

Improved Method in Lower Tailoring of Self-collision Detection of Arm Cloth Simulation

Yue Lv^a, Bing He^b and Mi Jing^c

^alv Yue2014@buaa.edu.cn, ^bhebing@buaa.edu.cn, ^cjingmi2014@163.com

Keywords: cloth simulation; lower tailoring; collision detection

Abstract. After the stage of higher tailoring, the candidate triangle pairs are still need to be further reduced to eliminate redundant tests in the stage of lower tailoring. This paper aims to propose a method to improve performance and efficiency in lower tailoring for self-collision detection of arm cloth simulation. Lower tailoring is a primitive level of cutting off redundant and non-colliding primitive pairs. Our method includes two stages. Firstly this paper eliminate redundant primitive pairs and remove non-intersecting primitive pairs by carrying out feature-based distribution and making intersection test of primitive bounding box. Secondly a non-coplanar filter is used to filter out non-coplanar primitive pairs within time step through non-coplanar test which further reduce candidate primitive pairs. Test results show that, our method can effectively remove redundant and non-colliding primitive pairs and has, to some extent, improved the efficiency for self-collision detection.

Introduction

By higher tailoring, we obtain all possibly colliding candidate triangle pairs (PCTPs), each of which needs to be distributed with 15 primitive pairs for basic intersection test, including 6 VF primitive pairs and 9 EE primitive pairs. Since the vertex and edge of the mesh are shared by multiple triangles, same primitive pair may be distributed for multiple times, which leads to redundancy of primitive pairs in basic intersection test. As for basic lower tailoring methods, two databases need to be maintained in the distribution process, to record separately the distributed VF and EE primitive pairs. Such method can effectively eliminate repeated basic intersection tests of primitive pairs. But when distributing primitive pairs for each candidate triangle, we need to query the databases. This requires a lot of time and affects the efficiency for self-collision detection. Curtis[1] proposed representative triangular mechanism to eliminate redundant primitive pairs. Hutter and Fuhrmann[2] proposed the method of primitive bounding box, to effectively cut off non-colliding primitive pairs. Moreover, Tang[3][4][5] et al. proposed the method of filter, to filter out non-coplanar primitive pairs.

Improved Method Based On Feature Distribution

Since the vertex/edge of the mesh is shared by multiple triangles, which results in redundancy of primitive pairs. To eliminate redundant primitive pairs, Curtis[1] proposed a thought of representative triangle, to distribute each vertex/edge of the mesh solely to a triangle including them. The distribution principle is as follows:

- (1) Each primitive feature should be distributed to a triangle.
- (2) Each primitive feature should be distributed to only one triangle.
- (3) Each primitive feature should be distributed to the triangle including it.

The distribution principle ensures that each primitive feature can be distributed solely to the triangle including it.

In the process of self-collision detection, when distributing primitive pairs to the candidate triangles obtained from higher tailoring, we need to judge firstly whether such triangle includes compatible feature pairs. If at least a triangle has been distributed with one or more vertexes, or two triangles have been distributed with one or more edges, such triangle pair contains compatible feature pair. If the candidate triangle pair does not contain compatible feature pairs, it is not

necessary to distribute primitive pairs of basic intersection test. As for each candidate triangle pair (A, B) containing compatible feature pairs, primitive pairs for basic intersection test are distributed as per the following method:

(1) Each vertex represented by triangle A forms, together with triangle B, a VF candidate primitive pair.

(2) Each vertex represented by triangle B forms, together with triangle B, a VF candidate primitive pair.

(3) Each edge represented by triangle A forms, respectively with each edge represented by triangle A, an EE candidate primitive pair.

This distribution mechanism can ensure that, the same primitive pair can only be distributed once. In case of basic intersection test, there will be no repeated primitive pairs. Representative triangular mechanism can effectively cut off redundant primitive pairs.

