Journal of Visualization, Vol. 9, No. 4 (2006)373-380

# Cyber Surgery: Parameterized Mesh for Multi-modal Surgery Simulation

Liu, Q.* and Prakash, E. C.*

*1 School of Computer Engineering, Nanyang Technological University, Singapore 639798.
E-mail: liuqiang@pmail.ntu.edu.sg, asprakash@ntu.edu.sg

Received 31 October 2005
Revised 1 September 2006

*Abstract* : This paper presents a new method for modeling of virtual organs for surgery simulation. The organ models have a parameterized representation that supports real-time interactive deformation. To accomplish real-time interaction with deformable organs, instead of computing the deformation on the 3D organ models in 3D space we use a novel yet simple and fast free-form deformation on the 2D parameterized representation itself. With the parameterized mesh, we also demonstrate that realistic visual and haptic rendering can be provided for interactive surgery simulation.

*Keywords* : Surgery Simulation, Parameterization, Free Form Deformation (FFD), Haptics.

## 1. Introduction

One important issue of surgery simulation is to model the virtual human organs in a realistic way, not only visually, but also haptically. Visually, the virtual organs should have adequate number of polygons so that the surfaces can be rendered smoothly. Textures are applied to add colors and details. Some deformation methods should be applied to allow interaction between the user and the virtual organ. To add haptical realism, a haptic device, such as Phantom developed by SensAble Technologies, can be used to control the movement of the virtual surgery tool. The feedback forces should be calculated and updated in a high frequency (1 kHZ at least for Phantom). Therefore, the deformation methods must be fast enough to maintain the desired update rate.

In this paper, we introduce a parameterized representation of the virtual organs for surgery simulation. The parameterized mesh has the feature of compact and implicit data representation. This feature is utilized to handle collision detection in an efficient way. A simple free-form deformation method is introduced. Bound with the parameterized mesh, it provides not only realistic visual deformation, but also fast feedback force update.

## 2. Parameterization and Resampling

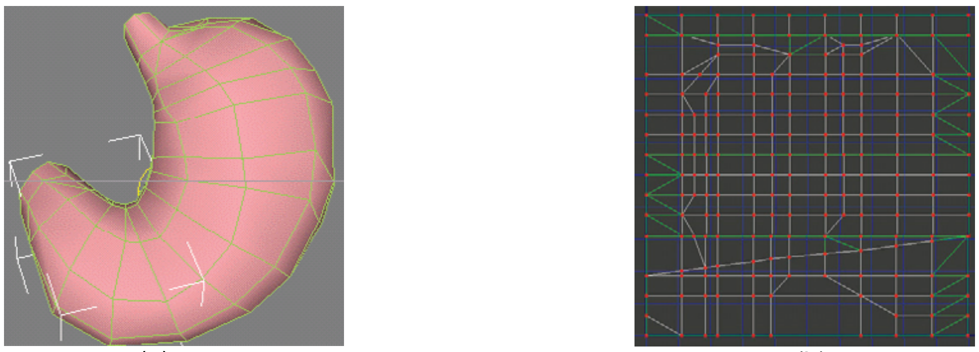
Gu et al. (2002) proposed to take advantage of parameterization and remesh an arbitrary surface onto a completely regular structure called Geometry Image, which captures geometry as a simple 2D array of quantized points. Surface information like normals and colors are stored in similar 2D arrays. The connectivity between sample vertices is implicit and therefore the data is more compact. In this paper, we utilize parameterized mesh, which is of the same idea as geometry image, to represent virtual organs for the application of surgery simulation. A parameterized mesh is a 2D array of quantized points in the 3D space of the form

$$V_{ij} = \left(x_{ij}, y_{ij}, z_{ij}\right),\ 0 \le i \le m, 0 \le j \le n$$

where $i$ and $j$ are the surface parameters, $m$ and $n$ are the dimensions of the parameterized mesh, and $x_{ij}$, $y_{ij}$, $z_{ij}$ are the x, y, z coordinates of the points, respectively. The virtual organ meshes available are usually arbitrary with low resolution. To be used for surgery simulation, they are first parameterized, then resampled into regular point arrays with high resolution.

