A Hybrid Deformation Model for Virtual Cutting

Xuqiang Shao[1,2], Zhong Zhou[1,2],Wei Wu[1,2]
[1]State Key Laboratory of Virtual Reality Technology and Systems
[2]School of Computer Science and Engineering
Beihang University, Beijing, China

Abstract—**In this paper, we present a novel hybrid deformation model, which can simulate large deformations and non-linear behaviors of soft tissues in real time. The model partitions a soft tissue into operational and non-operational regions, and simulates the large deformations by coupling the geometry constrained meshless method FLSM (fast lattice shape matching) with finite element algorithm TLED (total Lagrangian explicit dynamics). With the virtual node cutting method, the soft tissues can be cut realistically and robustly. The virtual cutting simulation features real time performance with the GPU-based acceleration. In combination with our multi-camera based real-time modeling framework, the deformable model and its interaction of virtual cutting can be extended to support remote virtual surgery.**

Keywords- Hybrid deformation model, virtual cutting, real-time simulation，virtual surgery

I. INTRODUCTION

Virtual surgery is a safe and effective method for surgical training. Surgeons can practice operation skills via a virtual surgery system. To date several practical surgical simulation systems in the virtual medical field have been presented. In 1998 Bro-Nielsen et al. described the HT Abdominal Trauma Simulator for open surgery [1]. This simulator is a PC based soft tissue simulation with instrumental interaction of tissue through the Phantom haptic feedback device. In 2001 Webster et al. presented a PC-based suturing simulator for open wounds [2]. The system supports haptic feedback through the Phantom Omni device. In 2002 Bielser et al. demonstrated an *Open Surgery Simulator* that could basically simulate interaction between skin and surgical hooks or surgical scalpels [3]. In 2004 Paul Gasson et al. developed a system for hernia repair, supporting use of virtual knives and clamps [4]. Their system, employing the Phantom for interaction with tissue, uses a non-linear Spring-Mass system to accurately create the basis for the haptic interaction.

Virtual cutting is an essential 3D interactive manipulation in many aspects of surgical simulation systems. There are many factors that affect the fidelity of the virtual cutting. The deformation model needs to support large deformations and nonlinearity of objects representing soft tissue in humans. The deformable objects also need to be progressively cut to provide smooth incision. Current simulation methods for cutting soft tissue lack in realism.

In this paper, we presented a novel hybrid deformation model simulating soft objects that are subject to large deformations. The model couples the geometry constrained meshless method FLSM to non-linear finite element method TLED. The real scalpel is reconstructed by our multi-camera based real-time modeling system, while its acquired 3D model is applied to cut deformable objects simulated by the proposed model. Integrated with the virtual node cutting algorithm, the cutting interaction achieves realistic visual performance. Using CUDA (Compute Unified Device Architecture) to accelerate the computation of the model, the virtual cutting can be performed at interactive frame rate (20fps). The presented model in context to our multi-camera based capturing method can in the future facilitate remote virtual surgery.

II. RELATED WORK

Producing realistic deformations of soft objects in real time is a critical task for virtual surgery. FEM (finite element method) is a physically based deformable model which has been widely used in surgical simulation. O'Brien et al. developed models for brittle [5] and ductile fracture [6] based on FEM discretization of continuum mechanics equations. They demonstrated the generation and propagation of cracks from eigen analysis of the stress tensor. Miller et al. recently presented an efficient TLED finite element algorithm and illustrated that moderately sized model solutions could be obtained at rates approaching real time performance [7]. The algorithm accounts for all physically accurate geometric nonlinearities associated with the large deformations which occur in a surgical scenario. Besides, it can be implemented on GPU in parallel because it need not to assemble global stiffness matrix and can completely implement on elements' level. With an appropriate cutting method, The TLED model can also be used to simulate the cutting of soft tissues.