Primitive level bounding box method proposed by Hutter and Fuhrmann[2] can effectively cut off non-intersecting primitive pairs of primitive bounding box, with high cutting efficiency. But it needs to maintain multiple primitive level bounding boxes, which affects simulation efficiency. To make use of the advantage of high cutting efficiency of the primitive level bounding box, at the stage of lower tailoring, we have optimized lower tailoring based on feature distribution, by representative triangular mechanism and primitive bounding box. Feature distribution is firstly processed to cut off redundant primitive pairs by representative triangular mechanism. When distributing primitive pairs for the candidate triangles containing compatible feature pairs, we will calculate bounding box of primitives, make intersection tests of primitive bounding box and cut off non-intersecting primitive pairs of primitive bounding box.

The lower tailoring optimization method here is for, in the process of distributing primitive pairs, processing intersection test of primitive bounding box, to cut off non-coplanar primitive pairs of primitive bounding box and further improve the efficiency of lower tailoring. Comparing with the method of primitive level bounding box, the lower tailoring optimization method here does not need to maintain multiple primitive level bounding boxes, with high simulation efficiency. The lower tailoring optimization method of the first stage here is achieved as follows:

Preprocessing stage:

Data structure of each triangle is added with two attributes, to record respectively the quantities of feature vertex of feature edge distributed to such triangle. Feature distribution is done to the mesh. In the distribution process, greedy algorithm is employed. Then it is done to every triangle. If a triangle contains an undistributed primitive (vertex, edge), then it will be distributed to such triangle. In the process of feature distribution, vertex list and edge list of each triangle are rearranged, to make the distributed feature primitives in the front of vertex/edge list.

Collision test stage:

During each time step, as for each candidate triangle pair obtained from higher tailoring, the candidate primitive pairs of subsequent intersection test will be distributed, of which the process is as follows:

Step I, firstly, judge whether the candidate triangle contains compatible feature pairs. If at least a triangle is distributed with one or more vertexes, or two triangles are distributed with one or more edge, such triangle contains compatible feature pairs. We continue Step II. If two triangles do not contain compatible feature pairs, such triangle pair will be neglected.

Step II, when distributing candidate primitive pairs, intersection test will be processed on primitive bounding box.

(1) Each vertex represented by triangle A is processed intersection test of VF primitive bounding box with triangle B. If the primitive bounding boxes intersect, a VF candidate primitive pair is formed;

(2) Each vertex represented by triangle B is processed intersection test of VF primitive bounding box with triangle A. if the primitive bounding boxes intersect, a VF candidate primitive pair is formed;

(3) Each feature edge represented by triangle A is processed intersection test of EE primitive

bounding box with each feature edge represented by triangle B. If the primitive bounding boxes intersect, an EE candidate primitive pair is formed.

As for each time step of intersection test of EE primitive bounding box, we firstly form AABB bounding box for two edges and then process intersection test of AABB bounding box. As for the intersection test of VF primitive bounding box, since in each time step of continuous collision test, the vertex trajectory is a line section, the AABB bounding box of such line section is not compact, the tailoring efficiency for bounding box intersection test is not high, in case of intersection test of VF primitive bounding box, intersection test is processed directly by line section and triangular bounding box. We employed the method of disengaging shaft here for the intersection of line section and bounding box.

Improved Method Based on Non-coplanar Filter

As for continuous self-collision detection, we process basic intersection test to 6 VF primitive pairs and 9 EE primitive pairs needed to be distributed for each candidate triangle. Each basic intersection test is divided into two steps: coplanar test and internal test. No matter for VF primitive pair and EE primitive pair, coplanar test involves in four vertexes. Only four vertexes are coplanar, the primitive pair can possibly collide. Provot[6] proposed that, the coplanar test of four vertexes needed to solve a cubic equation. If we can judge in advance that four vertexes will not be coplanar within the time step, the primitive pair will definitely not collide and can be cut off, so we do not need to solve a cubic equation for coplanar test. As for VF primitive pair, if the vertexes are always on the same side of a triangle within the entire time step, such VF primitive pair will not collide. This is because within the entire time step, the vertex and the triangle will not be coplanar, so they will not intersect. Similarly, as for EE primitive pair, if in the entire time step, two edges do not intersect, such EE primitive pair will not be coplanar.

