Improving Triangle Mesh Quality with SurfaceNets

P.W. de Bruin[1], F.M. Vos[2], F.H. Post[1], S.F. Frisken-Gibson[3], and A.M. Vossepoel[2]

[1] Computer Graphics & CAD/CAM group, Faculty of Information Technology and Systems, Delft University of Technology
[2] Pattern Recognition Group, Department of Applied Physics, Delft University of Technology
[3] MERL – a Mitsubishi Electric Research Laboratory, Cambridge, MA, USA

Abstract. Simulation of soft tissue deformation is a critical part of surgical simulation. An important method for this is finite element (FE) analysis. Models for FE analysis are typically derived by extraction of triangular surface meshes from CT or MRI image data. These meshes must fulfill requirements of accuracy, smoothness, compactness, and triangle quality. In this paper we propose new techniques for improving mesh triangle quality, based on the SurfaceNets method. Our results show that the meshes created are smooth and accurate, have good triangle quality, and fine detail is retained.

Keywords: *Surgical simulation, surface extraction, tissue deformation modelling, visualization, SurfaceNets*

1 Introduction

In recent years, endoscopic surgery has become well established practice in performing minimally-invasive surgical procedures. In training, planning, and performing procedures, pre-operative imaging such as MRI or CT can be used to provide an enhanced view of the restricted surgical field. Simulation of intra-operative tissue deformation can also be used to increase the information provided by imaging. However, accurate simulation requires patient-specific modeling of the mechanical behavior of soft tissue under the actual surgical conditions.

To derive an accurate and valid model for intra-operative simulation, we propose a five-stage process:

1. Image data acquisition (MRI, CT)
2. Image segmentation
3. Deformable tissue model generation
4. Intra-operative simulation of tissue deformation, guided by actual surgical conditions and/or intra-operative measurements conditions
5. Enhanced intra-operative visualization

In order to simulate tissue deformation, many authors have proposed finite element (FE) analysis of the relevant structures (see for example [2, 1, 8]). The FE models are commonly initialized by supervised segmentation of preoperative image data, resulting in a classification accurate to the pixel level. Using a surface

S.L. Delp, A.M. DiGioia, and B. Jaramaz (Eds.): MICCAI 2000, LNCS 1935, pp. 804–813, 2000.

extraction technique such as the Marching Cubes algorithm [9], the result is converted into a set of triangular meshes representing the surfaces of relevant organs. Such a representation can then be imported into an environment for FE analysis.

For optimal mechanical modelling and visualization, the triangular surface models should meet the following requirements :

- *Accuracy:* the representation of the organ surface geometry should be sufficiently accurate;
- *Smoothness:* the model should conform to the smooth organ boundaries. Sharp corners should be avoided as these can cause disturbing artifacts such as stress concentrations;
- *Compactness:* to achieve fast response times, the number of elements (triangles) in the model should be minimal; the resolution of the triangle mesh should be considerably lower than the medical image, with minimal loss of accuracy;
- *Triangle quality:* the shape of the triangles in the mesh should be as near as possible to equilateral to avoid FE errors and visualization artifacts.

Segmentation commonly results in a binary image (i.e., classification at pixel level). Extracting a surface from these binary data results in a triangulated surface model that does not meet all of the requirements above. The smoothness of the mesh can be poor due to quantization effects, showing ridges or terraces. Some solutions to this problem are inadequate. For example, Gaussian prefiltering of the binary image (before surface extraction) reduces accuracy, and significant anatomical detail (such as narrow ridges and clefts) may be lost, while insufficient smoothness is achieved [7].

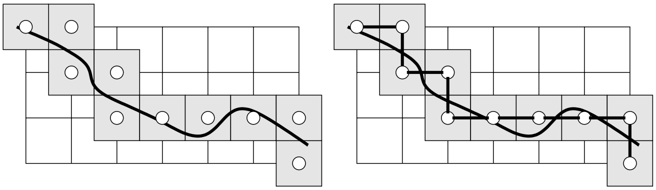
In addition, the number of triangles generated by surface extraction may be very large. Compactness may be improved using mesh decimation techniques [5, 10], but these techniques are usually most effective with smooth meshes. Thus, smoothing of a surface mesh with minimal loss of accuracy is useful to avoid errors in FE analysis and for reducing mesh size. Exploiting the original greyscale data rather than binary segmented data can help to achieve this.

