A Hybrid Collision Response in a Haptic Virtual Surgery System

Xiaobo. Li[1], Lixu. Gu[1, 2], Shaoting. Zhang[2], Jingsi. Zhang[2], Guangchao. Zheng[2]
[1]Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai, China
[2]School of Software, Shanghai Jiao Tong University, Shanghai, China

Abstract—**Collision response is an important component in a virtual surgery system, due to the requirements of real-time response and simulation realism. In this paper, we propose a novel hybrid collision response algorithm for generating a smooth feedback force. This algorithm takes the advantages of three traditional methods as well as gets rids of drawbacks of them. It also provides an approach to generate a reasonable feedback force. After collision response processing the variance of feedback force has been reduced from 30026 to 0.0003, form 39095 to 0.06 and from 5455 to 0.008 in x, y and z dimension respectively. The result shows that our methods can be used in a real-time response and simulation realism required haptic surgery simulator.**

Keywords—**Collision response, feedback force, virtual surgery**

I. INTRODUCTION

With the development of virtual reality (VR) technology and haptic devices, a VR based haptic virtual surgery system (HVSS) can be implemented for repeatedly surgery training. But there are still many bottlenecks. One of them is collision response. The function of collision response is to compute the behavior of two objects after a collision between them was detected. It also computes the feedback force results from collision in a HVSS. We have developed a HVSS for laparoscope based minimally invasive surgery (MIS) training, and a hybrid collision response algorithm is implemented in this system.

Collision response schemes were mainly categorized into three distinct approaches which are called penalty force approaches, contact surface computation and analytical solutions respectively. The idea of a penalty force is first mentioned in [1]. This approach is widely used in interactive real-time simulations due to its computational efficiency. However, one of the biggest drawbacks of this method is penetration, which is not wanted in a HVSS. Contact surface based algorithms have been discussed in [2]. and [3-9].The biggest advantage of the contact surface computation approach is that it avoids penetration; however, it needs a high sampling frequency, which cannot meet the requirement of real-time response in a HVSS. Analytical collision response approach applied to deformable models is discussed in [10]. and [11]. One advantage of the analytical solution is that it's very precise and it avoids the penetration too. But it will cause an extreme computational overhead, which is intolerable for a real-time response required HVSS. Another important task of HVSS is to provide a feedback force approximation to the real operation. Variety of forces for haptic simulation has been used [12]. A reasonable feedback force should be continuous, not only on scale, but also on direction, otherwise user will feel a discontinuous force which cannot be appeared in the real word. TABLE I shows the comparison of the three approaches

II. METHODOLOGY

As discussed in previous section, collision response plays an important role in a HVSS, but there are still some drawbacks on the traditional algorithms. We proposed a hybrid collision response method which takes advantages of the three approaches already existed and avoids some disadvantages of them. The main idea of the method is that to divide the whole procedure into three steps which are called preprocessing step, deformation step and force feedback step. See Fig. 1.

A. *Preprocessing Step*

In the preprocessing step, analytical solution is used. We just add some backtracking steps at the time when collision being detected that it lends no effect on the whole algorithm's time complexity but increases the frequency of computation on judgment. The backtracking method is described as algorithm I, see Fig. 2. where *step* is the time of backtracking, *stepAmount* is a predefined threshold of step, *timeFactor* is a coefficient of time step, *epsilon* is a predefined small number, and when *timeFactor* $\leq$ *epsilon*, the process will be stopped. *Inside* is a coefficient that marks if a point is inside an object, *currentPos* is the coordination of current point and *moveDir* is the vector of movement.