Many approaches have been proposed for dealing with the splitting of FEM mesh elements as the result of cutting or fractures. Popescu et al. delete elements as they are crossed by a blade, however their approach would produce artificial incisions in the tissue [8]. Nienhuys et al. snapped nodes of the elements to the trajectory swept by the blade [9]. The splitting of tetrahedral elements by planar cuts can be described by a small set of configurations [10, 11]. However, the two methods above may produce inaccurately-shaped sub-elements that destabilize the simulation. The virtual node algorithm [12] is a compromise method which separates visual and physical representation. It detaches the surface representation from the FEM mesh, and replicates nodes for simulation purposes. It enables arbitrary cuts through mesh elements, as long as one original node falls on each side of the cut. When combined with the dense underlying FEM meshes, the virtual node algorithm enables practically arbitrary cutting trajectories, at the price of slow

978-0-7695-4217-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ISM.2010.41

computations. This algorithm can be used for real-time cutting soft tissue with nonlinear properties.

Compared with FEM, meshless methods can handle large deformations and topological changes without the need for re-meshing the object. These methods have become popular during the last few years. Muller et al. introduced a meshless, continuum mechanics based model for animating elasto-plastic objects [13]. Pauly et al. extended their work to simulate fractures by using surface and volume sampling methods [14]. The meshless methods above are based on complex physics and are therefore costly to compute in real time. Meshless shape matching deformation model proposed by Muller et al. has recently been extensively investigated for its robust and efficient computation of deformations in real time [15]. It is a pure geometry constrained meshless simulation approach resulting in large deformations and extremely high efficiency in computational complexity.

For deformation simulation in interactive applications, efficiency is crucial. Multi-resolution models [16] as an acceleration strategy have been proposed for lattice-based FEM. For meshless shape matching method, River et al. proposed FLSM which also adopted the multi-resolution models [17]. Wen et al. adopted hybrid condensed FEM model with GPU acceleration [18]. They partitioned the soft tissue into operational and non-operational regions, while simulated cutting only on the operational region. As operational region is just a small portion of the whole soft tissue, real-time simulation is achieved.

In virtual reality based medical training, user interaction needs to be mapped from the real world into the virtual. Markers in connection with optical methods can be used to obtain information on the position of the user's body. Such approach however restricts the movement of body parts. Markless methods on the other hand do not have those restriction, thus facilitating more natural interaction. Grimage developed by INRIA is a new platform providing interaction between virtual objects and reconstructed 3D models [19]. Shujun et al. developed a CUDA-accelerated real-time 3D modeling system [20]. Kurillo et al. developed a tele-immersive framework that would allow doctors to collaborate in more natural way by being immersed in the virtual space [21].

This paper proposes a novel hybrid deformation model which partitions a soft tissue into operational and non-operational regions. First, the hybrid model applies FLSM to the non-operational region and the interface. For the non-operational region, the model calculates the positions of the next time step. But for each node of the interface, it only calculates the goal position, and defines a external force by multiplying a factor to the vector difference between the goal position and the actual position. Then the hybrid model uses TLED to calculate deformable positions of the operational region and the interface. The hybrid model can get balance between physical realism of TLED and computational efficiency of FLSM, so it has the ability to simulate large deformations in real time. With the help of real-time 3D modeling technology, we develop the reconstructed scalpel driven immersive virtual cutting. Results of our experiments show that the virtual cutting can achieve realistic visual feedback robustly.

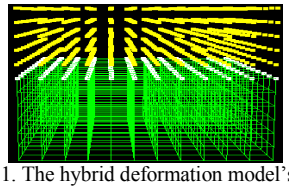


Figure 1. The hybrid deformation model's nodes

III. THE HYBRID DEFORMATION MODEL

The proposed novel hybrid deformation model couples the TLED and FLSM as a lattice-based shape deformer, and embeds object geometry into deformable lattices. This hybrid model divides the deformable object into the purely deformable region and the cutting region. As for the purely deformable region, the deformation need not to be physically accurate, so we adopt FLSM to simulate this region and the results are perceptually plausible. The cutting region is the interactive area and the topology will change. Only the physically accurate deformation model can provide correct information for the trainer, so the deformation of this region should comply with physical laws. Using non-linear finite element method TLED to simulate the cutting region is satisfactory.