## *2.1 Parameterization*

In surface parameterization, an arbitrary mesh is cut along a network of edge paths, and the resulting single chart is parameterized onto a unit square, so that there is a one-to-one correspondence between each vertex $V$ on the original mesh and one point on the parameter space $P = (u, v)$, with $0 \le u \le 1$ and $0 \le v \le 1$. There are many solutions readily available for the parameterization process, including the one advocated by Lee et al. (2005). For easy implementation, we parameterize the 3D virtual organ meshes semi-automatically with the pelting mapping feature provided by Autodesk 3DS Max 8®. Figure 1 shows the parameterization of a stomach mesh.



(a) (b)

Fig. 1. Parameterization of a stomach model: (a) The original 3D mesh, (b) The stomach UV map.

## *2.2 Resampling*

After parameterization, the model is resampled in $m \times n$ points which are evenly distributed in the parameter space. Each sampling point is indexed as $P_{ij} = \left(u_i, v_j\right)$, in which $1 \le i \le m$, $1 \le j \le n$, and $0 \le u_i \le 1$, $0 \le v_j \le 1$. For each sampling point, we check which triangle it lies in. The sampling point lies either on the vertex, edge or interior of a triangle. If it is on the vertex, we pick the first triangle that shares this vertex. If it is on the edge, we pick the first of the two triangles that share this edge. If a sampling point $P_{ij}$ lies in a triangle $V_1'V_2'V_3'$, which correspond to triangle $V_1V_2V_3$ on the 3D mesh, we calculate the barycentric coordinates $\left(w_1, w_2, w_3\right)$ of $P_{ij}$ as shown in Fig. 2(a).

$$w_1 = \left(u_i(v_2 - v_3) + v_j(u_3 - u_2) + u_2v_3 - u_3v_2\right) / A$$

$$w_2 = \left(u_i(v_3 - v_1) + v_j(u_1 - u_3) + u_3v_1 - u_1v_3\right) / A$$

$$w_3 = \left(u_i(v_1 - v_2) + v_j(u_2 - u_1) + u_1v_2 - u_2v_1\right) / A$$

in which $A$ is the area of triangle $V_1'V_2'V_3'$ and $A = u_1v_2 + u_2v_3 + u_3v_1 - u_1v_3 - u_2v_1 - u_3v_2$. $w_1$ is the

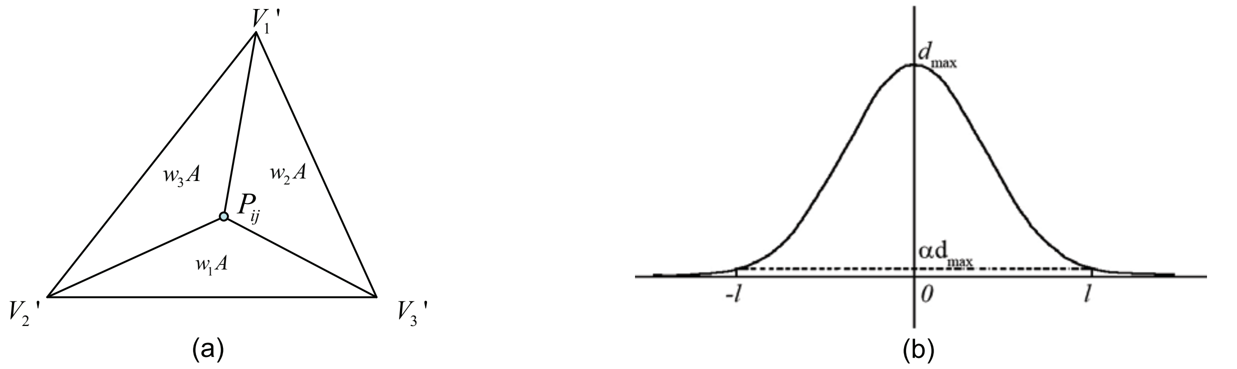



Fig. 2. (a) The barycentric coordinate $P_{ij}$ lies inside the triangle $V_1'V_2'V_3'$. (b) The shape of deformation.

area proportion of the triangle $P_{ij}V_2'V_3'$ to $V_1'V_2'V_3'$. Similarly $w_2$ and $w_3$ are the area proportion of the triangle $P_{ij}V_3'V_1'$ and $P_{ij}V_1'V_2'$, respectively. Therefore $w_1 + w_2 + w_3 = 1$.

