

# Embedding Segmented Volume in Finite Element Mesh with Topology Preservation

Kazuya Sase<sup>1</sup>, Teppei Tsujita<sup>2</sup>, and Atsushi Konno<sup>1</sup>(✉)

<sup>1</sup> Hokkaido University, Sapporo, Japan  
sase@scc.ist.hokudai.ac.jp, konno@ssi.ist.hokudai.ac.jp

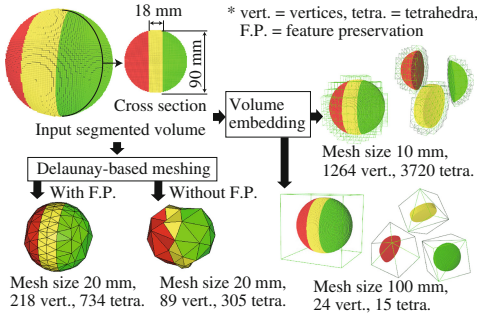
<sup>2</sup> National Defence Academy, Yokosuka, Japan  
tsujita@nda.ac.jp

**Abstract.** The generation of a patient-specific finite element (FE) model of organs is important for preoperative surgical simulations. Although methods for generating a mesh from a 3D geometric model of organs are well established, the reproduction of complex structures, such as holes, branches, and jaggy boundaries, remains difficult. To approximate the deformation of complex structures, an approach for embedding a fine geometry in a coarse volumetric mesh can be used. In this paper, we introduce a volume embedding method that preserves the topology of a complicated structure on the basis of segmented medical images. Our evaluation shows that the generated FE model precisely reproduces the topology of a human brain according to a segmented medical image.

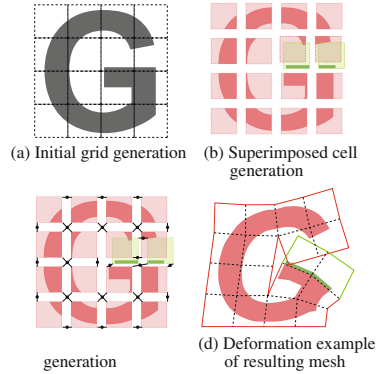
## 1 Introduction

The progress in modeling and simulation techniques of soft-tissue deformation has enabled the prediction of the mechanical behaviors of organs before actual surgery in the operation room. One of the problems in applying these techniques to clinical use is the generation of patient-specific biomechanical models including volumetric meshes. In general, the geometry of a patient's organs is obtained using medical imaging and segmentation techniques. To obtain patient-specific volumetric meshes, meshing methods that use segmented medical images as the input have been developed [2, 8]. However, preserving the fine geometry in a coarse mesh resolution is still difficult. The left side of Fig. 1 illustrates examples of Delaunay-based meshing of a segmented volume. As shown in this figure, the meshing with feature preservation provides a good volume mesh with boundary conformity. However, the number of vertices tends to become large when the feature constraints are applied. Finer mesh resolutions limit the range of applications because of their high computational costs.

In the community of computer graphics, embedding is a popular approach for approximating the deformations of fine geometry. By embedding a fine geometry in a coarse volumetric mesh, the deformation of the fine geometry is interpolated by the deformation of the volumetric mesh. Usually, coarse volumetric meshes are simple grid meshes, and thus, do not meet the boundary conformity. Instead of being limited by the inaccurate boundary conformity, it is easy to reduce



**Fig. 1.** Comparison between Delaunay-based meshing [2,8] and our volume embedding.



**Fig. 2.** Algorithm overview.

the computational cost of mechanical simulations by changing the resolution of the grid. The right side of Fig. 1 illustrates examples of the volume embedding approach. They are able to reduce the number of vertices down as far as 24 vertices in this example. One known issue is that a simple grid mesh cannot separate disconnected parts if proximate parts are included in the same cell. Nesme et al. [7] solved this problem by the separation and superposition of elements, but their method was intended for embedding a polygon surface model in a finite element (FE) mesh. In order to apply the method to volume data (medical images), they need to be converted into a surface polygon, which results in loss of volume information. Additionally, the method proposed in [7] did not consider the separation of a completely attached boundary that should be separated.

To solve these problems, we propose an embedding method that directly handles the volume data. Furthermore, we introduce a method of separating completely attached areas on the basis of user-defined segment pairs. The proposed method is evaluated using a brain atlas, and the generated mesh is tested whether it can be used for our interactive surgery simulator [9], which aims to plan the approaching process to the affected area on the insula. The simulator requires a mesh of which the Sylvian fissure is separated, and the fissure is generally completely attached in MR images.