When the hybrid model combines TLED with FLSM to simulate the deformable object as a whole, the key is in tight coupling of simulation nodes from different models. FLSM applies the shape matching method to regular lattices via a region-based convolution, and in TLED we adopt the hexahedron as the calculating element, so all simulation nodes are regularly placed. As shown in Fig. 1, the TLED model's elements and nodes are colored green; the FLSM nodes from the FLSM model colored yellow; and the interfacial nodes between two models colored white.

These three different node displacements should be calculated by different methods. The hybrid deformation model algorithm is described as follows:

1) Pre-computation: The triangle mesh of each object is the input. We use kd-tree constructed on GPU (through CUDA) to accelerate the calculation of distance scalar field. For discretization and nodal arrangement, we use distance scalar field to sample the volume of the objects. Then we calculate pre-computed data for different regions and load this information into the texture memory on the GPU. Finally, we initialize the nodal variables for the first time step.

2) Calculation of Displacements of FLSM Nodes in parallel: The interfacial nodes belong to the regions situated in the interface. The remaining work is to compute the goal positions of all the interfacial and FLSM nodes in each region.

For each region R_i we find the optimal rotation matrix R_r and the translation vectors C_r^t and C_r^0 which minimize equation (1).

$$\sum_{i \in R_i} w_i \left(R_r \left(x_i^0 - C_r^0 \right) + C_r^t - x_i^t \right) \qquad (1)$$

In equation (1) w_i are the weights of individual points. The optimal translation vectors turn out to be the center of mass of the initial shape C_r^0 and the center of mass of the actual shape C_r^t.

$$C_r^0 = \frac{\sum_{i \in R_i} w_i x_i^0}{\sum_{i \in R_i} w_i} \qquad C_r^t = \frac{\sum_{i \in R_i} w_i x_i^t}{\sum_{i \in R_i} w_i} \tag{2}$$

We estimate the least squares rotation R_r for particles in region R_i using the rotational part of A_r.

$$A_r = \sum_{i \in R_i} \tilde{m}_i \left(x_i^t - C_r^t\right)\left(x_i^0 - C_r^0\right)^T \tag{3}$$

We obtain the rotational part using the polar decomposition.

$$A_r = R_r U \tag{4}$$

In equation (4) matrix U is a unique 3-by-3 symmetric stretch matrix. Each region's least-squares rigid transformation of the rest positions x_i^0 is a rotation by R_r and a translation that shifts the rotated C_r^0 to C_r^t. This transformation is stored as the matrix T_r calculated by (5).

$$T_r = R_r \left(C_r^t - R_r C_r^0\right) \tag{5}$$

Each node goal position g_i can be restated as the transformation of the particle's rest position x_i^0 by the average rigid transformation over the regions the particle belongs to.

$$g_i = \frac{1}{\left|\Re_i\right|} \sum_{r \in \Re_i} T_r x_i^0 \tag{6}$$

In equation (6) $\Re_i$ denotes the set of regions these nodes belong to.

Once the goal positions of the interfacial nodes have been computed by the FLSM method, we can define an external force exerted on each interfacial node. The force is calculated by multiplying a factor to the vector difference between the goal position and actual position.

$$F_i^{'} = \beta(g_i(t) - x_i(t)) \tag{7}$$

In equation (7) the parameter β is an constant factor.

Finally, as for the FLSM nodes, the positions x_i and velocities v_i are updated using the goal positions g_i.

$$v_i(t+h) = v_i(t) + \partial \frac{g_i(t) - x_i(t)}{h} + h \frac{f_{ext}(t)}{w_i} \tag{8}$$

$$x_i(t+h) = x_i(t) + h v_i(t+h) \tag{9}$$

The parameter ∂ represents the stiffness of the non-operational region. During the FLSM procedure, we take advantage of the programmability of GPU to do the tasks in parallel. Information about particles and regions is stored into texture memory and the calculation is transferred into GPU by the shader programmed by the Cg language.