Non-coplanar test of VF primitive pairs:

As Figure 1 shows, assume at the start point, end point and any point within time step $[0,1]$, the positions of vertex P are P_0 , P_1 and P_t respectively; the positions of three vertexes of triangle T are (a_0, b_0, c_0) , (a_1, b_1, c_1) and (a_t, b_t, c_t) respectively; normal vectors are n_0 , n_1 and n_t respectively. As for VF primitive pair, to judge whether four vertexes are coplanar, we calculate their projection distance on normal vector direction. At any time within the time step, if the projection distance $(P_t - a_t) \cdot n_t$ is 0, the four vertexes of the primitive pair are coplanar. Otherwise, they are not coplanar.

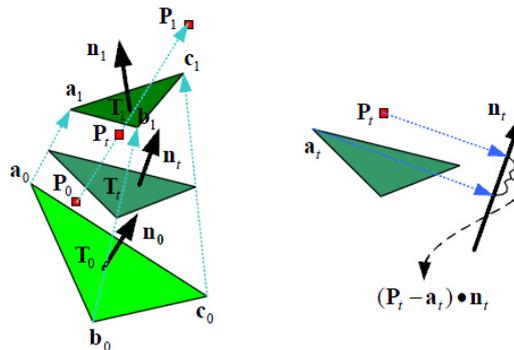


Fig.1. Projection Distance of VF Primitive Pair on Normal Vector Direction

We define marks A , B , C , D , E and F as follows:

$$A = (P_0 - a_0) \cdot n_0 \quad B = (P_1 - a_1) \cdot n_1 \quad (1)$$

$$C = (P_0 - a_0) \cdot \hat{n} \quad D = (P_1 - a_1) \cdot \hat{n} \quad (2)$$

$$E = (P_0 - a_0) \cdot n_1 \quad F = (P_1 - a_1) \cdot n_0 \quad (3)$$

The projection distance along n_t is then:

$$\begin{aligned}
(P_i - a_i) \cdot n_i &= A * (1-t)^3 + C * 2 * (1-t)^2 * t + E \\
&\quad * (1-t) * t^2 + B * t^3 + D * 2 * t^2 * (1-t) \\
&\quad + F * t * (1-t)^2 \\
&= A * B_0^3(t) + \frac{2 * C + F}{3} * B_1^3(t) \\
&\quad + \frac{2 * D + E}{3} * B_2^3(t) + B * B_3^3(t)
\end{aligned} \tag{4}$$

Where, $B_i^3(t)$ is the coefficient of Bernstein polynomial with degree of 3. Scalars A, B, $\frac{2 * C + F}{3}$ and $\frac{2 * D + E}{3}$ are the four control points of Bernstein polynomial.

Determining condition of non-coplanar VF primitive pairs: as long as the marks for the four scalars A, B, $2 * C + F$ and $2 * D + E$ are the same, namely they are all positive or all negative, the projection distance $(P_i - a_i) \cdot n_i$ will not be 0 and the vertex and the triangle will not be coplanar.

Non-coplanar test for EE primitive pairs:

As Figure 2 shows, at the start point, end point or any point within the time step [0,1], the positions of vertexes of two edges are respectively: (u_0, v_0, k_0, l_0) , (u_1, v_1, k_1, l_1) , (u_t, v_t, k_t, l_t) . When testing whether the four vertexes are coplanar, the calculating method for projection distance $(l_t - k_t) \cdot n_t$ is the same as that for VF primitive pairs, but l_t, k_t, u_t and v_t replace P_t, a_t, b_t and c_t respectively.

Marks A', B', C', D', E' and F' are defined as follows:

$$A' = (l_0 - k_0) \cdot n_0' \tag{5}$$

$$C' = (l_0 - k_0) \cdot \hat{n}' \quad D' = (l_1 - k_1) \cdot \hat{n}' \tag{6}$$

$$E' = (l_0 - k_0) \cdot n_1' \quad F' = (l_1 - k_1) \cdot n_0' \tag{7}$$

Where,

$$n_0' = (u_0 - k_0) \times (v_0 - k_0) \quad n_1' = (u_1 - k_1) \times (v_1 - k_1) \tag{8}$$

$$n_0' = (u_0 - k_0) \times (v_0 - k_0) \quad n_1' = (u_1 - k_1) \times (v_1 - k_1) \tag{9}$$

$$\hat{n}' = \frac{n_0' + n_1' - (\vec{v}_u - \vec{v}_k) \times (\vec{v}_v - \vec{v}_k)}{2} \tag{10}$$

$$\vec{v}_k = k_1 - k_0 \quad \vec{v}_u = u_1 - u_0 \quad \vec{v}_v = v_1 - v_0 \tag{11}$$

