver on the meshes produced by the two Delaunay meshers does not fluctuate significantly by the different fidelity values. That means that the need for good surface approximation does not seem to affect the accuracy of the solver. Meshes approximating very crudely the object surface yielded an error less than half the voxel size.

The main characteristic of the optimization-based mesher (i.e., PBM) is the high minimum and average dihedral angles, even in the case of very good fidelity. The reason is because relatively dense initial BCCs can easily capture the object surface without so much compression, thus preserving the good angles of the BCC triangulation. Of course, the number of elements increases significantly, which makes the mesh generation time extremely slow (see Table 1). We also observe that the solver on PBM's meshes exhibit the least error which in fact is achieved when fidelity is very good (less than 5 units approximately). This is reasonable because good fidelity does not deteriorate the quality as much as is the case of the two Delaunay meshes. Notice, however, that even when the PBM meshes have very bad fidelity, the error does not increase significantly.

5 Acknowledgments

This research is supported in part by NSF grants CCF-1139864, CCF-1136538, CSI-1136536, and by NIH/NINDS Grant No. 5R43NS067742-02. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NSF and National Institute of Neurological Disorders and Stroke or the National Institutes of Health. A preliminary version of this work appeared and is partially reproduced (including original Figures 1, 2, 3, 4) with kind permission from Springer Science+Business Media: Computational Biomechanics For Medicine (A. Wittek et al., eds.), An Evaluation of Tetrahedral Mesh Generation for Nonrigid Registration of Brain MRI, Part 2, 131-142, 2011, Panagiotis A. Foteinos, Yixun Liu, Andrey N. Chernikov and Nikos P. Chrisochoides, ©Springer Science+Business Media, LLC 2011.

References

1. CGAL, Computational Geometry Algorithms Library. URL http://www.cgal.org
2. Archip, N., Clatz, O., Whalen, S., Kacher, D., Fedorov, A., Kot, A., Chrisochoides, N., Jolesz, F., Golby, A., Black, P., Warfield, S.: Non-rigid alignment of preoperative mri, fmri, dt-mri, with intra-operative mri for enchanced visualization and navigation in image-guided neurosurgery. NeuroImage **35**(2) (2007)
3. Bajaj, C., Oden, J.T., Diller, K.R., Browne, J.C., Hazle, J., Babuška, I., Bass, J., Bidaut, L., Demkowicz, L., Elliott, A., Feng, Y., Fuentes, D., Kwon, B., S., P., Stafford, R.J., Zhang, Y.: Using cyber-infrastructure for dynamic data driven laser treatment of cancer. In: Proceedings of the 7th international conference on Computational Science, Part I: ICCS 2007, pp. 972–979. Springer-Verlag, Berlin, Heidelberg (2007)
4. Boissonnat, J.D., Oudot, S.: Provably good sampling and meshing of surfaces. Graphical Models **67**(5), 405–451 (2005)
5. Bordas S. Rabcsuk T., Z.G.: Three-dimensional crack initiation, propagation, branching and junction in non-linear materials by an extended meshfree method without asymptotic enrichment. Engineering Fracture Mechanics **75**(5), 943–960 (2008)
6. Cheng, S.W., Dey, T.K., Edelsbrunner, H., Facello, M.A., Teng, S.H.: Sliver exudation. J. ACM **47**(5), 883–904 (2000). DOI http://doi.acm.org/10.1145/355483.355487
7. 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 Transactions on Medical Imaging **24**(11), 1417–1427 (2005)
8. Delingette, H., Ayache, N.: Soft tissue modeling for surgery simulation. Computational Models for the Human Body pp. 453–550 (2004)
9. Doğan, G., Morin, P., Nochetto, R.H.: A variational shape optimization approach for image segmentation with a mumford-shah functional. SIAM J. Sci. Comput. **30**, 3028–3049 (2008). DOI 10.1137/070692066. URL http://portal.acm.org/citation.cfm?id=1461726.1461741
10. Fedorov, A., Chrisochoides, N.: Tetrahedral mesh generation for non-rigid registration of brain mri: Analysis of the requirements and evaluation of solutions. In: Proceedings of the 17th International Meshing Roundtable, pp. 55–72. Springer, Pittsburgh, PA (2008)
11. Ferrant, M.: Physics-based deformable modeling of volumes and surfaces for medical image registration, segmentation and visualization. Ph.D. thesis, Universite Catholique de Louvain (2001). Http://spl.harvard.edu:8000/pages/papers/ferrant/thesis/ferrant.pdf

