A Method for High-level Culling Optimization in Self-collision Test of Arm Cloth Simulation

LvYue[1, a], He Bing[1, b], Jing Mi[1, c]

[1]Beihang University, Beijing, China

[a]lvyue2014@buaa.edu.cn, [b]hebing@buaa.edu.cn, [c]jingmi2014@163.com

Keywords: Cloth simulation, self-collision, culling optimization

Abstract. This Paper presents an algorithm to optimize high-level culling for self-collision test of arm cloth grid. High-level culling is at the triangle level, to cut off non-colliding areas and triangular pairs and obtain all possibly colliding triangular pairs. Here, we processed basic high-level culling firstly by bounding box and continuous normal cone information. Then, we perform radiation angle test to cut off non-colliding triangular pairs through radiation angle test. At last, we cut off all neighbouring triangular pairs in the candidate triangular pairs based on Orphan set. Experimental results show that, the high-level culling optimization method here in the Paper can effectively cut off non-colliding triangular pairs and has greatly improved the efficiency for self-collision test.

Introduction

The most common method for high-level culling in arm cloth simulation is bounding box algorithm to obtain all possible colliding triangle pairs. To reduce the times of intersection test in traversing bounding box, many researchers have proposed different culling optimization method. Volino[1] employed topology and connectivity of grid and judged self-collision status of grid by contour test. Provot[2] proposed the method of normal vector cone based on grid curvature, which could effectively cut off non-colliding areas, but was only applicable for discrete collision test. Tang[3] extended normal vector cone method to continuous collision test and calculated continuous normal vector cone within the entire time step.

In the Paper, we improve the efficiency of self-collision test by the following strategies. Based on bounding algorithm and continuous normal vector cone, we conduct radiation angle test to reduce candidate triangular pairs for self-collision test. Besides we establish an orphan set so as to cut off the neighbouring triangular pairs in the candidate triangular pairs.

Basic High-level culling

Bounding box is the most common method in the basic high-level culling. But it needs a lot of computational time. In this paper, we combine the continuous normal vector cone method with bounding box in the continuous self-collision test. The cone is a set that includes all normal vectors of a given surface. And the *semi-cone angle* α is sufficient to build a test that discriminates zones that cannot self-intersect and zones that may: if $\alpha < \pi / 2$ the zone cannot self-intersect. While traversing the bounding box, we firstly use continuous normal vector cone to cut off non-colliding areas ,then do intersection test of bounding box in the rest areas, and finally obtained candidate triangular pairs. To utilize continuous normal vector cone information in ergodic process, each node of bounding box should contain the two extra attributes, *semi-cone angle* α and *cone axis* vector.

The steps of basic high-level culling are as follows:

Step I, at the preprocessing stage, establish bounding box for cloth grid;

Step II, within each time step at the stage of collision test, update the bounding box from bottom up, including bounding box information and continuous normal vector cone information of nodes;

Step III, culling of continuous normal vector cone. In the traversal of bounding box, we firstly judge the size of semi-cone angle for continuous nominal vector cone of current node. If $\alpha < \pi / 2$, we cut off the areas corresponding to the node. Otherwise, we process the left and right node iteratively by continuous normal vector cone and intersection test of bounding box described in Step IV.

Step IV, intersection test of bounding box. According the intersection test result of bounding box, we obtained all candidate triangular pairs whose bounding box intersects

Optimization Based on Radiation Angle

To further improve the efficiency of self-collision test, we apply radiation angle test on the basis of the above method. The radiation angle method is: given a closed manifold grid, if there is an observing point to which all triangles in the grid are visible (which means the lines between the observing point to every triangle do not intersect with other triangles), such grid will not produce self-collision. Based on this method, Wong[4] conducted a skeleton-driven radiation angle culling algorithm, which set up an observing point at each joint in the grid, calculate skeleton weight of the vertex by using thermal diffusion mechanism [5]to distribute the triangle to the observing point. This method, achieve grid clustering by calculating skeleton weight of vertex, is relatively complicated and affects the efficiency of collision test.

We have improved the method of Wong by setting up two observing points at centers of upper arm and forearm according to structural feature of arm clothing grid. The initial locations of observing points are determined by the boundary points at upper and lower end face of arm grid. The grid is divided into two clusters, each of which corresponds to an observing point, according to the relative position of middle point on the line between the triangle and two observing point. We defined two cross sections at the initial positions of the observing points, cut the arm grid and then obtain two sectional rings. In every time step, we calculate the position and speed of the observing point by vertex interpolation of corresponding sectional ring.

