

行政院國家科學委員會專題研究計畫 成果報告

在多核心平台上配合圖形處理單元達成互動可變形物體連續碰撞檢測的研究 研究成果報告(精簡版)

計畫類別：個別型
計畫編號：NSC 97-2218-E-009-040-
執行期間：97年10月01日至98年09月30日
執行單位：國立交通大學資訊工程學系(所)

計畫主持人：黃世強

計畫參與人員：碩士班研究生-兼任助理人員：陳蔚恩
碩士班研究生-兼任助理人員：鄭游駿
碩士班研究生-兼任助理人員：劉政旻
碩士班研究生-兼任助理人員：葉喬之

處理方式：本計畫可公開查詢

中華民國 98 年 12 月 31 日

REPORT: 在多核心平台上配合圖形處理單元達成互動可變形物體連續碰撞檢測的研究

計畫編號	NSC 97-2218-E-009-040
------	-----------------------

In the project, we intended to perform parallel collision detection on multicore platforms with GPUs for two years. However, the project was only done in one year. We managed to improve the elementary tests on a multicore platform but we didn't have enough time to develop techniques using GPUs. Our students learnt programming and computer graphics.

We published one paper which won the best paper award (最佳論文獎).

Sai-Keung Wong, *Adaptive Continuous Collision Detection for Cloth Models using a Skipping Frame Session*, National Computer Symposium (全國計算機會議 NCS), Workshop on Image Processing, Computer Graphics, and Multimedia Technologies, National Taipei University, 27-28 November, 2009.

We propose a novel adaptive pipeline for continuous collision detection (APCCD) in simulating cloth models. The proposed pipeline consists of four components: bounding volume hierarchy (BVH) update, BVH traversal, a skipping frame session, and elementary test processing. It supports both inter- and self-collision detection. A skipping frame session is activated adaptively for skipping both BVH update and BVH traversal. Our method tracks all interacting primitive pairs. Experimental results show that the proposed method significantly improves the performance of collision detection in simulating cloth models.

Summary of results: We propose a novel adaptive pipeline for continuous collision detection (APCCD) in the simulation of cloth models. The results are listed as follows:

1) The framework of APCCD: It consists of four components: BVH update, BVH traversal, a skipping frame session and elementary test processing. By employing the skipping frame session both BVH update and BVH traversal stages can be skipped. In order to employ the skipping frame session, the bounding volumes of BVH nodes are inflated based on both the local and global information of the cloth models.

2) The partial traversal scheme: As the inflated bounding volume of each node is not

tight any longer and they may be kept for several frames, there will be many redundant potentially colliding pairs. Instead of performing a full BVH traversal to eliminate the redundant pairs, a partial traversal scheme is proposed to handle them. The partial traversal scheme is conservative. Thus, we will not miss any colliding primitive pairs.

3) Robustness: We adopt a history-based approach to keep track the relative orientation of all the primitive pairs in close proximity. These primitive pairs are maintained in a hash table throughout the simulation.

Our Algorithm

We assume that the topology of the cloth models does not change and the simulation time step is t . Two primitives collide if their shortest distance is smaller than or equal to a predefined threshold which is larger than or equal to the thickness of cloth. If the bounding volumes of two triangles overlap, the two triangles form a potentially colliding pair (PCP). Before the simulation is performed, we employ the primitive assignment scheme to assign each primitive to its incident triangle: a triangle assigned to itself and a vertex or edge assigned to one of its incident triangles. Each triangle record stores a six-bit assignment mask for the primitive assignment. The assignment mask of the triangle indicates the vertices or edges assigned to the triangle. Based on the assignment mask, the potentially colliding primitive pairs of two triangles can be computed.

Algorithm 1 shows the runtime phase of APCCD.

Algorithm 1 APCCD Algorithm

- 1: **if** `flagSkippingFrame` **then**
- 2: collect dangling vertices
- 3: collect dangling triangles
- 4: perform traversal for dangling triangles
- 5: **else**
- 6: perform BVH update
- 7: perform BVH traversal
- 8: **end if**
- 9: perform front-end PCP record filtering
- 10: perform back-end PCP record filtering

At each simulation time step, the speed of each vertex and the speed of each triangle are computed. The speed of a triangle is the maximum speed of its three vertices.