The barycentric coordinates are used to interpolate the coordinate of a 3D point $V_{ij}$, which is the corresponding vertex of $P_{ij}$ on the 3D mesh, $V_{ij} = w_1V_1 + w_2V_2 + w_3V_3$.

### *2.3 Reconstruction*

After sampling at each sampling point, we get a $m \times n$ point array. Each point corresponds to a vertex on the 3D space, with coordinates and normal information. With this point array, the 3D mesh of the virtual organ can be reconstructed, by connecting the neighboring vertices, as shown in Fig. 3(a) and Fig. 3(b). The connectivity of vertices is implicit, and requires no additional storage space.

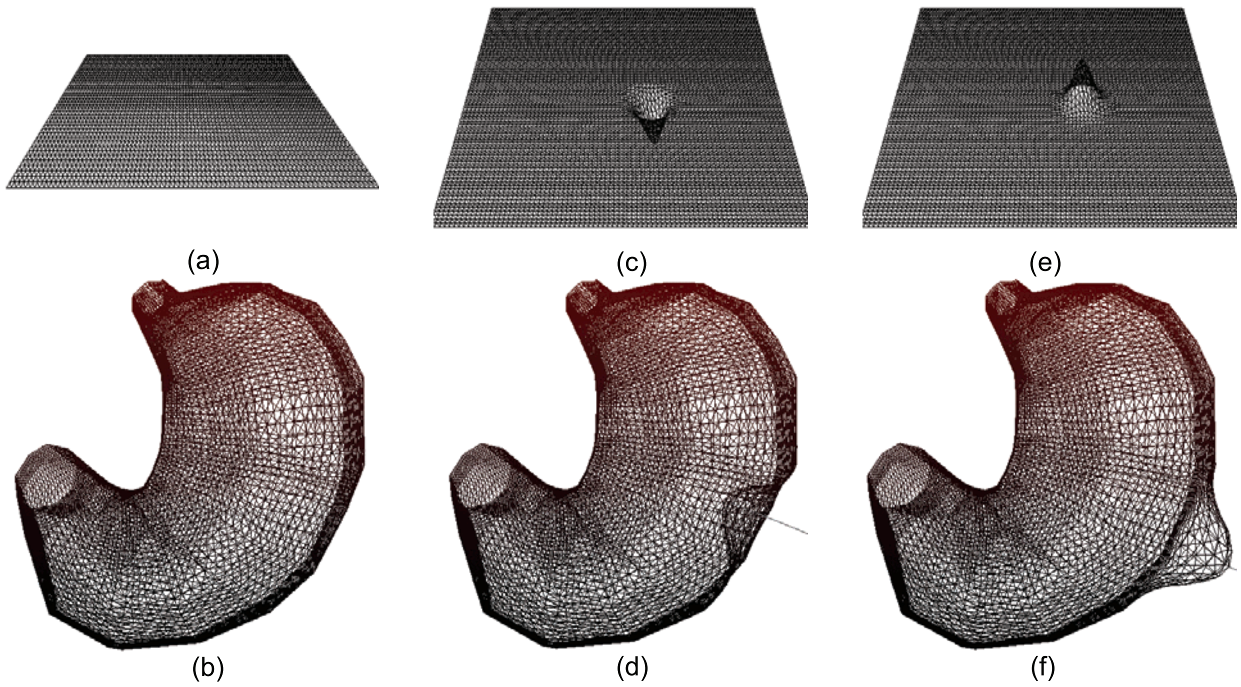



Fig. 3. Reconstruction of the 3D mesh. (a) The 2D point array. (b) The reconstructed 3D stomach model. (c), (d), (e), and (f) show the deformation of the stomach model with free form deformation.

# 3. Tool-Organ Interaction

Hirota et al. (2003) introduced a novel penalty method to simulate mechanical contact between elastic objects based on the concept of material depth. The penalty method is used for finite-element simulation. This method results in a reliable and smooth simulation. However, it is not practical for interactive simulation due to the exorbitant computational cost. In this section, we present the approaches to simulate the interaction between the surgery tool and the organ. We took advantage of the parameterized mesh and proposed a fast collision detection technique, and a FFD method to simulate the deformation of the organ. With parameterized meshes, the deformations and the feedback forces can be calculated quickly to provide fast update rate.