Determining condition of non-coplanar EE primitive pairs: as long as the marks of the four scalars $A', B', 2 * C' + F'$ and $2 * D' + E'$ are the same, the projection distance $(l_t - k_t) \cdot n_t$ will not be 0. Within the time step, two edges will not be coplanar.

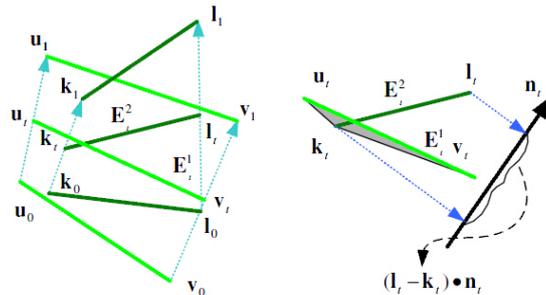


Fig.2. Projection Distance of EE Primitive Pair on Normal Vector Direction

At the last step of lower tailoring, we process lower tailoring optimization based on non-coplanar filter; cut off the non-coplanar primitive pairs within the entire time step through non-coplanar test;

further reduce the quantity of candidate primitive pairs and improve the efficiency of collision test.

At the stage of collision test, within each time step, we process non-coplanar test to VF and EE primitive pairs through calculating projection distance of primitive pairs.

As for VF primitive pairs, the steps to achieve non-coplanar test are as follows:

Step I, normalize time step as [0,1], the speed of each vertex multiplies by time step Δt ;

Step II, calculate the three control points of triangle normal vector within time step n_0 , n_1 and \hat{n} :

$$n_0 = (b_0 - a_0) \times (c_0 - a_0) \quad (12)$$

$$n_1 = (b_1 - a_1) \times (c_1 - a_1) \quad (13)$$

$$\hat{n} = \frac{n_0 + n_1 - (\vec{v}_b - \vec{v}_a) \times (\vec{v}_c - \vec{v}_a)}{2} \quad (14)$$

Step III, solve the values of the four control scalars of projection distance A , B , $2 * C + F$ and $2 * D + E$, judge whether they are positive or negative at the same time. If they are the same, VF primitive pairs will not be coplanar.

As for EE primitive pairs, the four vertexes on the two edges replace the four vertexes in the VF primitive pairs respectively for non-coplanar test, to solve the values of four control scalars of projection distance A' , B' , $2 * C' + F'$ and $2 * D' + E'$. We also judge whether they are positive or negative at the same time, if so, EE primitive pairs will not be coplanar.

Test results

There are 693 vertexes and 1,345 triangle patches in the arm clothing mesh here. Blue vertexes and yellow triangles are used to mark VF primitive pairs and green edges are used to mark EE primitive pairs.

At the stage of lower tailoring, we firstly process lower tailoring optimization based on feature distribution, cut off redundant primitive pairs that do not intersect with bounding boxes. To further improve tailoring efficiency, we have processed lower tailoring optimization based on non-coplanar filter, to cut off non-coplanar primitive pairs. The final test results are shown in Figure 3.

