Fast Continuous Collision Detection using Deforming Non-Penetration Filters

Min Tang[*]
Zhejiang University

Dinesh Manocha[†]
University of North Carolina at Chapel Hill

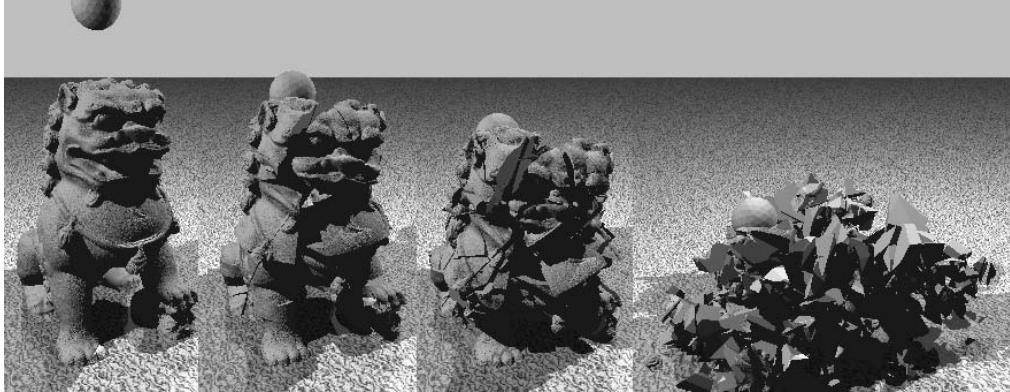Ruofeng Tong[‡]
Zhejiang University

Figure 1: Benchmark Lion: *In this simulation, a spherical ball falls on top of a Chinese statue of a lion and the lion gradually breaks into a high number of colliding pieces. This model has $805K$ vertices and $1.6M$ triangles. In this scene with changing topologies, our novel collision detection algorithm based on a deforming filter increases the culling efficiency and reduces the number of elementary tests by $10x$ when compared to prior methods and improves the performance of the CCD algorithm by $4.1x$.*

Abstract

We present a novel culling algorithm that uses deforming non-penetration filters to improve the performance of continuous collision detection (CCD) algorithms. The underlying idea is to use a simple and effective filter that reduces both the number of false positives and the elementary tests between the primitives. This filter is derived from the coplanarity condition and can be easily combined with other methods used to accelerate CCD. We have implemented the algorithm and tested its performance on many non-rigid simulations. In practice, we can reduce the number of false positives significantly and improve the overall performance of CCD algorithms by $1.5 - 8.2x$.

1 Introduction

Continuous collision detection (CCD) is widely used in different applications, including physically-based simulation, CAD/CAM, and robot motion planning. Its main purpose is to check for collisions between two discrete positions of the objects or primitives by using some form of interpolating trajectory. Some algorithms use a linearly interpolating trajectory between the corresponding vertices at two different positions and CCD computations reduce to performing elementary tests between the primitives based on this formulation. Specifically, the CCD test between two deforming triangles reduces to performing 9 vertex-face (VF) and 6 edge-edge

---
[*]e-mail: tang_m@zju.edu.cn
[†]e-mail: dm@cs.unc.edu
[‡]e-mail: trf@zju.edu.cn

(EE) elementary tests based on coplanarity conditions. Each elementary test can be reduced to computing the roots of a cubic equation [Provot 1997; Bridson et al. 2002].

There is an extensive amount of research on accelerating the performance of CCD algorithms between complex deformable models and reducing the number of elementary tests. Most algorithms use bounding volume hierarchies (BVHs) or a combination of high-level and low-level culling that use bounds on normals or mesh connectivity or GPU-based accelerations. However, the current CCD algorithms spend a high fraction of the total query time in performing the elementary tests between the primitives [Hutter and Fuhrmann 2007; Curtis et al. 2008; Tang et al. 2009a; Sud et al. 2006; Wong and Baciu 2006] and result in a very high number of false positives (i.e., 95% or more).