## 2 Method

### 2.1 Overview

The proposed method uses superimposed nodes and cells to preserve the topology of the structure. As illustrated in Fig. 2, the degrees of freedom of the deformation are added by superimposing nodes and cells. The topology, i.e., the connections and separations between the local volumetric areas, can be preserved using

superimposed nodes and cells. Additionally, we consider the separation of the connection between multiple segments. For example, the “Temporal lobe” segment and the “Parietal lobe” segment should be separated. Such separations are realized by separation label pairs (SLPs), which are explicitly specified by the user. The hexahedral mesh generation is processed according to the segmented volume and SLPs. The definitions of the input and output are given below.

**Segmented medical image.** A segmented medical image constitutes volume data that contain labels at aligned voxels. A label is an integer value associated with a segment, for example, 0 is empty space, 1 is white matter, and so on. A label at a specific voxel coordinate is denoted by  $L(\mathbf{i}) \in \mathbb{Z}$ , where  $\mathbf{i} \in \mathbb{Z}^3$  is a voxel coordinate. Voxel coordinate  $\mathbf{i}$  can be mapped to spatial coordinate  $\mathbf{p} \in \mathbb{R}^3$  using a vector of volume origin  $\mathbf{p}_0 \in \mathbb{R}^3$  and a vector of spacing values  $\mathbf{s} = [s_x, s_y, s_z] \in \mathbb{R}^3$  as  $\mathbf{p} = \mathbf{p}_0 + \mathbf{s} \odot \mathbf{i}$ , where  $\odot$  denotes element-wise vector multiplication.

**Mesh size.** Mesh size  $H \in \mathbb{R}$  is the approximate edge length of a hexahedral cell. The actual edge lengths in the  $x$ ,  $y$ , and  $z$  directions  $\mathbf{h} = [h_x, h_y, h_z]$  are determined by multiples of the spacing value of a volume, e.g.,  $h_x = \text{ceil}(H/s_x)s_x$  for the direction  $x$ , where  $\text{ceil}(H/s_x)$  is the number of voxels along the  $x$  axis.

**SLPs.** An SLP is a pair of labels  $\{L_a, L_b\}$ , where  $L_a$  and  $L_b$  are the labels of segment  $a$  and  $b$ , respectively. The algorithm for generating a hexahedral mesh is applied to separate the segments that are specified by SLPs. The SLPs need to be generated by users.

**Hexahedral mesh.** A mesh is represented as nodes and cells. A node has a corresponding position and a cell has references to nodes.

**Superimposed nodes and cells.** A superimposed node/cell is a node/cell that coexists at the same position as another node/cell. There is no limit to the number of superimposed nodes/cells that can exist at the same position.

## 2.2 Hexahedral Mesh Generation

First, initial nodes and cells are generated without considering superposition (Fig. 2(a)). The bounding box of the volume is calculated, and an orthogonal lattice is generated, the origin of which is on the corner of the bounding box; the lattice bases are  $[h_x, 0, 0]^T$ ,  $[0, h_y, 0]^T$ ,  $[0, 0, h_z]^T$ .

Second, a cell that includes multiple regions is divided into multiple cells such that each divided cell includes only one region (Fig. 2(b)). To do so, local segmentation is executed by voxel-level region growing in each cell, as illustrated in Fig. 3. For the region growing, first an initial seed point is arbitrarily selected from voxels inside the cell. Then, it is determined whether the labels of the neighbor are connected to one of the faces of the seed voxel. If the label of a neighbor voxel is not “empty” and is not listed in the SLPs, the voxel is added to the region. These procedures are iterated until no connected voxel is found. After one region is extracted, the region growing procedure is iterated until all non-empty voxels are added to a region. If more than one region is detected,

the cell for each region is superimposed. At this time, all cells refer to the same set of nodes. For the subsequent procedure, the set of voxels is stored on the associated region for each superimposed cell. If no region is detected, e.g., all the labels are “empty,” the cell is deleted.

Third, nodes shared by superimposed cells are also superimposed (see Figs. 2(c) and 4 for details). At each node, cells that share the node are searched and cell-level region growing is executed. The cell-level region growing is initiated from a seed point (cell), evaluates connectivities with surrounding cells, and finds connecting cells. To determine the cell connectivity, the pairs of neighbor labels on the boundary of two cells are used. If a label pair does not include an empty label and is not listed in the SLPs, the pair is recognized as a connected pair. If at least one connected pair is detected, the two cells are determined to be connected. If more than one region (connected cells) is detected, the node is replicated for each region and the corresponding node references are changed in the cells of the region to the superimposed node. If no region is detected, the node is deleted.

