VOLUMETRIC MESH CONSTRUCTION FROM SCATTERED PRIOR DATA: APPLICATION TO CARDIAC MR IMAGE ANALYSIS

Joël Schaerer, Arnaud Gelas, Rémy Prost, Patrick Clarysse, Isabelle Magnin

CREATIS-LRMN, CNRS UMR 5220, Inserm U630, INSA de Lyon, UCB Lyon 1
www.creatis.insa-lyon.fr

ABSTRACT

This paper approaches the problem of volumetric mesh creation from scattered prior data for biomechanical simulations using the Finite Element Method. In order to create high quality meshes, we investigate the use of implicit surface reconstruction methods, together with a novel "biting" meshing algorithm. The proposed method is fast, creates good quality meshes, and works in any dimension, even for complex topologies.

1. INTRODUCTION

The use of patient-specific models is becoming increasingly popular in medical image analysis. In particular, shape models of patients' organs are particularly useful, and can be used either for simulation or medical image analysis. Such models usually rely on a biomechanical model. A popular approach for simulating the physics is to use the finite element method (FEM). However, the FEM requires the discretization of the domain into a mesh. Moreover, mesh regularity is critical to the accuracy and stability of the method. The creation of such a 3D mesh can be a tedious task, especially for complex geometries and/or topologies.

In this paper we propose to use an implicit surface reconstruction method (in the 3D case, a *Multilevel Partition of Unity* [1]) combined with an original "biting" algorithm in order to solve this problem. Implicit surfaces naturally handle the scattered nature of the input data (often composed of manually created priors). The meshing algorithm we propose works well with the implicit surface approach, is fast, simple to implement and generates meshes of good quality.

The paper is organized as follows: In section 2 we describe the deformable model application that motivated our approach. Section 3 discuses previous work in this domain, and in section 4 we describe our proposed method in detail. Finally, we show some sample results in section 5.

2. BACKGROUND: ELASTIC MODEL

A recurring problem in medical image analysis is the automated extraction of structure shape from image data, also known as image segmentation. In order to ease the contour extraction process, one possibility is to use an *a priori* model of the object to be extracted that will be deformed iteratively to fit the image content. This approach is commonly known in image processing as deformable models.

In this paper we focus on a model designed for the simultaneous extraction of both the endocardial and epicardial surfaces in cardiac magnetic resonance images [2]. The concept, named *Deformable Elastic Template*, is a combination of :

- A topological and geometric model of the object to be segmented. In this context, this *a priori* model is a bicavity geometrical mesh that results from the manual segmentation of cardiac ventricles in a reference data set.
- A constitutive equation (elasticity) defining its behavior under applied external image forces that push the model's interfaces towards the image edges.

The equilibrium of the model is obtained through the minimization of the following global energy functional :

$$E = E_{elastic} + E_{data}$$

where $E_{elastic}$ represents the deformation energy of the model and E_{data} is the energy due to the external image forces. Both terms depend on two continuous functions: the displacement u, and a force field f.

2.1 Discretization

These energy terms can be approximated by discretizing the underlying functions. To perform the discretization, we use the finite element method (FEM): the elastic domain Ω is approximated by a polyhedron divided into tetrahedral elements. The displacement and forces are approximated to linear functions on these elements.

Under this approximation, the minimum of the energy must satisfy the following equation:

$$\mathbf{KU} = \mathbf{F}$$

where $\mathbf{K}$ is the stiffness matrix, corresponding to the response of the elastic material, and $\mathbf{U}$ and $\mathbf{F}$ are respectively the displacement and force vectors. This equation is a sparse linear system and can be solved using standard numerical methods.

2.2 Mesh quality

Since the precision of the approximation is directly related to the size of elements, in order to have the same precision everywhere in the domain we need to have uniformly-sized elements. Moreover, the conditioning of the matrix $\mathbf{K}$ depends on the *aspect ratio* of the tetrahedra. The quality of the mesh is thus a very important factor for the precision and the robustness of finite element methods.