Main Result: In this paper, we present a simple culling algorithm that can significantly reduce the number of false positives in terms of elementary tests and improve the overall performance of CCD algorithms. We introduce a new non-penetration filter which can remove many false positives that cannot be culled by BVHs or other culling methods. The main idea is to exploit the coplanarity condition of the elementary tests and check for overlap between the primitives at every deforming instance of the continuous trajectory between the two discrete time steps. We derive two formulations for this deforming non-penetration filter: one for the VF test and the other for the EE test.

Our approach is complementary to prior CCD algorithms and can be easily combined with BVHs or algorithms that use higher level culling based on normal bounds [Tang et al. 2009a] or lower level culling based on mesh connectivity [Curtis et al. 2008]. We have tested its performance on complex benchmarks corresponding to cloth simulation, breaking objects, and N-body simulations with high model complexity and observed up to 58x reduction in the number of elementary tests and up to 8.2x improvement in the performance of the overall CCD algorithm.

Organization: The rest of the paper is organized as follows: Section 2 gives a brief survey of prior work. We introduce our notation

and describe the coplanarity-based culling method and the deforming non-penetration filter in Section 3. We present the overall CCD algorithm and its implementation in Section 4. We compare our approach with prior algorithms in Section 5.

2 Related Work

Many efficient algorithms have been designed for CCD between rigid models and deformable models [Govindaraju et al. 2005; Hutter and Fuhrmann 2007; Zhang et al. 2007; Curtis et al. 2008; Tang et al. 2009a; Tang et al. 2009b]. Some of these approaches linearly interpolate between the vertices of the model and compute the first time-of-contact based on hierarchical culling and performing elementary tests between the triangle pairs. Based on linearly interpolating motion of the vertices, the elementary tests reduce to solving cubic equations. However, these tests can be expensive and also sensitive to tolerance [Brochu and Bridson 2009]. Other CCD formulations use a different formulation of the interpolating motion [Redon et al. 2002; Redon et al. June, 2004; Kim and Rossignac 2003]. Most of these methods reduce to computing roots of polynomial functions and can be accelerated using Béizer clipping [Sederberg and Nishita 1990], recursive subdivision [Taubin 1994], or eigenvalue computations [Manocha and Demmel 1995].

Most of the recent work on CCD algorithms has been on designing high-level and low-level culling algorithms that can reduce the number of elementary tests between the primitives.

High-level culling: Many high-level methods use BVHs to cull away the non-overlapping primitives and they can use tight fitting bounding volumes such as k-DOPs or OBBs to obtain higher culling efficiency. Other culling methods check for self-collisions based on surface normals and curvature [Volino and Thalmann 1994; Provot 1997; Mezger et al. 2003]. Recently, Tang et al. [2009a] extended these ideas to CCD and presented continuous normal cones (CNCs).

Low-level culling: Hutter and Fuhrmann [2007] used bounding volumes (k-DOPs) of the primitives to avoid performing elementary tests between different features. Curtis et al. [2008] and Wong and Baciu [2006] used masking schemes to remove the redundant elementary tests. However, neither adjacent triangles nor their features can be culled by bounding volumes. Govindaraju et al. [2005] eliminate some of the elementary tests associated with adjacent triangles based on some of the tests between the non-adjacent primitives. Tang et al. [2009a] proposed the concept of orphan sets to eliminate almost all the redundant elementary tests between adjacent triangles. The idea has been further extended to procedural representation triangles (PR-Triangles) to remove all redundant elementary tests between non-adjacent triangles. Although these methods can lower the number of elementary tests and false positives, the current CCD algorithms can still result in a high number of false positives (e.g., 95% or more).

3 Deforming Non-Penetration Filter

In this section, we introduce the notation used and present our culling algorithm that reduces the number of false positives.