Radiation Angle Test. We firstly divide the grid into some clusters according to the relative position to middle point of two observing points. Each cluster corresponds to an observing point. Then we conduct radiation angle test to each cluster and judge self-collision status according to the direction of the triangle to the observing point, which is described as follows:

Given time step $[0, \Delta t]$, triangle $T(p_0(t), p_1(t), p_2(t))$, normal vector of triangle pointing at outside the grid $n_T(t)$, and observing point $q(t)$, feature function of triangle about observing point is defined as:

$$\emptyset(T,q,t) = n_T(t) \cdot (p_0(t) - q(t)) \tag{1}$$

According to the feature function, the direction of triangle to observing point includes the following three categories:

(1) Positive, if in the entire time step $[0, \Delta t]$, $\phi(\mathrm{T,q,t}) > 0$

(2) Negative, if in the entire time step $[0, \Delta t]$, $\phi(\mathrm{T,q,t}) < 0$

(3) Nondirective, other situation

If two triangles in the grid collide, one of the following conditions should be met:

(1) In the entire time step, one triangle is positive and the other is negative

(2) In the entire time step, at least one triangle is nondirective.

Each cluster of grid is tested in terms of radiation angle with the method as follows:

Given the observing point q, grid triangles are divided into three sets $H^+(q)$, $H^-(q)$ and $H^u(q)$ within each time step, which represent respectively the sets of positive triangle, negative triangle and nondirective triangle. Based on the above two self-collision conditions, we only need to test whether collision occurs between $H^+(q)$ and $H^-(q)$, or between $H^u(q)$ and all triangles within each time step:

(1) If the triangles in a cluster are all positive to the observing point, this cluster will not self-collide and can be cut off.

(2) If some triangles in a cluster are negative or nondirective to the observing point, we need to test collision of triangles in $(H^{+}(q), H^{-}(q))$, $(H^{+}(q)+H^{-}(q)+H^{u}(q), H^{u}(q))$.

The algorithm are as follows:

Preprocessing stage:

Step I, determine position of observing point;

According to the characteristics of arm clothing grid, we have set up two observing points q_1 and q_2, respectively located at centers of upper arm and forearm. In this paper, the initial positions of the two points are respectively $\frac{3}{4}mid1+\frac{1}{4}mid2$ and $\frac{1}{4}mid1+\frac{3}{4}mid2$ (mid1 and mid2 respectively present the center point of two boundary face).Then we can calculate two cross sections at the initial positions of the observing points to get two sectional rings formed by two groups of intersects. Position and speed of observing points are acquired by interpolating the vertices of each sectional ring. To simplify calculation, we conduct interpolation directly through triangular ring strips where every triangle intersecting with cross section. The positions of observing points are acquired by interpolating the three vertices of these triangles.

Step II, divide the grid into two clusters based on the relative positions to center point.

Step III, establish bounding box for each cluster of grid. Besides add two extra attributes *neg* and *unc* for each node, which indicate the quantities of negative and nondirective triangles of the node.

Self-collision test stage:

Step I, update observing points in every time step by interpolating the position and speed.

Step II, classify the triangles in terms of observing points, determine plus-minus of feature function based on plus-minus of upper and lower bounding functions of feature function; group triangles of each cluster into positive, negative and nondirective triangles.

Step III, update the bounding box of each cluster of grid. The quantities of negative and nondirective triangles are obtained by combing left and right nodes. As for leaf node, if the corresponding triangle is positive, then *neg*=0, *unc*=0; if it is negative, then *neg*=1, *unc*=0; if it is nondirective triangle, then *neg*=0, *unc*=1;

Step IV, cluster-internal test. Conduct radiation angle test to each cluster and determine self-collision status. We combine radiation angle test with bounding box algorithm and continuous nominal vector cone information. In the traversal of bounding box, we firstly judge the direction of a triangle to the observing point. If the triangles within the cluster are all positive (*neg*=0, *unc*=0), self-collision will not occur in the cluster; if there are negative triangles in the cluster (*neg*>0), turn to step (1); if there are nondirective triangles in the cluster (*unc*>0), turn to step (2);

(1) If there are negative triangles in the cluster, we need to test self-collision of the triangular pairs in $(H^{+}(q), H^{-}(q))$. We process plus-minus test to the nodes of bounding box from the root node down to ergodic cluster. The method for plus-minus test is as follows:

1. Firstly, judge the size of semi-cone angle of continuous nominal vector cone of current node; if $\alpha < \pi / 2$, the area current node corresponded to will not self-collide and can be cut off; otherwise, turn to step 2;

2. Judge the quantity of negative triangle in current node; if *neg*=0, the area the node corresponded to will not self-collide and can be cut off. If there are negative triangles, the positive and negative triangles between the left and right nodes are tested firstly, then the left and right nodes are iteratively tested for plus-minus until it reaches leaf node. The testing to positive and negative triangular pairs between the left and right nodes is processed thoroughly within each level:

① If both the two nodes are leaf nodes, one of them is positive and the other negative, and the bounding boxes of them intersect, then these two triangles form a PCTP;

② If at least one node is an internal node, the left and right sub-node of such node and also another node are used for testing positive and negative triangular pairs;

③ Process from the top thoroughly, carry out steps ① and ② for iteration, obtain all positive and negative triangular pairs that may possibly collide in the cluster;

(2) If there are nondirective triangles in the cluster, we need to test collision of triangular pairs in $(H^{+}(q)+H^{-}(q)+H^{u}(q),H^{u}(q))$. We process the bounding box from root node of the cluster thoroughly upside down and also process nondirective test to nodes of bounding boxes; the method of nondirective test is the same as that of plus-minus test in step (2). Through nondirective test, we acquire all the triangular pairs that may possibly collide and includes nondirective triangles in the cluster.

Step V, inter-cluster test. The method is the same as that of bounding box intersection test in bounding algorithm: conduct intersection test of bounding box from the root node of two clusters of bounding box thoroughly upside down, until we obtain all inter-cluster triangular pairs that may possibly collide.

On the basis of basic higher tailoring method, we processed higher tailoring optimization based on radiation angle, cut off non-colliding areas and triangular pairs through cluster-internal and inter-cluster tests and obtained all candidate triangular pairs that may possibly collide.

Optimization Based on Orphan Set

From techniques discussed above we obtain all candidate triangular pairs intersecting with bounding box. Since bounding boxes of neighbouring triangular pairs always intersect, the efficiency of high-level culling is low. In this section, we take advantage of concept of orphan set introduced by Tang[3], to cut off neighbouring triangular pairs in the candidate triangular pairs.

At preprocessing stage, we establish orphan set. Only the upper and lower end faces of arm clothing grid are not closed, so there are less boundary elements and smaller quantity of primitive pairs in orphan set. At collision test stage, we cut off all neighbouring triangular pairs and obtain all non-neighbouring intersecting triangular pairs of bounding boxes, to form candidate triangular pairs.

Preprocessing stage:

Step I, establish data structure and store information of vertex, edge and triangle respectively. Vertex structure *vec3f* stores vertex position, *incident* stores the set of triangular index accessory to vertex, and *edge2f* stores two vertex indexes of the edge and two accessory face index. Triangular structure *tri3f* stores three vertex indexes of the triangle, *tri3e* stores the indexes of three edges of the triangle. Through vertex list and triangle list of the grid, we obtained *vec3f* list, *tri3f* list, *edge2f* list, *tri3e* list and *incident* list.

Step II, determine boundary edge and boundary point. Through *edge2f* list, we obtain all boundary edges. If an edge has only one accessory face, such edge is a boundary edge and the two vertexes of the boundary edge are boundary points.

Step III, as for each boundary point, we establish neighbouring sets OIS and OAS based on *incident* list and *tri3f* list and find out triangles in OAS intersecting with all triangles in OIS, to form a VF primitive pair of orphan set, together with boundary point.

Step IV, as for each boundary edge, we establish a set ES including the neighbouring sets OIS and OAS and all edges in set OAS based on *edge2f* list, *tri3f* list and *incident* list. As for every edge in ES not neighbouring boundary edge, if every of its accessory triangle neighbours each triangle edge in boundary edge OIS, such edge and boundary edge form an EE primitive pair in the orphan set.

Collision test stage:

In the process of intersection test of bounding box, if the bounding boxes of two leaf nodes intersect, we judge whether two triangles neighbour each other, namely whether they share a common vertex. Only two triangles do not neighbour each other, can a candidate triangular pair be formed. Or, such triangular pair should be cut off.

