The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (2005) *Short Presentations*
M. Mudge, N. Ryan, R. Scopigno (Editors)

Smooth Surface and Detailed Voxel Construction with Volumetric Implicit Function for Neurosurgical Simulation

Mayumi Shimizu and Yasuaki Nakamura

Department of Computer and Media Technologies
Hiroshima City University
3-4-1, Ozuka-higashi, Asa-minami-ku,
Hiroshima, 731-3194, JAPAN

Abstract

We present boundary surface construction and interactive feedback force generation for a neurosurgical simulation system. In the system, voxel data of a head created with a series of CT or MR images are used as a volume model. The Octree and volumetric implicit function are utilized to manage the voxel data, and to represent piecewise continuous volume. By sub-sampling the continuous volume, the detailed sub-voxel representation can be obtained. To provide interactive tactile sensation, a sophisticated smooth surface construction method applying Marching Cubes is proposed. Segmentation of voxel data is also performed by finding level surfaces on the volumetric function. Experimental results reveal the effectiveness of the proposed method.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Virtual reality

1. Introduction

Since surgery is a human activity that employs the use of visual and manual investigations, surgical simulation systems must integrate both haptic (manual) and visual information that provides temporal and spatial coherence with data manipulation. Recently, in surgical simulation systems [QS01, SY01, KTT00, GH91], volume rendering in place of surface rendering is applied to generate realistic visualization of 3D volumetric data.

The inside structural properties of an object can be seen by making other parts transparent. Haptic display devices are developed to present a touch sensation to a virtual object. Combining the volume rendering and haptic display devices, several surgical simulation systems are proposed [GH91, HQ04, SN04, SN05]. We have also been developing a neurosurgical simulation system [SN04, SN05] with both volume visualization and haptic presentation. Volume visualization of human organ utilizes a series of CT or MR images as voxel data. Voxel representation has such advantages that any 3D shape of objects can be modeled, and that deformation can be handled by modifying values of some voxels. However, there are some problems in voxel representation. The first problem is that the boundary of the object is not smooth but jaggy; object does not have smooth surface representation. The second is voxel data are not segmented into organs. Segmentation must be performed by human or computer to classify the voxel data into organs. The third, some haptic device such as “PHANToM” requires smooth surface of the object to calculate tactile sensation.

Historically, segmentation of image and surface reconstruction are researched as general techniques [RK82, Gal00]. We apply these techniques to resolve the problems in voxel representation. In our simulation system, a combined method of the Octree [Sam90] and volumetric implicit function is introduced to manage an object of voxel data. This method provides functions of efficient spatial searches, smooth surface construction, comprehensive boundary surface generation, and detailed volume rendering.

In the followings, we describe these functions. Configuration of the system is presented in Section 2. In Section 3, representation using the Octree and volumetric implicit function is described. In Section 4, we provide several algorithms to generate boundary surfaces, interactive smooth surface, and realistic tactile sensation. In Section 5, experimental results and some discussions are described.

2. Neurosurgical Simulation System

A head model constructed with CT or MR images is represented as voxel data. A surgical simulation proceeds by modifying the voxel data interactively with tactile sensation. The voxel representation is useful for presenting the irregularly deformed shape. Our neurosurgical simulation system employs the volume rendering board "VolumePro1000(VP1000)" [Ter01] to visualize the voxel head model in detail, and haptic device "PHANToM" [Sen00]to generate tactile sensation. The Octree is used to manage the whole voxels and to perform spatial searches such as nearest neighbor and range searches efficiently. Figure 1 shows a configuration of our system.

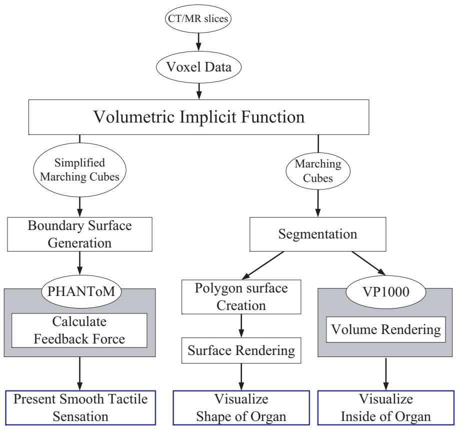


Figure 1: *System configuration*

VP1000 provides volume rendering. Casting rays and resampling data along the ray, the inside structure of a head is visualized in detail by using the translucent representation. However, the voxels with density values are defined as cubes at discrete locations. There is no explicit boundary surface on a set of voxels representing an organ. The classification of organs existing inside a head is a complicated and difficult task. Additionally, the rendered image becomes rough when zooming in the voxel model.