3) Calculation of Displacements of the Interfacial and TLED Nodes in Parallel: Because the interfacial nodes also belong to the finite elements situated in the interface, its deformations can be computed by TLED method as the TLED nodes.

For each element, we first compute the deformation gradient ${}_0^t X$ using (10).

$${}_0^t X_{ij} = \frac{\partial {}^t x_i}{\partial {}^0 x_j} \tag{10}$$

In equation (10) the left superscript *t* represents the current time and the left subscript 0 represents the time of the reference configuration. The indices *i* and *j* range over the three coordinates in three-dimensional space.

Next, the strain-displacement matrix at time *t* is computed by transforming a stationary matrix using the deformation gradient.

$${}_0^t B_{\mathbf{L}}^{(a)} = {}_0 B_{\mathbf{L}0}^{(a)} \, {}_0^t X^{\mathbf{T}} \tag{11}$$

In equation (11) ${}_0 B_{\mathbf{L}0}^{(a)}$ and ${}_0^t B_{\mathbf{L}}^{(a)}$ are the a^{th} submatrices of the stationary and complete strain-displacement matrices respectively, with *a* ranging from 1 to the number of nodes per element.

We compute second Piola–Kirchoff stress at integration points using (12).

$$S_{ij} = \mu\left(\delta_{ij} - {}_0^t C_{ij}^{-1}\right) + \lambda {}^t J \left({}^t J - 1\right) {}_0^t C_{ij}^{-1} \tag{12}$$

λ and μ denote Lame constants, and δ_{ij} represents Kronecker's delta.

Next, we compute element nodal reaction forces using Gaussian quadrature.

$${}^t F = \int_{{}^0 V} {}_0^t B_L^T \, {}_0^t \hat{S} \, d \, {}^0 V \tag{13}$$

In equation (13) ${}_0^t \hat{S}$ is the vector form of the second Piola-Kirchoff stress, ${}^0 V$ is the initial volume of the elements.

Finally, for TLED nodes, obtain net nodal reaction forces at time *t*, ${}^t F$ and explicitly compute displacements using central difference formula.

$${}^{t+\Delta t} u_i^{(k)} = \frac{\Delta t^2}{M_k}\left({}^t R_i - {}^t F_i^{(k)}\right) + 2 {}^t u_i^{(k)} - {}^{t-\Delta t} u_i^{(k)} \tag{14}$$

In equation (14) M_k is a diagonal entry in k^{th} row of the diagonalized mass matrix, Δt is the time step, and R_i is an external nodal force, for the interfacial nodes, R_i should contain the item $F_i^{'} = \beta(g_i(t) - x_i(t))$.

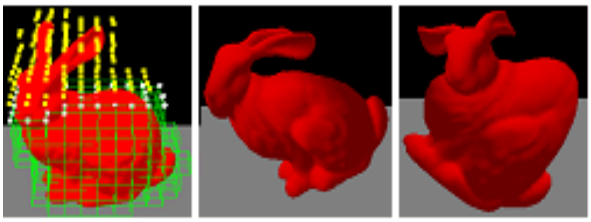


Figure 2. The deformations of bunny simulated by the hybrid deformation model

Force contributions and displacements may be computed for elements and nodes independently of their neighbors. The algorithm therefore exhibits data-parallelism. This allows an implementation programmed by CUDA to take advantage of the parallel processing capability of GPUs. So this method launches two kernels to accomplish these tasks. The first kernel computes element nodal force contributions, and the second kernel sums the element nodal forces uses them to compute new nodal displacements.

4) Interpolatory Calculation of Embedded Object Displacemens: The hybrid deformation model embeds the complex geometry of objects into the deformable lattices. For each vertex, the displacement can be computed via interpolatory calculation from the value of the lattice's nodes.

