An Adaptive Collision Detection and Resolution for Deformable Objects Using Spherical Implicit Surface

Sunhwa Jung[1], Min Hong[2], and Min-Hyung Choi[1]

[1] Department of Computer Science and Engineering,
University of Colorado at Denver and Health Sciences Center,
Campus Box 109, PO Box 173364, Denver, CO 80217, USA
`sjung@ouray.cudenver.edu, Min-Hyung.Choi@cudenver.edu`
[2] Bioinformatics, University of Colorado at Denver and Health Sciences Center,
4200 E. 9th Avenue Campus Box C-245, Denver, CO 80262, USA
`Min.Hong@UCHSC.edu`

Abstract. A fast collision detection and resolution scheme is one of the key components for interactive simulation of deformable objects. It is particularly challenging to reduce the computational cost in collision detection and to achieve the robust treatment at the same time. Since the shape and topology of a deformable object changes continuously unlike the rigid body, an efficient and effective collision detection and resolution is a major challenge. We present a fast and robust collision detection and resolution scheme for deformable objects using a new enhanced spherical implicit surface hierarchy. The penetration depth and separating distance criteria can be adjusted depending on the application specific error tolerance. Our comparative experiments show that the proposed method performs substantially faster than existing algorithms for deformable object simulation with massive element-level collisions at each iteration step. Our adaptive hierarchical approach enables us to achieve a real-time simulation rate, well suited for interactive applications.

1 Introduction

Physically based simulation for deformable objects is indispensable for many modern character animation and medical simulation. Deformable object is usually discretized into a set of basic meshed elements to model the geometry and behavior of the object. It often consists of thousands of nodes to avoid undesirable sharp folds and to better represent their detailed dynamic behaviors. The collision detection and contact resolution often cost more than 90% of the total simulation time per iteration and it is considered as a major bottleneck for any deformable object simulation. Although many researchers achieved robust collision detection schemes recently [4,5,7], they are often not applicable to the interactive applications due to the complexity of their algorithms. Recently stochastic collision detection methods [17] have been proposed to achieve the interactive rate of simulation. These approaches guarantee the desirable computation time for the collision detection, but they miss significant amounts of collisions, thus the robust behavior can not be expected. In addition, more critical issue is that the collision resolution scheme can not be separated from the collision

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3514, pp. 735 – 742, 2005.

detection methods. If the collision is not resolved properly in current step, the collision detection query in the next step will start with unacceptable initial conditions and consequently the collision resolution will fail. Therefore the collision detection method must guarantee the comprehensive collision check and it has to provide enough information about the colliding and penetrating objects so that the collision resolution scheme can handle the collisions properly. So the collision detection and resolution is inseparable and the accurate collision response can only be achieved with a well coordinated detection and resolution scheme. In addition there must be a way to adjust the error tolerance and a strategy to resolve a degenerate penetration situation that guarantees the penetration free status after the collision resolution, since unavoidable numerical drift and other geometric errors could result in element level intersections.

Our proposed method is based on a spherical implicit surface that provides fast and robust collision detection and the resolution approximation between deformable objects. It extends the hierarchical spherical surface to the sub-divided triangle level adaptively to detect and resolve collisions within a tight error bound. It is designed to handle massive collisions at a given iteration step so its applicability is wide across variety of deformable structures as long as their surface is meshed with triangles. Due to the controllability of collision tolerance, the view dependent adaptive application of collision resolution can be applied to further reduce the computational cost.

2 Related Works

To reduce collision detection query space, several bounding volume hierarchies and spatial subdivision schemes have been proposed [10,18,19,20]. Most of them require pre-computation to generate efficient tree structures, and they are designed to perform a series of quick rejection tests to reduce the collision query space. But they can not provide enough collision information to resolve the collisions, i.e. proximity information, separating and penetrating depth, and involved features, for an accurate collision resolution. Therefore after finding the primitives that violate the bounding volume conditions, a geometry based exact collision detection method should be applied. For an exact collision detection between surface geometric primitives, the continuous collision detection using a coplanar condition is a de-facto standard. Originally the continuous collision detection for a moving triangle and a point was proposed by Moore [1]. It required solving a fifth polynomial, but Provot [8] reformulated the problem with a cubic equation and they extended it to include edge-edge cases. Bridson et al. [7] further addressed the numerical error handling method to achieve robust collision detection. But these methods are still expensive and have no ability to reduce the computation time due to the nature of the triangle-triangle geometric computation. Moreover the numerical error is inevitable in degenerated cases such as severely stacked and pinched triangles, often found in complex cloth patches [4].

