A Method for Collision Response between Deformable Objects in Virtual Surgery

Bo Zhu[1], Lixu Gu[1,2], Jingsi Zhang[1]
[1]School of Software, Shanghai Jiaotong University, Shanghai, China
[2]Computer Science and Engineering Department, School of Electronic, Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai, China

Abstract — **Virtual surgery has drawn a lot of attention during the past few years for its repeatability and low cost. In a virtual surgical environment, collision between organs or between organs and surgical instruments occurs frequently. Therefore, an efficient and stable method is needed to simulate these collision processes. Nevertheless, contrary to rigid solid animation where complete analytical solutions have been found, simulating colliding response process between deformable objects still remains a challenge.**
In this paper, we present an advanced method to model collision response between tissues. It computes response force and generates an exact contact surface between colliding objects. Our method considers both geometric features such as penetration depth and physical properties such as relative velocity and friction to compute proper collision forces. Further, a breadth-first searching algorithm is applied in distributing these calculated reaction forces in different parts of the object in order to generate exact contact surface between interacting objects. Its application in a minimal invasion virtual surgery is used for case study.

Keywords — **contact surface, collision response, deformation, penalty force, penetration depth, virtual surgery.**

I. Introduction

Realistic and real-time tissue-tissue interaction and tool-tissue interaction is a critical part to build a successful virtual surgery system. Therefore, an efficient and robust method for collision response is necessary. Among the solutions used to cope with collision response in interactive simulation environments, penalty method and contact surface method are probably the most widely used. However, each of these two approaches has its own flaws. Penalty force method computes response forces based on penetration depth between the two colliding objects. Despite of its computational efficiency, it cannot generate contact surfaces of the interacting objects based on these forces properly and thus cannot avoid the mutual penetration which is visually unrealistic. An alternative solution which directly computes contact surfaces of colliding deformable objects based on the geometric information could address this problem, nevertheless, it is difficult to calculate the continuous collision forces between deformable objects, which are crucial in computer simulation of haptic response.

This paper present an advanced method to generate plausible contact surfaces based on penetration depth and some simplified physical laws. Firstly, we calculate penalty forces of the colliding deformable objects based not only on the interpenetration depths but also on velocity, mass and colliding property of the objects. Second, differently with the existing approach in which the distribution of response force is limited in colliding points, we use a breath-first searching algorithm (BFS) to distribute the force in a wider scope. Our main point is that merely handling points from intersected area of the colliding points is not sufficient to simulate realistic collision response – more points related to the contact surface are needed to be calculated in a global level.

II. Methodology

This section provides an overview of the proposed algorithm followed by a detailed description of its four stages.

3.1 Algorithm Overview

Our method is used in discrete-time simulation systems, which means that the deformation process between colliding objects is simulated in a series of discrete time slices. It takes a set of potentially colliding tetrahedral meshes as input. When colliding occurs, it first computes the response forces based on the points in intersected areas. The method proceeds in four consecutive stages:
Stage 1 detects all colliding points in the scene by using a dynamic special hashing approach and calculates the penetration depths for each colliding points.
Stage 2 computes the penalty force for each colliding point based on physical properties involving relative velocity and mass of the point.
Stage 3 finds the point with the maximal penetration depth and regards it as the seed point of the redistribution. Then it redistributes the penalty forces calculated in stage 2 within the scope of colliding points by using a width-first searching algorithm.
Stage 4 identifies all colliding points adjacent to one or more non-colliding points as border points. These border points are used as the root points of another width-first traversing process within the scope of non-colliding points.

As a result of this algorithm, all colliding mesh objects in the scene have an appropriate feedback force as a collision response. More importantly, these forces are distributed in a specific way (described in section 3.3 and 3.4) among mesh points, involving all colliding points and part of the non-colliding points, which enable the generation of a proper contact surface. The redistribution process of the response forces could help to directly compute the deformed surface at equilibrium more quickly and simply because it doesn't

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

care the complexity of the deformable model itself and the physical and spatial information of the collided object.

3.1 Points Collisions and Penetration Depths

The first stage contains two phases: detecting all object points that collide with any tetrahedral mesh in the scene and calculating their penetration depths and directions. First, the volumetric collision detection is accomplished by the spatial hashing approach as presented in [3]. In the second phase, we calculate the penetration depth for each of the colliding points. There are several methods which have been proved to be efficient in computing consistent penetration depths and directions with triangulated surfaces. One approach is to find one closest surface to the point and take the minimal distance from the point to that surface as penetration depth. Another approach is to approximate the penetration based on a set of close surface features. In our method we choose the latter approach. (see more details of the two approaches in [2])

