An Integral Equation Based Multiresolution Modeling Scheme for Multimodal Medical Simulations

J. Kim[1], S. De[2], M. A. Srinivasan[1]
[1]The Touch Lab,
[1]Dept. of Mechanical Engineering and Research Laboratory for Electronics
Massachusetts Institute of Technology, Cambridge, MA 02139
[2] Department of Mechanical, Aerospace and Nuclear Engineering,
Rensselaer Polytechnic Institute, Troy, NY 12180
jungkim@mit.edu, des@scorec.rpi.edu, srini@mit.edu

Abstract

Novel modeling paradigms are necessary to cope with the requirement of physically based real time simulation of laparoscopic surgical procedures using force feedback.

The requirement of real time force feedback precludes the use of a very high high-resolution model over the entire domain. In this paper we propose a method to address this issue by introducing a multiresolution modeling technique, where a reasonably coarse global model is locally enhanced using mesh subdivision and smoothening. The global model is based on a discretization of the boundary integral representation of the problem. The use of precomputation and structural reanalysis techniques result in a very rapid computation procedure. The local refinements are provided in the vicinity of the tool-tissue interaction area by adaptive subdivision of the boundary element mesh. This technique results in interactive graphical as well as haptic rendering rates for reasonably complex models.

1. Introduction

The simulation of deformable objects has been widely studied in various fields including computer animation, computer aided design (CAD) and medical virtual reality[1]. The simulation of deformations is computationally expensive because the computation involves discretizing complex three-dimensional domains and solving large systems of differential equations. In computer animation of deformable objects, for example, detailed computations are carried out off line and each scene is constructed frame-by-frame from these simulations thus producing the breathtaking special effects seen in movies. However, such an endeavor requires large computational times and storage.

In contrast, simulation of deformations in interactive applications like VR-based medical simulators requires real time performance. In surgical simulation with force feedback, a much higher update rate of about 1 kHz is required as well as a 30 Hz visual update rate. The need for interactive performance severely delimits the complexity of the models that can be used to perform the computations.

Various techniques have been proposed to model the behavior of deformable anatomical objects in real time [2]. Cover *et al.* presented a methodology for interactive deformation of surfaces for surgical simulation by using energy-minimizing splines [3]. Delp *et al.* developed a 3D model of the human lower body to simulate gunshot injuries to the thigh [4]. They used computational geometry techniques to develop interactive deformable tissue models. Basdogan *et al.* [5] developed a deformable organ model for a laparoscopic simulator, which computes reaction forces proportional to the penetration depth of surgical tools and displays visual deformations locally by using spline functions.

"Physically based models" such as mass–spring models or finite element models are more desirable since they are based on the underlying mechanics of biological tissues. Although mass-spring models are computationally inexpensive and easy to implement, it is difficult to extract parameters for the individual springs, masses or dampers from experiments. Additionally, the construction of an optimum 3D network of springs, masses and dampers is a complicated process and the change of resolution is not a computationally easy task.

Techniques based on partial differential equations governing the deformation of continua, like the finite element method, are more appealing since only a few material parameters need to be extracted from experimental data to develop a consistent model.

0-7695-1890-7/03 $17.00 © 2003 IEEE

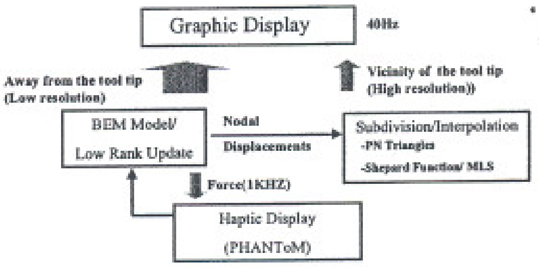

Figure 1. The conceptual diagram of the multiresolution approach

Bro-Nielson and Cotin developed a simulation system based on three dimensional linear elastic finite element models [6-8]. Real time performance was achieved by the use of condensation, and precomputation of the stiffness matrix governing tissue behaviors. Wu and Tendick [9] modeled material nonlinearities in FEM to handle large deformation for 3D deformable bodies. However, these finite element techniques are computationally quite inefficient in real time applications and the graphics hardware currently available is not very efficient for real time volumetric rendering.

