Mixed-element mesh for an intra-operative modeling of the brain tumor extraction

Claudio Lobos[1], Marek Bucki[1], Nancy Hitschfeld[2], and Yohan Payan[1]

[1] TIMC-IMAG Laboratory, UMR CNRS 5525, University J. Fourier, 38706 La Tronche, France [claudio.lobos|marek.bucki|yohan.payan]@imag.fr
[2] Departamento de Ciencias de la Computación, FCFM, Universidad de Chile, Santiago, Chile nancy@dcc.uchile.cl

Summary. This paper presents a modified-octree technique that generates a mixed-element mesh. The final output mesh consider cubes, prisms, pyramids and tetrahedra. This technique is optimized for brain tumor extraction simulation in a real-time application. The proposed technique is based on the octree algorithm with a specific constraint: elements will be split only if they intersects a certain region of interest. With this approach we pursued a refined mesh only in the path from the skull opening point to the tumor. Fast computation by the Finite Element Method (FEM) is achieve thanks to the local refinement. Examples are given and comparison with other approaches are presented.

Key words: Modified-Octree, Mixed-elements, Finite Elements, Real-time Application, Patient-specific Simulation, Region Of Interest.

1 Clinical Background

Accurate localization of the target is essential to reduce morbidity during a brain tumor removal intervention. Image guided neurosurgery nowadays faces an important issue for large skull openings, with intra-operative changes that remain largely unsolved.

Once the skull is open a deformation of the brain naturally occurs. This phenomena is known as "the brain-shift". The causes of this deformation can be grouped by:

- physical changes (dura opening, gravity, loss of cerebrospinal fluid, actions of the neurosurgeon, etc) and
- physiological phenomena (swelling due to osmotic drugs, anesthetics, etc).

As a consequence of this intra-operative brain-shift, pre-operative images no longer correspond to reality. Therefore the neuro-navigation system based on those images doesn't necessarily represent the current situation.

In order to face this problem, various teams have proposed to incorporate into existing image-guided neurosurgical systems, a biomechanical modeling to compensate the brain deformations by updating the pre-operative images and planning according to intra-operative brain shape changes. For this, such measured changes (for example the changes of the external shape of the brain tissues in the opening skull region) are given as new boundary conditions to the biomechanical model of the brain tissues that infers the new position of the tumor. Such intra-operative use of a biomechanical model implies that a strong modeling effort must be carried out. Our group tries to provide some clues towards such a modeling effort. Three steps are followed to design the brain model:

- The segmentation of pre-operative images (MRI) to build the external surface mesh of the brain, including the tumor.
- The generation of a volume mesh optimized for real-time simulation.
- The creation of a model of the brain-shift with Finite Elements (FE).

The focus of this paper is the second point. The proposed meshing technique starts from a surface representation of the brain with the tumor and produces a volume mesh of it. Some results about the third point are also shown in section 6.2.

2 Meshing constraints

Time is crucial in surgery thus, to produce an intra-operative real-time modeling of the brain-shift the FEM computation must be fast. The speed of the FEM directly depends on the number of degrees of freedom the system has, thus an optimal mesh in terms of quantity of nodes must be provided.

A good representation of the tumor as well as the Skull Opening Point (SOP) and the path between them is mandatory because here is where a greater deformation is expected [16]. This path will from now on be referred to as the Region of Interest (RoI). A mesh without quality elements can lead to errors in the computation of the FE thus quality is also an important issue in this problem. Therefore the constraints to model the brain-shift in a real-time application are:

1. The final mesh must be refined enough in the RoI and coarse elsewhere.
2. Achieve surface representation for the input FE mesh.
3. Guarantee element quality throughout the entire mesh.

3 Meshing Background

This section gives a description of several meshing techniques in order to find the most suitable one to solve the brain-shift simulation problem.

3.1 Advancing Front

The Advancing Front (AF) technique [11, 7] uses as input a closed surface. All the faces that describe the input domain are treated as fronts and are expanded into the volume in order to achieve a final 3D representation. The selection of points to create the new faces encourages the use of existing points. Figure 1 shows some steps of the AF technique where only a portion of the surface has been taken for clarity purposes. Additional process to improve the quality of the elements can be made.

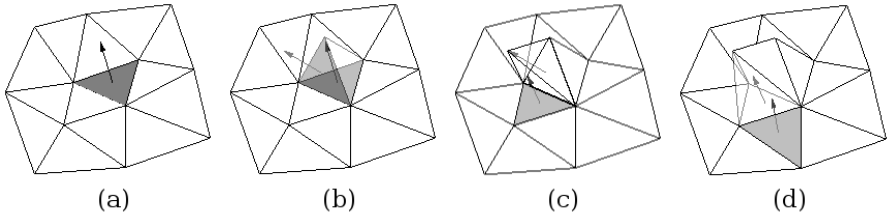

Fig. 1. The advancing front technique: (a) A portion of a surface mesh with one front to expand, (b) the tetrahedron is created and the new faces can be treated as fronts, (c) another expansion using a recently inserted front and (d) another expansion using already inserted points.