3.1 Notations

We use following notations in the rest of the paper:

- $SV(X)$ is the swept volume of a deforming primitive X along the linear interpolating path between the vertices.
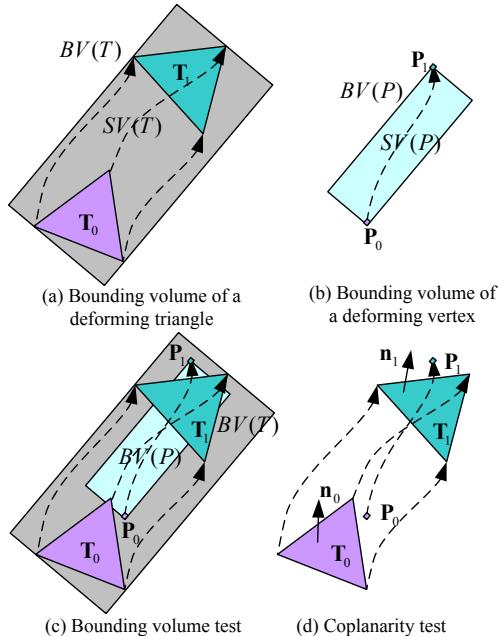- $BV(X)$ is the bounding volume of a deforming primitive X.

(a) Bounding volume of a deforming triangle

(b) Bounding volume of a deforming vertex

(c) Bounding volume test

(d) Coplanarity test

Figure 2: Deforming Filter: *For a deforming triangle T and a deforming vertex P defined by T_0,T_1 and P_0, P_1 respectively, the bounding volume test (c) becomes quite conservative. The coplanarity test (d) checks whether a penetration between the primitives during the time interval. If the vertex is always on the same side of the triangle during the entire time interval, then the four vertices associated with that elementary test cannot be coplanar during the time interval $[0, 1]$ and therefore, no collision occurs.*

- T_0, T_1 and T_t represent the instances of a deforming triangle T at $t = 0$, $t = 1$, and arbitrary $t \in [0, 1]$, respectively.

- a_0, b_0, and c_0 are the three vertices of T_0. a_1, b_1, and c_1 are the vertices of T_1. a_t, b_t, and c_t are the vertices of T_t.

- n_0, n_1, and n_t are the normal vectors of T_0, T_1, and T_t.

- P_0, P_1 and P_t are the instances of a deforming vertex P at $t = 0$, $t = 1$, and arbitrary $t \in [0, 1]$, respectively.

- Operator '$*$', '$\cdot$', and '$\times$' denote multiple of two scale values, dot product of two vectors, and cross product of two vectors, respectively.

3.2 Motivation

Given the linear interpolating motion between the vertices, the CCD test between a triangle pair can be reduced to two types of elementary tests: 6 VF tests and 9 EE tests. Each elementary test can be further broken down into two parts: coplanarity test and inside test. Both the VF tests and EE tests involve the use of four deforming vertices, and a necessary condition for a collision is that these four vertices be coplanar. Provot [1997] showed the coplanarity test of four vertices can be reduced to finding roots of a cubic equation. Instead of solving the cubic equation, we will deduce a sufficient condition that these four vertices are non-coplanar during the time interval. By using this condition, many elementary tests can be culled and we do not need to solve the cubic equations.

We use bounding volumes for each primitive (edge, face, vertex), and perform bounding volume based culling before the elementary tests. However, the bounding volume tests can become quite con-
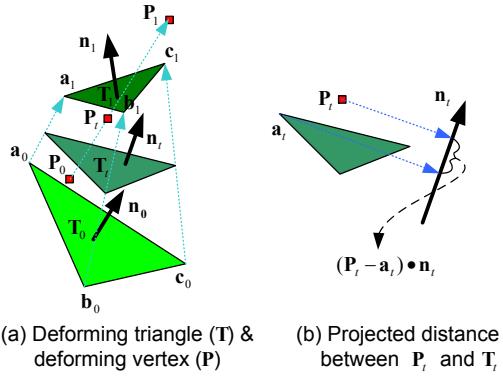
(a) Deforming triangle (**T**) &
deforming vertex (**P**)