3. PREVIOUS WORK

While elastic models have been quite popular in the field of cardiac medical image analysis, the construction of the meshed model has not been discussed a lot. Authors usually rely on existing code [2, 3] or very simple ad-hoc methods [4]. However, these methods are usually sub-optimal for a number of reasons:

- They require a surface mesh to be created first, and are highly dependent on the quality of the surface mesh.
- They lack flexibility and require a quite long tool chain.
- They usually have a non-uniform sizing of elements, smaller near the surface and bigger in the interior. While this is desirable for some applications, it can be detrimental in the context of deformable meshes, where simulation accuracy is important everywhere.

In particular, the creation of a surface mesh from sparsely sampled contours is non-trivial. Indeed surface mesh reconstruction methods require some knowledge about the topology of the surface, or some condition on the sampling. This case is frequently encountered when considering 3D shape reconstruction from medical imaging data.

A good review of volumetric meshing approaches can be found in [5]. "Biting" approaches are considered in [6] in the context of surface meshing. [7] explores the context for 2D domain meshing, and presents some interesting theoretical results. In this paper, we propose to use a similar approach to mesh the inner part of a 3D implicitly defined object, and obviously our approach can be easily extended to n-dimensional meshing.

4. PROPOSAL

Instead of reconstructing a surface mesh from given samples, we first reconstruct an implicit surface, which handles limitations described in the previous section, and offers some interesting properties:

- the reconstructed surface is guaranteed to be manifold,
- an easy and efficient classification test to know whether one point is inside, outside or on the surface.

Thus to create the volumetric mesh from sparse samples, we first reconstruct an implicit function f (section 4.1) which provides continuous information about interior and border of the Domain Ω. Then to mesh Ω, we use a simple yet efficient algorithm to place mesh nodes, inspired by methods described in [7] and [6] (section 4.2.1), and compute a Delaunay triangulation(section 4.2.2) to get an initial volumetric mesh which could be further improved by relaxation methods.

The resulting meshes are very regular, with a quasi-uniform sampling of control points. They are very well suited for solving the finite-element problems required for elastic modeling.

4.1 Implicit Function Reconstruction

An implicit function f representing a given object is defined such that its zero level set represents the shape, i.e. one point p is on the surface if $f(p) = 0$, or inside the object if $f(p) > 0$, else outside.

Implicit function reconstruction has been intensively studied over the last decade [8, 9, 1]. Among all these methods, we decide to use a *Multilevel Partition of Unity* method [1] which allows to reconstruct an implicit function with a given reconstruction error bound and can represent features and high order corners.

It is important to mention that our proposed method can work with any implicit surface reconstruction method, as long as they provide a well defined implicit function f on the whole domain of interest Ω.

4.2 Mesh Creation

4.2.1 *Node Selection*

The first and most important part of mesh creation is the selection of the node points. The approach we propose is based on the extension of the "biting" method described in [7] to k dimensions. Our implementation is based on a grid discretization.

Consider a given implicit function f, and a given d dimensional grid I on which f is sampled. Depending on the implicit function value at its corresponding location p, one grid node n is labeled, by the LABELELEMENTS procedure, as ALIVE (if $f(p) > 0$), or DEAD.

We select one node u from the border between ALIVE and DEAD labeled elements, via the SELECTNODE procedure. The corresponding location p of this node is added to the center set $\mathscr{C}$. Then, all ALIVE nodes n such that the Euclidean distance to u is below R are labeled as DEAD by the KILLNODES procedure. These last processes are iteratively repeated until there are no more ALIVE nodes. The algorithm is summarized in Algorithm 1. An illustration of the algorithm is presented in Figure 1.