An alternative technique for solving differential equations on geometrically complex domains is the point collocation-based method of finite spheres scheme introduced by De and Bathe [10]. This is a truly meshfree scheme and the laborious numerical integration in the finite element method is circumvented by satisfying the governing partial differential equations only at a discrete set of nodal points. De, Kim and Srinivasan applied a localized version of this technique to compute the deformation fields and reaction forces in the vicinity of the tool tip in laparoscopic surgical procedures [11, 12]. While this technique does not provide global deformation fields, it can handle nonlinear behavior of soft tissues and does not require any precomputations.

Another well known technique of numerically solving boundary value problems is the boundary element scheme in which integral equations are discretized on the boundary of the domain using piece-wise polynomial approximations [13]. One of the earliest applications of the boundary element technique to the field of real time simulation of deformable objects is the work of James and Pai [14]. They modeled real-time quasi-static deformations using the boundary element method and achieved real time performance using a simple structural reanalysis technique employing the Sherman-Morrison-Woodbury formula for inversion of low-rank updates of the stiffness matrix. However, the approach is limited to linear models which are homogeneous and isotropic.

It is therefore evident that a single high-resolution model is insufficient to perform in real time. This necessitates the use of multiresolution models to reduce the burden of computation. Astley and Hayward proposed a primitive multirate simulation technique for haptic interactions [15] using Norton equivalents. Cavusoglu and Tendick developed a multirate simulation technique in the context of the finite element technique [16]. Zhang at el. proposed a level of detail modeling technique with haptic subdivision [17]. Several researchers have implemented wavelet transform techniques for multiresolution modeling [18, 19]. Wavelets offer a natural multiresolution modeling approach, but are not computationally efficient for application to deformation modeling.

In this paper, we present a novel multiresolution approach to balance computational speed and accuracy of computations as shown in Figure 1. The justification of using such a multiresolution approach is the following. When a tool -tissue interaction occurs, the "scene of action" is restricted to only a zone in the vicinity of the tool tip. Therefore, a high-resolution model is required for this region and this high-resolution model is governed by user's choice of tool-tissue contact location. The rest of the domain can be modeled using a relatively coarse model.

In section 2 we explain our approach of using the boundary element method for the global model. In section 3 we describe the local enhancement techniques providing a better quality of local deformations. In section 4 we show some examples of the application of our technique.

2. The boundary integral equation based global modeling scheme

We use the boundary element method [13] for our global model. The boundary element technique is based on the discretization of the integral equations of motion posed on the surface of the model and has significant advantages over volumetric techniques such as the finite element method for the modeling and analysis of linear problems. Anatomical organs are usually available as surface models on which texture mapping is used to render realistic graphics. Voxel-models are also available, but they require much higher computational efforts and special graphics hardware. Moreover, the boundary element technique reduces the dimensionality of the problem by one and generates a dense, but smaller set of equations compared to the finite element technique. Using accelerated iterative solvers like the GMRES [20], the boundary element equations may be solved quite efficiently. However, we use a technique similar to the technique in [14] and precompute the inverse of the stiffness matrix for a predefined set of boundary conditions and use rapid update techniques for real time computations. In the following paragraphs we will discuss the details of the formulation.