### *3.1 Collision Detection*

The user can control the tool interactively to touch, poke, and grasp the virtual organ. Like most surgery simulators, we assume that only the tool tip touches the organ, and only the collision between the tool tip and the organ is checked. Since the organ mesh is resampled in a high density, we assume that the collision happens at the vertex points only. Therefore, only point-point collision detection is required.

Before the simulation, the distance between the tool tip and each vertex is calculated. The vertex nearest to the tool tip is the one with shortest distance. During simulation, the nearest vertex $V_n$ is constantly traced and updated. After each time step, we check the neighborhood of the vertex $V_n^{t-1}$, which is the nearest vertex in the previous time step. The distance between the tool tip and each vertex in the neighborhood is calculated and compared. The new nearest vertex $V_n^t$ is thus updated. From this vertex two vectors are evaluated. First vector is the one from this vertex to the tool tip, noted as $N_t$. The second is the surface normal vector at that vertex, noted as $N_n$, as shown in Fig. 4. The dot product of these two vectors determines the location of the tool tip. If $N_t \bullet N_n < 0$, a collision is detected. The vector from $V_n^t$ to the tool tip is the displacement of the vertex $V_n^t$. This displacement value is transferred to the parameter space to calculate the displacements of the neighboring vertices. The virtual organ is thus deformed.

Since the movement of the tool in one time step is limited, only vertices in a small area need to be checked. The collision detection only involves a small number of point distances calculation and one dot product calculation. Therefore the collision detection is fast and efficient.

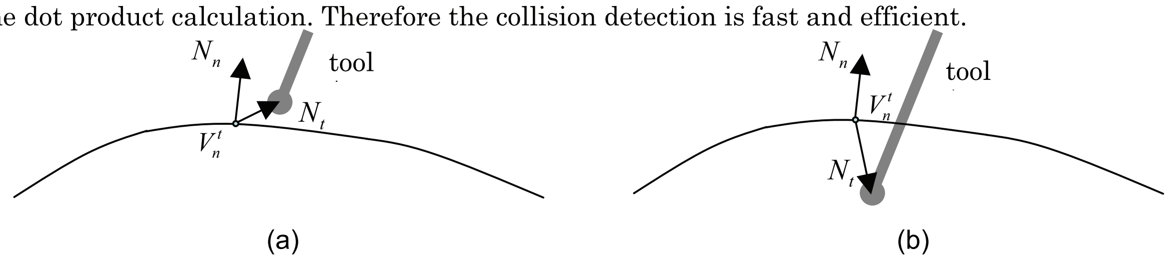



Fig. 4. The collision detection between the organ and the tool. (a) $N_t \bullet N_n > 0$, no collision detected. (b) $N_t \bullet N_n < 0$, collision detected.

### *3.2 Organ Deformation*

We designed a fast and easy free form deformation method which is based on the parameterized representation of the organ model. Although the scheme is simple, realistic deformation can be achieved.

Deformation is calculated in 2D parameter space. When one vertex is touched by the surgery tool and moved, the amount of displacement of this vertex is transferred to the corresponding point on the 2D parameter space. A Gaussian distribution function is evaluated at each surrounding point, and the displacements of these points are calculated with this distribution function. The 3D coordinates of the corresponding vertex on the mesh are updated based on the calculated displacement of each point. The shape of the 3D organ model is changed accordingly, and hence deformed, as shown in Fig. 3(c), (d), (e), and (f). The Gaussian distribution function is evaluated on a discrete 2D mesh. Moreover, the Gaussian distribution function has the shape of a "bell-shaped" hump. Therefore, before the simulation we evaluate the Gaussian distribution function at a local area only, and store the values in a look-up table. During the simulation, we only need to check the pre-computed look-up table.

### *3.3 Haptic Force Feedback*

For rapid and reasonably accurate haptic feedback, we adopt the model used in (Mahvash, 2002). The feedback force $F$ is computed directly based on the displacement $d_c$ of the vertex in contact with the tool tip. The displacement is decomposed into two components: $d_n$ which is the component in the normal direction, and $d_t$ which is the component in the tangential direction. Suppose the tool tip is of a hemisphere shape with radius $r$. The force feedback in the normal direction is computed analytically as $F_n = \frac{4}{3}\sqrt{r}E^*(d_n)^{3/2}$, where $E^*$ is defined in terms of the Young's modulus, and Poisson's ratio of the material. The force in the tangential direction is modeled as a mass-spring behavior: $F_t = kd_t$, where $k$ is the spring constant. As we can see, it requires little calculation. Therefore, fast update rate can be guaranteed for the haptic device.