(b) Projected distance
between $\mathbf{P}_t$ and $\mathbf{T}_t$

Figure 3: VF test filter: *To perform a VF test between a deforming triangle (defined by a_0, b_0, c_0 at $t = 0$, and a_1, b_1, c_1 at $t = 1$) and a moving vertex (defined by P_0 at $t = 0$ and P_1 at $t = 1$), we need to check for coplanarity between the vertex and the triangle by finding a t ($t \in [0, 1]$) when the projected distance along the normal vector of the triangle is equal to zero, i.e., $(P_t - a_t) \cdot n_t = 0$.*

servative for primitive level CCD tests. For the example shown in Fig. 2, $BV(T)$ and $BV(P)$ are the bounding volumes of a deforming triangle T and a deforming vertex P, respectively. In order to perform culling, $BV(T)$ and $BV(P)$ must contain the swept volumes of deforming primitives, i.e., $SV(T)$ and $SV(P)$, respectively (Fig. 2(a-b)). The VF pair $\{T, P\}$ will be tested for exact CCD test, since their bounding volumes overlap during the time interval $[0, 1]$ (Fig. 2(c)).

We make use of the following observation: if the vertex P is always on the same side of the triangle T along the entire $[0, 1]$ trajectory, this pair should be classified as a false positive since it cannot be coplanar (Fig. 2(d)). Based on this observation, we derive a sufficient condition to check whether P_t will always be on the same side of T_t during the time interval $[0, 1]$. If a pair of primitives satisfies this condition, then we don't need to perform the exact test in terms of solving a cubic equation.

The idea can be similarly extended for the EE tests: if there is no crossing between the two deforming edges along the continuous trajectory, they cannot be coplanar during the time interval.

3.3 Coplanarity Test

In order to check the coplanarity of a vertex P_t and a triangle T_t, we need to calculate the projected distance between them along the direction of n_t, as shown in Fig. 3(b). If this distance becomes zero at any time interval, than the four vertices of the two primitives are classified as coplanar.

Non-coplanar Theorem for VF tests: *For a triangle T_t and a vertex P_t defined by the start and end positions during the interval $[0, 1]$, these positions are linearly interpolated in the interval with respect to the time variable, t. If the following four scalar values: A, B, $\frac{2*C+F}{3}$, and $\frac{2*D+E}{3}$ have the same sign, T_t and P_t will not be coplanar during the interval:*

$$
\begin{aligned}
A &= (P_0 - a_0) \cdot n_0, & B &= (P_1 - a_1) \cdot n_1 & (1) \\
C &= (P_0 - a_0) \cdot \hat{n}, & D &= (P_1 - a_1) \cdot \hat{n} & (2) \\
E &= (P_0 - a_0) \cdot n_1, & F &= (P_1 - a_1) \cdot n_0 & (3)
\end{aligned}
$$

And:

$$
\begin{aligned}
n_0 &= (b_0 - a_0) \times (c_0 - a_0), n_1 = (b_1 - a_1) \times (c_1 - a_1) \\
\hat{n} &= \frac{(n_0 + n_1 - (\vec{v_b} - \vec{v_a}) \times (\vec{v_c} - \vec{v_a}))}{2} \\
\vec{v_a} &= a_1 - a_0, \vec{v_b} = b_1 - b_0, \vec{v_c} = c_1 - c_0.
\end{aligned}
$$

Proof. The normal vector n_t of the deforming triangle at time t can be represented as following:

$$n_t = n_0 * B_0^2(t) + \hat{n} * B_1^2(t) + n_1 * B_2^2(t) \quad (4)$$

where $B_i^2(t)$ is the ith basis function of the Bernstein polynomials of degree 2.