Algorithm 1 Biting Algorithm

```
procedure ComputeMeshNodes($f$, $R$, $I$)
    $X \leftarrow \emptyset$
    Alive, Dead $\leftarrow$ LabelElements($f$, $I$)
    repeat
        $u \leftarrow$SelectNode(Alive, Dead)
        $X \leftarrow X \cup \{u\}$
        KillNodes($u$, $R$, Alive, Dead)
    until $Card$(Alive) $= 0$
end procedure
```

The two main parameters of this step are the size of the grid d, and the radius R. The radius R is the minimum distance between nodes, and is very close to the average distance between adjacent nodes (in the Delaunay sense). It is thus related to the precision of the FEM approximation, and should therefore be selected considering the desired simulation accuracy. The size of the grid d determines how closely the reconstructed mesh's geometry will match the original implicit geometry. It should therefore be as high as possible. The limiting factor is available memory, since memory use increases with d^k where k is the considered dimension. It should however be noted that each point of the grid only requires one bit of memory.

4.2.2 *Triangulation*

A Delaunay Triangulation is a triangulation that maximizes the minimum internal angles of tetrahedra. Therefore, for a given set of points $\mathscr{P}$, Delaunay Triangulation provides the best aspect ratio and is very popular for FEM applications.

The *Voronoi cell* of the *site* $p \in \mathscr{P}$ is given as $\mathscr{V}_p = \{x \in \mathbb{R}^3 : \forall\, q \in \mathscr{P} - \{p\},\ \ \|x - p\| \leqslant \|x - q\|\}$. The sets $\mathscr{V}_p$ are convex polyhedra. Faces shared by two Voronoi cells are called *Voronoi faces*, and edges shared by three Voronoi cells are called *Voronoi edges*. The points shared by four or more Voronoi cells are called *Voronoi vertices*. The *Voronoi diagram* $\mathscr{V}_{\mathscr{P}}$ of $\mathscr{P}$ is the collection of all Voronoi cells, faces, edges and vertices.

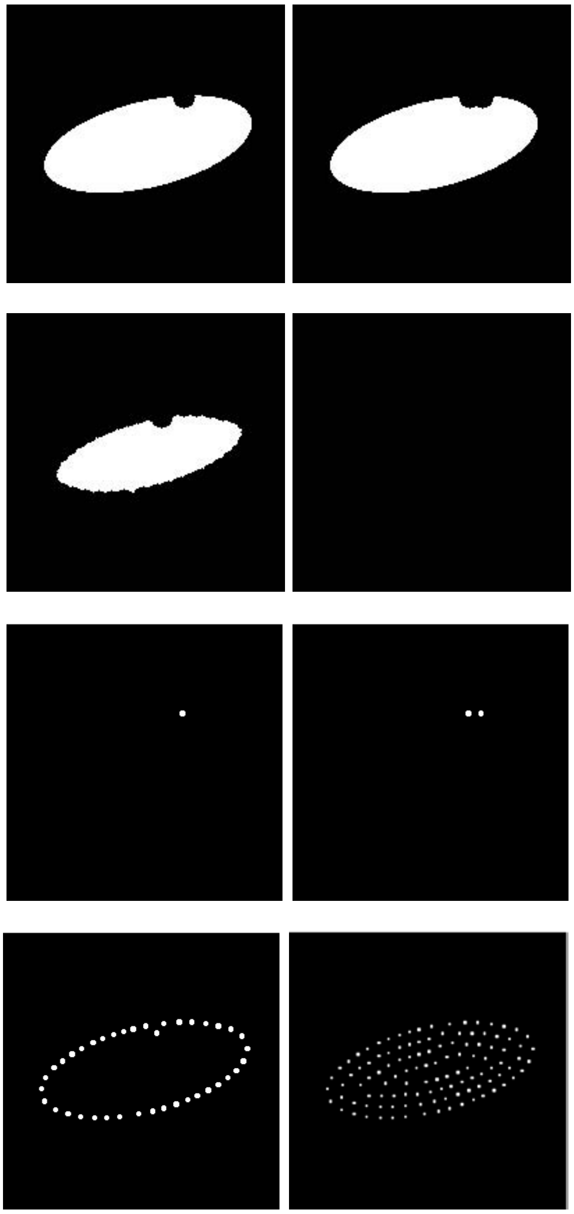


Figure 1: Illustration of the algorithm on a simple 2D problem: the top four figures show nodes labeled as ALIVE, with the corresponding mesh node set X on the bottom four figures, at the two first iteration steps, one intermediary iteration, and upon termination of the algorithm.