## 4. Experiments and Results

We implemented the interaction between the tool and organ. We parameterized and resampled two organ models, one stomach model and one liver model, with relatively low resolution. Each is resampled by a $81 \times 81$ point array. Both resampled meshes can be simulated in real time with user interaction. Mouse input to the system controls the movement of the virtual tool. Figure 5 shows a few snapshots of the surgery simulation with the interaction between the grasper tool and the organs.

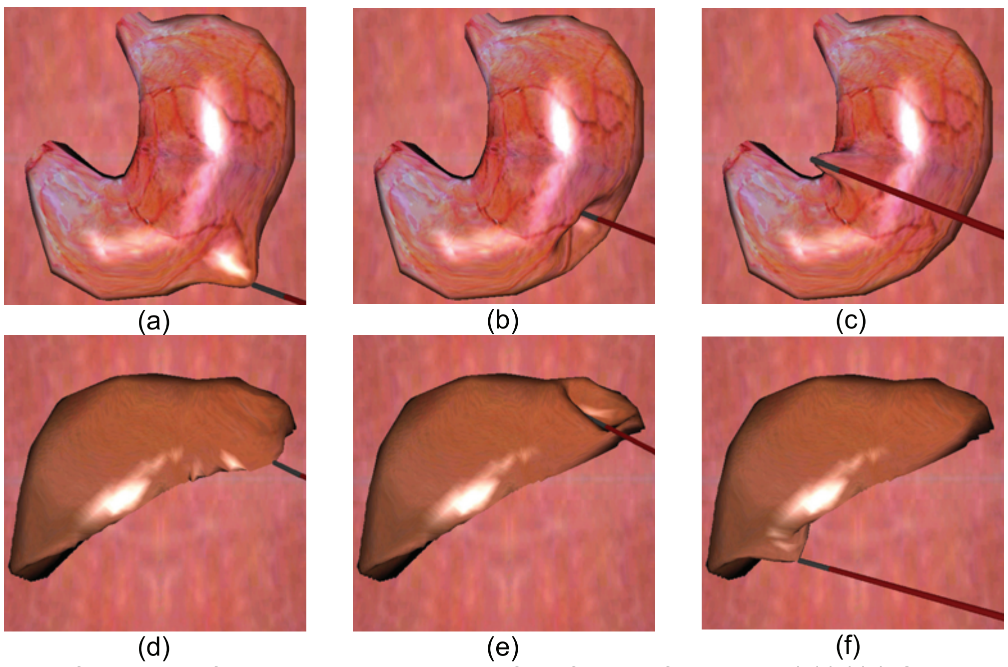



Fig. 5. The deformation of the organ models with free form deformation. (a)(b)(c) Simulation with the stomach model. (d)(e)(f) Simulation with the liver model.

## 5. Comparison and Discussion

There are two parameters that control the shape of the deformation of the organ with tool interaction. The first parameter is the maximum allowed deformation $d_{\max}$, which can be represented as a value relative to the size of the virtual organ:

$$d_{max} = \beta \frac{(x_{max} - x_{min}) + (y_{max} - y_{min}) + (z_{max} - z_{min})}{3}$$

where $x_{max}$, $x_{min}$, $y_{max}$, $y_{min}$, $z_{max}$, $z_{min}$ are the maximum and minimum coordinates in the $x$, $y$, $z$ directions respectively. The other parameter is the standard deviation of the Gaussian distribution $\sigma$. As shown in Fig. 2(b), at a point $E$ whose distance to the deformation center is $l$, the maximum length of the displacement is $d_E = \alpha d_{max}$, where $\alpha$ is the Gaussian distribution value at $E$. If value of $\alpha$ at this point is small enough, the displacement of point $E$ is almost unnoticeable (in our implementation, $\alpha = 0.01$), and $l$ can be considered as the radius of the local deformation area. $\sigma$ can be represented as a function of $l$ as $\sigma = -l^2 / \ln \alpha$.