We define: $\alpha = B_0^2(t) = (1 - t)^2$, $\beta = B_1^2(t) = 2 * t * (1 - t)$, and $\gamma = B_1^2(t) = t^2$. Then Equation (4) becomes:

$$n_t = n_0 * \alpha + \hat{n} * \beta + n_1 * \gamma$$

For the moving vertex $P_t = P_0 * (1 - t) + P_1 * t$ and a vertex of the deforming triangle $a_t = a_0 * (1 - t) + a_1 * t$, their projected distant along n_t is:

$$
\begin{aligned}
(P_t - a_t) \cdot n_t &= ((P_0 - a_0) * (1 - t) + (P_1 - a_1) * t) \cdot n_t \\
&= ((P_0 - a_0) * (1 - t) + (P_1 - a_1) * t) \\
&\quad \cdot (n_0 * \alpha + \hat{n} * \beta + n_1 * \gamma) \\
&= (P_0 - a_0) \cdot n_0 * (1 - t) * \alpha \\
&+ (P_0 - a_0) \cdot \hat{n} * (1 - t) * \beta \\
&+ (P_0 - a_0) \cdot n_1 * (1 - t) * \gamma \\
&+ (P_1 - a_1) \cdot n_1 * t * \gamma \\
&+ (P_1 - a_1) \cdot \hat{n} * t * \beta \\
&+ (P_1 - a_1) \cdot n_0 * t * \alpha
\end{aligned}
$$

Substitute α, β, and γ, we have:

$$
\begin{aligned}
(P_t - a_t) \cdot n_t &= (P_0 - a_0) \cdot n_0 * (1 - t)^3 \\
&+ (P_0 - a_0) \cdot \hat{n} * 2 * (1 - t)^2 * t \\
&+ (P_0 - a_0) \cdot n_1 * (1 - t) * t^2 \\
&+ (P_1 - a_1) \cdot n_1 * t^3 \\
&+ (P_1 - a_1) \cdot \hat{n} * 2 * t^2 * (1 - t) \\
&+ (P_1 - a_1) \cdot n_0 * t * (1 - t)^2 \quad (5)
\end{aligned}
$$

Using the symbols defined by Equations(1)-(3), Equation (5) becomes:

$$
\begin{aligned}
(P_t - a_t) \cdot n_t &= A * (1 - t)^3 + C * 2 * (1 - t)^2 * t \\
&+ E * (1 - t) * t^2 + B * t^3 \\
&+ D * 2 * t^2 * (1 - t) + F * t * (1 - t)^2 \\
&= A * B_0^3(t) + \frac{2 * C + F}{3} * B_1^3(t) \\
&+ \frac{2 * D + E}{3} * B_2^3(t) + B * B_3^3(t) \quad (6)
\end{aligned}
$$

where $B_i^3(t)$ is the ith basis function of the Bernstein polynomials of degree 3.

By using the convex hull property associated with control points of the Bernstein basis, the range of the projected distance between P_t and T_t is bounded by the *control vertices*. In our case, these control vertices are the four scalar values: A, B, $\frac{2*C+F}{3}$, and $\frac{2*D+E}{3}$.
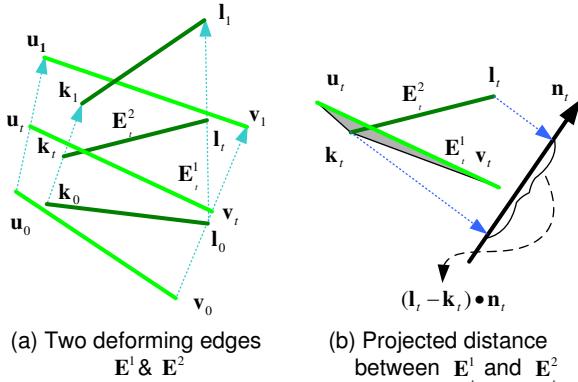
(a) Two deforming edges E^1 & E^2

(b) Projected distance between E_t^1 and E_t^2

Figure 4: **EE test filter:** *To perform a EE test between the two edges E^1 and E^2 (defined by u_0, v_0 and k_0, l_0 at $t = 0$, u_1, v_1 and k_1, l_1 at $t = 1$), we need to check the coplanarity conditions of these vertices by finding all t ($t \in [0, 1]$) where the projected distance between l_t and the triangle defined by k_t, u_t, and v_t is equal to zero, i.e., $(l_t - k_t) \cdot n_t = 0$.*

So we obtain the sufficient condition for non-coplanar: If all the four scalar values are negative/positive, the vertices P_t will never be on the plane defined by the three vertices (a_t, b_t, and c_t) of the deforming triangle T_t. □

The geometric interpretation of symbol A and B is that they represent the projected distances at $t = 0$ and $t = 1$, respectively. From Equation(6), we conclude that it is not sufficient to only check the signs of A and B, the values of other symbols C-F also affect the coplanarity condition.