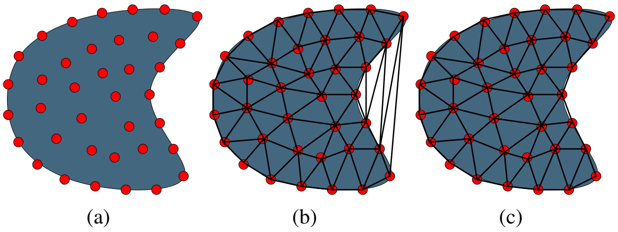


Figure 2: Node selection (a), Delaunay triangulation (b), after spurious cell removal (c).

The *Delaunay Triangulation* $\mathscr{D}_{\mathscr{P}}$ of a set of points $\mathscr{P}$ is dual to the Voronoi diagram $\mathscr{V}_{\mathscr{P}}$. The convex hull of four or more points in $\mathscr{P}$ defines a *Delaunay cell* if the intersection of the corresponding Voronoi cells is not empty and there exists no superset of points in $\mathscr{P}$ with the same property. Analogously, the convex hull of $1 \leqslant k \leqslant 3$ points define a $(k-1)$-dimensional *Delaunay face* if the intersection of their corresponding Voronoi cells is not empty. A 0-, 1-, and 2-dimensional Delaunay face is also called a Delaunay *vertex*, Delaunay *edge*, Delaunay *triangle* respectively. The collection of Delaunay cells and their faces defines a decomposition of the convex hull of all points in $\mathscr{P}$. This decomposition is a triangulation where the Delaunay cells are tetraedra if the points are not aligned.

To compute the Delaunay triangulation of a given point set, several algorithms are available. Here, we used the algorithm proposed in the QHULL library [10].

The resulting mesh contains tetrahedra that are outside of the domain and that should thus be removed. Following [5], one tetrahedron is labeled as outside if its circumcenter c is outside the domain, i.e. $f(c) < 0$, and if the ratio between the distance of the circumcenter to the surface and the radius of the circumsphere is inferior to a certain threshold (see Fig.2). In our experiments, we used 0.4 as the threshold, which is the value used in [5].

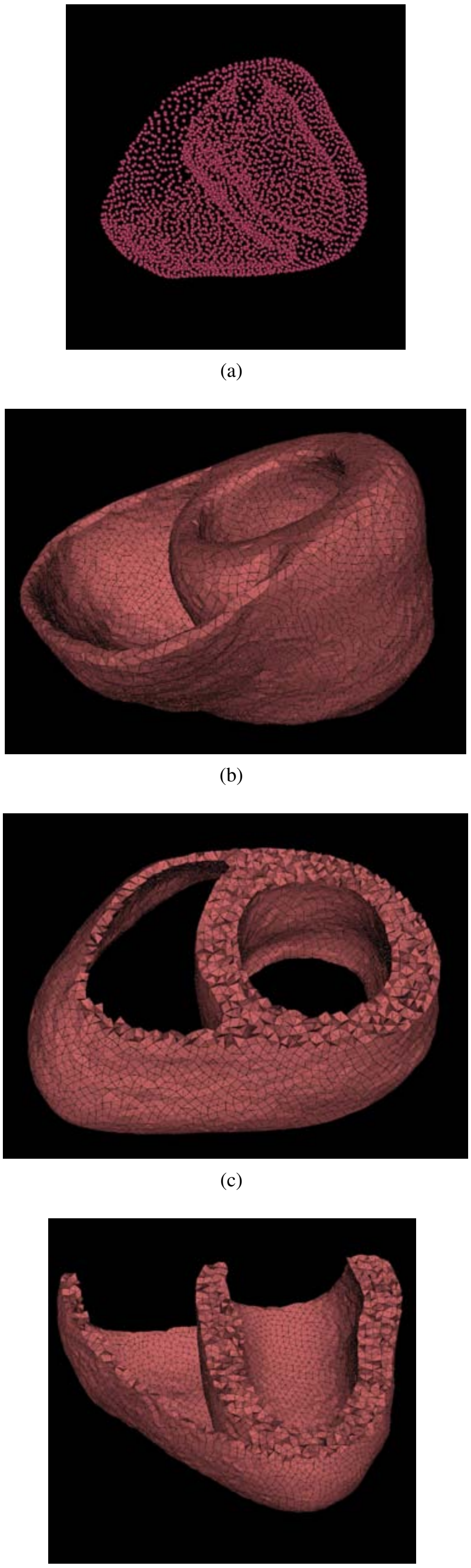
In practice, this criterion is evaluated using the following approximation of the Euclidean distance from c to the implicit surface $\partial\Omega$ [11] :