Hubbard [14] proposed a method to approximate polyhedra using a hierarchical bounding sphere for collision detection between rigid polyhedra. This method achieved the accuracy through searching down the collision spheres in the hierarchical structure and it provided the ability to control the computational cost by adjusting the level of detail of the bounding spheres. But building a hierarchy that fits the model with minimal overestimation is computationally expensive. Bradshaw et al. [16] proposed methods that generate better fit sphere trees for rigid polyhedra. However deformable objects are difficult to generate proper sphere trees because fixed size spheres can not cover the changing geometry of the deformed structure effectively. Recently James et al. [21] proposed a method to build an efficient bounding tree for reduced deformable objects but it can not be used for general deformable objects with large deformation since it only works for simple and limited deformation based on modal analysis.

Collision resolution is also a crucial component to achieve accurate post-collision behavior. Baraff et al [3] applied the penalty force to resolve cloth and cloth collision. But it could be difficult to find the correct magnitude of forces for the collision resolution, and when the contact spring force is too strong or weak it will generate oscillations between collided regions. The other popular method is to resolve geometrically after the penetration happens. Volino [5] and Provot [8] used this method and most deformable objects with volume can utilize this method. Bridson et al [7] combined these two methods to take advantages of them. These approaches require accurate collision information from the previous iteration steps. Baraff et al [4] proposed a method that does not require historical information but it has a limitation when the colliding objects do not have closed surfaces.

3 Collision Detection

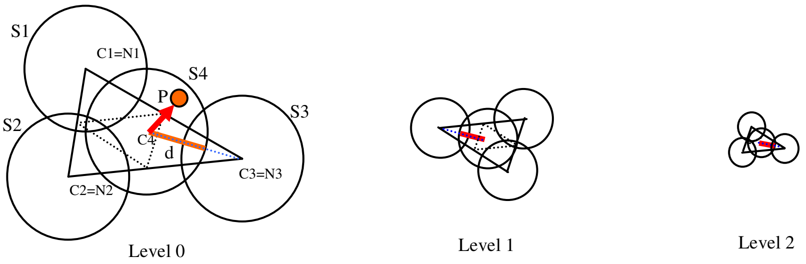
Conventional continuous collision detection [5,6,7,8] in triangle/node and edge/edge cases requires solving a cubic equation of the coplanar condition and solving one 2 by 2 linear system. Although the individual computation is simple, massive collision often creates a big numerical system and it becomes a bottle neck of the simulation. In addition, the collision result has numerical error because of many numerical operations (320 multiplications and 156 subtractions or additions) to generate the cubic equations. When a triangle is relatively small, the relative error can be a serious problem. Our method can substitute this method with quick distance calculation in an adaptive fashion.

In our method we used Axis Aligned Bounding Box (AABB) hierarchy for reducing the collision detection query space. AABB works efficiently for deformable objects because of the quick updating time. After searching the collision through AABB, the collision detection algorithm will find the two triangles which are close enough to require an exact collision check. In this chapter, our method to decide the collision state of the two triangles is described.

3.1 Approximation Using Spherical Implicit Surface

Triangle and node collision test. First we calculate the proper radius of the sphere (d) to generate the spherical implicit surface of the triangle for collision detection. d is the 2/3 of the distance between the triangle's center of mass (C4) to one of its nodes. The distance fields (spheres S1-S3) are created from each triangle's node using radius d and S4 is created from the triangle's center of mass. In figure 1 (level 0) four spheres