3.2 Response Forces of Colliding Points

The second stage computes the penalty force for each colliding points based on the penetration depth and on some physical properties of the colliding objects including relative velocity and mass. Usually, response force of one mass point is computed based on the assumption that there exists a linear relationship or an exponential relationship between penetration depth and penetration force. Nevertheless, our method takes into consideration not only penetration depth but also the mass and the relative velocity of the point. The force computing process is based on one simple physical law – conservation law of momentum. In discrete-time simulations, the penalty force satisfies the equation as follows:

$$\left(F_{penalty} - f(\mathbf{p}_i)\right) \cdot \Delta t = m_i \left(V(\mathbf{p}_i) - V(c) \right) \qquad (1)$$

To ensure that the response process is consistent and stable, the relative separate velocity should be as low as possible, so that the disturbance of collision response is within a very fine scope. In order to reduce the relative separate velocity and to increase the continuity of the response force, we introduce the friction term f in equation (1). When a collision occurs, the friction and damping force f at a colliding point P is expressed by:

$$f(\mathbf{p}_i) = \lambda_i \lambda_j (V(\mathbf{p}_i) - V(\mathbf{c})) \qquad (2)$$

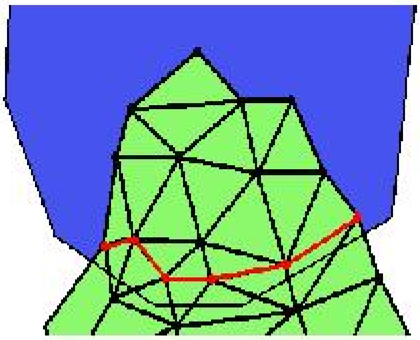
After using the momentum equation and adding the friction term, we could get a smooth and continuous response force as demonstrated in part two of the experiment.

3.3 Response Force Redistribution among Colliding Points

As our experiment shows, simply adding the colliding forces onto the points in intersected area cannot ensure that the deforming process is suitable. In particular, it cannot ensure that generated contact surface is plausible. Because in that case, the impact of the collision force has been limited in a small scope–the intersected area–and therefore the deformation of the non-colliding part, which also have an great influence on contact surface generation, relies more on the internal interaction of the deformable model itself than on the colliding response force.(see experiment one) .

To solve this problem, we distribute the colliding force onto colliding points and part of non-colliding points related to the contact surface instead of merely onto points in intersected area. In stage 3 and stage 4, we present an innovative force-distributing method which could simulate one of the important characteristics of collision response between deformable objects – the attenuation of the internal forces on force diffusing process.

Internal force attenuation, which helps to generate a proper contact surface in our method, means that the colliding forces on the mesh points attenuate gradually from points with deep penetration depth to those with shallow one, from points closer to the border surface(see figure 2 and figure 3) to those further from it. And therefore the attenuation process is possessed of the marked directivity.



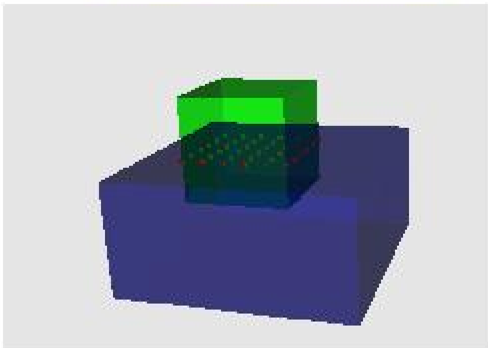


Figure 2 and 3: Border surface (illustrated by red lines in Fig2 and by points in Fig3). In our method, we identify all colliding points adjacent to one or more non-colliding points as border points. And border surface is composed of all border points.

We divide a force on one mesh point into two parts: one part is acting on the point itself and the other is passed to the points around it along the attenuation direction. To control the proportion of the two parts, an attenuating coefficient λ has been introduced. Further, the (1- λ)f part is distributed into all adjacent points along the attenuation direction. This influence is dependent on the distance between the two points. The respective weighting function has to be positive for all non-zero distances and increasing

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

for decreasing distances. The weighting function for influence $\omega(\mathbf{p}_i, \mathbf{p})$ is expressed by:

$$\omega(\mathbf{p}_i, \mathbf{p}) = \frac{1}{\|\mathbf{p}_i - \mathbf{p}\|^2} \quad (3)$$

Based on the influence $\omega(\mathbf{p}_i, \mathbf{p})$, the response force on the adjacent point $\mathbf{p}_j$ is updated as:

$$\mathbf{F}_i' = \frac{(1-\lambda)\mathbf{F} \cdot \omega(\mathbf{p}_i, \mathbf{p})}{\sum_{j=1}^{k} \omega(\mathbf{p}_j, \mathbf{p})} + \mathbf{F}_i \quad (4)$$

with k denoting the number of adjacent points of point p along the attenuation direction.