PHANToM presents interactive tactile sensation as a feedback force. The feedback force is calculated using a surface contact point, where an instrument touches the head, and a surface normal vector at the contact point. In calculating the feedback force, PHANToM assumes a continuous smooth surface model. Therefore, the tactile sensation cannot be generated on the voxel model without smooth surfaces.

To solve these drawbacks of voxel model, we introduce a tri-cubic parametric function, called Volumetric Implicit Function, in short VIF, to represent voxel data as piecewise continuous solid. Using VIF, a smooth continuous surface is prepared to calculate a feedback force, and circumstantial voxel data are generated to obtain a smoother rendering image. When the voxel model is zoomed in, more detailed voxel data are reconstructed by re-sampling VIF and rendered. Moreover, approximating the whole voxel data by VIF, voxel values are defined on continuous functions and a classification of organs can be performed by finding level surfaces. The boundary surface is generated locally by extracting a level surface of VIF using the simplified algorithm of Marching Cubes [LC87]. Constructing the smooth boundary surface, the smooth and continuous normal vector is computed. As a result, the smooth and continuous reaction force can be obtained from voxel data.

We provide a function that generates polygon surfaces to render a head model as a surface model. Polygon surfaces are generated by applying the Marching Cubes algorithm to VIF. To understand a shape of an organ, only a certain organ can be extracted using VIF and viewed by rendering the boundary polygon of the organ. When a surgeon wants to see the shape of organ, the surface model created by Marching Cubes algorithm is displayed. On the other hand, the inside of an organ is visualized by VP1000 when a surgeon wants to see inside an organ.

Simulation of a surgical incision is performed by making the voxels underlying the instrument to be transparent and updating the modified part.

3. Octree and Volumetric Implicit Function Representation

Transferring voxel data representing a head model to VP1000, volume rendered image is displayed. The voxel representation is useful for the simulation, but there are some problems in voxel representation. These problems are caused by the roughness and jaggedness of the boundary of voxel model. The first problem is in the case of enlargement. When the voxel model is zoomed in, the boundary of an object becomes more rough and jaggy. The second one is concerned with feedback force generation. In calculating the feedback force, the collision with an instrument and the voxel model must be checked. However, the collision check with every voxel requires a huge amount of calculation. Even if the collision is detected, the smooth and continuous reaction force cannot be obtained because the boundary of voxels is discrete. The third is the classification of voxels into each organ. Since a head model is not composed of individual organs but a set of voxels, it is very difficult to classify the voxels to each organ. To overcome these problems, a new algorithm that approximates the voxel values using the Octree is developed.

3.1. Management of Voxel Model by the Octree

An Octree is a hierarchical data structure that can manage 3D spatial data efficiently. Our system employs the Octree to manage the entire volume and to perform the spatial search of the nearest voxel from PHANToM cursor position. Generally, an Octree is constructed by dividing the space into eight sub-spaces until the sub-space becomes homogenous.

In our system, an Octree subdivision stops when the voxel values in the sub-space are approximated by a VIF within a proper tolerance. The details of implicit representation are mentioned in the next sub-section.

The procedure of constructing an Octree and VIFs is shown in Figure 2. Firstly, the entire voxel data are approximated by a tri-cubic parametric function. The root node of an Octree corresponds to the entire volume space. If the square error ε exceeds threshold ε_0, the region is divided into eight sub-regions, and then the voxels in each sub-region are approximated. Otherwise, parameters of an approximated implicit function are assigned to the corresponding leaf node. The voxel data can be managed by an Octree in which each leaf has a tri-cubic parametric function representing a part of the voxel data. Namely, an Octree manages the voxel model in a hierarchical manner as a collection of tri-cubic functions.