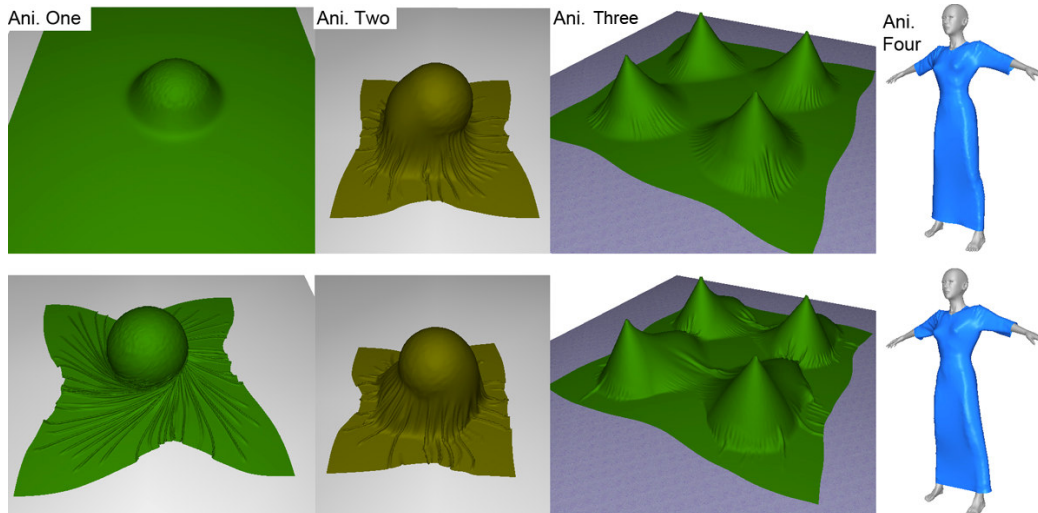
At the runtime phase, there is a skipping frame session. During the skipping frame session, both BVH update and BVH traversal are skipped. The number of frames that the skipping frame session lasts is nf and the value nf is determined adaptively. In the first frame of the skipping frame session, *flagSkippingFrame* is set as false. Otherwise, it is set as true.

If *flagSkippingFrame* is false, we perform a full BVH update and a full BVH traversal. In BVH update, the size of the bounding volume is extended adaptively according to the moving state of the objects. After that, BVH traversal is performed to gather potentially colliding triangle pairs (PCTPs) and the PCTPs are stored in the PCP pending list.

If *flagSkippingFrame* is true, we will collect the vertices and triangles which move farther from their estimated movement distance. We call the vertices *dangling vertices* and the triangles *dangling triangles*. A partial BVH traversal scheme is applied for processing the dangling triangles. We proceed to perform elementary test processing which consists of two sub-phases: front-end PCP filtering and back-end PCP filtering. In the frontend PCP filtering phase, non-colliding pairs will be further eliminated based on the distance heuristic. The PCTPs in the PCP pending list are inserted into the admissible PCP list if they pass the check of the distance heuristic. The PCTPs in the admissible PCP list will be passed to the back-end PCP filtering phase and their contact information is computed.

Our method guarantees to detect all PCTPs whose shortest distance is smaller than or equal the predefined threshold.

Experimental Results



We performed experiments on different sets of objects. The cloth models are interacting with some other rigid objects. Interactive performance can be achieved.

Drawbacks

As our method is a history-based method, the required memory is mainly used for storing the potentially colliding triangle pairs. The memory size is proportional to the number of potentially colliding triangle pairs. For example, in the spinning ball benchmark, the average number of potentially colliding triangle pairs was 142 k and the memory size was 48 M.

Conclusions

There are two limitations in our method. First, in order to employ the skipping frame session, the movement distance of the vertices of the cloth models should be small compared to the size of bounding volumes of other objects. However, our experiment results show that when a wind drag model with moderate strength, the skipping frame session can still be employed. If the external forces are too strong, the skipping frame session can be disabled. By employing the distance heuristic alone, our method also performs efficiently. Second, as our method is a history-based method, all colliding pairs and potentially colliding pairs in close proximity are hashed, the memory space is quite demanding. On the other hand, tracking pairs in close proximity is necessary in order to reliably compute the relative orientation of colliding pairs.

If we would have enough time, we would proceed to investigate methods on GPUs.

REFERENCES

- [1] D.-E. Baraff and A. Witkin, "Large Steps in Cloth Simulation," *Computer Graphics*