This technique is strongly dependent on the input surface mesh. If the input is accurate, it will have a high number of faces producing a refined volume mesh in the entire input domain. This is contradictory with our first constraint: the mesh must be refined in the RoI and coarse elsewhere. On the other hand, if the input mesh is coarse, this technique should be combined with others strategies of local refinement to respect the first constraint.

In relation to the second constraint, this technique achieves a perfect surface representation because it uses the input surface to produce the final volume mesh. However a little remark can be made: If the input surface mesh is coarse and the RoI is constrained to have a large quantity of points, the resulting final volume mesh will continue to be coarse in the portion of the RoI that intersects the input surface mesh.

Two solutions are proposed: (1) pre-treatment of the input surface mesh to be more refined in the RoI or (2) local refinement in the surface portion of the RoI once the volume mesh is generated.

The third constraint is not managed by this technique. However different algorithms can be applied to improve the quality of the resulting volume mesh, for example in the case of tetrahedral AF meshes, constraint the new tetrahedra to fulfill the Delaunay [5] property as long as the fronts continues their expansion.

3.2 Mesh Matching

Mesh matching is an algorithm that starts with a generic volume mesh and tries to match it onto the input domain [4, 13]. The generic volume can be obtained from an interpolation of several sample models. To generate a new mesh, in our case of the brain, the problem is reduced to finding a 3D nonuniform transformation function that will be applied to the entire generic mesh (atlas) and produce in this way the final volume mesh for the current patient. This is shown in figure 2 in a maxillofacial meshing example, from Chabanas et al. [2]

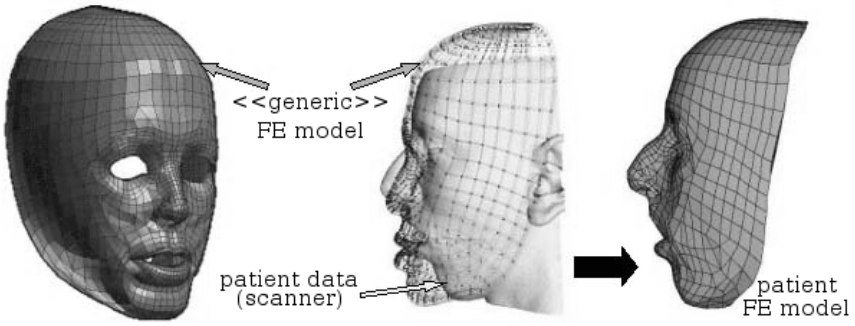

Fig. 2. The mesh matching algorithm in a maxillofacial simulation.

This technique does not satisfy the first constraint. The atlas does not consider the tumor information thus the mesh is not refined in the RoI. It would be necessary to combine this technique with others strategies of local refinement in order to respect the first constraint. In other words, a generic solution cannot be applied to a problem that is patient dependent, i.e. with positions and sizes of tumor and the SOP that change from one patient to another.

The second constraint is satisfied because the atlas normally resembles the input domain. This can be not true for very specific cases where the brain is malformed.

The third constraint like the second one, is also satisfied because the atlas has a great quality coefficient in the entire mesh. It is a quite perfect model that sometimes has modifications usually made by hand in order to produce the best starting point. A loss of quality can occur only in cases where the input domain and the atlas are not alike.

3.3 Regular Octree

The octree meshing technique starts from the bounding box of the surface [17, 14]. This basic cube or "octant" is split into eight new octants. Each

octant is then iteratively split into eight new ones, unless it resides outside the input surface mesh, in which case it is deleted. The algorithm stops when a predefined maximum level of iterations is reached or when a condition of surface approximation is satisfied.

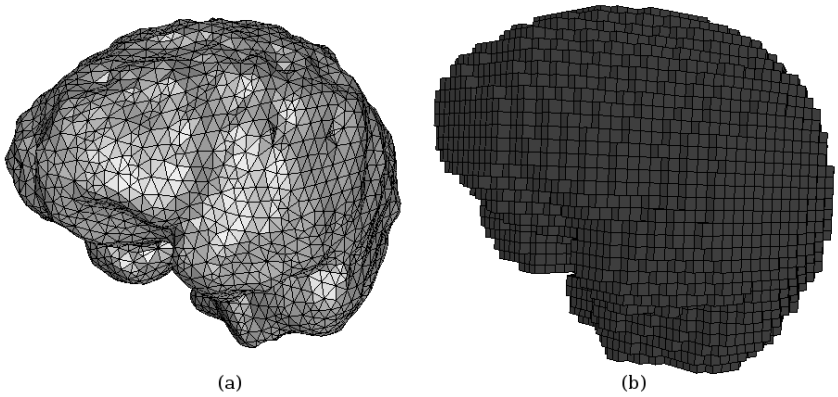

Fig. 3. (a) Input surface triangle mesh of 3152 points and (b) output hexahedral mesh of 20,000 points.

The problem with a regular octree mesh for our problem is that it result in a high number of points even in regions where very few deformation of the brain is expected [9]. Therefore, a non-optimal mesh would be the input for the FEM producing unnecessary time consumption for the entire simulation. For example the mesh shown in figure 3 is unacceptable for the brain-shift modeling due to excessive quantity of points. Therefore the first constraint is not fulfilled.