In general, the proposed hybrid deformation model can improve the physical accuracy of the model and at the same time maintain the efficiency and robustness of shape matching algorithms. We can regulate the balance between physical accuracy and computational efficiency by varying the ratio between TLED and FLSM nodes. The model can simulate large deformations in real time. The nodal arrangement and large deformation simulated by the hybrid model are shown in Fig. 2.

IV. VIRTUAL CUTTING

When the soft object is cut, the topology of its geometry model will change. The object also deforms under its inner stress and external forces applied by the scalpel. The cutting of soft objects simulated by the FEM model should avoid the generation of the ill-conditioned elements and get a good visual feedback. The virtual node algorithm shown in Fig. 3 replaces a cut element by two half void elements whose geometry is the same as the original one. It preserves the quality of the elements while achieving smooth surface.

The existing cutting simulations are usually based on the linear models, yet the linear models are not accurate enough to simulate cutting in surgical operation since soft tissue usually displays nonlinear property. To address this drawback of the linear models, we combine the presented hybrid deformation model with the virtual node algorithm to

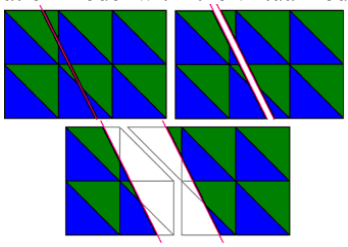
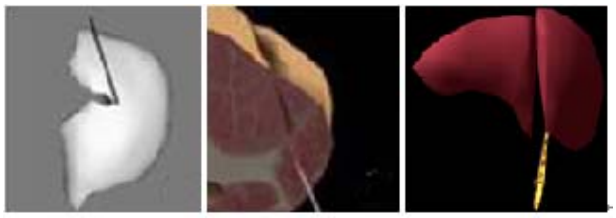


Figure 3. The principle of the virtual node algorithm



(a) (b) (c)

Figure 4. The cutting visual feedback comparison: (a) removing method (b) decomposition method (c) our method

support the cutting interaction. The reconstructed scalpel cuts the cutting region of soft objects simulated by the TLED. The TLED does not require the global stiffness matrix to be assembled in every frame, but is rather pre-computed by related quantities on the element level. In this way, we can locally modify topologies of soft tissue after cutting, avoiding re-computation of pre-computed data. The virtual node cutting method can be integrated into the TLED to simulate cutting of soft objects. The cutting procedure is shown as follows.

First, the collision detection procedure is implemented to find the elements intersecting with the cutting plane swept by the reconstructed scalpel. In our work we adopted the spatial hashing algorithm [22] to implement the collision detection.

Next, we check if cut was performed and use virtual node algorithm to change the mesh topology. Only completely cut elements are handled in this step. A copy of the completely cut elements is made and virtual nodes according to the crosspoints are generated. The pre-computed quantities are copied from the cut element to newly added elements. If the element is not cut, or the cutting is handled, the procedure will switch to the next step.

Finally, we calculate the deformations of all nodes by the hybrid model and render the scene. Fig. 4 shows the rendering of three different methods, removing method, decomposition method and our method. Compared with the removing method showed in the picture, our method can produce smooth incisions. The result of the decomposition method showed in Figure 4b compares to our method in the smoothness of the cut. Our method, however, is more stable because no ill-conditioned element is produced during the cutting process.

From the experiments with different simulation methods, we can conclude that our hybrid model integrated with the virtual node cutting method can achieve realistic visual feedback of the cutting interaction efficiently.

V. EXPERIMENTS RESULTS

Two of the most important issues in deformable simulat-

TABLE I. EXECUTION TIME OF DEFORMATION

Model	*Number of nodes*	*Ratio*	*Total cost(ms)*
Bunny	2250	0.4	13.5
Apple	2250	0.6	16.2
Liver	2886	0.4	20.4
Kidney	2886	0.6	24.6