$$d(c, \partial\Omega) \approx \frac{|f(c)|}{\|\nabla f(c)\|}$$

5. RESULTS

The method was evaluated on both synthetic shapes and real heart data. Figure 4 shows the results of the method on a synthetic die shape. The resulting mesh has approximately 10,000 cells. Note that the tetrahedra have a good aspect ratio and are uniformly sized. The whole process took about 5 seconds on a Pentium M 2.2GHz workstation, without any particular effort to optimize the code.

Figure 3 shows the results on a real dataset composed of the shape of the left and right ventricle of a healthy volunteer. The creation of a high resolution heart mesh took about 5 minutes, in the same conditions as above. The final mesh is comprised of approximately 136,000 cells. The resulting mesh can then deformed using the finite element method to match the heart of a patient. This scheme can be used, for example, for image segmentation.



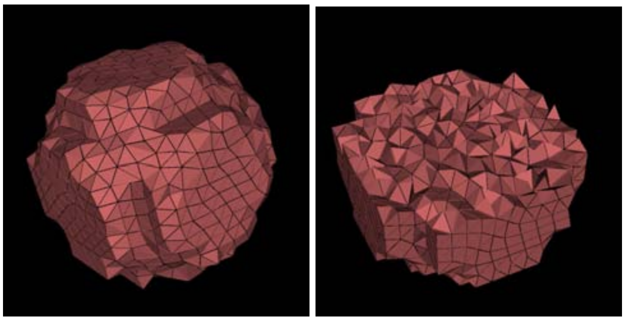
(a)

(b)

(c)

(d)

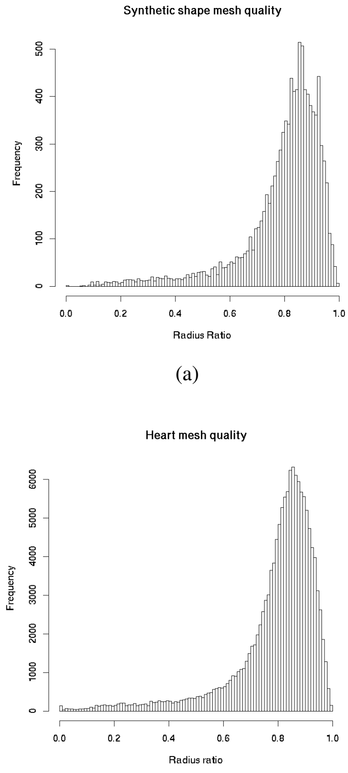
Figure 3: Original scattered point set (lower portion of the heart) (a), Reconstructed volumetric mesh (b), Cut along the short axis (c), Cut along the long axis (d)



(a)

(b)

Figure 4: Exterior mesh of a synthetic die shape (a), Inside view (b)



(a)

(b)

Figure 5: Radius ratio histograms for (a) synthetic and (b) heart meshes

In order to quantitatively assess the quality of elements, a quality criterion Q was computed for each tetrahedron in the generated mesh. We used the ratio between the radius of the insphere and the radius of the circumsphere (multiplied by a factor of 3 for normalization). This criterion has been shown in the past to be a good measure for any kind of degeneracy [5], and varies between 0 (worst) and 1 (perfect mesh). For the synthetic shape mesh, the mean of Q was of 0.80, with a standard deviation of 0.15. For the heart mesh, the mean was of 0.79 with a standard deviation of 0.16. In comparison, [5] obtains average values between 0.86 and 0.88, with a much more complex and computationally-intensive method. Figure 5 shows the radius ratio histograms for both meshes. One significant drawback of our method is that a significant number of slivers remain: the radius ratio of the worst element is very low. This problem could be addressed by adding a sliver removal step, such as the one described in [12], which is proven to eliminate all slivers provided the original points are well spaced.