The octree by itself does not consider a surface approximation algorithm once the split process is done. Therefore it has to be combined with other techniques in order to achieve a surface representation. Two main approaches are considered:

- Marching cubes [12]: this algorithm crops the cubes that lie within the surface and produces, in most cases, tetrahedra.
- Surface projection: this technique projects the points of those elements that intersect the surface, onto it. The main problem is that this can produce degenerate elements unless a minimal displacement is needed.

The quality of the mesh is normally acceptable. The elements with a bad quality can be found only at the surface of the input domain because they have changed in order to achieve the correct representation of the domain.

3.4 Delaunay Mesh

In the literature a Delaunay [5] mesh is said to be a mesh of tetrahedra that respects the Delaunay constraint. An example mesh of the Delaunay property can be seen in figure 4

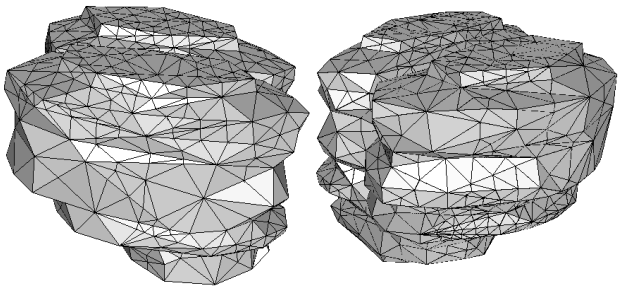

Fig. 4. One output mesh that respect the Delaunay properties. It was generated with TetGen.

A Delaunay mesh can be useful to solve the brain-shift (meshing) problem, however it must be combined with a point insertion strategy that consider the constraints presented in section 2. Current implementations, like TetGen [19], do not offer as much control over point quantity and quality as the constraints needs. One alternative implementation would be the following work flow:

- generate a basic tetrahedralization of the input domain (that doesn't necessarily respect the Delaunay constraint).
- improve the quality of the elements in the RoI and constrain them to a certain size.
- without inserting an excessive number of points, increase the quality of the others elements (outside the RoI).

What is critical in this process is to control the number of nodes. A classical problem of Delaunay meshes is to control the insertion of new nodes when going from one refined zone into another not so refined.

Constraint two can be satisfied by the constrained Delaunay meshes that consider the representation of a domain by using certain predefined points. Those points would be the input surface domain in the brain-shift case. And finally, the third constraint requires the insertion of points to fulfil the underline quality requirement. See also [1] for a Delaunay based approach to isotropic tetrahedral meshing.

4 Discussion Regarding Presented Techniques

The mesh matching and the advancing front techniques are ruled out because it is not easy to incorporate a RoI i.e. to generate a mesh that is refined in a specific region and coarse elsewhere. These techniques need major adjustments in order to achieve the desired mesh.

The Delaunay approach is closer to achieving the desired features, however its emphasis is on the quality of the tetrahedra (i.e. the achievement of the Delaunay property) and not on the quantity of nodes.

Even though the octree technique doesn't accomplish the constraints by itself it allows the refinement of some elements directly by the splitting process. This is a very powerful feature that easily help to fulfill the constraint on local refinement in the RoI. However a process to manage the transition between regions of different refinement levels is needed. A method for surface representation is also needed for the octree in order to achieve an optimal mesh for real-time simulation.

The following section proposes some adaptations provided to the octree technique in order to fulfill all the desired constraints. We consider that this is the best starting point to respect the constraints mentioned in section 2.

5 Meshing Technique for the Brain Shift

The basic octree algorithm is applied but with two main modifications:

- The condition to stop the refinement is the number of nodes the mesh has. The quantity of nodes is provided as input to the algorithm.
- One element will be refined only if it is inside the RoI.

The first modification corresponds to a basic constraint of real-time simulation. Even with high quality elements, a mesh that would have too many nodes would increase the FE time response during surgery and this is not acceptable.

The second modification is due to the fact that the RoI is the area where the predictions of the developed model should be the most accurate one. Consequently only those elements are refined.

This basic octree mesh is not suitable for the FEM. Figure 5 shows an example of different refinement levels for neighbor elements. The implemented algorithm deals with the different possible cases by adding mixed-elements: pyramids, prisms and tetrahedrons. At the end of this subprocess the mesh is valid for the FEM.

The final step is to achieve a surface representation regarding the quality of the elements. Next part details each step of our algorithm.

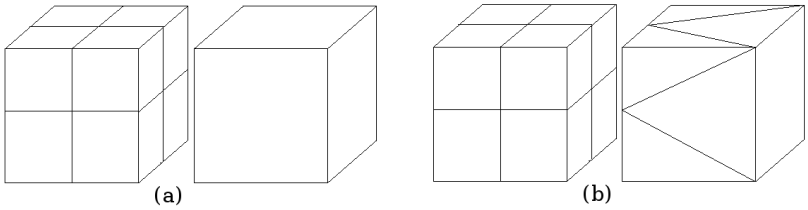

Fig. 5. An example of transition management between different levels of refinement. (a) Two neighbor octants, the left one is more refined. (b) The same two octants but now the right one has a subdivision: 5 pyramids and 4 tetrahedra, making the mesh congruent and suitable for the FEM.