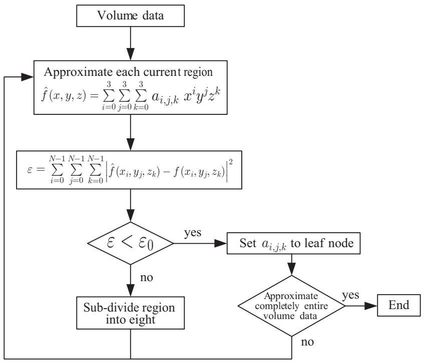


Figure 2: *An Octree and implicit solid modeling*

3.2. Volumetric Implicit Function

Voxel data are given at discrete locations and with discrete values. Approximating the voxel values by VIFs, we solve the problem as mentioned. The boundary surfaces, which are necessary for calculating the smooth and continuous reaction force, can be obtained by constructing a smooth surface locally using VIF. When the voxel data are zoomed in, the sub-voxels are generated by re-sampling VIF and rendered by VP1000. A detailed volume image with smooth boundaries can be displayed. The voxels can be classified by finding level surfaces using VIF. In the followings, the formal representation of implicit solid model by VIF is described.

3.2.1. Definition of Volumetric Implicit Function

Each voxel has density value d. Let the location of a voxel be $(x,y,z), x,y,z$ = 0...N-1. Voxel data are given at discrete locations and are expressed as Eq.(1). In our system, voxel value $d(x,y,z)$ is positive if the voxel is inside the object, while voxel values outside the object are assigned negative values depending on the distance from the object to the voxels.

VIF, which approximates voxel data, is a tri-cubic parametric function expressed as Eq.(2); a three dimensional smooth and continuous solid can be calculated by Eq.(2).

$$f(x,y,z) = d_{xyz}, \quad x,y,z = 0...N-1 \tag{1}$$

$$\hat{f}(x,y,z) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} a_{i,j,k}x^i y^j z^k \tag{2}$$

$\hat{f}(x,y,z)$ is determined so as to minimize the following least square error between $\hat{f}(x,y,z)$ and $f(x,y,z)$.

$$\varepsilon = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} \left|\hat{f}(x_i,y_j,z_k) - f(x_i,y_j,z_k)\right|^2 \tag{3}$$

3.2.2. Octree Construction with VIF

In constructing the Octree, firstly, the whole voxel data is approximated such that Eq.(3) becomes minimal. If ε is greater than threshold ε_0, the region is divided into eight sub-regions. Then, the sub-regions are approximated. If ε is less than or equal to ε_0, the sub-division process terminates, and the parameter vector $a_{i,j,k}(i,j,k$=0,...,3) in Eq.(2) are set to the corresponding leaf node. Until an Octree representing the collection of regions becomes to approximate the entire volume data precisely, the rest sub-regions in the Octree are sub-divided and approximated.

4. Surface Construction Using VIF

4.1. Interactive Surface Construction for Tactile Sensation

When a surgical instrument intersects with a head model, our system generates a feedback force as a tactile sensation by PHANToM. To obtain the smooth tactile sensation, the smooth boundary surfaces must be prepared for PHANToM. A local smooth surface around the contact point is created interactively and used to calculate a feedback force. A simplified Marching Cubes method is applied to find precise boundary points, where $\hat{f}(x,y,z)$ = 0, on a voxel model. Then, mapping the boundary points to *S*-*T* coordinate, a smooth surface is calculated. The procedure of this method is as follows.

Let P and $\vec{v}$ be a point where PHANToM collides on the volume model, and a normalized vector from the center of the volume model to P, respectively. As the head model is a spherical shape, we assume that the direction of normal vector at point P is almost equal to the normal of the sphere. We represent a boundary surface around P as a bi-cubic parametric surface. The bi-cubic parametric surface is determined by boundary points on the head model and represented by parameters s and t.

Firstly, the directions of vector $\vec{v}$ are classified into 26 directions, $\vec{v}_i$ (i=0...25). $\vec{v}_i$ is a vector from the center of volume to a surface point on a unit sphere. The classified surfaces on the unit sphere are shown in Figure 3(a). *S*-*T* coordinate is determined by $\vec{v}_i$ as illustrated in Figure 3(b). $\vec{v}_i$ corresponds

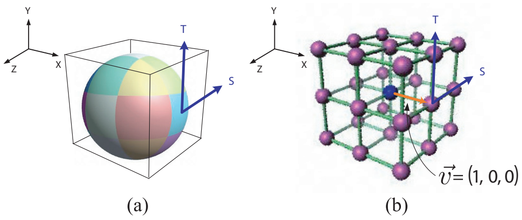


Figure 3: *S-T coordinate*

to one of the surrounding 26 points in Figure 3(b). $\vec{v}$ and $\vec{v_i}$ can be expressed by rotation angles θ about Z axis and φ about Y axis. Vector $\vec{v_i}$ is expressed as Eq.(4).