B. *Deformation Step*

In the deformation step, soft tissues' deformation is calculated. Three algorithms could be chosen which are

TABLE I
COMPARISION OF THREE COLLISION RESPONSE APPROACHES

Parameters to be compared	Approach		
	Penalty Force Based	Contact Surface Based	Analysis Based
Penetration	Penetration	No penetration	No penetration
Real-time	Good	Moderate	Bad
Stability	Bad	Moderate	Good
Sampling Frequency	Moderate	High	No mentioned

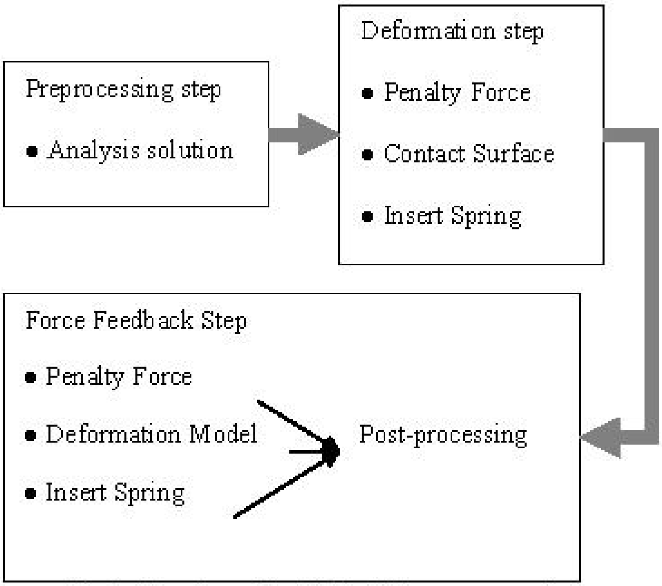
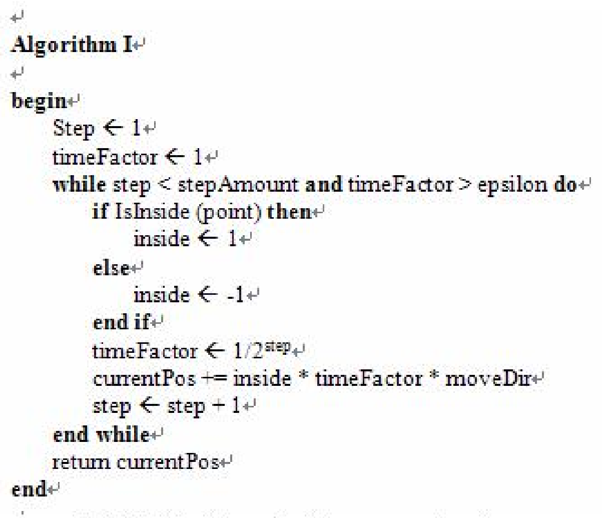


Fig. 1. Three steps of the hybrid collision response method

penalty force based algorithm [13]. contact surface based algorithm [2]. and insert spring based algorithm which adding some springs between two objects when they are colliding, see Fig. 3. Those springs' original length is 0, so points will finally return to the balance position, and then the collision between these two objects will be released. Which algorithm should be selected depends on the requirement of different system. The penalty force based algorithm and the insert spring based algorithm is suitable for a real-time response required system, the contact surface based algorithm is suitable for a high quality of vision simulation required system and the insert spring based algorithm is also suitable for a system which requires high stability.

C. *Force Feedback Step*

In the force feedback step, response force is computed. There are also three selections: penalty force based method [13]. deformable model based method which calculates internal force of each mass point, and then sums them up to get the collision area's response force, and insert spring system, whose basic idea is the same as that discussed in the previous section. In a real-time response required system, deformable model based method is more suitable, whereas in a force continuity required system, insert spring based method is more suitable. After working out a response force, we smooth the force in a post-processing approach to get a feedback force. In our HVSS, a force feedback device called PHANToM Desktop is used. It provided by company Sensable, can move in 3D space and let us feel the feedback force bodily. It's coefficients of rigidity in 3 dimensions are 1.86N/mm, 2.35N/mm and 1.48N/mm respectively which basically fulfill the requirement of surgery simulation. However, the force it gives us is between 0N and 7.9N, but the response force we worked out is far out of this range. So we have to do some post-processing of the force. This post-processing includes two steps: map step and smooth step.



```
Algorithm I

begin
    Step ← 1
    timeFactor ← 1
    while step < stepAmount and timeFactor > epsilon do
        if IsInside (point) then
            inside ← 1
        else
            inside ← -1
        end if
        timeFactor ← 1/2^step
        currentPos += inside * timeFactor * moveDir
        step ← step + 1
    end while
    return currentPos
end
```