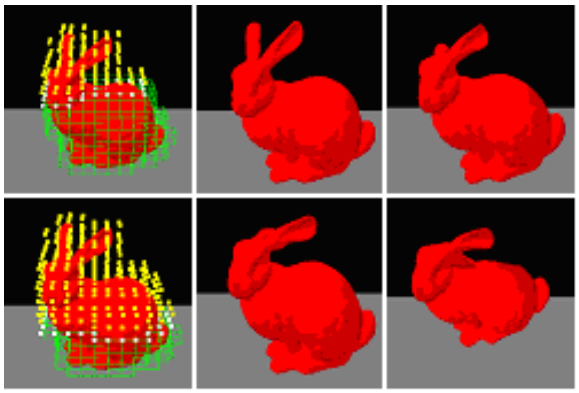


Figure 5. Parameter adjustment

ion are physical accuracy and computational efficiency. Global deformation simulated by the new hybrid deformation model can be easily controlled by adjusting the parameter ratio. A larger value of ratio yields more physics-based deformation while a smaller ratio leads to less realistic performance. The new hybrid deformation model can simulate deformable objects of different stiffness by adjusting parameters Lame constants λ and μ in (1) and ∂ in (11). Fig. 5 illustrated different behaviors when the objects suffer from the same size external force. The ratio for the first row is 70%, the second row 30%; for the second column, λ =200000.0, μ =200000.0, ∂ =0.8; for the third column, λ =20000.0, μ =20000.0, ∂ =0.5. The execution times of deformation simulation with different parameters are shown in Table 1. All experiments have been performed on an Intel Core 2 PC with 2.13 GHz , 2.0 GB of memory and GeForce 9800 graphics card.

The hand reconstructed from multiple cameras' images can interact with the deformable bunny simulated by the novel hybrid deformation model in real time. The results are shown in Fig. 6. Our real-time modeling system is equipped

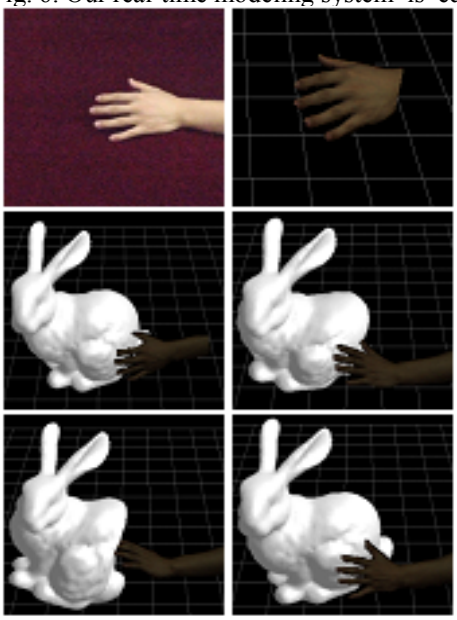


Figure 6. The reconstructed hand interaction with the deformable bunny

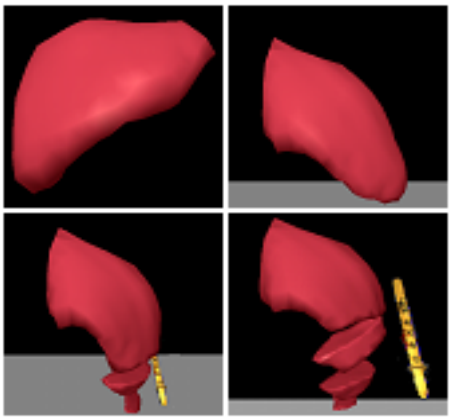


Figure 7. Cutting the virtual liver

with six cameras, six PCs, one graphic workstation and three screens.

We have designed two real time virtual cutting scenarios. In the first scenario, we use the reconstructed scalpel to cut the deformable soft liver in the virtual world. The liver is simulated by the new hybrid deformation model. The parameter values are λ =20000.0, μ =20000.0, ∂ =0.5. The cutting results are shown in Fig. 7. From the results, we can conclude that the hybrid deformation model can simulate large deformation efficiently, and the cutting method can produce smooth incisions.