The displacement ($\mathbf{u}$) and traction ($\mathbf{p}$) vectors at a point $\mathbf{x}$ in the domain or on the boundary may be represented by their three Cartesian components

$$\mathbf{u} = \mathbf{u}(\mathbf{x}) = (u_x, u_y, u_z)^T$$

$$\mathbf{p} = \mathbf{p}(\mathbf{x}) = (p_x, p_y, p_z)^T$$

Using piece-wise constant elements (i.e., the displacements and tractions are assumed to be constant over each element), the BEM equations of linear elastostatics with 'E' elements is given by [13]

$$c\,\mathbf{u}_i + \sum_{i=1}^{E}\Big(\int_{\Delta_i} p^* d\Gamma\Big)\mathbf{u} = \sum_{i=1}^{E}\Big(\int_{\Delta_i} u^* d\Gamma\Big)\mathbf{p}$$

where Δ_i is the surface of the i^th^ element, u^* and p^* are the 'fundamental solutions'(see [13]). The coefficient 'c' depends on the smoothness of the boundaries and can be found in literature (for a Lipschitz boundary, c = 0.5). Satisfying this equation at the centroids of each of the elements and incorporating the boundary conditions, we obtain the following system of linear algebraic equations:

$$\mathbf{AY} = \mathbf{F}$$

where **Y** is a vector of length N and contains the unknown deformations and tractions at the centroids of the boundary elements. **A** is N x N dense matrix. **F** is the known right hand side vector containing boundary conditions and depth of tool indentation. The solution of this system is symbolically represented as,

$$\mathbf{Y} = \mathbf{A}^{-1}\mathbf{F}$$

Usually, computing the inverse of **A** is an $O(N^3)$ process (see table 1 for the time for building system matrix and the time for solving system equations for three different models). Therefore it is expedient to precompute it.

If $\mathbf{A}_o$ is the matrix corresponding to a set of predefined boundary conditions, then, corresponding to a new set of boundary conditions, the matrix **A** may be expressed as a low-rank update of $\mathbf{A}_o$

$$\mathbf{A} = \mathbf{A}_o + \mathbf{U}\mathbf{V}^{\mathbf{T}}$$

where **U** and **V** are matrices defined exactly as in [19]. The matrix $\mathbf{A}_o$ and its inverse are precomputed and stored. The inverse of **A** is given by the Sherman-Morrison-Woodbury formula [21] as follows.

$$\mathbf{A}^{-1} = \mathbf{A}_o^{-1} - \mathbf{A}_o^{-1}\mathbf{U}\mathbf{C}^{-1}\mathbf{V}^T\mathbf{A}_o^{-1}$$

where $\mathbf{C} = \mathbf{I} + \mathbf{V}^T\mathbf{A}_o^{-1}\mathbf{U} \in \mathbf{R}^{3s\times 3s}$ is known as the "capacitance matrix" and 's' is number of nonzero boundary conditions. In case of localized contact, 's' is small and this ensures a computational complexity that scales linearly with the number of unknowns in the system. Table 1 shows the update times for three different models and they are shorter than 1 millisecond.

When a tooltip interacts with the organ, the force of interaction is obtained using the following formula:

$$\mathbf{F} = a_{tip}\mathbf{p}_c$$

where a_{tip} and $\mathbf{p}_c$ are the effective area of tool tip and the traction vector at node 'c' where the interaction occurs. It is noteworthy that $\mathbf{p}_c$ is an element of **Y** and can be computed without extra computational cost.

Table 1. The computing times for various models (A winNT workstation with dual Pentium III 1GHz is used)

Model	Points/ Triangle	Modeling Time	Solution Time	Update Time
Cube	98/192	0.11s	11s	0.17 msec
Kidney (Low resolution)	346/ 688	1.51s	7m21s	0.54 msec
Kidney (High resolution)	1378/ 2752	26.5s	36h 46m 21s	2.3 msec

3. Local Enhancement of Deformation

One of the major difficulties encountered during the display of deformation fields using a model with a single, fixed spatial resolution is that visual artifacts like flatness appear in the vicinity of the tool–tissue interaction point experiencing large displacements. To enhance or refine this, the subdivision algorithm is frequently used in many techniques [9, 17, 22]. But these subdivision algorithms introduce another numerical complexity. To expedite the computation, we employ a local interpolation technique coupled with the simplest subdivision algorithm named "PN Triangle". It is advantageous compared to the multiresolution techniques widely used in off-line graphics applications because we can avoid heavy memory requirements to store multi-resolution information and complicated switching rules for display. The steps in this procedure are summarized as follows:

1. Detect a collision between a tool and an organ model
2. Find neighborhood triangles around the contact triangle that need local enhancement
3. Subdivide each coarse triangle into a finer set of triangles
4. Read displacements of nodal points from the global model. Usually, nodal points are vertices or centroids of each triangle
5. Compute interpolation functions generating smooth deformation fields
6. Compute displacements of points in refined triangles by using the interpolation functions
7. Display the deformed entire object

In the following paragraphs we explain the subdivision algorithm that generates a set of triangles which have a smaller size with varying normal distribution from a larger and flat triangle Among various subdivision algorithms in literature (e.g., Loop [23], Catmull-Clark [24]), we choose an algorithm called the PN algorithm developed by Vlachos *et al.* [25]. It divides a triangle into a set of triangles, which have variation of normal vectors. The geometry of a PN triangle is based on one cubic Bizier patch. The set of subtriangles matches the position and normal vector at the vertices of the flat triangle. Since no additional data other than the position and normal vector of three vertices are necessary, it is computationally efficient compared to the other techniques requiring the information of neighborhood polygons. In Appendix A, the details of the PN triangulation algorithm are explained.

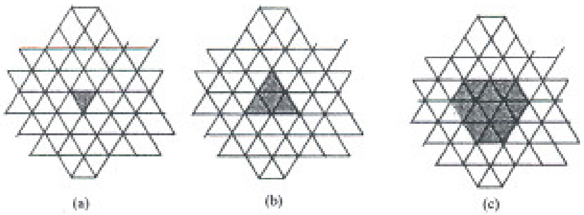

Figure 2. The subdivision area (a) A triangle in touch with a tool (b) Triangles sharing edges with the contact triangle (c)Triangles sharing vertices with the contact triangle

We define the "the edge neighborhood" of triangles around a contact triangle as those that share the same edge with the contact triangle. "The vertex neighborhoods" is also defined as the group of triangles sharing a vertex with the contact triangle. Figure 2 shows the definition of the two groups of triangles. (The vertex neighborhood triangles are identical to the group of triangles "the first ring" in [17].) When the contact occurs in one triangle, the triangles in the neighborhood are selected and subdivided. We can repeat this process if finer resolution is required.

Although the PN triangulation algorithm significantly reduces the storage requirements and is easy to implement, the flatness from original triangles is still remained in subdivided triangles and it is quite noticeable. Hence the extra smoothening or interpolation process is necessary to remove the flatness.

The interpolation of a variable $\mathbf{u(x)}$ is:

$$\mathbf{u}^{h} = \sum_{i=1}^{N} a_{i}(\mathbf{x})\hat{\mathbf{u}}_{i}$$

where $\hat{\mathbf{u}}_i$ is the value of $\mathbf{u(x)}$ and a_i is a basis function at node '*i*'. The selection of a set of basis functions determines the performance of the interpolation. In our application, we require local approximations, high convergence rate, and, of course, computational efficiency. We select two techniques for generating the basis functions: the moving least squares technique and the Shepard partition of unity approach [26]. The moving least squares functions reduce to the Shepard functions as a special case.

Both these techniques use weight (or window) functions ($W_i(\mathbf{x})$) which are radial functions with compact support (i.e. these functions are highly localized, being nonzero in only a small region of the domain). They allow the localization of the approximation and result in a banded matrix. The influence of a node is governed by a decreasing radial weight function, which vanishes outside the domain of influence of the node. Figure 3 shows the interpolation functions in one-dimension.