$$\vec{v_i} = (\cos\theta_i\cos\varphi_i, \sin\theta_i, \cos\theta_i\sin\varphi_i) \qquad (4)$$
$$\begin{pmatrix} \theta_i = \frac{n\pi}{4}, \varphi_i = \frac{m\pi}{4} \\ (-2 \le n \le 2, -4 \le m \le 3) \end{pmatrix}$$

If $\vec{v}$ is classified into $\vec{v_0}$=$(1,0,0)$, S and T axes are toward $-Z$ and Y axes, respectively. Then, boundary points are sought on section planes, $T = 0, \pm1, \ldots, \pm k$, that are parallel to S axis and perpendicular to T axis. S-T coordinate in each direction vector $\vec{v_i}$ can be defined according to the rotation angles, θ_i and φ_i.

We do not seek boundary points in 3D space but 2D plane because of the processing speed. The boundary points are extracted by the following simplified Marching Cubes algorithm that works in 2D plane.

Simplified Marching Cubes Algorithm As shown in Figure 4, the boundary candidate points can be classified into five patterns with 4-neighbor points except the rotation. Symbols "+" and "–" in Figure 4 stand for the signs of voxel values. Boundary points denoted by tiny circles that satisfying equation $\hat{f}(x,y,z) = 0$ are extracted. Namely, between a voxel with positive value and a voxel with negative value, the boundary point P_0 satisfying Eq.(5) is computed as illustrated in Figure 5.

$$P_0 = \frac{P_- V_+ - P_+ V_-}{V_+ - V_-} \qquad (5)$$

where P_+ and P_- are positions of voxels with a positive value and a negative value, and V_+ and V_- are positive and negative values, respectively.

The boundaries are determined by starting from the contact point and following the candidate points clockwise and counter clockwise in a plane parallel to S axis. Candidate points matching to the patterns in Figure 4 are searched by the same algorithm of 8-neighbor border following [RK82]. Figure 6 illustrates a tracking example starting from $V1$. Small circles are boundary points determined on a plane. When the boundary following for a plane is finished, the same algorithm is applied to the next plane. Figure 7 shows the process of local surface construction. The left figure shows planes on which boundary points are followed. When the enough number of boundary points along parameter s and t for determining a local surface around P is found, a bi-cubic parametric surface is defined by these points.

The boundary points are represented as $Q(s_i,t_j) = (x_{ij}, y_{ij}, z_{ij})$ (i=$-n \ldots n$, j=$-m \ldots m$). In our experiment, n and m are equal to 2.

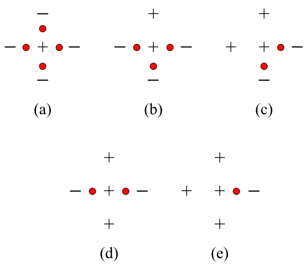


Figure 4: *Patterns of boundary candidate points*

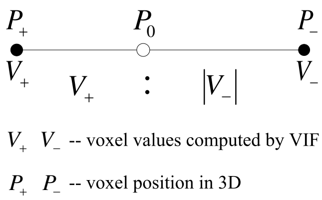


Figure 5: *Position of boundary point*

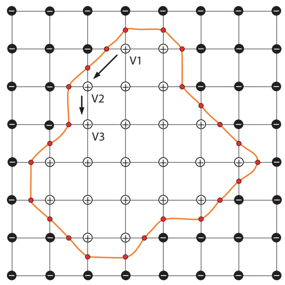


Figure 6: *Example of boundary following*

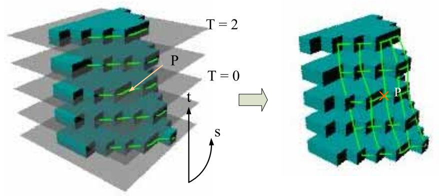


Figure 7: *Construction of local boundary surface*

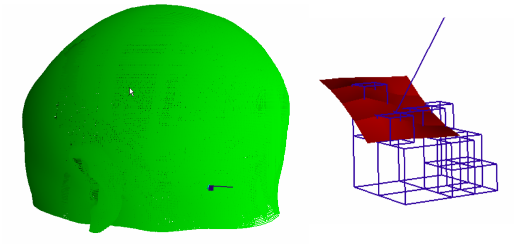


Figure 8: *Constructed local surface*

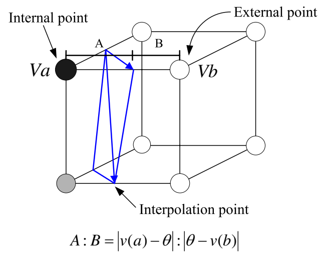


Figure 9: *Polygon surface creation*

Bi-cubic parametric surface expressed as Eq.(6) is computed so that the square error ρ in Eq.(7) becomes minimal.