Now the shape of the deformation is controlled by two parameters, $l$ and $\beta$. In Table 1, we compared the shape of the deformation of the stomach model with different $l$ values and $\beta$ values. Similarly, in Table 2, we compared the shape of the deformation of the liver model with different $l$ values and $\beta$ values.

Table 1. The shape of the virtual stomach deformed with different parameter values.

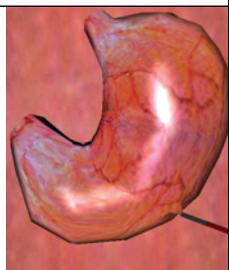
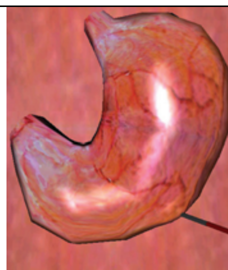
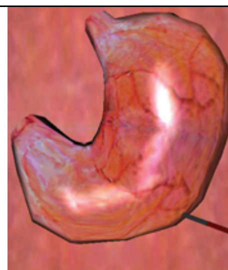
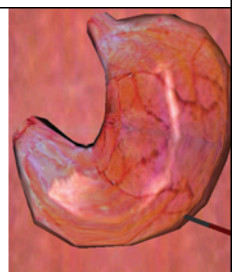
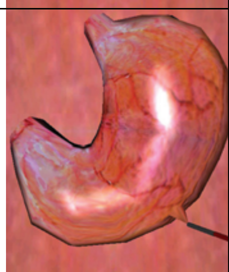
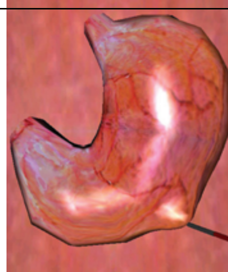
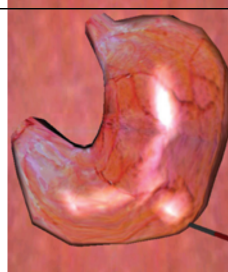
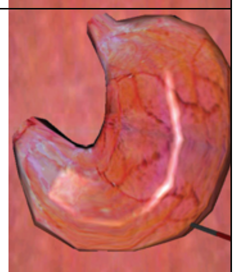
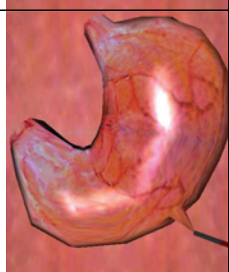
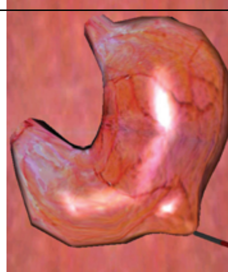
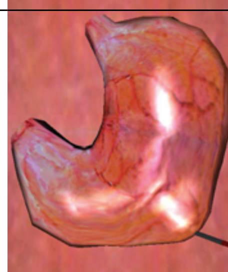
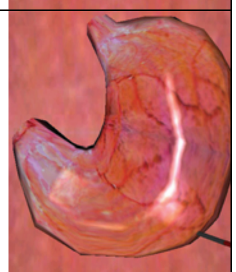
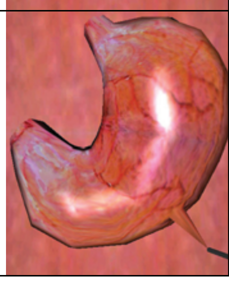