Optionally, floating cells are deleted. In many cases, a segmented medical image includes isolated small segments. Such segments may cause unmeaningful small pieces in mechanical simulation. To avoid this, we count the number of cells for each cell island (a set of connected cells). If the number is smaller than a user-specified threshold, the cells are deleted. From our experience, an adequate threshold is two cells.

### 2.3 Physics Simulation and Visualization

In this study, the generated meshes were validated by performing finite element method (FEM) simulation based on a corotational formulation and implicit time integration scheme for the calculation of dynamic soft-tissue deformation [6]. Each generated hexahedral cell is divided into five first-order tetrahedral elements and a linear elastic property is applied. The values related to material properties are the same for all elements and we adopts the property of a soft material: Young’s modulus 1000 Pa, Poisson’s ratio 0.4, density 1.0 g/cm<sup>3</sup>.

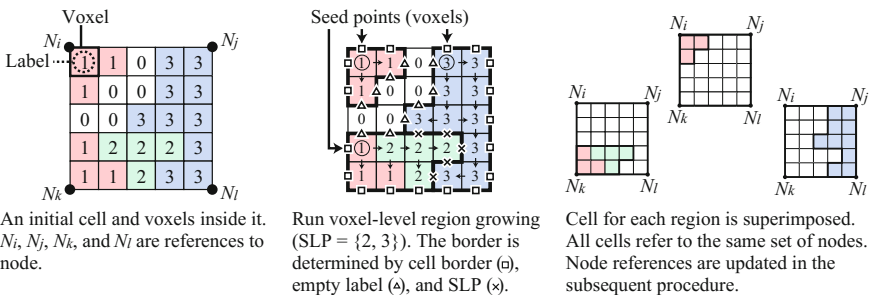


Fig. 3. Superimposed cell generation using voxel-level region growing.

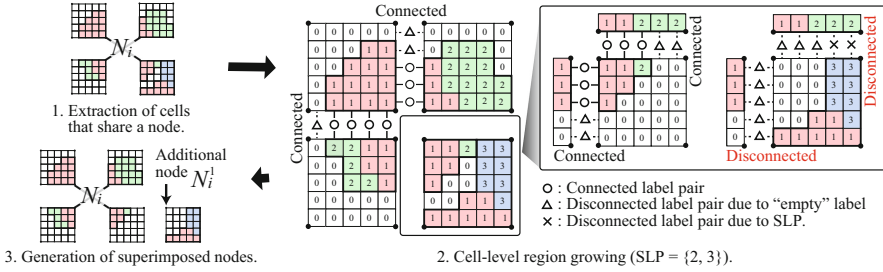


Fig. 4. Superimposed node generation using cell-level region growing.

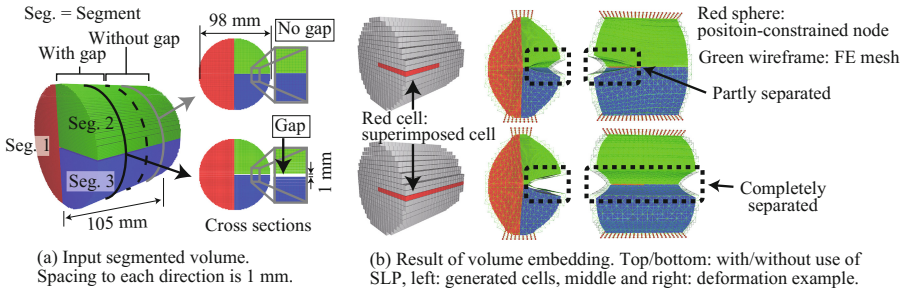


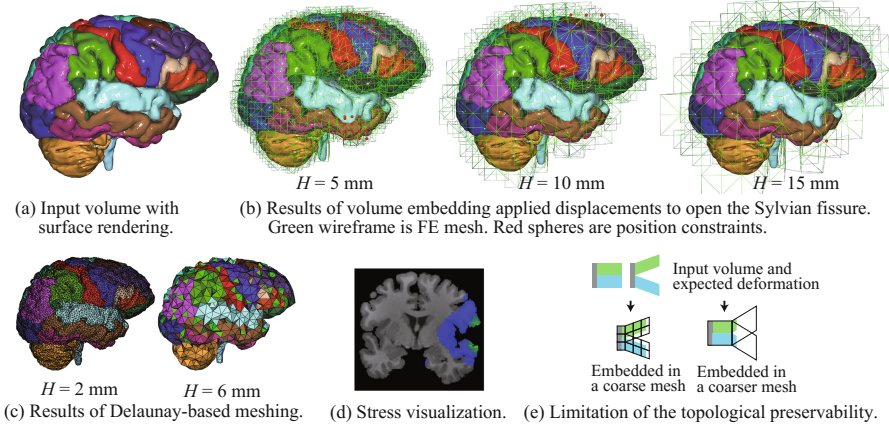
Fig. 5. Evaluation using a cylinder-shaped segmented volume. (Color figure online)