- (*SIGGRAPH'98*), pp. 43–54, 1998.
- [2] G. van den Bergen, “Efficient Collision Detection of Complex Deformable Models using AABB Trees,” *Journal of Graphics Tools*, vol. 2, no. 4, pp. 1–14, 1999.
 - [3] R. Bridson, R. Fedkiw, and J. Anderson, “Robust treatment of collisions, contact and friction for cloth animation,” *ACM ToG*, vol. 21, no. 3, pp. 594–603, 2002.
 - [4] K.-J. Choi and H.-S. Ko, “Stable but responsive cloth,” in *ACM SIGGRAPH*, 2004, pp. 604–611.
 - [5] M. Courchesne, P. Volino, and N. Magnenat-Thalmann, “Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects,” in *SIGGRAPH*, 1995, pp. 137–144.
 - [6] S. Curtis, R. Tamstorf, , and D. Manocha, “Fast collision detection for deformable models using representative-triangles,” in *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, 2008, pp. 61–69.
 - [7] S. Gottschalk, M.-C. Lin, and D. Manocha, “OBBTree: A Hierarchical Structure for Rapid Interference Detection,” in *ACM SIGGRAPH*, 1996, pp. 171–180.
 - [8] M. Hutter and A. Fuhrmann, “Optimized continuous collision detection for deformable triangle meshes,” in *WSCG*, 2007, pp. 25–32.
 - [9] J.-T. Klosowski, M. Held, S.-B.-J. Mitchell, H. Sowizral, and K. Zikan, “Efficient collision detection using bounding volume hierarchies of k-DOPs,” *IEEE Trans. on Vis. and Comp. Graphics*, vol. 4, no. 1, pp. 21–36, Jan. 1998.
 - [10] D.-E. Knuth, *Sorting and Searching: The Art of Computer Programming*, 2nd ed. Addison-Wesley, 1997, vol. 3.
 - [11] T. Larsson and T. Akenine-Moller, “Efficient collision detection for models deformed by morphing ,” *Visual computer*, vol. 19, no. 2, pp. 164–174, May 2003.
 - [12] T. Larsson and T. Akenine-Moller, “A dynamic bounding volume hierarchy for generalized collision detection,” *Computer and Graphics*, vol. 30, pp. 451–460, 2006.
 - [13] J.-D. Liu, M.-T. Ko, and R.-C. Chang, “Collision Avoidance in Cloth Animation,” *Visual Computer*, vol. 12, no. 5, pp. 234– 243, 1996.
 - [14] J. Mezger, S. Kimmerle, and O. Etmuss, “Hierarchical techniques in collision detection for cloth animation,” *Journal of WSCG*, vol. 11, no. 1, pp. 322–329, 2003.
 - [15] M. Moore and J.-P. Wilhelms, “Collision detection and response for computer animation,” *Computer Graph.*, vol. 22, no. 4, pp. 289–298, 1988.
 - [16] X. Provot, “Deformation constraints in a mass-spring model to describe rigid cloth behaviour,” in *Graph. Interface*, 1995, pp. 147–154.
 - [17] X. Provot, “Collision and self-collision handling in cloth model dedicated to design garments,” in *Computer Animation and Simulation*, 1997, pp. 177–189.
 - [18] A. Selle, J. Su, G. Irving, and R. Fedkiw, “Robust High- Resolution Cloth Using Parallelism History-Based Collisions and Accurate Friction,” *IEEE TVCG*, 2008.
 - [19] A. Smith, Y. Kitamura, H. Takemura, and F. Kishino, “A simple and efficient method for accurate collision detection among deformable polyhedral objects in arbitrary motion,” in *Virtual Reality*

Annual International Symposium, 1995, pp. 136–145.

[20] M. Tang, S. Curtis, S.-E. Yoon, and D. Manocha, “ICCD: Interactive Continuous Collision Detection between Deformable Models Using Connectivity-Based Culling,” *IEEE TVCG*, vol. 15, no. 4, pp. 544–557, 2009.

[21] M. Tang, , S.-E. Yoon, and D. Manocha, “Adjacency-based culling for continuous collision detection,” *Visual Computer*, vol. 24, pp. 545–553, 2008.

[22] M. Teschner, S. Kimmerle, G. Zachmann, B. Heidelberger, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, and W. Strasser, “Collision detection for deformable objects,” in *Eurographics State-of-the-Art Report*, 2005.

[23] P. Volino and N. Magnenat-Thalmann, “Efficient self-collision detection on smoothly discretised surface animation using geometrical shape regularity,” *Computer Graph. Forum*, vol. 13, no. 3, pp. 155–166, 1994.

[24] S.-K. Wong and G. Baciú, “Dynamic interaction between deformable surfaces and nonsmooth objects,” *IEEE TVCG*, vol. 11, no. 3, pp. 329–340, 2005.

[25] S.-K. Wong and G. Baciú, “A Randomized Marking Scheme for Continuous Collision Detection in Simulation of Deformable Surfaces,” in *ACM International Conference on Virtual Reality Continuum and Its Applications*, 2006, pp. 181–188.

[26] S.-K. Wong and G. Baciú, “Robust Continuous Collision Detection for Interactive Deformable Surfaces,” *Computer Animation and Virtual Worlds*, vol. 18, no. 3, p. 179, 2007.