5.1 Inputs and basic mesh

In order to produce a final mesh several inputs are requested:

- A surface mesh of the input domain (S_d).
- A desired quantity of points. This input will be used as the condition to stop the octree subdivision.
- A surface mesh that describes the RoI (S_{RoI}). This mesh has to intersect the input domain.

The first step is to obtain the bounding box of the domain. A global search is performed to detect the maximum and minimum coordinates of each axis. The result is an hexahedron that will correspond to the root octant of the octree. Then a point (P_{out}) that is outside this bounding box is calculated to be used as a reference in several tests that are described later.

The algorithm continues with the classical octree subdivision but with one difference. An octant will be split only if it intersects or is completely inside the RoI (defined by S_{RoI}). The process continues until the maximum number of nodes, provided as input, is reached.

When an element is created the P_{out} is used to check if the element is completely outside the domain. A virtual segment is created between the P_{out} and each vertex of the element. If one of those segments crosses an odd quantity of S_d faces, this element is said to be inside. If not, the element is removed from the mesh. For each new element there is also a test to check if it intersects the S_{RoI}. If not it won't continue to split.

Figure 6 plots the mesh generated for a given quantity of points, the input S_d and the S_{RoI}. Note that the mesh is strongly refined in the RoI.

At this step, the mesh is not suitable to carry out a classical Finite Element Analysis (FEA) because of incongruent elements (an element is said to be incongruent when it has one or more points inserted in each face or edge of it). Some modifications of the FE basis functions (see [15] for example) could enable this mesh to be used for an FEA. However, it was chosen in this paper to remain with a classical FEM approach, which means that the mesh has to be modified in order to manage transitions between incongruent elements.

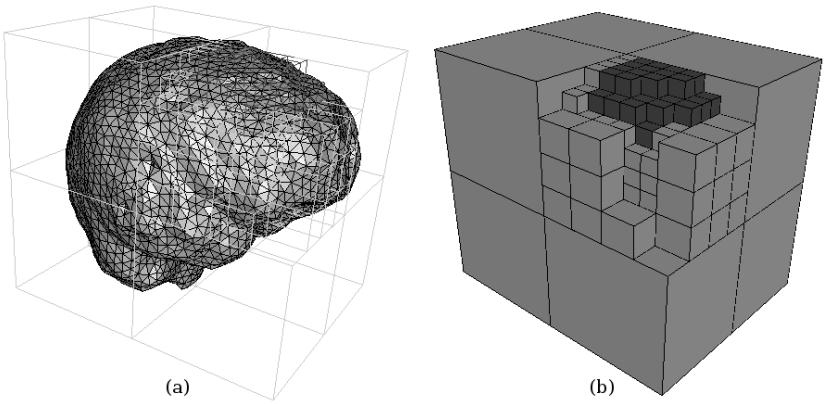

Fig. 6. Octree: only the elements that intersects the RoI are split in order to achieve a higher density in that region: (a) the input and the generated mesh and (b) the generated mesh in solid: dark elements corresponds to the RoI. This mesh has 416 points and cannot be used for classical FEA.

5.2 Managing transitions

The developed application has several patterns [8] that detect and split the incongruent elements. In particular, we can address to incongruent elements that has at most one point inserted in each face or edge of it. This is a property called "one-irregular". In order to produce a one-irregular mesh, all the octants that do not respect this property are split. In this manner, regions outside the RoI have more points to represent the volume. In other words, the density of points outside the RoI is a consequence of the RoI itself. The resulting mesh can be seen in figure 7.

For each new element generated from this subdivision a test is run to check if it is outside the contour of the brain, as described in the previous section.

It is important to keep the faces as planar as possible and the triangle faces are the only ones that preserve this property all the time. In order to represent the surface the points outside the S_d must be projected into it. For this reason, all the elements that intersects the input domain are tetrahedralized. In this manner the outside points can be moved without losing the planar property of the faces.

The octree hierarchical structure is used to assign a default subdivision orientation for each type of element. The neighbor face information is also used to test if there are faces that were already triangulated in which case this orientation is used.

An hexahedron will be split into five tetrahedra as shown in figure 8a. The prism will be divided into three tetrahedra like in figure 8b and the pyramid into only two as figure 8c shows.

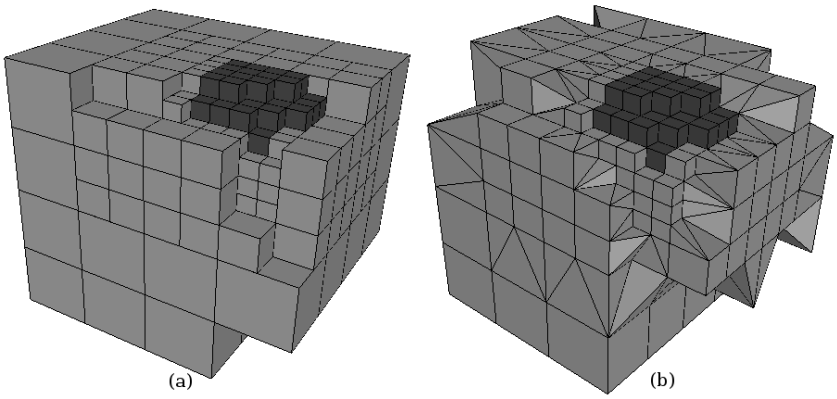

Fig. 7. (a) A one-irregular mesh: each cube has at most one point inserted in each face or edge of it. This mesh has 594 points. (b) A mixed-element mesh that handle the transitions. This mesh has 660 points and can be used for classical FEA.