For graphics rendering, the surface polygons of segmented volumes are generated using the marching cubes method. As in [6, 7], the polygons are deformed according to the deformed FE mesh obtained by the FEM simulation. In this method, each surface vertex is associated with a tetrahedral element in advance and affine transformation is applied to the vertex to hold the initial barycentric coordinate in the associated tetrahedron. The deformation of volume data is realized by considering voxels as particles. The transformations of the particles are performed in the same way as for surface polygon vertices.

### 3 Results and Discussions

The proposed method was evaluated using a cylinder-shaped segmented volume and brain atlas dataset published in [4]. All numerical experiments were executed by an implementation on a workstation with an Intel Core i7-3960X (6 cores, overclocked to 4.5 GHz), 64 GB of RAM, and two GPUs, an NVIDIA K20c (2,496 CUDA cores) for the computing of FEM and an NVIDIA Quadro K5000 (1536 CUDA cores) for graphics processing. The proposed meshing algorithms were parallelized using OpenMP.

**Cylinder Model.** Figure 5 shows the result of the evaluation using a cylinder-shaped segmented volume. As shown in Fig. 5(a), the input volume has three



**Fig. 6.** Evaluation using a brain atlas [4].

segments labeled 1, 2, and 3. In the front-half part of the volume, there is a groove between segments 2 and 3. The width of the groove is equal to the length of a voxel. In the rear-half part, segments 2 and 3 are completely attached without any gap. Figure 5(b) shows the two results of the proposed volume embedding: the top part shows the result without SLP and the bottom part shows the result with SLP  $\{2, 3\}$ . Both of the results were obtained using a mesh size of  $H = 5$  mm. The left part of Fig. 5(b) shows the generated cells. In the figure, gray and red cells indicate normal and superimposed cells, respectively. The right part of Fig. 5(b) shows the deformation examples of the meshes from two different views. In this figure, it is clear that the groove was separated even when the SLP was not used. However, without SLPs, the completely attached boundary of segments 2 and 3 was not separated. In contrast, the separation of the boundary was achieved using the SLP. This result shows that the use of SLPs is effective for separating completely attached segments.

**Brain Atlas.** Figure 6 shows the results of the evaluation using a brain atlas [4]. Figure 6(b) shows the results of the proposed method. The input SLPs were manually specified by the author to separate the Sylvian fissure in order to utilize the generated mesh in our neurosurgery simulator [9]. The displacements to open the Sylvian fissure was imposed after the FE mesh generation and calculated the shape of the equilibrium. To compare the proposed method with a well-established method, Delaunay-based meshing without feature preservation [8] implemented in CGAL library [1] was also conducted (Fig. 6(c)). The quantitative results are described in Table 1. Note that the measurement of the computational time of the FEM simulation (global matrix assembly and linear system solving with 20 CG iterations [9]) is performed only for our embedding method, and in the case of our method with  $H = 2.0$ , the simulation was not executed because of out of GPU device memory due to too large mesh.

**Table 1.** Quantitative results of the evaluation using the brain atlas ( $256 \times 256 \times 256$ ) [4].  $N_{\text{vert}}$  and  $N_{\text{tet}}$  are the number of vertices and tetrahedra,  $T_{\text{mesh}}$  and  $T_{\text{fem}}$  are the computational time taken for the mesh generation and a loop of FEM simulation, respectively.

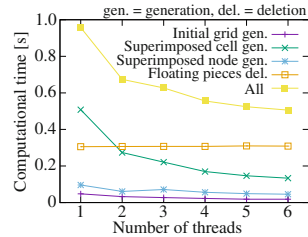
Method	Mesh size (mm)	$N_{\text{vert}}$	$N_{\text{tet}}$	$T_{\text{mesh}}$ (s)	$T_{\text{fem}}$ (ms)
Pons, et al. [8]	2.0	149,506	799,165	22.23	–
Pons, et al. [8]	4.0	44,421	95,740	7.43	–
Pons, et al. [8]	6.0	6,106	29,144	1.28	–
Ours	2.0	227,225	925,360	5.78	–
Ours	5.0	22,999	79,650	0.57	133.3
Ours	10.0	4,829	14,130	0.19	30.8
Ours	15.0	2,072	5,530	0.16	11.9

Delaunay-based meshing with a resolution that is too low, as noted in [8], produced a non-manifold mesh that included singular vertices for which the thickness is zero, which leads to instability in FEM. Further, small or thin segments vanished in the resulting mesh. In contrast, the proposed method generated meshes that cover the entire body of the volume and FEM simulation was conducted without instability. Figure 6(c) shows that the frontal and temporal lobes are separated and the topology of the Sylvian fissure is correctly preserved.