Based on the computed penalty force information for colliding points in the second stage, the third stage propagates the information to all border points. Border points plays an important role in stage 3 and 4 because the surface composed of these points is directly related with the prospective deformed shapes after the colliding response. In stage 3, the penalty forces on the border points are computed by a breadth-first searching algorithm. Firstly, point with the deepest penetration depth in all colliding points is selected as the root point for the searching. During the process of breadth-first searching, we use two queues, Qroot and Qbuffer, to preserve the information of current root points and the information of root points in next circle. Qroot is initialized with the selected root point, and Qbuffer is initialized with empty. The pseudo-code description of the algorithm is presented:

```
for each colliding point pi
    mark[ i ] = false
repeat
  while Qroot is not empty
    pop pi from Qroot
    for each adjacent point pj of pi
        if mark[ i ] = false
            then update Fj
                push pj into Qbuffer
    mark[ i ] = true
  for each point pk in Qbuffer
    pop pk from Qbuffer
    push pk into Qroot
until Qroot is empty
```

At the end of the third stage, penalty forces computed in stage 2 is redistributed among colliding points. In contrast to existing penalty force approaches that merely consider forces on each point separately, the weighted averaging of distances provides a continuous behavior of the penalty force for colliding points that are adjacent to each other. Non-plausible penalty forces due to the discrete meshes of the penetrated object are avoided.

3.4 Response Force Redistribution among Non-Colliding Points

Stage 4 computes the penalty forces for part of the non-colliding points based on the breadth-first searching algorithm similar to the one used in stage 3. We introduce a coefficient µ to denote the proportion of the non-colliding points responding to the collision. The force-diffusing algorithm is similar to the algorithm in stage 3. The searching scale is refined by µN-Nc, in which N denotes the total number of points in the object, and Nc denotes the number of colliding points. The algorithm starts with border points computed in stage 3:

```
for each non-colliding point pi
    mark[ i ] = false
for each border point pi
    push pi into Qroot
repeat
  while Qroot is not empty
    pop pi from Qroot
    counter++
    if counter > µN-Nc
        return
    for each adjacent point pj of pi
        if mark[ i ] = false
            then update Fj
                push pj into Qbuffer
    mark[ i ] = true
  for each point pk in Qbuffer
    pop pk from Qbuffer
    push pk into Qroot
until Qroot is empty
```

At the end of the fourth stage, the colliding forces have been distributed into colliding points and related non-colliding points properly.

III. Results

We have integrated our method in a haptic virtual surgery system for laparoscope based Minimal invasive surgery (MIS) training. Many experiments have been carried out to compare the quality and performance of the proposed method with the standard approach. All test scenarios presented in this section have been performed on a PC Pentium 4, 2.8GHz, GeForce 7300 LE GPU.

In a first test, we test the collision response process between two deformable tumors consisting of 583 tetrahedrons in the virtual surgical environment. At the beginning of the experiment, large penetrations between the objects have occurred. As illustrated in 3 images on the right hand of Fig 4, the standard approach, which merely takes points in intersected areas into consideration (illustrated by red points), fails to simulate a smooth and consistent response process. After the collision response the generated contact surfaces are also non-plausible (picture 3 on the right). However, employing our approach to the same scenario, as illustrated in the 3 left-hand images, results in a more consistent response process. Because we use both colliding points (red points) and some non-colliding points(blue points) to calculate the deformed shape, the generated contact surfaces are more smooth and precise.

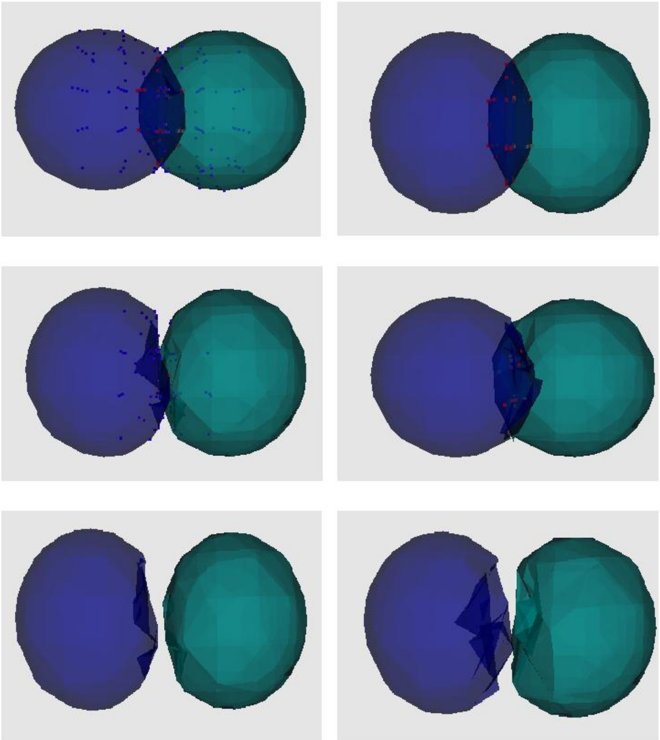


Figure 4: Two colliding deformable tumors. The standard approach is shown on the right hand and our approach is on the left hand.