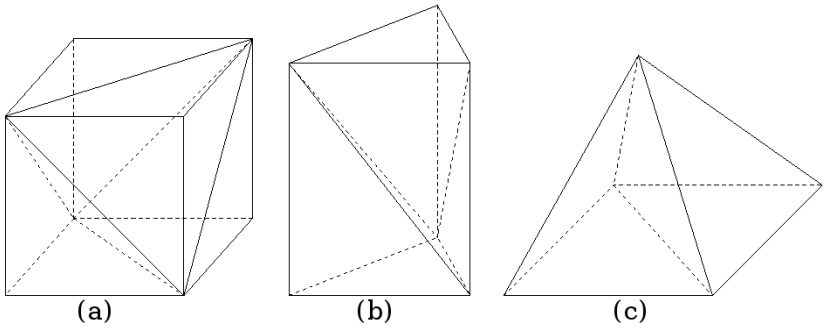

Fig. 8. (a) The subdivision of an hexahedron into five tetrahedra, (b) The subdivision of a prism into three tetrahedra and (c) The subdivision of a pyramid into two tetrahedra.

After the local partition of a surface element into tetraedra, it is possible that some neighbors (such as an hexahedron) get a diagonal in their square face. This diagonal must be incorpored and the neighbor element must be tessellated. This new tessellation depends on the type of the element with a diagonal in their square face. For example, if the element is a pyramid: the base face is triangulated congruently with the neighbors and the propagation (of the triangulation) stops. If the element is a prism it depends on the number of already triangulated faces it has:

- 0: the prism is split in one tetrahedra and one pyramid. The face that wasn't with a diagonal before is updated with the information to be handled by the neighbor element.

- 1: if it is possible to build one pyramid and one tetrahedra this option is preferred and the propagation of diagonalizations stops. If not, one diagonal is added and the corresponding face is updated.
- 2: there is only one option and the propagation stops.

In the case of the hexahedron the inner middle point of it is inserted. With this new point the hexahedron is replaced by 6 pyramids. Then the faces that have diagonals trigger the split of the corresponding pyramid and the propagation stops.

5.3 Mesh Quality

An extraordinary number of measures has been proposed, ranging from bounds on solid angles to more complex geometric ratios (See [10, 18, 20] for more details). To our knowledge, only one team [6] dealing with the brain-shift problem through a biomechanical FE modeling of the brain have proposed a mesh quality measure. We decided to use the same measure: the aspect-ratio coefficient (ARC). This coefficient is obtained in different manners for each type of element.

The ARC is normalized so that ARC = 1 corresponds to an ideal element and ARC $\rightarrow \infty$ as the element becomes increasingly distorted.

The ARC of the tetrahedra is obtained as follows [6]:

$$\text{ARC} = \frac{(\frac{1}{6}\sum_{i=1}^{6}(l_i^2))^{3/2}}{8.47867V^{el}}$$

Where $l_i (i = 1, \ldots, 6)$ are its edge lengths and V^{el} is the volume of the tetrahedron. The value 1 corresponds in this case to an equilateral tetrahedra.

For the Hexahedron the segments between oppose faces middle points are obtained. The ratio between the longest and the shortest of the three segments will be used as the ARC of this element.

Three segments are also used in the case of the pyramid. The two firsts are constructed using the middle points of opposite edges in the base face. The third segment is the one that represents the height of the pyramid. Like in the case of the hexahedron the coefficient will be the ratio between the longest and shortest segment.

To obtain the ARC of the prism a selection of a square face of it is made. Two segments are obtained as in the case of the pyramid base face. The third segment is obtained using the height of the triangle face that corresponds to the average of the two original triangle faces of the prism. This can be seen in figure 9. The ARC, as in previous cases, is obtained from the ratio between the longest and shortest segment.

Measures over the quality of the mesh are described in section 6.1.

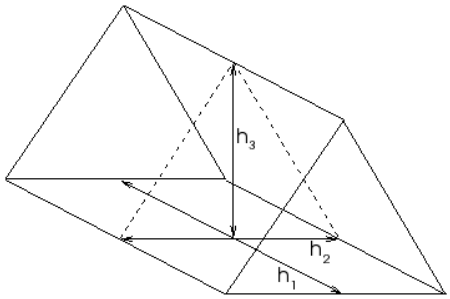

Fig. 9. The aspect-ratio coefficient for the prism is obtained from the ratio between the longest and shortest segment of the three heights presented.

5.4 Surface representation

We propose a novel mesh adaptation technique. The goal is to achieve an acceptable surface representation after a quality adaptative octree-based mesh has been generated. The nodes of the mesh obtained until this point of the algorithm do not necessarily lie on the object surface, as described in figure 7. It is thus necessary to improve the surface representation of the mesh while keeping an acceptable level of quality of its elements.

The inputs to our algorithm are:

- The source mesh: a generic volumetric mesh comprising quality elements (in our case the octree-based mixed-element mesh generated above).
- Destination mesh: the surface we want our source mesh to be adapted to (in our case S_d.