Using a set of nodal points $\{\mathbf{x}_i\}$ i =1 ,2…,N the approximation $\mathbf{u}^h(x,y,z)$ is

$$\mathbf{u}^{\mathbf{h}} = \Phi^{\mathbf{T}} \cdot \hat{\mathbf{u}}$$
$$\hat{\mathbf{u}} = [\mathbf{u}_1, \cdots \mathbf{u}_N]$$

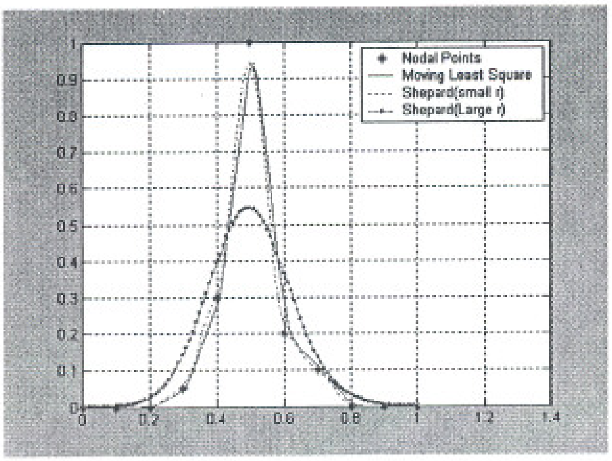

Figure 3. The interpolation functions in 1D

If moving least squares functions are used, then

$$\mathbf{\Phi}^{\mathbf{T}} = \mathbf{p}^{\mathbf{T}}(\mathbf{x})\mathbf{A}^{-1}\mathbf{B}$$
$$\mathbf{p}^{\mathbf{T}} = [1 \quad x \quad y \quad z\ ]$$
$$\mathbf{P} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & z_N \end{bmatrix}$$
$$\mathbf{A} = \mathbf{P}^{\mathbf{T}}\mathbf{W}\mathbf{P}, \qquad \mathbf{B} = \mathbf{P}^{\mathbf{T}}\mathbf{W}$$
$$\mathbf{W} = \mathrm{Diag}(w_1, \cdots w_N)$$

On the other hand, if the Shepard partition of unity approach is taken, the Shepard function at node 'i' is defined as

$$\Phi_i = \frac{W_i}{\sum_j^N W_j}$$

For both the cases, the choice of a radial weight function W_i determines the degree of continuity and differentiability of the approximation as well as the cost of computation. We choose quartic spline weight functions:

$$W_i = \begin{Bmatrix} 1 - 6m^2 + 8m^3 - 3m^4 & 0 < m < 1 \\ 0 & m > 1 \end{Bmatrix}$$
$$where \ \ m = (\|\mathbf{x} - \mathbf{x}_i\| / r_i)$$

The choice of the radius of influence r_i at node 'i' is also important as it determines the number of nodal points included in the influence of node 'i'.

Both these techniques can generate smooth local enhancements. The moving least squares method offers higher convergence rates but is computationally more expensive than the Shepard partition of unity approach. In this work we have used the Shepard function for real time implementation.

We would like to note that our technique is for purely geometric smoothening since no mechanics or material properties are considered in the computation as shown in [9, 27]. However, this leads to a significantly improved visual display of the deformation field.

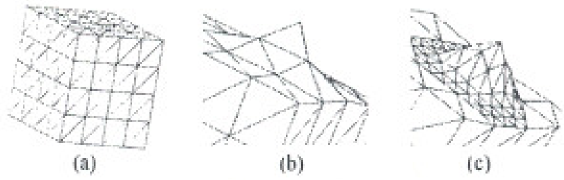

Figure 4. The comparison of deformations. (a) Unmodified cube model (b) Deformation with the coarse model (c) Coarse model with local subdivision and smootheing using Shepard functions

4. Examples

Figure 4 compares the deformation of a cube using (b) a coarse model and (c) a coarse model with local subdivision and smoothing using Shepard functions. In (c) a smooth and visually plausible deformation field is observed compared with (b).