In the second scenario, we cut the high resolution bunny shell. The hybrid deformation model gets the voxelization presentation of the Stanford bunny, then embeds the geometry of the bunny into the voxels and simulates the cutting interaction using the proposed method. The embedded bunny's high quality surface representation is based on point samples [23, 24] and draws heavily from the CSG rendering technique for point-sampled surfaces proposed by Wicke et al. [25]. The parameter values are λ =200000.0, μ =200000.0, ∂ =0.8. The results are shown in Fig. 8. This technology can be used to simulate the cutting of virtual blood vessels.

During the cutting, the number of nodes increases in the operational region, thus reducing the speed. On average the interactive rate can be up to 20 Fps. Table 2 shows the detailed execution time of the whole process.

From the two virtual cutting scenarios, we can conclude that the proposed camera-based virtual cutting interactive framework can achieve realistic and stable feedback in real time.

VI. CONCLUSION AND FUTURE WORK

In this paper we proposed a novel hybrid deformation model which partitions soft tissue into operational and non-

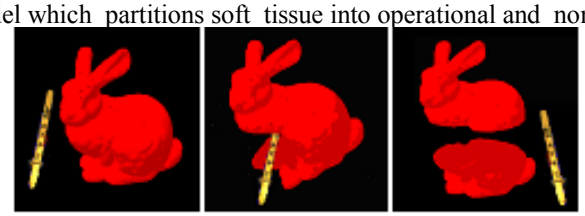


Figure 8. Cutting the Bunny shell

operational regions, and couples FLSM with TLED to simulate deformation. The novel hybrid deformation model can simulate the large deformations and nonlinearity of soft tissues in real time. Integrated with the virtual node algorithm, the hybrid deformation model can support stable and realistic virtual cutting. With the help of our multi-camera based real-time modeling system Dream World [20], we design a new type of immersive virtual cutting system. Using the reconstructed scalpel to cut the soft tissues in any view angle and depth, we can achieve realistic and stable visual feedback of virtual cutting. In the future, we will extend the system to be a new type of remote surgical simulator.

ACKNOWLEDGMENT

This paper is supported by the National Natural Science Foundation of China (Grant No.60903064), and the National Grand 973 Program of China (Grant No.2009CB320805), 2008 China Next Generation Internet Application Demonstration sub-Project (Grant No.CNGI2008-123), and the National High-tech 863 Program of China (Grant No.2009AA01Z333).

REFERENCES

[1] M. Bro-Nielsen, D. Helfrick, B. Glass, Xiaolan. Zeng and H. Connacher, VR simulation of abdominal trauma surgery, Proceedings of Medicine Meets Virtual Reality, Studies in Health Technology and Informatics, vol. 50, pp.117–123, 1998.

[2] R. W. Webster, D. I. Zimmerman, B. J. Mohler, M. G. Melkonian and R. S. Haluck, A prototype haptic suturing simulator, Proceedings of Medicine Meets Virtual Reality 2001, Studies in Health Technology and Informatics, vol. 81, pp. 567–569, 2000.

[3] D. Bielser and M. H. Gross, Open surgery simulation, Pro-ceedings of Medicine Meets Virtual Reality 02/10, Studies in Health Technology and Informatics, vol. 85, pp. 57–63, 2002.

[4] P. Gasson, R. J. Lapeer and A. D. Linney, Modelling techniques for enhanced realism in an open surgery simulation, In IV'04: Proceedings of the Information Visualisation, Eighth International Con-ference on (IV'04), pp. 73–78, Washington DC, USA, 2004. IEEE Computer Society.

[5] J. F. O'brien and J. K. Hodgins, Graphical modeling and animation of brittle fracture, Proc. of ACM SIGGRAPH, 1999.

[6] J. F. O'brien, A.W. Bargteil and J. K. Hodgins, Graphical modeling and animation of ductile fracture, Proc. Of ACM SIGGRAPH, 2002.

[7] K. Miller, G. Joldes, D. Lance and A. Wittek, Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation, Communications in Numerical Methods in Engineering 23 (2), 2007, pp. 121-134.