Recently, a technique called *SurfaceNets* was proposed to optimize a triangle mesh derived from binary data [6]. In this paper, the SurfaceNets method is extended to incorporate greyscale data. Several new techniques are examined and compared with Marching Cubes.

The paper is organized as follows. Section 2 briefly describes the basic SurfaceNets method, the extension to incorporate grey-scale data and new techniques for achieving smoothness, accuracy, and good triangle quality. In Section 3 these techniques are evaluated with respect to the requirements for mechanical modeling and visualization listed above. Finally, Section 4 summarizes our findings and draws conclusions.

2 Techniques

This section presents a brief explanation of the original SurfaceNet method (largely following [6]) which assumes that a binary segmentation of the original data exists. Then, two techniques will be introduced that utilize the greyscale image data during relaxation of the SurfaceNet.



(a) Before linking the nodes.

(b) After linking the nodes.

Fig. 1. Building a SurfaceNet. The white squares represent voxels, the thick black line represents the edge of an object and the grey squares are cells with nodes represented by white circles in the center.

2.1 Generating a SurfaceNet from Binary Data

The goal of the SurfaceNet approach is to create a globally smooth surface that retains the fine detail present in the original data. Generation of the surface net for binary data consists of the following four steps [6]:

1. Identify nodes of the SurfaceNet;
2. Create links between the nodes;
3. Relax node positions while maintaining constraints on node movement;
4. Triangulate the SurfaceNet for visualization and FE analysis.

The first step in creating a SurfaceNet is to locate the cells that contain the surface. A cell is formed by 8 neighbouring voxel centers in the binary segmented data (Figure 1 presents the 2D case as illustration). If all eight voxels have the same binary value, then the cell is either entirely inside or entirely outside of the object. If, however, at least one of the voxels has a binary value that is different from its neighbours, then the cell is a surface cell. The net is initialized by placing a node at the center of each surface cell (step 1). Subsequently, links are created with nodes that lie in adjacent surface cells (step 2). Assuming only face connected neighbours, each node can have up to 6 links (corresponding to right, left, top, bottom, front and back neighbours). Once the SurfaceNet has been defined, each node is moved to achieve better smoothness and accuracy ("relaxation", step 3) subject to the constraint that each node must remain within its original cell. The relaxation process is described in more detail in the next section.

2.2 Improving Smoothness

Once a SurfaceNet has been defined, the node positions are adjusted to improve the smoothness of the surface. This is often desirable to remove furrows and terraces due to the binary segmentation. Let us first only consider the smoothness of the net.

One way to smooth the surface is to move every node to the average position of its linked neighbours [4]. The vector $\vec{a}$ pointing from the current position of the node $\vec{p}_{\text{old}}$ to the average position is calculated as:

$$\vec{a} = \frac{1}{N} \sum_{i=1}^{N} \vec{p}_i - \vec{p}_{\text{old}} \tag{1}$$

where $\vec{p}_i$ corresponds to the position of a linked neighbour and N is the total number of neighbours of this node.

It may well be that the average position is outside the original cube and therefore diverges from the initial segmentation. To impose conformance, the relocation vector $\vec{a}$ is constrained to stay within the boundaries of the original cell by the function c (see Figure 2):

$$\vec{p}_{\text{new}} = \vec{p}_{\text{old}} + c\left(\vec{a}\right) \quad . \tag{2}$$

Here, c is defined to satisfy the proper constraint of the node position such that $\vec{p}_{\text{new}}$ is always within the boundaries of the cell. Note that this approach is different from the original SurfaceNet method which simply clips the new position's x, y, and z coordinates to cell boundaries when the new position falls outside the cell.

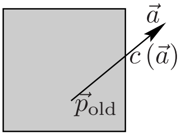


Fig. 2. Position constraint of a node. If $\vec{p}_{\text{old}} + \vec{a}$ is outside the cell boundary, the function c is used such that $\vec{p}_{\text{old}} + c(\vec{a})$ is on the cell boundary.