Fig. 2. Backtracking method in preprocessing step

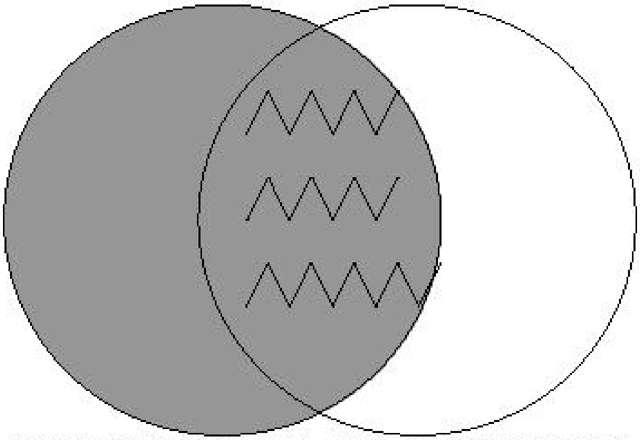


Fig. 3. Insert spring based approach. When the gray entity colliding with the white one, some springs are inserted between them

Equation (1) is used to map force into a smaller range in the map step. Here *froceX*, *forceY*, *forceZ* are response forces in 3 dimensions respectively, *scaledForceX*, *scaledForceY*, *scaledForceZ* are the forces after mapping, *forceThreshold* is the biggest force sends to PHANToM Desktop, *forceMax* represents the biggest absolute value of the forces in 3 dimensions and *commonForce* is the force appears with a high frequency.

$$\begin{cases} scaledForceX = \dfrac{2forceX \cdot \arctan[(forceMax \cdot \pi)/(4commonForce)]forceThreshold}{\pi \cdot forceMax} \\ scaledForceY = \dfrac{2forceY \cdot \arctan[(forceMax \cdot \pi)/(4commonForce)]forceThreshold}{\pi \cdot forceMax} \\ scaledForceZ = \dfrac{2forceZ \cdot \arctan[(forceMax \cdot \pi)/(4commonForce)]forceThreshold}{\pi \cdot forceMax} \end{cases} \quad (1)$$

In the smooth step, the force is smoothed to increase its quality of continuity. Here a linear interpolation method is used. The algorithm is described as algorithm II, see Fig. 4. where *responseForce* is the force from collision response, *feedbackForce* is the recent force in set to PHANToM Desktop, *stepForce* is the increment of every time step, and

```
Algorithm II↵
↵
begin↵
    if responseForce != GetForceFromCR() then↵
        responseForce ← GetForceFromCR()↵
        step ← 0↵
        stepForce ← (responseForce-
        feedbackForce)/stepAmount↵
    end if↵
    while step < stepAmount do↵
        feedbackForce += stepForce↵
        step ++↵
    end while↵
    setPhantomForce(feedbackForce)↵
end↵
```

Fig. 4. linear interpolation for smoothing feedback force

stepAmount is a pre-defined number of interpolation. This algorithm has a low time complexity, and performs well in practice. And what should be noticed is the value of *stepForce*, if it's too big, users will always feel a transitional force instead of real feedback force; and if it's too small, user will feel a stagnant all the same. After some experiment, the result is good if the value is between 50ms and 100ms.

III. RESULTS

The results presented here are measured on a PC with Pentium IV 2.6GHz CPU, 2GB RAM, and GeForce 6800 display card, the Operation System is Microsoft Windows XP. The implementation is written in C++ and compiled with the Microsoft Visual Studio .Net 7.0. We chose a clinical kidney data which contains 1922 points and 6538 tetrahedra, a created tumor data which contains 206 points and 583 tetrahedra, and a sphere shape tool which contains 54 points and 164 tetrahedra. The parameters are shown in TABLE II. The aim here is to remove a tumor from kidney with a surgery tool in a laparoscope based MIS environment. Fig. 5. shows the result when the instrument touches the tumor.

TABLE III shows the variance of feedback force in three dimensions. The variance is a measure of how spread out a distribution is, i.e. the measure of variability. It is defined as follows where v is the variability, f_i is the ith force, n is the total number of forces to be computed.