6. DISCUSSION

The proposed method was shown to generate quality meshes well suited for finite element analysis, using implicit surface reconstruction methods and a "biting" algorithm. The remaining slivers could prove to be a problem; however, according to Cheng [12], they can be completely removed. One advantage of the proposed method is that it can work for any dimension and any domain topology, although in this paper we only showed results on simple topologies in 3D, which are sufficient for our application.

One improvement could be to add a relaxation step after the initial mesh creation. Indeed, the biting scheme ensures good quality elements almost everywhere in the domain, but in a few points adjacent nodes can be more distant than the radius of the biting element. A relaxation scheme would certainly improve the mesh at these locations.

7. CONCLUSION

We have proposed a method to create a volumetric mesh from scattered data points using a implicit surface approach, focusing on a deformable model application. The proposed method creates meshes of good quality and well suited for the envisioned application. It is also very generic and can work on complex topologies in any dimension.

The resulting mesh can be used for finite-element analysis, and is particularly well suited for methods which imply deforming the mesh. In the future we hope to further improve the method by adding a sliver removal method and a relaxation step.

Acknowledgements

The authors would like to thank the ACI-AGIR (http://acimd.labri.fr) and GWENDIA projects for their support. GWENDIA is a project funded by the French National Agency for Research under contract number ANR-06-MDCA-009. This work has also been supported by the Région Rhône-Alpes in the context of project PP3 /I3M of cluster ISLE.

REFERENCES

[1] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, "Multi-level partition of unity implicits," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 463–470, 2003.

[2] Timo Mäkelä, Quoc Cuong Pham, Patrick Clarysse, Jukka Nenonen, Jyrki Lötjönen, Outi Sipilä, Helena Hänninen, Kirsi Lauerma, Juhani Knuuti, Toivo Katila, and Isabelle E. Magnin, "A 3-D model-based registration approach for the PET, MR and MCG cardiac data fusion," *Medical Image Analysis*, vol. 7, pp. 377–389, 2003.

[3] Sermesant M, Forest C, Pennec X, Delingette H, and Ayache N, "Deformable biomechanical models: application to 4d cardiac image analysis.," *Medical Image Analysis*, vol. 7, pp. 475–88, 2003.

[4] A. Montillo, D.N. Metaxas, and L. Axel, "Automated Segmentation of the Left and Right Ventricles in 4D Cardiac SPAMM Images," in *MICCAI*, 2002.

[5] Pierre Alliez, David Cohen-Steiner, Mariette Yvinec, and Mathieu Desbrun, "Variational tetrahedral meshing," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 617–625, 2005.

[6] J. Lötjönen, P.-J. Reissman, I.E. Magnin, J. Nenonen, and T. Katila, "A triangulation method of an arbitrary point set for biomagnetic problems," *IEEE Transactions on Magnetics*, vol. 34, pp. 2228–2233, 1998.

[7] Xiang-Yang Li, Shang-Hua Teng, and Alper Üngör, "Biting: advancing front meets sphere packing," *International Journal for Numerical Methods in Engineering*, vol. 49, no. 1-2, pp. 61–81, 2000.

[8] S. Muraki, "Volumetric shape description of range data using "blobby model"," in *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. 1991, pp. 227–235, ACM Press.

[9] G. Turk and J. F. O'Brien, "Shape transformation using variational implicit functions," in *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1999, pp. 335–342, ACM Press/Addison-Wesley Publishing Co.

[10] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996, http://www.qhull.org.

[11] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, 1991.

[12] Siu-Wing Cheng, Tamal K. Dey, Herbert Edelsbrunner, Michael A. Facello, and Shang-Hua Teng, "Sliver exudation," in *Symposium on Computational Geometry*, 1999, pp. 1–13.