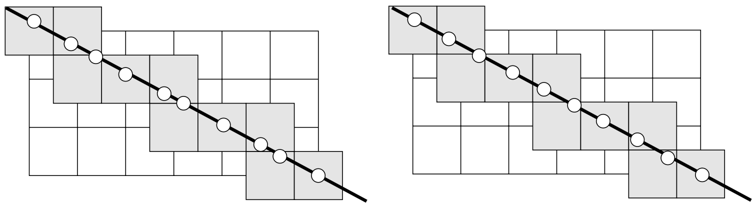
The relaxation is implemented in an iterative manner by considering each node in sequence and calculating a relocation vector for that node. The SurfaceNet is updated only after each node in the net has been visited. This procedure is repeated until the number of iterations has reached a preset threshold, or when the largest relocation distance is less than a given minimum value.

2.3 Increasing Accuracy Using Greyscale Data

The technique described above ignores all greyscale information in the dataset after building the SurfaceNet. The nodes shrink-wrap around the object without trying to conform to an iso-surface of the data. This is reasonable when the binary segmentation is the best estimate of the object. However, if the object surface can be estimated to lie at an iso-surface of the image data, this iso-surface can be used to increase the accuracy of the SurfaceNet.

Let us assume that the true object surface can be obtained by drawing an iso-surface (at I_{iso}) in the original greyscale data. For instance, in many CT based applications the Marching Cubes algorithm is used to approximate the object shape in this way. By definition, at a given point the greyscale gradient vector is perpendicular to the iso-surface through that point. Thus, to enhance accuracy; a node can be displaced along the gradient vector to the iso-surface (see Figure 3(a)). This is expressed as:

$$\vec{g} = \text{SIGN}\left(I_{\text{iso}} - I\left(\vec{p}_{\text{old}}\right)\right) \nabla \vec{p}_{\text{old}} \quad . \tag{3}$$



(a) Using the gradient method (b) Combining the methods

Fig. 3. Using the gradient method for relaxation the nodes (white circles) are projected onto the iso-surface (thick line, left). The combined relaxation technique also spaces out the nodes along the iso-surface (right).

Here, SIGN is a function that returns the sign of its argument, $I(\vec{p}_{\mathrm{old}})$ is the interpolated intensity and $\nabla\vec{p}_{\mathrm{old}}$ is the normalized gradient vector at $\vec{p}_{\mathrm{old}}$. The latter vector is obtained either by a central difference gradient method or by convolution with Gaussian derivatives.

The node position is updated by:

$$\vec{p}_{\mathrm{new}} = \vec{p}_{\mathrm{old}} + c\,(d\vec{g}) \quad . \tag{4}$$

In this equation, d is a scaling parameter representing the distance to the iso-surface. The value of d can be estimated by assuming a linear image field near the iso-surface and interpolating the greyscale values at the node and at a point sampled along the vector $\vec{g}$. As in Equation 2, c imposes a position constraint on the node to stay within the boundaries its cell.

2.4 A Combined Approach

Combining the methods presented in Section 2.2 and Section 2.3, we obtain a surface that fits the iso-surface of the data and is also globally smooth. To combine these features, a node should be displaced to obtain better smoothness within the iso-surface. The combination is made by first calculating the projection $\vec{a}_p$ of the averaging vector $\vec{a}$ on the plane perpendicular to the gradient $\vec{g}$ (cf. Equation 1, Equation 3):

$$\vec{a}_p = \vec{a} - \vec{g}\,(\vec{a} \cdot \vec{g}) \quad . \tag{5}$$

Subsequently, the combined displacement function is defined as:

$$\vec{p}_{\mathrm{new}} = \vec{p}_{\mathrm{old}} + c\,(\vec{a}_p + d\vec{g}) \quad . \tag{6}$$

This formula combines relocation towards the iso-surface with smoothing in the orthogonal plane (i.e., on the surface). This can be seen in Figure 3 where the nodes are first projected onto the line and then evenly spaced out along the

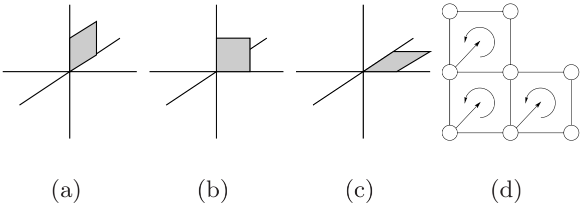


Fig. 4. Three possible configurations for a quadrilateral. For each main direction only the top right corner has to be checked for a quadrilateral (right).

line by the averaging. Again, c ensures that the new position of each node is always within the boundaries of its original surface cell. Note that there may be some tension between the goals of a smooth SurfaceNet and one that fits the iso-surface. One of these goals can be favored over the other either by weighting the independent contributions differently or by applying them sequentially rather than simultaneously; ending with the favored goal.