The key idea of our algorithm is to use a mechanical simulation to constraint the deformation between the source and the destination mesh. We drive the surface nodes of the source mesh with a step-by-step displacement towards the destination mesh, leaving a mechanical model perform the inner nodes relaxation throughout the deformation. It is also important to mention that it was chosen to implement a compressible material behavior (poisson $\nu = 0.3$), in order to allow a mechanical compression of the 3D mesh.

Our approach, unlike direct projection of surface nodes that disregards inner nodes position, is guaranteed to not produce element inversions. To preserve the mesh overall quality we check the elements quality at each step of the deformation. If necessary we artificially increase the mechanical resistance, or stiffness, of those elements that suffer the greatest quality loss before proceeding to the next deformation step. This can result in oscillations between neighboring elements. Therefore in order to guarantee the algorithm termination, a constraint was used: we stop increasing the stiffness of an element if it's $Young$'s modulus value reaches a predefined threshold. Note that our goal is not to compute a realistic deformation of the mesh. The virtual mechanical medium is merely used to compute the inner relaxation of the

Algorithm 1 Surface representation

Require: source mesh and destination mesh.
  Let E be the set of elements in the source mesh, defining the mechanical model.
  Let S be the set of surface nodes in the source mesh.
  Let I be the set of inner nodes in the source mesh.
  Let D be the destination surface mesh.
  for all surface nodes P in S **do**
    compute the projection vector of P on D: $U(P)$.
  end for
  Let $Step$=0.
  repeat
    for all surface nodes P in S **do**
      Compute displaced node P': $P' = P + U(P)/\text{MAX\_STEP}$.
    end for
    Let S' be the set of resulting surface nodes positions
    Compute the inner nodes deformations using the mechanical model E, constrained by the new surface nodes positions S'
    Let I' be the resulting inner nodes positions
    for all elements in E **do**
      Let Q be the quality of E given the new nodes positions S' and I'
      Let Y be the stiffness of the current element
      if Q is not acceptable and $Y <$ Y_MAX **then**
        increase Y in E: $Y = 2 * Y$
      end if
    end for
    if no change in element stiffness have occurred in E **then**
      accept the deformation and proceed to next step: $S = S'$, $I = I'$ and $Step = Step + 1$
    end if
  until $Step$ = MAX_STEP

nodes at each deformation step: the $Young$s modulus arbitrary incrementation prevents the most exposed elements from being excessively deformed and possibly degraded.

The entire process can be seen in algorithm 1. Note that in the Do-While loop, if some elements need to be stiffened, the deformation for a given step is redone using the same initial node positions S and I along with the updated elements E. Another very important remark is that this method do not insert new points to the input mesh.

6 Results

6.1 Quality of the mesh

Measurements of the elements quality are performed before and after the projection onto the surface. Before the projection the quality of each element

ranges between 1.3 and 2.7, which is quite normal since the octree-based algorithm can generate high quality elements. After the projection, 97% of the elements has a quality value between 1.3 and 7. This can be seen in figure 10. In addition, table 1 shows the quality measurement before and after the projection of the nodes onto the input domain. The average quality values don't drastically change from one to another.

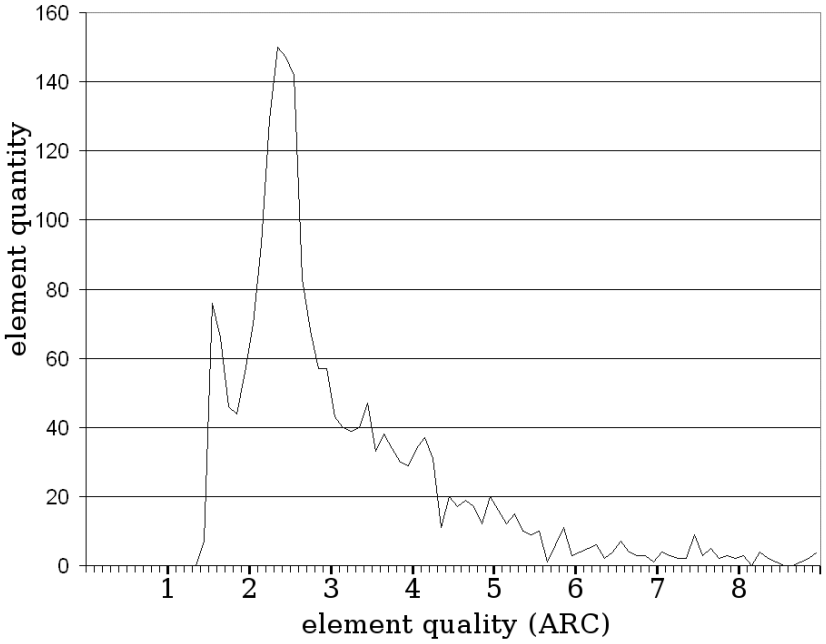

Fig. 10. Quality measurement of the elements after the projection.