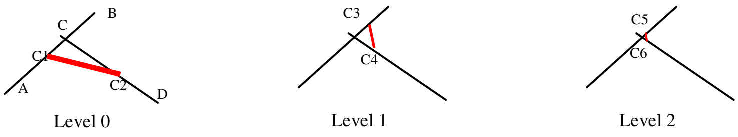
(S1-S4) represent the spherical implicit surface of the triangle. The spherical implicit surface of the triangle is the sum of the distance field from the three nodes of the triangle and the triangle's center of the mass. Node P is the one of three nodes in the other triangle in the exact collision detection. After performing the distance calculations from the center of each sphere to the node P, we can conclude whether node P is inside of the spherical implicit surface or not. Then we can check the distance between the closest location of the triangle and node P. If node P is in the one of spheres S1, S2, S3, and S4, the approximated distance between the triangle and the node is the distance from node P to one of the closest sphere centers. The collision normal for collision resolution is the vector from the selected center of the sphere to the node P. But if node P is in the two spheres, the approximated distance between triangle and node P is the distance between the mid point of the center of the two involved spheres and node P. If node P is in the three spheres, the approximation can be done from the three involved spheres' center. This collision result can be used to decide the collision resolution direction. The key idea of our method is using the result of the quick distance check to determine the collision resolution information (direction and distance). If more accuracy for the exact collision location and distance is required and the error tolerance (floating distance between triangles) is set below the size of the radius of the spherical implicit surface, we can perform the same algorithm recursively using the subdivided triangles.



Level 0
Level 1
Level 2

Fig. 1. Level 0 collision detection in triangle C1, C2, and C3 and with the node P. N1, N2, and N3 are the node positions of the triangle. Level 1 and Level 2 illustrates the detailed levels

Edge and edge collision test. After checking triangle and point collision, the possibly missed collision might be the edge and edge case which the points of the edge are outside of the implicit surface. To check these collisions, the distance between two mid points of the edges is checked. If they are overlapped, the distance between two mid points of the edge is tested whether it is within the threshold. Figure 2 illustrates that the collision detection can be performed in different levels. To select the collision check points (C3 and C4) at the next level, we use a binary search method. For example, in figure 2 (level 0) we select the minimum distance between four nodes (AC, AD, BC, and BD). In this case distance between BC is the minimum distance, thus the next level sets are edge C1 and B and edge C2 and C. We can pre-store the distance data in the previous triangle and node collision test and can reduce the computation time.



Level 0 Level 1 Level 2

Fig. 2. Level 0 collision detection between two edges. C1 and C2 are the mid points of the edges. If the distance between C1 and C2 is less than the threshold, the collision is determined

3.2 Adaptive Re-sampling

Typically we found that the level 0 collision result would be sufficiently accurate enough for the robust collision handling. If we use the relatively finer mesh, we do not need any further collision detection query. But when the triangle is too coarse, we need to adaptively search down to the next level. The collision detection algorithm on level 0 can be applied recursively until we get a satisfying result. No extra storage is required for the substructure to achieve this. Since we have already four distance calculations at the upper level, we need three more distance calculations for the next lower level collision detection. The beauty of our method is that we can have enough collision resolution information whenever we stop collision detection processing. The decision of further searching can be done using human collision perception, which O'Sullivan [15] researched with psychophysical experiment. We changed the threshold according to the distance between the camera and the object to reduce the computational burden. If the user wants to achieve more accuracy, it can be done by simply expanding the tree structure but the cost is not substantial since the involved triangles will be subdivided and each operation only includes a quick distance comparison.

4 Collision Resolution

(A) (B)

Fig. 3. The red arrow in (A) is the collision normal in the point and triangle collision detection result from figure 1. The red arrow in (B) is the collision normal in the edge and edge collision detection result from figure 2 (A)

Figure 3 illustrates the collision results in level 0 when we consider that the distance is less than the threshold. We can use the distance vector as the collision normal to apply the collision resolution method. From the figure 1, the node P was detected as collided with sphere 4. Figure 3 shows the normal of the collisions in triangle-point case and edge and edge case. Using these collision reports, geometry method and penalty force method are used for the collision resolution. Our method utilizes both methods to handle the collisions depending on the collision situation. Penalty force method reduces the number of collisions and geometry method guarantees the collision-free states. For the penalty force method, we sum up the mass of the object (edge, node or triangle) and calculate the repulsion force for two objects and the forces are distributed to each node according to their masses. For the geometry method, we modified the velocity to the collision response velocity and moved the position to the legal location.