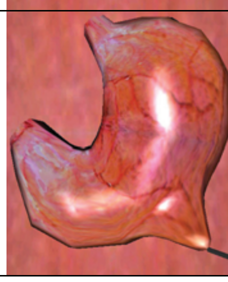
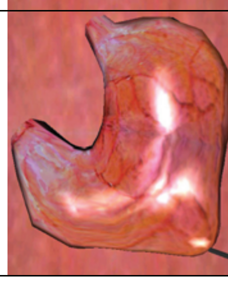
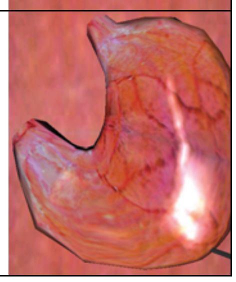
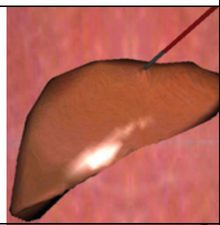
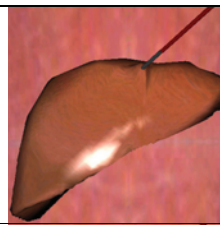
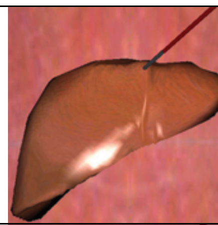
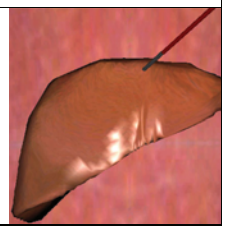
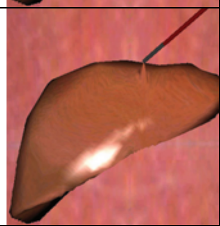
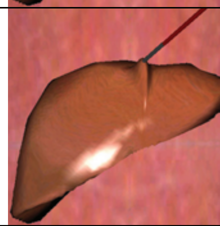
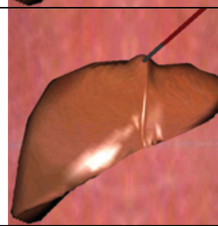
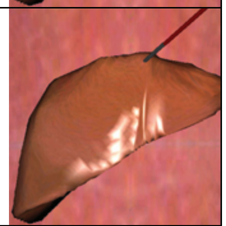
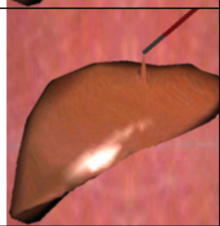
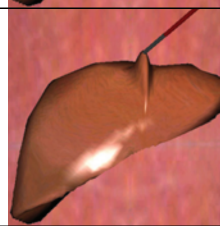
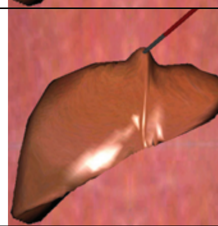
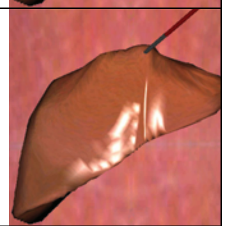
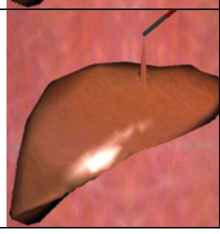
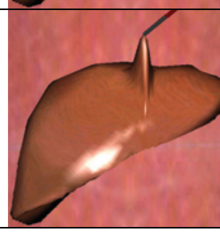
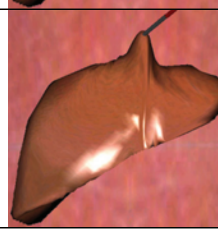
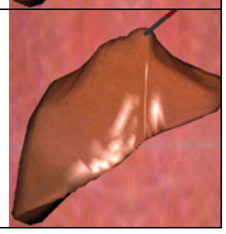
| | $l = 1$ | $l = 5$ | $l = 10$ | $l = 20$ |
|---|---|---|---|---|
| $\beta = 0.05$ |  |  |  |  |
| $\beta = 0.1$ |  |  |  |  |
| $\beta = 0.15$ |  |  |  |  |
| $\beta = 0.25$ |  |  |  |  |