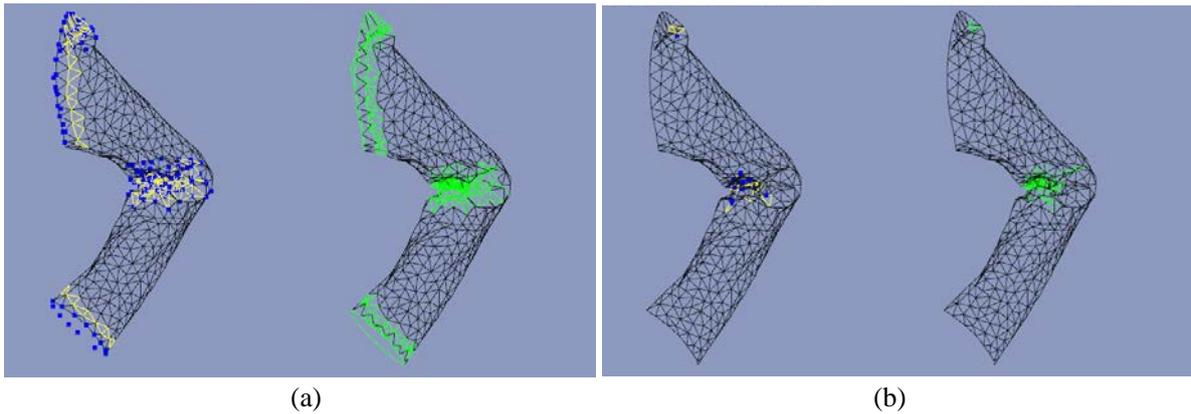


Fig.3. Results of Basic lower tailoring(a) and Our method(b)

Table 1 Comparison of Different Lower tailoring Optimization Methods

	Qty. of candidate VF primitive pair	Qty. of candidate EE primitive pair	Frame rate
Basic lower tailoring	1122	2056	12.70
2008 Curtis	74	475	32
2011 Tang	44	338	37.54
Ours	31	137	34.22

It can be seen from Table 1 that, comparing with basic lower tailoring method, the quantity of candidate primitive pair tested by our method decreased sharply and simulation frame rate has been improved by 3 times. Comparing with the tailoring method based on representative triangular mechanism of Curtis, since our method has filtered out non-coplanar primitive pairs and tested even fewer candidate primitive pairs, with higher tailoring efficiency. Our lower tailoring optimization method is lower than Tang's method in terms of simulation frame efficiency. But since our method has cut off redundant primitive pairs that do not intersect with primitive bounding box, fewer candidate primitive pairs have been tested. Thus, it is better than Tang's method in terms of tailoring efficiency.

Our lower tailoring optimization method is to process lower tailoring optimization firstly based on feature distribution, then tailoring optimization based on non-coplanar filter, so as to have effectively cut off redundant and non-colliding primitive pairs, reduced the quantity of primitive pairs for subsequent basic intersection testing and improved the efficiency of self-collision detection.

Conclusion

The Paper provides lower tailoring optimization method of self-collision detection. We processed lower tailoring optimization based on feature distribution firstly and cut off redundant primitive pairs through feature distribution. When distributing primitive pairs to candidate triangles, we processed intersection test of primitive bounding box, to cut off non-intersecting primitive pairs of primitive bounding box. Furthermore, on the basis of lower tailoring of feature distribution, we have processed tailoring optimization based on non-coplanar filter. Through non-coplanar test, we have cut off non-coplanar primitive pairs within time step, to have further reduced the quantity of candidate primitive pairs and improved the efficiency of self-collision detection.

Acknowledgement

In this paper, the research was sponsored by the National Natural Science Foundation of China (Project No. 201112400450401).

References

- [1] Curis S, Tamstorf R, Manocha D. Fast collision detection for deformable models using representative triangles[J]. In Proceedings of the 2008 symposium on Interactive 3D graphics and games, ACM,2008: 61–69.
- [2] Hutter M, Fuhrmann A. Optimized continuous collision detection for deformable triangle meshes[C]. Proc.WSCG07,2007: 25–32.
- [3] Tang M, Manocha D, Tong R. Fast continuous collision detection using deforming non-penetration filters[C]. In Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, ACM, 2010:7–13.
- [4] Tang C, Li S, Wang G. Reduced deforming filter culling for fast continuous collision detection[J]. VRST '10 Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology, 2010:79-82.
- [5] Tang C, Li S, Wang G. Fast continuous collision detection using parallel filter in subspace[J]. I3D '11: Symposium on Interactive 3D Graphics and Games,2011:71-80.
- [6] Provot X. Collision and self-collision handling in cloth model dedicated to design garment[J]. Graphics Interface, 1997:177–89.