Figure 7 shows the parallel scalability of each algorithm stage of the mesh generation by timing the executions with different number of threads. The algorithms for “superimposed cell generation” and “superimposed node generation” were parallelized straightforwardly because they are cell- and node-independent, and their good scalability can be observed. Thanks to this performance, hexahedral mesh generations were finished in several seconds, as shown in Table 1. This enables us to find a balance between accuracy and performance by modifying the mesh size effectively. In our implementation of FEM, a mesh with 5,000 nodes can be calculated in real-time rate (30 fps). Therefore, we were able to determine that a mesh with  $H = 10.0$  mm is adequate for interactive simulation.

Figure 6(d) shows a postprocess application example using 3D Slicer [3] that visualizes the stress field inside the brain. This was obtained by exporting a deformed volume with mesh size 5 mm. This is valuable, for example, when objectively assessing the damage of a retraction.

However, there is a limit on the topological preservability. When mesh size  $H$  increases, small boundaries with lengths that are smaller than  $H$  were connected (Fig. 6(e)). Thus, the mesh size needs to be determined considering the lengths of boundaries that should be preserved.



**Fig. 7.** Parallel scalability.

## 4 Conclusion and Future Work

In this paper, we proposed a volume embedding method that preserves the topology of a structure according to a segmented medical image and user-defined SLPs. The method was evaluated in terms of its ability to approximate deformations and computational time. The evaluation shows that the proposed method can generate topology-preserved meshes fast and robustly.

There are some problems to address in future. The proposed method ignores the boundary conformity of the FE mesh and decreases in the precision of the mechanical simulations. These issues can be mitigated by modifying the stiffness matrix according to the spatial distribution of labels inside each cell [7]. Furthermore, contact handling can be addressed by placing collision proxy points inside the volumetric mesh [5]. These treatments would enhance the quality of the simulations.

**Acknowledgements.** This work was supported by the Japan Society for the Promotion of Science (JSPS) through the Grant-in-Aid for Scientific Research (A) (15H01707) and the Grant-in-Aid for JSPS Fellows (15J01452).

## References

1. CGAL: Computational Geometry Algorithms Library. <http://www.cgal.org/>
2. Boltcheva, D., Yvinec, M., Boissonnat, J.-D.: Mesh generation from 3D multi-material images. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009. LNCS, vol. 5762, pp. 283–290. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-04271-3\\_35](https://doi.org/10.1007/978-3-642-04271-3_35)
3. Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., Buatti, J., Aylward, S., Miller, J.V., Pieper, S., Kikinis, R.: 3D Slicer as an image computing platform for the quantitative imaging network. *Magn. Reson. Imaging* **30**(9), 1323–1341 (2012)
4. Halle, M., Talos, I.F., Jakab, M., Makris, N., Meier, D., Wald, L., Fischl, B., Kikinis, R.: Multi-modality MRI-based atlas of the brain (2015). <https://www.spl.harvard.edu/publications/item/view/2037>
5. McAdams, A., Zhu, Y., Selle, A., Empey, M., Tamstorf, R., Teran, J., Sifakis, E.: Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.* **30**(4), 37:1–37:12 (2011)
6. Müller, M., Gross, M.: Interactive virtual materials. In: *Proceedings of Graphics Interface 2004*, pp. 239–246 (2004)
7. Nesme, M., Kry, P.G., Jeřábková, L., Faure, F.: Preserving topology and elasticity for embedded deformable models. *ACM Trans. Graph.* **28**(3), 52:1–52:9 (2009)
8. Pons, J.-P., Ségonne, F., Boissonnat, J.-D., Rineau, L., Yvinec, M., Keriven, R.: High-quality consistent meshing of multi-label datasets. In: Karssemeijer, N., Lelieveldt, B. (eds.) *IPMI 2007*. LNCS, vol. 4584, pp. 198–210. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-73273-0\\_17](https://doi.org/10.1007/978-3-540-73273-0_17)
9. Sase, K., Fukuhara, A., Tsujita, T., Konno, A.: GPU-accelerated surgery simulation for opening a brain fissure. *ROBOMECH J.* **2**(1), 1–16 (2015)