5 Experiments

In the comparative experiments we performed the collision detection at the level 0 with threshold 1, and the collision resolution scheme was applied differently to each object. Since cloth model is thin and generates numerous self-collisions, we need to treat collisions carefully utilizing penalty force method and geometry method. We applied the penalty force after calculating the internal force to reduce the number of collision and the geometry method applied after finishing the simulation and before displaying the scene to enforce the collision free. For the two cylinders simulation, the geometry method is enough to handle the collision because there is no chance to miss the collision detection completely and the number of collisions is small.

Cloth simulation. To compare the performance between existing continuous collision detection method using the coplanar conditions and our proposed method, we applied both methods to a patch of cloth simulation to check self-collision. The conventional method requires solving cubic equations for each coplanar condition. Figure 4 is the snapshot of the cloth piling up simulation. The cloth model contains 3000 (30 by 100) particles. This illustrates the robust collision resolution with a predefined cloth thickness. We used a 2.4 GHz Pentium 4 processor computer and it took about 0.5 second per iteration. The collision detection threshold which we provide became the thickness of the cloth model. Figure 5 shows the comparative performance in the piling cloth simulation. At the beginning of the simulation, less exact collision detections is required due to the low curvature of the surface and less self collision occasions. After the cloth model begins to fold, the computation time grows as the number of colliding elements increases and more cases for exact element level collision resolution are occurred. The conventional method takes approximately in the range of 0.3-1.8 seconds to resolve all collision conditions. However, our method takes in the range of 0.015-0.5 second. Throughout the simulation, our method shows a significant speedup, about 20 to 200 times faster. We can achieve more accuracy by applying the small threshold with adaptive re-sampling distance checking.

Cylinder and cylinder simulation. We applied our method to a collision simulation between two deformable cylinders in figure 4. These cylinders are coarsely meshed and they are modeled with soft material property using linear finite element method [21]. Although these cylinders are under large deformation, the result demonstrates that our proposed method handles robustly the collision detection and resolution in real-time. The computational cost of our collision detection and resolution is negligible for the total simulation time due to the small number of nodes and heavy computation time in FEM analysis. You can see the result animation from [23].

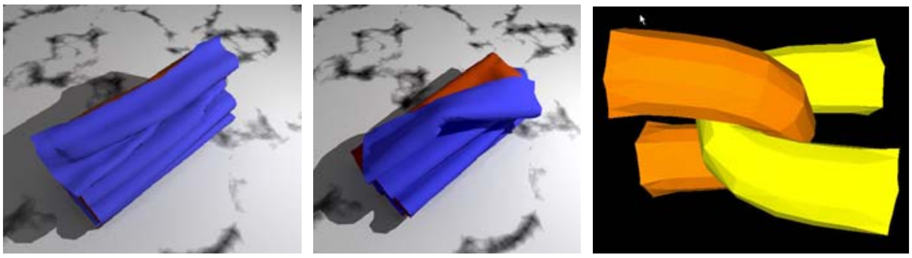


Fig. 4. Left two figures are the snapshots of the cloth simulation falling down and folded and crumpled. Due to the size of meshed triangles, our approximation in level 0 gives reasonably accurate collision detection result for the collision resolution. Right figure shows the deformation between two cylindrical volumetric models meshed with tetrahedra in contact

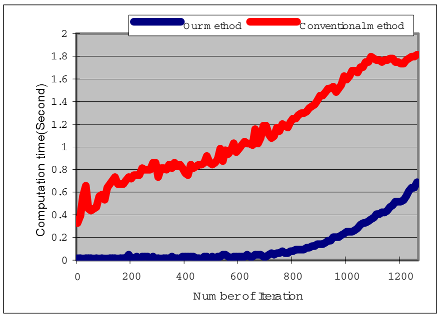


Fig. 5. The performance comparison between the conventional method and our method

6 Conclusion

We propose a fast and robust collision detection and contact resolution for deformable objects. We have utilized the human perception to determine the necessary accuracy of collision detection. Our experimental results demonstrate that the massive collisions between deformable objects can be effectively handled. Our method substantially reduces the complexity of the collision management and alleviates the numerical error in calculation. Due to the simplicity of our method, it is possible to achieve even better performance when it employs a GPU-based hardware acceleration. For coarsely meshed objects, the proposed method may have to go down to the numerous levels of details to achieve the necessary accuracy of a collision, and the depth of the tree is inversely proportional to the size of a triangle mesh. However, since deformable models should be meshed reasonably to generate plausible motion, the levels of detail are limited within a few steps.