12. Foteinos, P., Chernikov, A., Chrisochoides, N.: Guaranteed quality tetrahedral delaunay meshing for medical images. In: 7th International Symposium on Voronoi Diagrams in Science and Engineering. Laval University, Quebec City, Canada (2010)
13. Goksel, O., Salcudean, S.E.: Image-based variational meshing. IEEE Transactions on Medical Imaging **30**(1), 11–21 (2011). DOI 10.1109/TMI.2010.2055884
14. Hartmann, U., Kruggel, F.: A fast algorithm for generating large tetrahedral 3d finite element meshes from magnetic resonance tomograms. In: Proceedings of the IEEE Workshop on Biomedical Image Analysis, WBIA, pp. 184–192. IEEE Computer Society, Washington, DC, USA (1998)
15. Labelle, F., Shewchuk, J.R.: Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. ACM Transactions on Graphics **26**(3), 57.1 – 57.10 (2007)
16. Liu, Y., Foteinos, P., Chernikov, A., Chrisochoides, N.: Multi-tissue mesh generation for brain images. In: Proceedings of the 19th International Meshing Roundtable, pp. 367–384. Springer, Chattanooga, TN (2010)
17. Liu, Y., Foteinos, P., Chernikov, A., Chrisochoides, N.: Multi-tissue mesh generation for brain images. In: Proceedings of the 19^{th} International Meshing Roundtable. Chattanooga, Tennessee, USA (2010). To appear
18. Liu, Y., Foteinos, P., Chernikov, A., Chrisochoides, N.: Mesh deformation-based multi-tissue mesh generation for brain images. Engineering with Computers (2011). In press
19. Molino, N.P., Bridson, R., Teran, J., Fedkiw, R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In: Proceedings of the 12th International Meshing Roundtable, pp. 103–114 (2003)
20. Oudot, S., Rineau, L., Yvinec, M.: Meshing Volumes Bounded by Smooth Surfaces. In: International Meshing Roundtable, pp. 203–220. Springer-Verlag, San Diego, California, USA (2005)
21. Pizer, S.M., Fletcher, P.T., Joshi, S., Thall, A., Chen, J.Z., Fridman, Y., Fritsch, D.S., Gash, A.G., Glotzer, J.M., Jiroutek, M.R., Lu, C., Muller, K.E., Tracton, G., Yushkevich, P., Chaney, E.L.: Deformable m-reps for 3d medical image segmentation. International Journal of Computer Vision **55**, 85–106 (2003). URL `http://dx.doi.org/10.1023/A:1026313132218`. 10.1023/A:1026313132218
22. Reid, A.C., Langer, S.A., Lua, R.C., Coffman, V.R., Haan, S.I., Garcia, R.E.: Image-based finite element mesh construction for material microstructures. Computational Materials Science **43**(4), 989–999 (2008). DOI DOI:10.1016/j.commatsci.2008.02.016
23. Rineau, L., Yvinec, M.: Meshing 3D Domains Bounded by Piecewise Smooth Surfaces. In: International Meshing Roundtable, pp. 443–460 (2007)
24. Schneiders, R.: Quadrilateral and hexahedral meshes. In: J.F. Thompson, B.K. Soni, N.P. Weatherill (eds.) Handbook of Grid Generation. CRC Press (1998)
25. Shewchuk, J.R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In: M.C. Lin, D. Manocha (eds.) Applied Computational Geometry: Towards Geometric Engineering, *Lecture Notes in Computer Science*, vol. 1148, pp. 203–222. Springer-Verlag (1996). From the First ACM Workshop on Applied Computational Geometry
26. Shewchuk, J.R.: What is a good linear element? In: Proceedings of the 11th International Meshing Roundtable, pp. 115–126. Sandia National Laboratories (2002)
27. Si, H.: Tetgen a quality tetrahedral mesh generator and a 3d delaunay triangulator. http://tetgen.berlios.de/
28. Tournois, J., Srinivasan, R., Alliez, P.: Perturbing slivers in 3D Delaunay meshes. In: B.W. Clark (ed.) Proceedings of the 18th International Meshing Roundtable, pp. 157–173. Springer (2009)
29. Zhang, Y., Bajaj, C., Sohn, B.S.: 3d finite element meshing from imaging data. Computer Methods in Applied Mechanics and Engineering **194**(48-49), 5083–5106 (2005). DOI 10.1016/j.cma.2004.11.026