2.5 Triangulation

After relaxation the SurfaceNet is triangulated to form a 3D polygonal surface. We have simplified the original triangulation process described in [6]. Instead of directly building the triangles, first quadrilaterals are identified. There are three sets of four links of a node that lie in a plane (e.g., the left, right, top, and bottom links of a node lie in a plane) (see Figure 4). In each plane the connected nodes form quadrilaterals and each node is a vertex of at most four quadrilaterals. In order to find all quadrilaterals it is sufficient to check in each plane one “corner” of a node. For example, in Figure 4(d), all quadrilaterals are found by checking the upper right region of a node.

After relaxation each quadrilateral is triangulated using either a shortest diagonal or a Delaunay criterion [4]. Either of these criteria creates triangles that result in a smoother shape than choosing a fixed configuration. The resulting triangle mesh can be rendered using standard 3D graphics techniques.

3 Results

To evaluate the relative effectiveness of the presented techniques, the SurfaceNet is compared to Marching Cubes, which is the standard iso-surface extraction tool [9]. The effectiveness of each technique will be tested against the requirements listed in Section 1. Each of these requirements is measured as follows.

- A measure expressing the local smoothness of a polygon mesh is given in [11]. As a first step, the angles α_i of all triangles around a vertex are summed. If all triangles connected to a vertex are coplanar this sum is equal to 2π. A measure of the local smoothness at a vertex is defined by $2\pi - \sum \alpha_i$, the absolute value of which is then averaged over all vertices.

- A simple and direct measure for triangle quality is found upon division of the smallest angle of each triangle by its largest angle. If the triangle is equilateral this expression is equal to 1.
- The accuracy is expressed by the unidirected modified Hausdorff distance that represents the mean distance of the generated mesh to a reference shape [3]:

$$H_{\mathrm{ave}}(S_1, S_2) = 1/N \textstyle\sum_{p \in S_1} e(p, S_2) \tag{7}$$

  where e is the minimum distance between a point and a surface, and S_1 and S_2 are two surfaces.

Using these measures, the following experiments are conducted. Two volumes, containing greyscale images of distance maps of respectively a plane and a sphere were created, where the greyscale values were stored as floats. An iso-surface is extracted using Marching Cubes (MC), a SurfaceNet with averaging (SNA) and a SurfaceNet with the combined technique (SurfaceNet with Extended Relaxation and Triangulation SNERT) as presented in Section 2. These surfaces are compared to the exact reference shape. The results of this comparison are shown in Table 1.

Table 1. Measured results on the Plane and the Sphere. Methods are Marching Cubes (MC), SurfaceNet Averaging only (SNA) and SurfaceNet with Extended Relaxation and Triangulation (SNERT). The accuracy is measured respectively at the vertices and at the centers of the faces.

	Quality		Smoothness		Accuracy (vertices)		Accuracy (face centers)	
	Plane	Sphere	Plane	Sphere	Plane	Sphere	Plane	Sphere
MC	0.64	0.54	$0.25\ 10^{-6}$	0.0028	$5.59\ 10^{-6}$	$2.92\ 10^{-3}$	$4.81\ 10^{-6}$	0.028
SNA	0.92	0.74	$16.2\ 10^{-3}$	0.0108	0.092	0.204	0.788	0.426
SNERT	0.93	0.75	$0.15\ 10^{-6}$	0.0028	$20.9\ 10^{-6}$	$12.3\ 10^{-3}$	$12.5\ 10^{-6}$	0.043

Comparing the quality of the triangles for each method shows that both SNA and SNERT produce triangles of a higher quality than MC for the plane as well as the sphere. Also, in the case of the plane the MC and SNERT method produce a smoother (=flatter) surface than SNA. The sphere has a constant curvature that corresponds to the smoothness outcome of MC and SNERT. SNA shrinks the mesh and pulls the nodes away from the iso-surface accounting for the lower smoothness and accuracy. The accuracy of the SNERT surface is lower than MC because the nodes are placed according to the trilinearly interpolated values. However, the error at the vertices for SNERT is smaller than the error at the face centers for the Marching Cubes generated sphere.