References

1. Moore, M. and Wilhelms, J.: Collision detection and response for computer animation. In SIGGRAPH 1998, ACM Press / ACM SIGGRAPH, Computer Graphics Proc., 289-298.

2. Baraff, D. and Witkin, A.: Physically based modeling. SIGGRAPH 03' course notes, 2003
3. Baraff, D. and Witkin, A.: Large steps in cloth simulation. In Proc. of SIGGRAPH 1998, ACM Press / ACM SIGGRAPH, Computer Graphics Proc., 1-12
4. Baraff, D., Kass, M. and Witkin, A.: Untangling cloth. In Proc. of SIGGRAPH 2003, ACM Press / ACM SIGGRAPH, Computer Graphics Proc., 862-870.
5. Volino, P. and Thalmann, N. Magnenant: Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity. In Proc. of Eurographics, vol. 13 of Computer Graphics Forum, Eurographics Association, C–155–166, 1994.
6. Volino, P., Courchesne, M. and Thalmann, N. Magnenant: Accurate collision response on polygonal meshes. In Proc. of Computer Graphics, 179–188, 2000.
7. Bridson, R., Fedkiw, R. and Anderson, J.: Robust treatment of collisions, contact and friction for cloth animation, In Proc. of SIGGRAPH 2002, ACM SIGGRAPH, 594-603.
8. Provot, X.: Collision and self-collision handling in cloth model dedicated to design garment. In Graphics Interface, 177–89, 1997.
9. Lafleur, B., Thalmann, N. Magnenat and Thalmann, D.: Cloth animation with self-collision detection. In Proc. of the Conf. on Modeling in Comp. Graphics, Springer, 179–187, 1991.
10. Zhang, D. and Yuen, M.: Collision detection for clothed human animation. In Proceedings of the 8th Pacific Graphics Conference on Computer Graphics and Application (PACIFIC GRAPHICS-00), 328–337, 2000.
11. Lin, M. and Gottschalk, S.: Collision detection between geometric models: A survey. In Proc. of IMA Conf. on Mathematics of Surfaces, 1998.
12. Choi, K. and Ko, H.: Stable but responsive cloth. In Proc. of SIGGRAPH 2002, ACM SIGGRAPH, Comput. Graphics Proc., 604–611.
13. Choi, M., Hong, M. and Samuel, W.: Implicit constraint enforcement for stable and effective control of cloth behavior. A technical report form Univ. of Colorado at Denver, Computer Science and Engineering Department TR-03-01, 2004.
14. Hubbard, P. M.: Approximating polyhedra with spheres for time-critical collision detection. ACM Trans. Graph., 15(3):179-210, 1996.
15. O'Sullivan, C.: A model of collision perception for real-time animation. In Computer Animation and Simulation'99, pages 67–76.
16. Bradshaw, G. and O'Sullivan C.: Sphere-tree construction using dynamic medial axis approximation. In ACM SIGGRAPH Symposium on Computer Animation, pages 33-40. ACM Press, July 2002
17. Kimmerle, S., Nesme, M. and Faure, F.: Hierarchy Accelerated Stochastic Collision Detection. Vision, Modeling, and Visualization 2004
18. Gottschalk, S., Lin, M. C. and Manocha, D.: OBBTree: A hierarchical structure for rapid interference detection. In Proc. of SIGGRAPH 1996. ACM SIGGRAPH, 1996.
19. Klosowski, J. T., Held, M., Mitchell, J. S. B., Sowizral, H., and Zikan, K.: Efficient collision detection using bounding volume hierarchies of k-DOPs. IEEE Transactions on Visualization and Computer Graphics, v.4 n.1, p.21-36, 1998
20. Palmer, I. J., and Grimsdale, R. L.: Collision detection for animation using sphere-trees." Computer Graphics Forum, 14(2):105-116, 1995.
21. James, D. L. and Pai, D. K.: BD-Tree: Output-Sensitive Collision Detection for Reduced Deformable Models. In Proc. of SIGGRAPH 2004, ACM SIGGRAPH, 2004
22. Hong, M., Choi, M. and Lee, C.: Constraint-Based Contact Analysis between Deformable Objects. In Proceedings of International Conference on Computational Science, 2004.
23. http://graphics.cudenver.edu/~lab/iccs05.htm