After high-level culling optimization based on orphan set, the candidate triangular pairs do not include neighbouring triangular pairs. The primitive pairs distributed in the candidate triangular pairs and the primitive pairs in orphan set form candidate primitive pairs, which are used for subsequently basic intersection test.

Experimental Results

In our experiments, we use a PC with a Intel Xeon CPU and 4G of memory. There are 693 vertexes and 1,345 triangular facets used for arm clothing grid here. Take one gesture for example, the quantity and simulation frame rate of the candidate triangular pairs (PCTPs) tested in each step are shown in Tables 1 and 2.

Table 1 PCTPs Quantity of High-level culling Optimization

	100th frame	500th frame	1,000th frame
Basic high-level culling	4742	4854	5710
optimization I based on radiation angle	1093	1225	1740
optimization II based on orphan set	281	215	382

Table 2 Simulation Frame Rate (Fps) of High-level culling Optimization

	100th frame	500th frame	1,000th frame
Basic high-level culling	0.91	0.36	0.30
optimization I based on radiation angle	2.7	2.82	1.49
optimization II based on orphan set	12.79	15.72	6.85

It can be seen from Tables 1 and 2 that, when clothing deform greatly, if we only conduct basic high-level culling, the efficiency and simulation frame rate are both low. On the basis of basic high-level culling, 70% of candidate triangular pairs have been cut off in step I of optimization based on radiation angle, meanwhile simulation frame rate has been improved by 3 to 5 times. There are fewer candidate triangular pairs after step II of optimization based on orphan set, and simulation frame rate was improved by 6 times.

It can be known from the above experiment that, our high-level culling optimization method can effectively cut off non-colliding areas and triangular pairs and improve the efficiency of self-collision test and clothing simulation.

We also compared our high-level culling optimization method with other high-level culling methods, including bounding box method (BVH), orphan set method proposed by Tang in 2009 and skeleton-driven radiation angle culling method proposed by Wong in 2013. In case of larger deformation of clothing in gesture 2, we compared the experimental results of the 100th frame. The quantity and simulation frame rate of candidate triangular pairs are shown in Table 3.

Table 3 Comparison of Different Methods for High-level culling Optimization

	Qty. of candidate triangle	Simulation frame rate
Basic BVH	4742	0.91
2013 Wong	1093	2.7
2009 Tong	578	12.71
Our method	281	12.79

It can be seen from Table 3 that, comparing with basic bounding box method, the quantity of candidate triangular pairs tested by our high-level culling optimization method is much smaller and simulation frame rate is improved by 14 times. We improved the skeleton-driven radiation angle method of Wong and cut off neighbouring triangular pairs in candidate triangular pairs. The simulation efficiency by our method is 6 times higher than that of Wong and moreover, the tested candidate triangular pairs are much less. Comparing the culling method of Tang which only employs orphan set, since we processed self-collision culling to each cluster through radiation angle test, the candidate triangular pairs tested by our method is only a half of that tested by Tang's method, with simulation frame rate similar to that of Tang's.

Conclusions

We apply basic high-level culling firstly by bounding box and continuous normal vector cone, to cut off non-colliding areas and triangular pairs of bounding box. Then we perform radiation angle test to cut off non-self-colliding areas and triangular pairs and obtain all candidate triangular pairs that may possibly collide. At the last step we cut off all neighbouring triangular pairs in the candidate triangular pairs through orphan set. Experiments show that our high-level culling optimization method can improve the efficiency of self-collision test greatly.

Acknowledgement

The work is supported by National Natural Science Foundation of China (No.61272346).

References

[1] Volino P. Collision and Self-Collision Detection: Efficient and Robust Solutions for Highly Deformable Surfaces[A]．Computer Graphics, SIGGRAPH '95 Proc[C], 1995：137-144.

[2] Provot X. Collision and self-collision handling in cloth model dedicated to design garment[J]. Graphics Interface, 1997:177–89.

[3] Tang M., Curtis S., Yoon S., et al．ICCD: Interactive Continuous Collision Detection between Deformable Models Using Connectivity-Based Culling[J].IEEE Transactions on Visualization and Computer Graphics, 2009, 15(4):544-557

[4] Wong S.K, Lin W.C, Hung C.H, et al. Radial view based culling for continuous self- collision detection of skeletal models[J]. ACM Transactions on Graphics,Volume, 2013.

[5] Barron I., Poppvi C. Automatic rigging and animation of 3D characters[J]. ACM Transactions on Graphics, 2007:2087–2096.