To illustrate the effectiveness of our technique a graphical example is shown in Figure 7. Clearly, the SNERT surface is as flat as the MC surface and the triangles have higher quality. Figure 6 shows the mesh generated by MC and

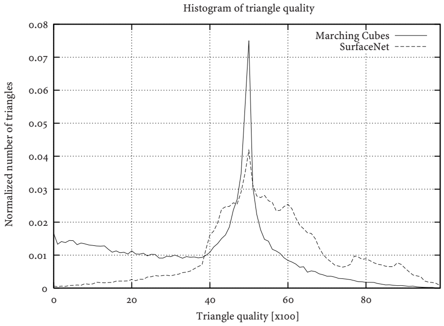


Fig. 5. Histogram of triangle quality for meshes generated using Marching Cubes and a SurfaceNet. The mesh was generated from a CT scan of an ankle. The peak at 0.5 indicates that many right triangles are generated (quality measure is smallest angle divided by largest angle of a triangle).

SNERT on a dataset containing two overlapping spheres. The average triangle quality for the Marching Cubes mesh is 0.64, for the SNERT mesh this number is equal to 0.93.

In addition to the results presented, several experiments were done on true greyscale MRI and CT data. Figure 5 shows a histogram of triangle quality for meshes generated by MC and SNERT from a greyscale CT dataset containing part of a human ankle. It can be seen that the SNERT mesh contains less low quality triangles and contains more high quality triangles. Figure 8 shows the meshes generated from a CT-scan of a human ankle. Lastly, Figure 9 shows a close-up of the bladder extracted from a 256x256x61 MRI dataset of the abdomen of a female patient.

4 Conclusions

Finite element analysis is a standard way to simulate soft tissue deformation. For proper modelling, triangular mesh models must satisfy requirements of accuracy, smoothness and triangle quality. Several approaches proposed in the literature do not meet these requirements (e.g., Marching Cubes in combination with low pass filtering).

In this paper we extending the SurfaceNet method, and evaluated two variants. Optimization of a triangle mesh was performed by averaging vertices, stepping in the direction of the gradient to the iso-surface, and a combined approach.

From visual inspection of test objects, the meshes generated by a SurfaceNet appear to be of similar quality as those created by Marching Cubes. This is backed up by measurements. The SurfaceNet meshes are more suitable for finite element modelling as they are significantly smoother and have a low number of poor quality triangles.

We conclude that SurfaceNet creates a globally smooth surface description that retains fine detail.

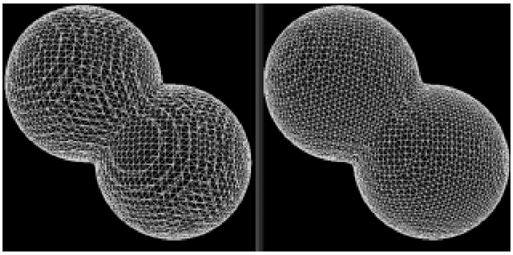


Fig. 6. Two spheres partly overlapping. Meshes generated by Marching Cubes (left) and SNERT (right). Both meshes have the same number of triangles.

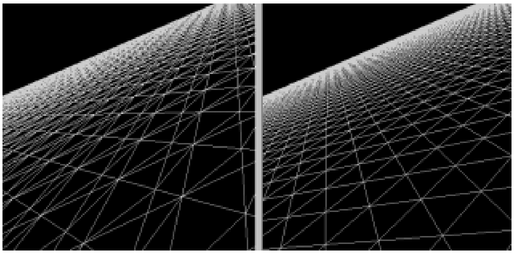


Fig. 7. Generated mesh using Marching Cubes (left) and SurfaceNets (smoothing+gradient) (right). A view of a plane is shown.

Future research will focus on improving the performance of the SurfaceNets technique and developing suitable mesh reduction techniques for finite element analysis.