TABLE II
EXPERIMENTAL SETUP OF COLLISION RESPONSE

Objects	Parameters		
	Shape	Number of points	Number of tetrahedra
Kidney	Clinical data	1922	6538
Tumor	Sphere	206	583
Device	Sphere	54	164

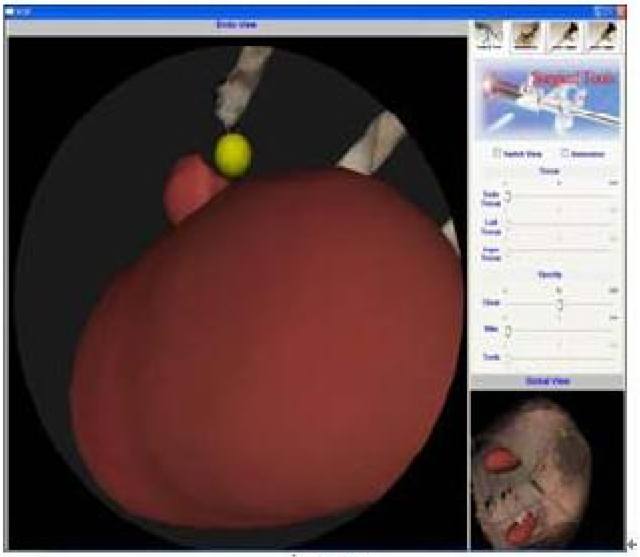


Fig. 5a

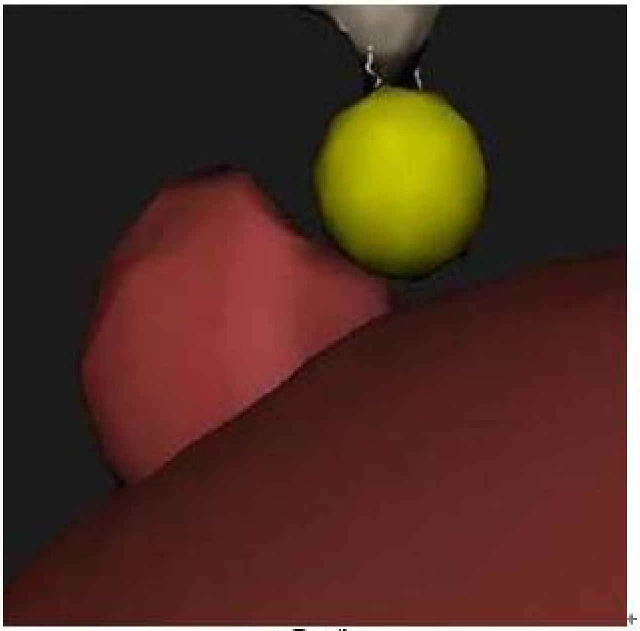


Fig. 5b

Fig. 5. Result of collision response
PHANToM Desktop is used to control a virtual object (the yellow sphere in the figure) to collide with a tumor on a kidney. The tumor is deforming when collision has been detected. Fig. 5b. magnifies the location where collision has been detected in fig. 5a and users can feel a generated feedback force through PHANToM Desktop.

$$v = \frac{\sum_{i=1}^{n-1}(f_{i+1} - f_i)^2}{n} \qquad (2)$$

IV. DISCUSSION

Modules in the three steps discussed in section II have high cohesion and low coupling, so there are many combinations, and which one will be selected depends on the requirement of different system. Our haptic virtual system requires high quality of visual and haptic simulation as well as real-time ability, so in the second step, the

TABLE III
VARIANCE OF FEEDBACK FORCE IN THREE DIMENSIONS

Forces	Variance		
	X dimension	Y dimension	Z dimension
Penalty force	59.18	14836	3242
Deformable model	30026	39095	5455
Insert spring	49.73	5556	587.9
Force after smoothing	0.0003	0.06	0.008

contact surface based algorithm has been used, and in the third step, the insert spring based method has been used, the force computed has been mapped to 0~2N and smoothed as the final feedback force. The process is described in Fig. 6.

In our case, a small variance indicates a smooth feedback force whereas a big one indicates a force not so smoothly. From TABLE III we see that after processing, the variance is sharply down, i.e. the feedback force is much smoother after processing, so it better represents a real word force.