The region of interest is mostly represented by hexahedron due to the local refinement of this zone as explained in section 5.2. After projection, the aspect-ratio of the hexahedra is not drastically increased thus the deformation of these elements is reasonable. The same happens with the pyramids and the situation only changes for the tetrahedra, although the aspect-ratio average is still acceptable. We can conclude that the quality in the RoI remains good enough for the simulation. Note that in the presented example no prism was generated. This is because, in our experience, even though they can be found in a mesh, these elements are very rare (just a few patterns consider this type of element).

before projection	Element type	Aspect ratio average	quantity of elements
	Hexahedra	1.27056	72
	Prism	0	0
	Pyramid	2.1387	386
	Tetrahedra	1.58253	1418
	Total	1.68499	1876
after projection	Element type	Aspect ratio average	quantity of elements
	Hexahedra	1.35187	72
	Prism	0	0
	Pyramid	1.7168	386
	Tetrahedra	3.27791	1418
	Total	2.88278	1876

Table 1. Aspect-ratio per element type average before and after the projection onto the surface.

6.2 Modeling

The final mesh is used to simulate the brain-shift with the help of MRI images. Figure 11 shows how the output mesh is finally put together with the initially scanned images.

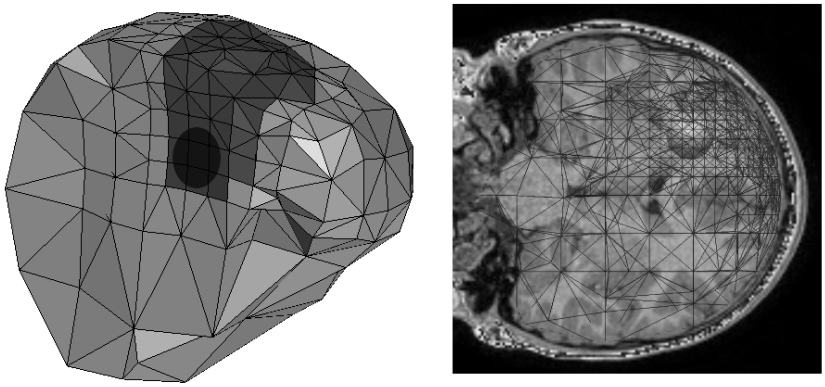

Fig. 11. MRI images with the final mesh.

The tumor is simulated by a sphere and it can be seen how the mesh is congruently more refined in the RoI, i.e. the path between the Opening Skull Point and the tumor.

The FEM deforms the mesh and with this information, the images are updated to show the tissue displacement. A linear elastic small deformation framework is chosen for this computation ($E = 1kPa, \nu = 0.45$). The set of

pictures in figure 12 shows how this is done. The first picture shows the initial shape of the brain following the skull aperture. The four points in the picture are control points used to change the model boundary conditions assuming some measurements of the intra-operative changes. The second picture shows the brain deformations induced by such changes in the model boundary conditions, using the control points. And the third picture simulates the resection (removal) of the tumor.

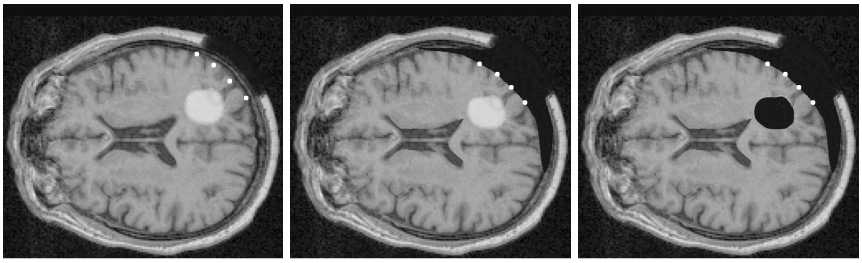

Fig. 12. (a) the initial input with skull aperture, (b) brain-shift simulation using 4 control points and (c) update of the model with tumor resection

As time is crucial in this simulation it is very important to mention that the computation of a deformation is done in 100 ms. This is the process (a) → (b) in figure 12. When there is resection of tissue a main update to the model must be done (re-computation of the elasticity matrix), this operation takes 12 sec for the presented case which is a very acceptable time for surgery. In figure 12 this corresponds to (b) → (c).

Finally figure 13 shows the outputs using different RoIs. In the case of the first one, at the end it has 628 points and the quality coefficient average in the entire mesh is 2.52882. In the second one it has 617 points and a quality coefficient average of 2.9747.

7 Conclusions

This paper aimed at proposing a method to automatically generate a 3D mesh of the brain adapted to the constraints of an intra-operative use, i.e. a good representation of the area between the targeted tumor and the Opening Skull Point, with a coarser mesh elsewhere in order to allow a fast Finite Element computation. The method was successfully evaluated on a given brain geometry with different simulations for tumor and SOP locations. The algorithm generates a mixed-element mesh that achieves correct surface representation. The overall mesh quality is preserved in the final mesh although the termination rule can lead to excessive quality degradation for some elements. In our