Table 2. The shape of the virtual liver deformed with different parameter values.

| | $l = 1$ | $l = 5$ | $l = 10$ | $l = 20$ |
|---|---|---|---|---|
| $\beta = 0.05$ |  |  |  |  |
| $\beta = 0.1$ |  |  |  |  |
| $\beta = 0.15$ |  |  |  |  |
| $\beta = 0.25$ |  |  |  |  |

In our experiments shown in Table 1 and Table 2, we find that for $\beta = 0.25$, the deformations are not realistic. Similarly, when $l = 20$, the deformations are not realistic. So for simulation we recommend the diagonal values to produce acceptable deformation. We can conclude that, to avoid unrealistic free-form deformation,

1. the value of $\beta$ should increase with the value of $l$, and
2. the value of $l$ and $\beta$ should not to be too large, i.e., the deformation area and the maximum allowed deformation should not be too large.

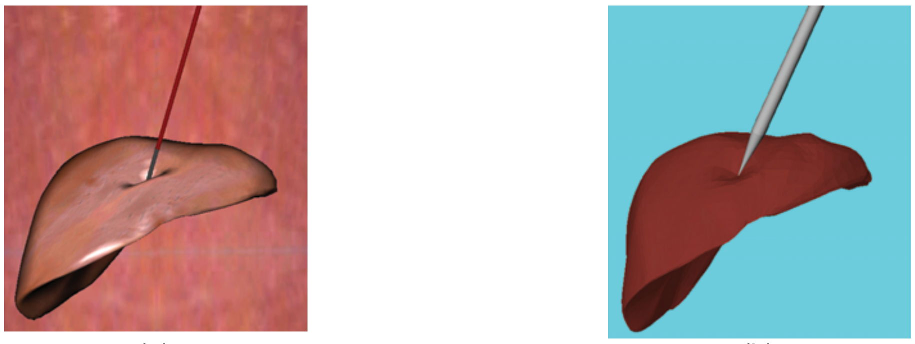
In practice, $\beta$ is set to increase with $l$, in a discrete function, e.g. $\beta = \beta_1$ if $0 < l < l_1$, $\beta = \beta_2$ if $l_1 \le l < l_2$, $\beta = \beta_3$ if $l_2 \le l < l_3$,...

The organ deformation result from our experiments is compared with the result of previous work (De, 2001) in Fig. 6. The example in De (2001) used finite spheres for real time simulation and display of deformation. In our case we use a parametric free form deformation, and use a bump mapped specular shading for visualization. The comparison shows that, although our method is simple, realistic result can be achieved. Moreover, our visual rendering effect is more realistic.

# 6. Conclusion and Future Work

We introduced a parameterized representation of 3D organ meshes for the simulation of laparoscopic surgery. Random input virtual organ meshes are parameterized and resampled into regular high resolution models. We introduced a free-form deformation approach for the parameterized mesh to simulate the interaction between the tool and the organ to achieve an update rate fast enough to provide haptic feedback.

In future work, we plan to improve the deformation with more sophisticated deformation method, such as mass-spring model (Zhang, 2003). The deformation can be calculated similarly in the 2D parameter space, then transferred back to the 3D mesh. Since the vertices are regularly spaced in the parameter space, no remeshing is required for mass-spring model or FEM. More realistic deformation can be produced with these physically based deformation methods. The parameterized representation can also be utilized to deal with contact of the deformable organ with other objects, such as a plane or another deformable organ, which will be discussed in our upcoming papers.



(a) (b)

Fig. 6. The comparison of the organ deformation results. (a) Our method. (b) Result from (De, 2001).

## *Acknowledgements*

We would like to thank Dr. M. A. Srinivasan, Director of Touch lab, MIT, who provided access to the Touch Lab at MIT, which helped this research immensely.

## *References*

De, S., Kim, J. and Srinivasan, M. A., A meshless numerical technique for physically based real time medical simulations, Medicine Meets Virtual Reality 2001, (2001), 113-118.

Gu, X., Gortler, S. J. and Hoppe, H., Geometry images, ACM SIGGRAPH 2002, (2002), 355-361.

Hirota, G., Fisher, S. and State, A., An improved finite-element contact model for anatomical simulations, The Visual Computer, 19-5 (2003), 291-309.

Lee, H., Tong, Y. and Desbrun, M., Geodesics-based one-to-one parameterization of 3D triangle meshes, IEEE Multimedia, 12-1 (2005), 27-33.

Mahvash, M., Hayward, V. and Lloyd, J. E., Haptic rendering of tool contact, Proc. Eurohaptics 2002, (2002), 110-115.

Zhang, Y., Prakash, E. C. and Sung, E., Hierarchical facial data modeling for visual expression synthesis, Journal of Visualization, 6-2 (2003), 313-320.

## *Author Profile*

Qiang Liu: He received his B. Eng. in computer science and technology in 2002 from Zhejiang University, China. He is currently a Ph.D student in the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include virtual reality, computer graphics, computer animation, and 3D visualization.

Edmond Cyril Prakash: He is an Assistant Professor of Computer Science at the Nanyang Technological University, Singapore. He received his Ph.D degree from Indian Institute of Science, Bangalore. Edmond's research focus is on exploring the applications of visualization in engineering, science, medicine, finance and entertainment. He also serves as a Research Director for the gameLAB at NTU, Singapore.