[8] D. C. Popescu, B. Joshi and S. Ourselin, Real-time topology modification for Finite Element models with haptic feedback, In Proceedings of The 11th International Conference on Computer Analysis of Images and Patterns, Springer, LNCS vol.3691, pp.846-853, 2005.

[9] H. W. Nienhuys and A. F. Van Der Stappen, Combining finite element deformation with cutting for surgery simulation, Proc. of Eurographics, 2000.

[10] D. Bielser, V. A. Maiwald, M. Gross, Interactive cuts through 3-dimensional soft tissue, Proc. of Eurographics, 1999.

[11] F. Ganovelli, P. Cignoni, C. Montani and R. Scopigno, A multiresolution model for soft objects supporting interactive cuts and lacerations, Proc. of Eurographics, 2000.

[12] N. Molino, Z. Bao and R. Fedkiw, A virtual node algorithm for changing mesh topology during simulation, ACM Trans, Graph. (SIGGRAPH Proc.) vol. 23, pp. 385–392, 2000.

[13] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross and M. Alexa, Point based animation of elastic, plastic and melting objects, In Proc. of the ACM SIGGRAPH/Eurographics symposium on Computer animation, 2004, pp. 141-151.

[14] M. Pauly, R. Keiser, B. Adams, M. Gross and L. J. Guibas, Meshless animation of fracturing solids, In Proc. of ACM Siggraph, 2005.

[15] M. Müller, B. Heidelberger, M. Teschner and M. Gross, Meshless Deformations Based on Shape Matching, Proceedings of ACM SIGGRAPH 2005, Los Angeles, USA, July 31 - August 4, 2005, pp. 471-478.

[16] S. Capell, S. Green, B. Curless, T. Duchamp and Z. Popovic, A Multiresolution Framework for Dynamic Deformations, Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, 2002, pp. 41–47.

[17] A. R. Rivers and D. L. James, FastLSM: Fast Lattice Shape Matching for Robust Real-Time Deformation ACM Transactions on Graphics (SIGGRAPH 2007), to appear.

[18] W. Wen and P. A. Heng, An improved scheme of an interactive finite element model for 3D soft-tissue cutting and deformation, The Visual Computer, vol.21, no.8-10, September 2005, pp.707-716.

[19] J. Allard, C. Ménier, B. Raffin, E. Boyer and F. Faure, Markerless 3D Interactions, In SIGGRAPH'07: ACM SIGGRAPH 2007 emerging technologies, ACM, New York, USA

[20] Z. Shujun, W. Cong, S. Xuqiang and W. Wei, Dream world: CUDA-accelerated real-time 3D modeling system, IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS), Hong Kong, China, May 11-13, 2009, pp.168-173.

[21] G. Kurillo, R. Bajcsy, O. Kreylos and R. Rodriguez, Teleimmersive environment for remote medical collaboration, In Proceedings of Medicine Meets Virtual Reality (MMVR17), Long Beach, CA, Jan 19-22, 2009, pp. 148-150.

[22] M. Teschner, B. Heidelberger, M. Müller, D. Pomeranets and M. Gross, Optimized spatial hashing for collision detection of deformable objects, Proc. of VMV, 2003.

[23] M. Alexa, J. Behr, D. Cohenor, S. Fleishman, D. Levin and C. T. Silva, Computing and rendering point set surfaces, IEEE TVCG 9, 1 , 2003,pp. 3–15.

[24] A. Adamson and M. Alexa, Approximating and intersecting surfaces from points, In Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing, 2003, pp. 230–239.

[25] M. Wicke, M. Teschner and M. Gross, Csg tree rendering of point sampled objects, In Proceedings of Paci_c Graphics, 2004.

TABLE II. EXECUTION TIME OF CUTTING

Scenarios	*Ratio*	*Reconstruction(ms)*	*Collision Detection(ms)*	*Cutting(ms)*	*Deformation(ms)*	*Total(ms)*
Scenarios 1	0.5	20.43	12.63	7.24	15.30	55.60
Scenarios 2	0.4	20.43	10.56	6.35	14.50	51.83