3.3.1 Extension to EE tests

In order to perform an elementary test between two edges E^1 and E^2 (defined by u_0, v_0 and k_0, l_0 at $t = 0$, u_1, v_1 and k_1, l_1 at $t = 1$), we need to check the coplanarity of these four points (Fig. 4). By replacing P_t, a_t, b_t, c_t with l_t, k_t, u_t, v_t, we can symmetrically deduce a non-coplanar theorem for EE tests:

Non-coplanar Theorem for EE tests: *For two deforming edges E^1 and E^2 defined by the start and end positions during the interval $[0, 1]$, these positions are linearly interpolated in the interval with respect to the time variable, t. If the following four scalar values: A', B', $\frac{2*C'+F'}{3}$, and $\frac{2*D'+E'}{3}$ have the same sign, E^1 and E^2 will not be coplanar during the interval:*

$$A' = (l_0 - k_0) \cdot n_0', \quad B' = (l_1 - k_1) \cdot n_1'$$
$$C' = (l_0 - k_0) \cdot \hat{n}', \quad D' = (l_1 - k_1) \cdot \hat{n}'$$
$$E' = (l_0 - k_0) \cdot n_1', \quad F' = (l_1 - k_1) \cdot n_0'$$

And: $\quad n_0' = (u_0 - k_0) \times (v_0 - k_0), \quad n_1' = (u_1 - k_1) \times (v_1 - k_1)$
$$\hat{n}' = \frac{(n_0' + n_1' - (\vec{v_u} - \vec{v_k}) \times (\vec{v_v} - \vec{v_k}))}{2}$$
$$\vec{v_k} = k_1 - k_0, \vec{v_u} = u_1 - u_0, \vec{v_v} = v_1 - v_0.$$

3.4 CCD Algorithm

We use the non-planarity filter defined above to perform coplanarity-based culling. We use Equations(1)-(3) to calculate the values of the four control vertices defined in those equations for a primitive pair. If these variables have the same sign, this pair will be identified as a false positive.
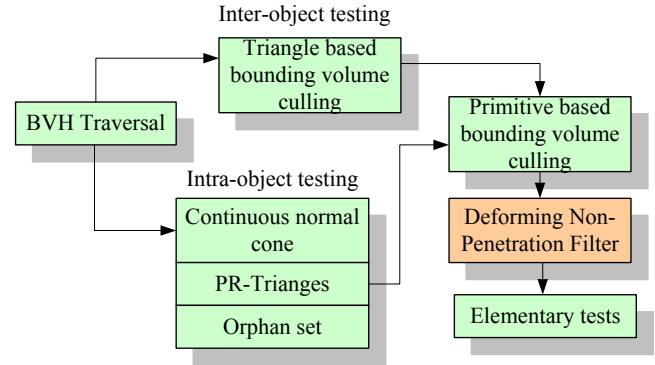

Figure 5: **CCD algorithm with deforming non-penetration filter:** *The deforming non-penetration filter is complementary to the overall CCD algorithm and can be performed right before the elementary tests.*
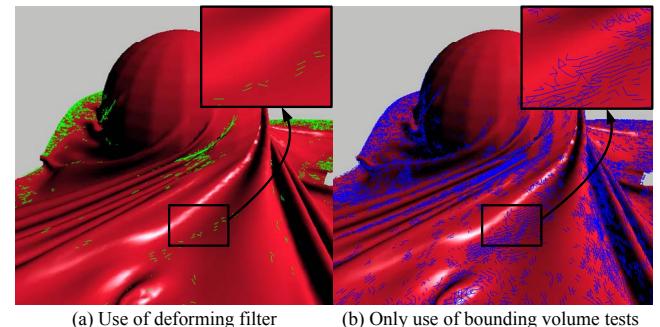