$$\hat{Q}(s,t) = (\sum_{k=0}^{3}\sum_{j=0}^{3} a_{k,j}s^k t^j, \sum_{k=0}^{3}\sum_{j=0}^{3} b_{k,j}s^k t^j, \sum_{k=0}^{3}\sum_{j=0}^{3} c_{k,j}s^k t^j) \tag{6}$$

$$\rho = \sum_{i=-n}^{n}\sum_{j=-m}^{m} |\hat{Q}(s_i,t_j) - Q(s_i,t_j)|^2 \tag{7}$$

The normal vector at the collision point on bi-cubic parametric smooth surface can be computed by Eq.(8). The normal vector is used to generate the feedback force.

$$normal = \frac{\frac{\partial \hat{Q}}{\partial s} \times \frac{\partial \hat{Q}}{\partial t}}{\left|\frac{\partial \hat{Q}}{\partial s} \times \frac{\partial \hat{Q}}{\partial t}\right|} \tag{8}$$

Figure 8 shows an extracted boundary and a constructed local surface. Left figure represent zero crossing points and right figure is an enlarged smooth surface constructed interactively and and Octree subdivision.

4.2. Surface Rendering and Segmentation

Polygon model is useful for understanding a shape of an object. The rendering speed of polygonal objects is much faster than the volume rendering of voxel model, as well. Our system provides a segmentation of organs and polygon surface construction. The shapes of organs or the boundary polygons can be obtained by applying the Marching Cubes algorithm to VIF. The Marching Cubes method detects the level zero points on the continuous function (Eq.(2)) efficiently. That is, the zero points constructing the boundary surfaces are extracted, and the polygon surfaces are rendered interactively. Moreover, the internal structure of the head model is segmented into each organ by considering the inside of level surfaces of the implicit representation to be the same tissue.

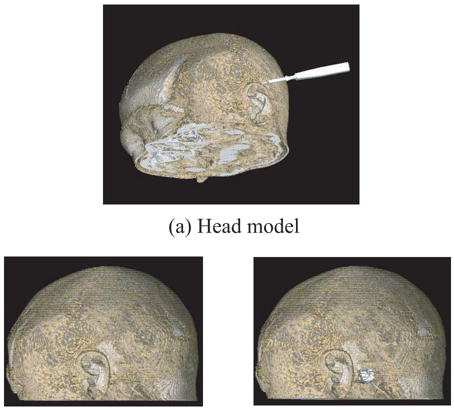
Marching Cubes [LC87] generates polygon facets from a set of points existing in 3D space. Marching Cubes uses the logical cube created from eight neighbor points which are middle of voxels. Determining how the surface intersects in the space of a small cube, surfaces are composed as many polygons. Let values of voxels Va and Vb in VIF be $v(a)$ and $v(b)$, respectively (Figure 9). Suppose that Va is an internal point and Vb is an external point of the surface. The interpolation point which is the vertex of the polygon exists where the edge Va-Vb is divided into $|v(a)| : |v(b)|$. If we want to create the polygon at level θ, the vertex of the polygon exists where the edge is divided by the ratio of $|v(a) - \theta| : |\theta - v(b)|$. Deciding the interpolation points, the boundary surfaces or the shapes of organs are visualized exactly even if the image is enlarged.

When a user wants to visualize the shape of only an organ, the boundary surfaces of the organ are rendered as surface model constructed by Marching Cubes method (see FIgure 11). On the other hand, when a user wants to visualize an inside of an extracted organ, the inside portion segmented with the level surfaces is transferred to VP1000, and visualized by volume rendering. That is, the volume visualization of only an organ can be realized.

5. Experimental Result and Discussion

The head model (voxel data) used in this experiment is created from CT slices of human head. Figure 10 (a) shows the created head model. Rotating the head model, inside structures of a head are visualized in detail interactively. Each organ (bone, skin, etc.) is classified by applying Marching Cubes algorithm to VIF. Figure 10 (b) and (c) show the result of an incision. When an instrument collides with the head model, 4x4x4 voxels underlying the instrument are made transparent in this experiment. Then, in order to update the scene, the voxel data in an Octree's node in which cutting portion is included are transferred to VP1000. At the same time, the reaction force is provided by PHANToM. As the head model is managed by the Octree, the hardness and the friction force are changed depending on the organ.