Now let us consider a realistic organ model. Figure 5 shows the deformations of a kidney model generated using the techniques described in this paper. We use a Microsoft WinNT based personal computer (Pentium III 900 MHz processor) with a high-end graphics accelerator (NVDIA TNT M64[1]) and the Phantom force feedback devices from SensAble Technologies Inc. The source code is written in C++ using the OpenGL library for graphics rendering and Ghost SDK for haptic rendering. An update of 40 Hz for the graphics loop and 1 kHz for the force feedback loop are obtained. Figure 5 shows the deformed human kidney model with only global model and multiresolution model. We used the material properties like Young's modulus from literature[28].

[1] Recently ATI Technologies Inc. (www.ati.com) has developed hardware support for global PN Triangulation.

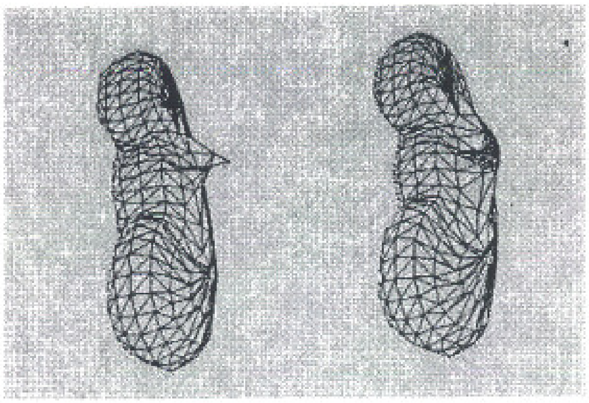

Figure 5. The deformed kidney with only global model (left) and the multiresolution model (right)

5. Concluding Remarks

In this paper, we have presented an efficient multiresolution modeling approach for the simulation of tool-tissue interactions in real time medical simulation. A coarse global model employing the boundary integral formulation is coupled to a refined local enhancement technique around the vicinity of tool-tissue contact area. A graphical smoothening scheme using localized interpolation functions such as Shephard functions are used in the vicinity of tool-tissue interactions. The global model is, however, used to compute the tool-tissue interaction forces. The local enhancement technique makes it possible to avoid the use of a high-resolution model over an entire domain, which is computationally inefficient. The techniques described in this paper are equally applicable to the finite element based discretization. The finite element technique has advantages over the boundary element technique in the modeling of nonhomogeneous and nonlinear media.

A limitation of the current approach is that the refinement in the tool-tissue interaction region is purely geometric. Physically based techniques are possible, but computationally expensive, and that is part of on-going research. Research also needs to be done in extending the model to the simulation of such processes as surgical cutting.

6. Appendix

A. Construction of PN Triangle

The geometry of a curved PN triangle starts from the general equation of a cubic patch **b**. The normal vector of each subdivided triangle follows the normal of the cubic path **b.**

$$b: \quad R^2 -> R^3, \; for \; w = 1-u-v, \quad u,v,w \geq 0$$

$$\begin{aligned} b(u,v,w) &= \sum_{i+j+k=3} b_{ijk} \frac{3!}{i!\,j!\,k!} u^i v^j w^k \\ &= b_{300} w^3 + b_{030} u^3 + b_{003} v^3 + b_{210} 3w^2 u \\ &\quad + b_{120} 3u^2 v + b_{102} 3wv^2 + b_{012} 3uv^2 \\ &\quad + b_{111} 6wuv \end{aligned}$$