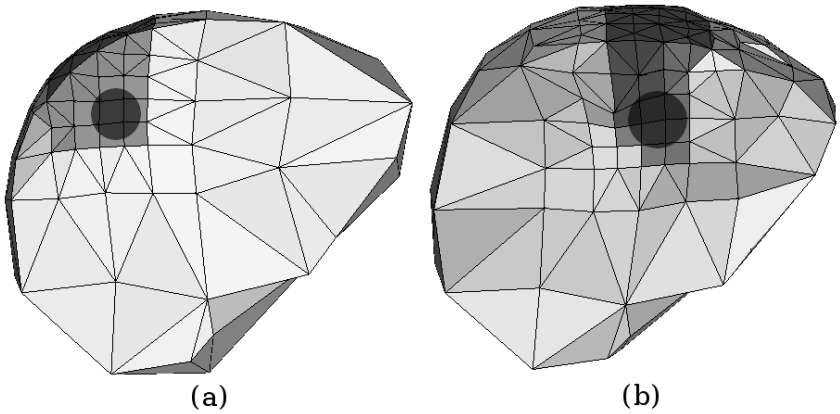

Fig. 13. (a) a refinement with a RoI in the top-back section of the brain and (b) a refinement with a RoI in the top-middle section of the brain.

experience the degraded elements quantity represents less than 0.1% of the total elements count and a subsequent sliver removal procedure [3] should be applied with little impact on the final mesh topology.

This method could also be used to adapt a mesh in an inter-subject context for fitting a generic mesh on specific patient data. In particular, the issues encountered in the Mesh-Matching procedure as presented in [4] and [13] should be overcome by the mechanical approach to the displacement of inner nodes.

8 Acknowledgement

Francisco Galdames and Fabrice Jaillet for the input surface mesh of the brain used in the explanation of the developed algorithm. This project has been financially supported by FONDECYT 1061227, ALFA IPECA project, FONDEF D04-I-1237 and ECOS-Sud C06E04.

References

1. P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun. Variational tetrahedral meshing. *ACM Transactions on Graphics*, 24:617–625, 2005. SIGGRAPH '2005 Conference Proceedings.
2. M. Chabanas, V. Luboz, and Y. Payan. Patient specific finite element model of the face soft tissue for computer-assisted maxillofacial surgery. *Medical Image Analysis*, 7:131–151, 2003. Issue 2.
3. S.W. Cheng, T.K. Dey, H. Edelsbrunner, M.A. Facello, and S.H. Teng. Sliver exudation. *Journal of the ACM*, 47:883–904, 2000.

4. B. Couteau, Y. Payan, and S. Lavalle. The mesh-matching algorithm: an automatic 3d mesh generator for finite element structures. *Journal of Biomechanics*, 33:1005–1009, 2000.
5. B. Delaunay. Sur la sphère vide. *Bull. Acad. Sci. USSR(VII)*, pages 793–800, 1934.
6. M. Ferrant, S.K. Warfield, A. Nabavi, F.A. Jolesz, and R. Kikinis. Registration of 3d intraoperative mr images of the brain using a finite element biomechanical model. In *Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 19–28, London, UK, 2000. Springer-Verlag.
7. P.J. Frey, H. Borouchaki, and P.L. George. Delaunay tetrahedralization using an advancing front approach. In *5th International Meshing Roundtable*, pages 31–46. Sandia National Laboratories, 1996.
8. N. Hitschfeld. Generation of 3d mixed element meshes using a flexible refinement approach. In *Engineering with Computers*, volume 21, pages 101–114, 2005.
9. J. Hu, X. Jin, and L. et al Zhang. Intraoperative brain shift prediction using a 3d inhomogeneous patient-specific finite element model. *Journal of Neurosurgery*, 106:164–169, 2007.
10. X. Li, J-F. Remacle, N. Chevaugeon, and M. S. Shephard. Anisotropic mesh gradation control. In *Thirteenth International Meshing Roundtable*, pages 401–412. Sandia National Laboratories, September 2004.
11. C. Lobos and N. Hitschfeld. 3d noffset mixed-element mesh generator approach. In *14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 47–52, 2006.
12. W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, volume 21, pages 163–169. ACM Press, July 1987.
13. V. Luboz, M. Chabanas, P. Swider, and Y. Payan. Orbital and maxillofacial computer aided surgery: Patient-specific finite element models to predict surgical outcomes. *Computer Methods in Biomechanics and Biomedical Engineering*, 8:259–265, 2005.
14. L. Marechal. A new approach to octree-based hexahedral meshing. *10th International Meshing Roundtable*, pages 209–221, October 2001.
15. M. Nesme, F. Faure, and Y. Payan. Hierarchical multi-resolution finite element model for soft body simulation. *Lecture Notes in Computer Science*, 4072:40–47, july 2006.
16. I. Reinertsen, M. Descoteaux, K. Siddiqi, and D.L. Collins. Validation of vessel-based registration for correction of brain shift. *Medical Image Analysis*, 11:374–388, 2007. doi: http://dx.doi.org/10.1016/j.media.2007.04.002.
17. R. Schneiders. Octree-based hexahedral mesh generation. *Int. J. of Comp. Geom. & Applications*, 10:383–398, 2000.
18. J. R. Shewchuk. What is a good linear element? interpolation, conditioning, and quality measures. In *Eleventh International Meshing Roundtable*, pages 115–126. Sandia National Laboratories, 2002.
19. TetGen. *A Quality Tetrahedral Mesh Generator*. http://tetgen.berlios.de.
20. Y. Zhang and C.L. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric imaging data. In *Thirteenth International Meshing Roundtable*, pages 365–376. Sandia National Laboratories, 2004.