V. CONCLUSION

We developed a haptic virtual surgery system for laparoscope based MIS training. An important approach which is called collision response is discussed. In this paper, a hybrid collision response method is proposed. This method not only takes advantage of the existing three approaches, but gets rids of their disadvantages as possible. Furthermore, after post-processing, it generates a continuous real-like feedback force. So a haptic simulator is implemented.

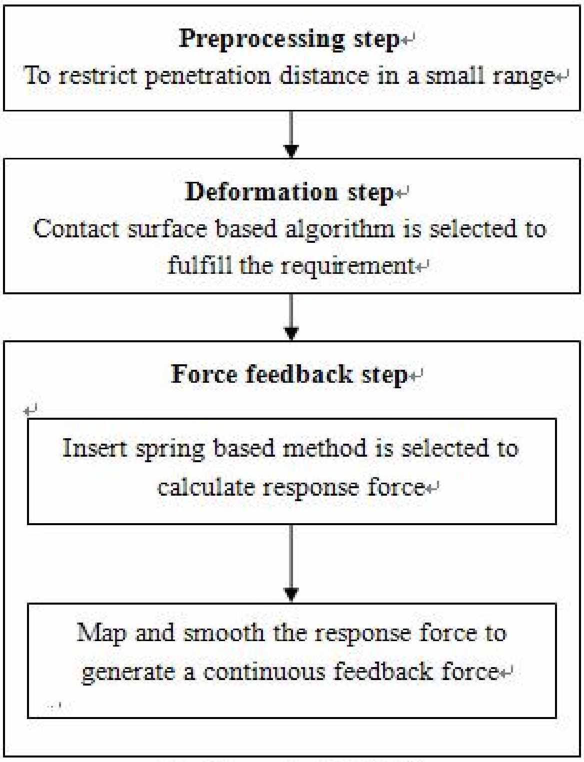


Fig. 6. Processing in the HVSS

ACKNOWLEDGMENT

We'd like to show our appreciation to Shanghai Ren Ji hospital, for the doctors there provided us lots of clinical data and gave us many clinical advices. We are also grateful to our team members Eitz Mathias, Jalda Dworzak and Jan Boehm with whom we had a lot of long discussions about our work and who provided a lot of helpful comments and hints.

REFERENCES

[1] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. "Elastically deformable models," *ACM SIGGRAPH Computer Graphics,* 21(4), pp. 205–214, 1987.

[2] J. Spillmann and M. Teschner. "Contact surface computation for coarsely sampled deformable objects." *Proc. Vision, Modeling, Visualization VMV'05, Erlangen, Germany*, pp. 289–296, 2005.

[3] D. Baraff, "Analytical methods for dynamic simulation of non-penetrating rigid bodies," *ACM SIGGRAPH Computer Graphics,* 23(3), pp. 223–232, 1989.

[4] D. Baraff, "Issues in computing contact forces for non-penetrating rigid bodies," *Algorithmica*, 10, pp. 292–352, 1993.

[5] D. Baraff, "Fast contact force computation for non-penetrating rigid bodies," *Siggraph*, pp. 23–34, 1994.

[6] F. Faure, "An energy-based method for contact force computation," *Siggraph*, pp. 357–366, 1996.

[7] C. Duriez, C. Andriot, and A. Kheddar. "Signorini's contact model for deformable objects in haptic simulations," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[8] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. "Realistic haptic rendering of interacting deformable objects in virtual environments," *IEEE Transactions on Visualization and Computer Graphics*.

[9] M. Pauly, D.K. Pai, and L.J. Guibas. "Quasi-rigid objects in contact," *Proc. Symposium on Computer Animation*, 2004.

[10] K.L. Johnson. "Contact mechanics," *Cambridge University Press*, ISBN0521255767, 1985.

[11] J.T. Oden N. Kikuchi. Contact problems in elasticity: A study of variational inequalities and finite element methods. ISBN 0898714680, 1988.

[12] http://www.sensable.com. 2005.

[13] M. Teschner, B. Heidelberger, M. Muller, and M. Gross. "A versatile and robust model for geometrically complex deformable solids," *Computer Graphics International*, 2004. Proceedings, pp. 312–319, 2004.

978-1-4244-1868-8/08/$25.00 ©2008 IEEE.