Coefficients are also called control points and become vertices of newly generated a set of smaller triangles. For given positions P1, P2, P3 and normal vectors N1, N2, N3 of three vertices as shown in Figure 6, control points of the curved PN triangle are defined as follows:

$$\begin{aligned} &\mathrm{b}_{300} = \mathrm{P}_1 \\ &\mathrm{b}_{030} = \mathrm{P}_2 \\ &\mathrm{b}_{003} = \mathrm{P}_3 \\ &\mathrm{w}_{ij} = (\mathrm{P}_j - \mathrm{P}_i)\mathrm{N}_i \\ &\mathrm{b}_{210} = (2\mathrm{P}_1 + \mathrm{P}_2 - \mathrm{w}_{12}\mathrm{N}_1)/3 \\ &\mathrm{b}_{120} = (2\mathrm{P}_2 + \mathrm{P}_1 - \mathrm{w}_{21}\mathrm{N}_2)/3 \\ &\mathrm{b}_{021} = (2\mathrm{P}_3 + \mathrm{P}_2 - \mathrm{w}_{32}\mathrm{N}_3)/3 \\ &\mathrm{b}_{102} = (2\mathrm{P}_3 + \mathrm{P}_1 - \mathrm{w}_{31}\mathrm{N}_3)/3 \\ &\mathrm{b}_{201} = (2\mathrm{P}_1 + \mathrm{P}_3 - \mathrm{w}_{13}\mathrm{N}_1)/3 \\ &\mathrm{E} \quad = (\mathrm{b}_{210} + \mathrm{b}_{120} + \mathrm{b}_{021} + \mathrm{b}_{012} + \mathrm{b}_{102} + \mathrm{b}_{201})/6 \\ &\mathrm{V} \quad = (\mathrm{P}_1 + \mathrm{P}_2 + \mathrm{P}_3)/3 \\ &\mathrm{b}_{111} = \mathrm{E} + (\mathrm{E} - \mathrm{V})/2 \end{aligned}$$

For a better surface appearance, the normal vectors can be computed separately with linear or quadratic variation (See [25]). But the computing cost for these computations would be doubled.

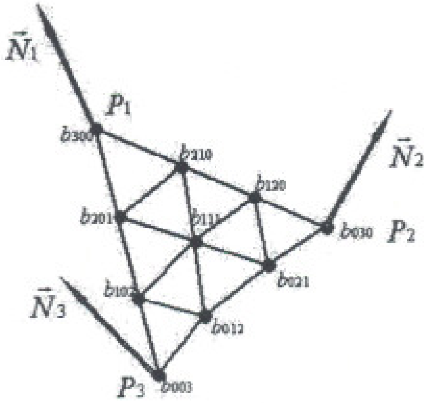

Figure 6. The inputs and outputs of the PN Triangle algorithm

References

[1] C. Basdogan, C. Ho, and M. A. Srinivasan, "Virtual Environments for Medical Training: Graphical and Haptic Simulation of Laparoscopic Common Bile Duct Exploration", *IEEE/ASME Transactions on Mechatronics*, vol. 6, pp. 269-285, 2001.

[2] S. F. F. Gibson and B. Mirtich, "A survey of Deformable Modeling in Computer Graphics," MERL Report TR -97-19, November 1999.

[3] S. Cover, N. Ezauerra, and J. O'Brien, "Interactively Deformable Models for Surgery Simulation", *IEEE Comput. Graphics Applicat. Mag.*, pp. 68-75, 1992.

[4] S. Delp, P. Loan, C. Basdogan, and J. Rosen, "Surgical Simulation: An Emerging Technology for Training in Emergency Medicine", *Presence*, vol. 6, pp. 147-159, 1997.

[5] C. Basdogan, C. Ho, and M. A. Srinivasan, "Force Interaction in Laparoscopic Simulation Haptic Rendering of Soft Tissue" , Proceeding of MMVR'6 Conference, 1998.

[6] M. Bro-Nielsen, "Finite Element Modeling in Surgery Simulation", *Proceeding of IEEE*, vol. 86, pp. 490-503, 1998.

[7] M. Bro-Nielsen and S. Cotin, "Real Time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation", *Proceeding of Computer Graphics Forum, Eurographics 96*, pp. 57-66, 1996.

[8] S. Cotin, H. Delingette, and N. Ayache, "Real-time Elastic Deformations of Soft Tissue for Surgery Simulation", *IEEE Trans. On Visualization and computer graphics*, vol. 5, pp. 62-73, 1999.