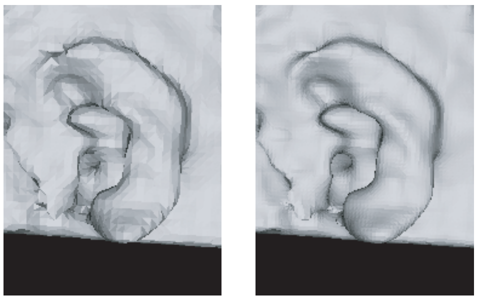
Figure 11 (a) and (b) are created polygon surfaces based on the original voxel values and the approximated voxel values, respectively. As for the resultant image (a), the coarseness is conspicuous. On the other hand, the resultant image (b) can represent the more precise surface. Hence, a high-quality smooth surface is extracted by approximating the whole voxels with continuous functions.



(a) Head model

(b) Before incision (c) After incision

Figure 10: *Head model and incision operation*



(a) Created from voxel values (b) Created from values of VIF

Figure 11: *Polygon surfaces*

It takes about 5 minutes to approximate the whole voxels in size of 512 cubic. This time includes file reading and Octree construction. The construction of an Octree is performed before starting the simulation. Therefore, the construction time does not matter for the interactive simulation. In the interactive operation, it takes about 10 milliseconds to construct the local surface and the normal vector. The frame rate in the surgical simulation with force feedback is 7 frames per second. This frame rate is not sufficient for the interactive environment because it is well known that at least 10 frames per second are needed for the interactive simulation. In order to obtain more interactive frame rate, we will improve the rendering algorithm.

6. Conclusion

In this study, we introduced the method of approximating voxels with continuous tri-cubic functions, called volumetric implicit function (VIF). Representing voxel data by VIF, the smooth and continuous reaction force, the visualization of smooth boundaries after the expansion, and the classification of voxels into each organ were realized.

The boundary points were extracted using the simplified Marching Cubes method. Then, by approximating these points with a bi-cubic parametric surface, smooth and continuous surfaces and normal vectors, which are necessary for calculating a feedback force, were generated interactively. As a result, the smooth and continuous reaction force was obtained.

Applying the Marching Cubes algorithm to VIF, zero crossing points were extracted efficiently and boundary polygon surfaces were constructed. At the same time, the inside of a head was segmented into organs although the segmentation of the internal structure of a head is difficult in voxel expression.

References

[Gal00] GALLIER J.: *Curves and Surfaces in Geometric Modeling*. Morgan Kaufmann, 2000.

[GH91] GALYEAN T. A., HUGHES J. F.: Sculpting: A interactive volumetric modeling technique. *Computer Graphics 25*, 4 (1991), 267–274.

[HQ04] HUA J., QIN H.: Haptics-based dynamic implicit solid modeling. *IEEE trans. Visualization and Computer Graphics 10*, 5 (2004), 574–586.

[KTT00] KOCKRO R. A. SERRA L., TSENG-TSAI Y.: Planning and simulation of neurosurgery in a virtual reality environment. *Neurosurgery 46*, 1 (2000), 118–137.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH 87* (1987), pp. 163–169.

[QS01] QUINGSON Z. KEONG K. C., SING N. W.: Interactive surgical planning using context based volume visualization techniques. In *Proc. MIAR 2001* (2001), pp. 21–25.

[RK82] ROSENFELD A., KAK A. C.: *Digital Picture Processing*. Academic Press, 1982.

[Sam90] SAMET H.: *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1990.

[Sen00] *SDK Programmer's Guide*. SensAble Technologies Inc., 2000.

[SN04] SHIMIZU M., NAKAMURA Y.: Virtual surgical simulation with tactile sensation for neurosurgery operation by volume graphics. In *Proc. CCCT 2004* (2004), pp. 112–117.

[SN05] SHIMIZU M., NAKAMURA Y.: Neurosurgical simulation system with smooth tactile sensation. In *Proc. CATA 05* (2005), pp. 190–195.

[SY01] SHI J.-Y., YAN L.-X.: Deformation and cutting in virtual surgery. In *Proc. MIAR 2001* (2001), pp. 95–102.

[Ter01] *VolumePro 1000 Principles of Operation*. TERARECON Inc., 2001.