Towards Time-Critical Collision Detection for Deformable Objects Based on Reduced Models

C. Mendoza
Grupo Modelado y Realidad Virtual
Universidad Rey Juan Carlos, Spain
cesar.mendoza@urjc.es

C. O’Sullivan
Computer Science Department
Trinity College Dublin
Carol.OSullivan@cs.tcd.ie

1 Introduction and related work

Topics of physically based modeling of deformable objects and collision detection have been extensively researched. Please refer to [1, 2, 3] for recent surveys. Nowadays, the combination of GPU techniques and multiresolution physical models [4] allows interactive simulations of complex deformable objects with a large number of polygons. To achieve this, the geometry of the object is separated from the deformable model in order to represent the latter at different levels of resolutions (*reduced models*). Recently, James and Pai [5] extended this idea to collision detection. They used the coordinates of the reduced model to update a sphere-tree hierarchy during object deformations. Moreover, many interactive simulations require a constant frame rate which can be obtained by using a time-critical approach. This approach allocates a given time-slice to the collision detection process, so it can be interrupted if the assigned time is expired. To the best of our knowledge, time-critical approaches have been only applied to rigid body collision detection [6, 7].

In this work, we propose an algorithm to carry out collision detection for deformable objects using an interruptible mechanism.

2 Deformable model

We use a bi-resolution model. A dense mesh is used for the graphical rendering of the object and a coarse tetrahedral mesh (reduced model) to simulate deformations using an explicit finite element method [8]. These meshes are connected in an off-line process by means of *rigid links*. Each rigid link joins a vertex, v_i, in the dense mesh to a point on the surface facet of the coarse mesh. Thus, at each deformation step of the coarse mesh, the rigid links are updated so that they keep their original orientation with respect to the facets of the coarse mesh. This will automatically move v_i and hence it will deform the dense mesh.

3 Interruptible mechanism

Our interruptible mechanism is based on a sphere-tree hierarchy constructed using an adaptive medial axis approximation [9]. Each sphere in the hierarchy is associated to a set of rigid links (those inside the sphere). The idea is to execute hierarchy traversals using a *breadth-first search*, instead of using a *depth-first search*, as used in traditional hierarchy traversals. In a depth-first search the tree is examined in vertical directions, going down to the leaves for each search. If the process is suddenly interrupted, many branches, from the root to the leaf, may remain untested, thus leading to missed collisions. Therefore, depth-first search is certainly not an option for time-critical processing, unless the search is capped at a given depth. On the other hand, a breadth-first search traverses the hierarchy horizontally, progressing successively through tighter approximations of the object. Hence, if the allocated time is expired, at least the algorithm would have tested collisions between approximations of the object providing enough information to compute collision responses. In order to do this, we create an entity, called *pair*, where we store spheres to be tested for overlapping. Additionally, two FIFO (First Input First Out) lists are used: PAIRLIST recursively stores the pairs of spheres to be tested for collision, according to the breadth-first search. If the colliding pairs of spheres are leaves then they are stored in COLLISIONLIST. The collision responses are *approximations* of the real ones since they are computed using the spheres representing the object (and contained in the obtained lists) instead of the primitives of its surface. We use the PAIRLIST and the COLLISIONLIST to compute the forces and consider untested pairs of spheres, in PAIRLIST, to be colliding. Each sphere, in the colliding pair, stores a repulsion vector which is computed by taking the difference vector between the center of the spheres. This force is applied to the associated rigid links of the sphere and hence applied to the coarse tetrahedral mesh to produce deformations. The spheres use a set of coarse mesh vertices, obtained through the rigid links associated to the spheres, as the reduced coordinates. These coordinates are used to update the sphere-tree hierarchy as in [5].

4 Results and Conclusions

In our tests, the algorithm provided approximate responses after interruptions while keeping a constant frame rate for the simulation. Acceptable force computations were obtained when the COLLISIONLIST was not empty otherwise the computed force was obtained using rough sphere-tree approximations stored in PAIRLIST thus leading to less satisfactory force responses.

There is a trade-off between accuracy and speed. Additional work has to be done in the hierarchy update, in particular when a sphere does not have explicitly associated a rigid link.

References

[1] A. Nealen, M. Muller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. In Y. Chrysanthou and M. Magnor, editors, *Proc. State of the Art Reports, Eurographics '05*, pages 71–94, 2005.

[2] P. Jimenez, F. Thomas, and C. Torras. 3d collision detection: A survey. *Computer and Graphics*, 21(3):269–285, 2001.

[3] M. Teschner, S. Kimmerle, B. Heidelberge, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino. Collision detection for deformable objects. In C. Schilick and W. Purgathofer, editors, *Proc. State of the Art Reports, Eurographics '04*, pages 119–140, 2004.

[4] M.-P Cani G. Debunne, M. Desbrun and A. Barr. Dynamic real-time deformations using space and time adaptive sampling. In *Proc. ACM SIGGRAPH '01*, pages 31–36, 2001.

[5] D. James and D. Pai. BD-Tree: Output-sensitive collision detection for reduced deformable models. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3), August 2004.

[6] J. Dingliana and C. O'Sullivan. Graceful degradation of collision handling in physically based animation. *Computer Graphics Forum*, 19(3):239–248, August 2000.

[7] J. Klein and G. Zachmann. Adb-trees: Controlling the error of time-critical collision, detection. In *8th International Fall Workshop Vision, Modeling, and Visualization (VMV)*, pages 19–21, University München, Germany, November 2003.

[8] J. O'Brien and J. Hodgins. Graphical modeling and animation of brittle fracture. In *Proceedings of ACM SIGGRAPH 1999*, pages 137–146. ACM Press/Addison-Wesley Publishing Co., August 1999.

[9] G. Bradshaw and C. O'Sullivan. Adaptive medial-axis aproximation for sphere-tree construction. *ACM Transactions on Graphics*, 23(1):1–26, 2004.