[9] X. Wu, M. S. Downes, T. Goktekin, and F. Tendick, "Adaptive Nonlinear Finite Elements for Deformable Body Simulation Using Dynamic Progressive Meshes", *Computer Graphics Forum, vol. 20*, vol. 20, pp. 349-58, 2001.

[10] S. De and K. J. Bathe, "Towards an Efficient Meshless Computational Technique: the Method of Finite Spheres", *Engineering Computations*, vol. 18, pp. 170-192, 2001.

[11] S. De, J. Kim, and M. A. Srinivasan, "A Meshless Numerical Technique for Physically Based Real Time Medical Simulations" , Proceeding of MMVR 2001, 2001.

[12] J. Kim, S. De, and M. A. Srinivasan, "Computationally Efficient Techniques for Real Time Surgical Simulation with Force Feedback" , Tenth Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems , IEEE VR 2002, 2002.

[13] C. A. Brebbia, J.C.F.Telles, and L. C. Wrobel, *Boundary Element Technique: Theory and Applications in Engineering*. New York: Springer-Verlag, 1984.

[14] D. James and D. K. Pai, "ArtDefo, Accurate Real Time Deformable Objects" , Computer Graphics (ACM SIGGRAPH 99 Conference Proceedings), 1999.

[15] O. R. Astley and V. Hayward, "Multirate Haptic Simulation Achieved by Coupling Finite Element Meshes through Norton Equivalents" , Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Belgium, 1998.

[16] M. C. Cavusoglu and F. Tendick, "Multirate Simulation for High Fidelity Haptic Interaction with Deformable Objects in Virtual Environments" , Proceedings of the IEEE International Conference on Robotics and Automation, 2000.

[17] J. Zhang, S. Payandeh, and J. Dill, "Haptic Subdivision: An Approach to Defining Level-of-detail in Haptic Rendering" , Proceedings of the 10th Symp. on Haptic Interfaces for Virtual Env. & Teleoperator Systems, 2002.

[18] A. Khodakovsky, P. Schroder, and W. Sweldens, "Progressive Geometry Computation" , SIGGRAPH 2000,Computer Graphics Proceeding, 2000.

[19] D. L. James and D. K. Pai, "A Unified Treatment of Elastostatic Contact Simulation for Real Time Haptics", *www.haptics-e.org*, vol. 2, 2001.

[20] Y. Saad and M. H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems", *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856-869, 1986.

[21] G. H. Golub and C. F. V. Loan, *Matrix Computations. (2nd Ed.)*. Baltimore: John Hopkins University Press, 1989.

[22] J. Warren and H. Weimer, *Subdivision methods for Geometric Design: A Constructive Approach*. San Francisco: Morgan Kaufmann Publishers, 2002.

[23] C. T. Loop, "Smooth Subdivision Surfaces Based on Triangles," in *Department of Mathematics*: University of Utah, 1987.

[24] E. Catmull and J.Clark, "Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes", *Computer Aided Design*, vol. 10, pp. 350-355, 1978.

[25] A. Vlachos, J. Peters, C. Boyd, and J. L. Mitchell, "Curved PN Triangles" , Symposium on Interactive 3D Graphics, 2001.

[26] S. De, J.-W. Hong, and K. J. Bathe, "On the Method of Finite Spheres in Applications: Towards the Use with ADINA and in a Surgical Simulator", *Computational Mechanics*, in Press, 2003.

[27] J. Kim, S. De, and M. A. Srinivasan, "Physically Based Hybrid Approach in Real Time Surgical Simulation With Force Feedback" , MMVR 2003, 2003.

[28] H. Yamada, *Strength of Biological Materials*. Baltimore: Williams and Wilkins Company, 1970.

[29] C. Ho, C. Basdogan, and M. A. Srinivasan, "Efficient Point-Based Rendering Techniques for Haptic Display of Virtual Objects", *Presence*, vol. 8, pp. 477-491, 1999.