Consistent Mesh Generation for Non-Binary Medical Datasets

Bernhard Reitinger, Alexander Bornik and Reinhard Beichel

Institute for Computer Graphics and Vision
Graz University of Technology, 8010 Graz, Austria
Email: breiting@icg.tu-graz.ac.at

Abstract. This paper presents a new approach for constructing a consistent (non-manifold) mesh based on non-binary datasets. Mesh generation is the method of choice if a surface is extracted based on labeled volumetric datasets. In most cases, the traditional Marching Cubes algorithm or one of its extensions is applied for generating meshes of binary labeled input data. However, for different – especially medical – applications, non-binary datasets with more than two differently labeled voxels can be adjacent in a rectilinear grid. For these cases, the traditional Marching Cubes algorithm fails. Additional to the mesh generation algorithm, an extension for a well-known geometric filter is presented avoiding ridges in the mesh.

1 Introduction

Mesh generation of labeled (segmented) volumetric datasets is an often used technique for medical applications where the surface of a target object is subject to visualization. The traditional method is to apply the Marching Cubes (MC) algorithm proposed by Lorensen and Cline in [1] or some of its variations like [2, 3, 4] to generate a triangular mesh. All of these methods are usually concerned about 2-manifolds which only allow surface extraction between two different labels (binary classification).

However, in many cases a non-binary classification (a volumetric dataset containing different objects, each with a different label) are provided where coherent surfaces must be reconstructed consistently. For instance, a liver dataset with eight different labels indicating the liver segments is taken as input. The goal is to generate a mesh model covering the liver’s boundary and all interfaces between segments. A naïve solution for this problem would be to apply the MC iteratively on each labeled region while masking out all other regions. Although all interfaces would be extracted, a lot of inconsistencies are expected between adjacent regions. The problem is that the MC algorithm can only produce 2-manifolds (homeomorphic to a sphere) and not non-manifolds required for multi-labeled datasets.
2 Related Work

Only few papers address methods for generating meshes based on multi-labeled datasets. Bloomental and Ferguson described in [5] one of the first approaches in this field. Their work is based on implicit surface modelling and computational solid geometry. By subdividing cubic cells into tetrahedra, a triangulation is constructed algorithmically. This approach generates a large amount of triangles which can also be degenerated and not well-shaped. Hege et al. presented a generalized Marching Cubes algorithm in [6]. Their method relies on a intensity-based dataset where probabilities are assigned to each voxel. By trilinear interpolation, a large amount of intermediate triangles are generated in order to fit the surface. Different to these methods, this paper presents an efficient approach to generate well-shaped triangular meshes based on labeled datasets.

3 Methods

The input for our method is a rectilinear (multi-labeled) volumetric dataset with a label assigned to each voxel. A zero label indicates background (outside of the domain) and all other labels unequal to zero denote objects. Eight adjacent voxels build one volumetric cell.

The main idea of our algorithm is based on a domain subdivision strategy. If a cell is homogeneous, nothing must be done. However, a non-homogeneous cell with two or more different labels is subdivided into eight equally sized sub-cells $S_i$ (see Figure 2(a)). The corresponding voxel label is assigned to each $S_i$. If two adjacent sub-cells $S_i$ and $S_j$ have different labels, an interface consisting of two triangles is generated (see Figure 2(b)). Within one cell, up to 12 interfaces (24 triangles) can be generated depending on the number of different labels.

For consistent mesh generation it is necessary to avoid duplicates. All generated vertices within one cell must be coordinated with vertices of adjacent cells. We use an efficient hash map for storing vertex locations only once in a compact data structure. The generated mesh itself only consists of indexed triangles pointing to this vertex map.

3.1 Filtering

Since no interpolation scheme is used for the generation algorithm, the mesh may show staircase characteristics especially for low input resolution. In order to get a visual appealing output mesh, filtering is applied. Surface filters are preferred as they only affect geometry and do not alter topology. Beside smoothing, geometric filters are also used for improving the overall quality of the mesh. The most simple but effective smoothing filter is the Laplacian filter [7]. Each vertex at position $x_i$ is smoothed iteratively by using the following formula:

$$x_{i+1} = x_i + \lambda \Delta_i$$

where $\Delta_i$ is defined as
**Fig. 1.** Cell subdivision into eight sub-cells. (a) Each sub-cell contains the label of its corner voxel. (b) Generated faces of the example cube shown in (a).

![Diagram](image)

(a)  
(b)

**Table 1.** Quantitative results of the mesh generation tested for two different medical datasets.

<table>
<thead>
<tr>
<th>dataset</th>
<th>size</th>
<th>#vtx</th>
<th>#f</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>liver</td>
<td>256 x 256 x 174</td>
<td>632,272</td>
<td>127,208</td>
<td>18.5</td>
</tr>
<tr>
<td>lung</td>
<td>128 x 128 x 148</td>
<td>331,034</td>
<td>668,624</td>
<td>6.1</td>
</tr>
</tbody>
</table>

\[
\Delta_i = \sum_{j=1}^{N} w_{ij} (x_j - x_i)
\]

\(w_{ij}\) specifies the weight between \(x_i\) and its neighboring \(x_j\). A good choice for \(w_{ij}\) is \(\frac{\lambda}{N}\), where \(N\) is the number of adjacent vertices. The scale factor \(\lambda\) (\(0 < \lambda < 1\)) influences the degree of smoothness and is defined constant for all vertices. The sum of all \(w_{ij}\) must be one.

### 3.2 Mesh Generation

The problem of this filter, if applied to a non-manifold, is that ridges can occur at objects’ boundaries. This is because each vertex is treated equally for filtering. An example can be seen in Figure 3(b) which shows a liver dataset with ridges at junctions of segment interfaces and liver boundary. For this reason, a constraint is introduced so that all vertices of the object’s boundary can only be moved within the boundary and cannot be attracted by vertices of interface boundaries. As a result, ridges at the boundary are avoided (see Figure 3(c)).

We observed that singularities can occur if equally labeled voxels are only 26-connected which cause problems when filtering the surface. Two differently labeled regions are only connected through one vertex and filtering would generate unwanted spikes. In the current implementation we assign a freeze label to these vertices which means that they are not moved during smoothing and avoids therefore these spikes.
4 Results

We tested our algorithm on various input datasets. However, the target application is the liverplanner [8] where liver segment boundaries based on a prior segment classification [9] need to be visualized. Additionally we also tested our method with a labeled sheep lung dataset.

Table 1 shows the summary of one liver and one lung dataset. The $\#\text{vtx}$ and $\#\text{f}$ indicate the number of generated vertices and faces (triangles). The timings are measured in seconds. The output result for the liver dataset can be observed in Figure 2. Figure 2(a) shows the raw output of the mesh generation, Figure 2(b) and 2(c) are the results after applying a standard/modified Laplacian filter
with \( \lambda = 0.88 \) and 30 iterations. In Figure 2(d) the interior interfaces between liver segments are displayed.

5 Discussion

This paper presented an algorithm for generating non-manifold meshes based on a non-binary classification of volumetric datasets. Since the mesh generation itself produces staircase artifacts, a modified version of the Laplacian filter is applied. This prevents from generating ridges at junctions between the object’s boundary and interior interface boundaries. As the output surface is consistent and does not contain duplicates, the result can be used for further volumetric mesh generation.

Acknowledgements

This work was supported by the Austrian Science Foundation (FWF) under grant P17066-N04.

References