Acknowledgements

This research is part of the MISIT (Minimally Invasive Surgery and Intervention Techniques) programme of the Delft Interfaculty Research Center on Medical Engineering (DIOC-9). The work described here was largely carried out at MERL – a Mitsubishi Electric Research Laboratory in Cambridge, MA (USA).

References

[1] Bro-Nielsen, M. Finite element modelling in surgery simulation. *Proceedings of the IEEE Special Issue on Virtual & Augmented Reality in Medicine 86*, 3 (Mar. 1998), 490–503.

[2] Cotin, S., Delingette, H., and Ayache, N. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics 5*, 1 (1998), 62–73.

[3] Dubuisson, M., and Jain, A. A modified Hausdorff distance for object matching. In *Proceedings, 12th IAPR International Conference on Pattern Recognition, Conference A (Jerusalem, Israel, October 9–13, 1994)* (Oct. 1994), IEEE Computer Society Press, Los Alamitos, CA, 1994, 566-568, pp. 566–568.

[4] Field, D. Laplacian smoothing and Delaunay triangulations. *Communications in Applied Numerical Methods 4*, 6 (1988), 709–712.

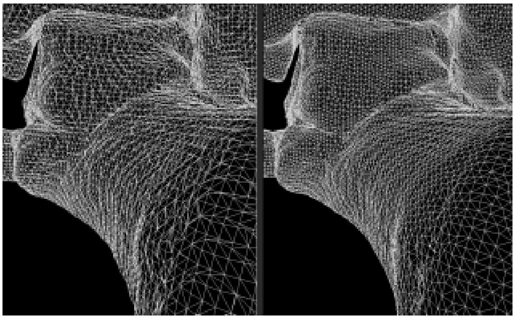


Fig. 8. Mesh generated by Marching Cubes (left) and SurfaceNet (right) on a greyscale image of an ankle. The dataset is a CT-scan with dimensions 132 × 141 × 69.

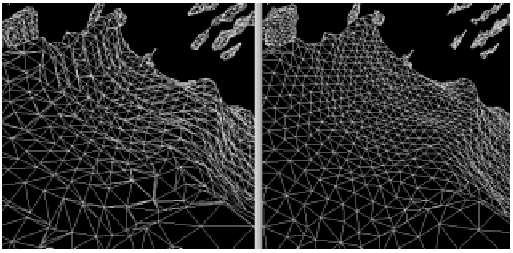


Fig. 9. View of a bladder extracted using Marching Cubes (left) and using SurfaceNets (right).

[5] Garland, M. *Quadric-Based Polygonal Surface Simplification.* PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1999. Software and thesis available at: `http://www.cs.cmu.edu/~garland/quadrics/`.

[6] Gibson, S. Constrained elastic surfacenets: generating smooth surfaces from binary sampled data. In *Proceedings Medical Image Computation and Computer Assisted Interventions, MICCAI '98* (1998), pp. 888–898. `http://www.merl.com/reports/TR99-24/`.

[7] Gibson, S. Using distance maps for accurate surface representation in sampled volumes. In *Proceedings 1998 IEEE Symposium on Volume Visualization* (Oct. 1998), pp. 23–30. `http://www.merl.com/reports/TR99-25/`.

[8] Koch, R., Gross, M., Carls, F., von Büren, D., and Fankhauser, G. Simulating facial surgery using finite element models. In *Proc. ACM SIGGRAPH 96* (1996), pp. 421–428. New Orleans, USA, August 4-9 1996, Computer Graphics Proceedings.

[9] Lorensen, W., and Cline, H. Marching cubes: a high resolution 3D surface construction algorithm. In *Proc. ACM SIGGRAPH'87* (July 1987), pp. 163–169.

[10] Montani, C., Scateni, R., and Scopigno, R. Decreasing isosurface complexity via discrete fitting. Tech. Rep. xx, Istituto per l'Elaborazione dell'Informazione - Consiglio Nazionale delle Ricerche, Pisa, Italy, Dec. 1997. `http://vcg.iei.pi.cnr.it`.

[11] Veron, P., and Leon, J. Shape preserving polyhedral simplification with bounded error. *Computers & Graphics 22*, 5 (1998), 565–585.