The second scenario tests the continuity of the response force. In our haptic virtual surgery system, the continuity of response force plays an important role in discrete-time simulation because a discontinuous response may cause the haptic simulation unrealistic. Two standard approaches, computing the response force respectively based on linear relationship and exponential relationship, has been compared with our method in aspects of continuity and consistent. As illustrated in the Fig 5, our method remarkably reduces the jumps of forces between consecutive time slices.

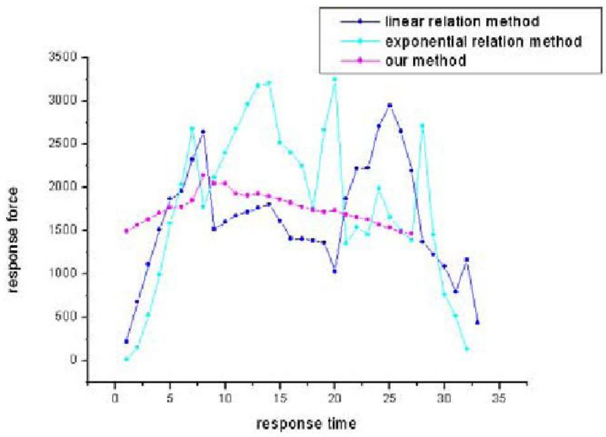


Figure 5: Comparison of continuity of response force computed by 3 different methods.

The third experiment tests the performance of our method in a more complex environment. In this scenario, several deformable objects are simulated and we control a virtual object (the yellow sphere) to collide with two tumors on a kidney. The tumor is deforming when collision has been detected and at the same time the feedback force has been computed and we can feel it through the tactile sensor.

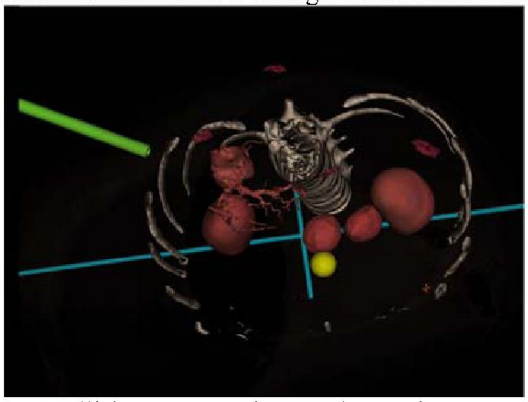


Figure 6: Collision response in complex environment.

ACKNOWLEDGMENT

Firstly, I'd like to thank Image-Guided Surgery and Therapy Laboratory in SJTU for the members in it provide us a lot of experiment data. I would also thank Xiaobo Li, Shaoting Zhang and Mathias Eitz for their good advice on our research.

REFERENCES

[1] M. Teschner, B. Heidelberger, M. Mueller, D. Pomeranets, and M. Gross. Optimized spatial hashing for collision detection of deformable objects. Proceedings of Vision, Modeling, Visualization VMV'03, pages 47–54, 2003.

[2] B. Heidelberger M. Teschner R. Keiser M. Muller M. Gross Consistent Penetration Depth Estimation for Deformable Collision Response 2004, VMV

[3] E. Mathias L.X.Gu, Hierarchical Spatial Hashing for Real-time Collision Detection, Shape Modeling and Applications, 2007. SMI '07. IEEE International Conference, 61-70, June 2007

[4] M. Teschner, B. Heidelberger, M. Muller and M. Gross A Versatile and Robust Model for Geometrically Complex Deformable Solids. CGI, 2004

[5] M. Gascuel An Implicit Formulation for Precise Contact Modeling between Flexible Solids, IMAG, BP53X 38041

[6] R. Bridson, S.Marino and R. Fedkiw. Simulation of Clothing with Folds and Wrinkles. Eurographics Association 2003

[7] L. Zhang, Young J.Kim, G. Varadhan and D.Manocha, Generalized Penetration Depth Computation. SPM, 2006

[8]Montgomery K., Bruyns C., Brown J., Sorkin S., Mazzela F., Thonier G.,Tellier A., Lerman B., Menon A. A general framework for collaborative, realtime surgical simulation. Medicine Meets Virtual Reality. 23-26, 2002

[9] Tuchschmid S., Grassi M., Bachofen D., Fruh P., Thaler M., Szekely G., M.Harders A flexible framework for highly-modular surgical simulation systems. ISBMS 2006, LNCS 4072.84-92, 2006

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