(a) Use of deforming filter

(b) Only use of bounding volume tests

Figure 6: **Edges involved by EE tests with different culling methods:** *This figure shows edges (highlighted in green and blue respectively) used by EE tests during a specific frame of the Cloth benchmark (Fig. 7(d)) with coplanarity-based culling and bounding volume based culling respectively. Comparing to only bounding volume based culling, the number of EE elementary tests are reduced by 80% with the deforming non-penetration filter.*

The deforming non-penetration filter can be combined with hierarchical representations (Fig. 5). During the traversal of BVH, we perform different high-level culling methods to remove redundant pairwise tests or primitive pairs that are not in close proximity to one another. Next, the deforming non-penetration filter is used as part of low-level culling to further remove the false positives.

Fig. 6 compares the effects of bounding volume based culling and coplanarity-based culling. The green line segments in Fig. 6(a) and blue line segments in Fig. 6(b) describe the distribution of edges involved in EE tests with deforming non-penetration filter and bounding volume based culling respectively. As shown in the figure, compared to the number of tests in the bounding volume based culling, we observe more than 80% reduction in the number of elementary tests on this scene.

4 Implementation and Performance

In this section, we describe our implementation and highlight the performance of our algorithm on several benchmarks.

4.1 Implementation

We have implemented our algorithm on a standard 2.4GHz Intel Pentium machine with 4GB RAM on 32-bit Windows/XP platform. The performance is measured using a single thread. We use k-DOPs (specifically 16-DOPs) as bounding volumes because they provide a good balance between tight fitting and rapid updating.

We use restructuring and refitting to update the hierarchy for deformable models. Intel SSE/SSE2 instructions are used to accelerate the updating of and overlap testing between bounding volumes. We use an implementation of the ICCD algorithm based on normal cones for high-level culling [Tang et al. 2009a] as a baseline for comparison. We also integrate our filters with the R-Triangle algorithm [Curtis et al. 2008], as it spends a large fraction of the query time on the elementary tests.

4.2 Benchmarks & Performance

In order to test the performance of our algorithm, we used six different benchmarks, arising from different simulations with different characteristics.

- **Lion:** For this breaking benchmark with 1.6M triangles (Figure 1), our algorithm reduces the number of elementary test by 10x as compared to prior CCD algorithms.

- **Balls:** A scene with hundreds of balls ($34K$ triangles) that are colliding with each other (Figure 7(a)). Our deforming non-penetration filter reduces the number of elementary tests by 17x.

- **Falling Alphabets:** Multiple deforming characters ($5K$ triangles) fall into a bowl and break into pieces (Figure 7(b)). Our CCD algorithm reduces the number of elementary tests by 58x.

- **Princess:** This model ($40K$ triangles) has many inter- and intra-object collisions (Figure 7(c)). Our algorithm reduces the number of elementary tests by 20.5x.

- **Cloth:** A cloth ($92K$ triangles) has a high number of self-collisions (Figure 7(d)). Our algorithm reduces the number of elementary tests by 11.5x.

- **Flamenco:** This benchmark ($49K$ triangles) has many inter- and intra-object collisions (Figure 7(e)). The CCD algorithm with the deforming non-penetration filter reduces the number of elementary tests by 17x.

Fig. 8 highlights the culling efficiency of our algorithm by comparing the number of elementary tests performed. As shown in the figure, the elementary tests are dramatically reduced by $10 - 58.7$x. Also, by preventing the computation of these false positives, we achieve $1.5 - 3.5$x improvement on overall performance in comparison with the ICCD algorithm [Tang et al. 2009a], and $2.4 - 8.2$x improvement over R-Triangles [Curtis et al. 2008](Table. 1). For Benchmark Flamenco, its average running time per frame in R-Triangles, ICCD, and our system are shown in Figure 9.

5 Analysis and Comparison

In this section, we analyze our results and compare against prior methods.

5.1 Analysis

Although the performance of our deforming non-penetration filter varies with different benchmarks, we are able to obtain high culling
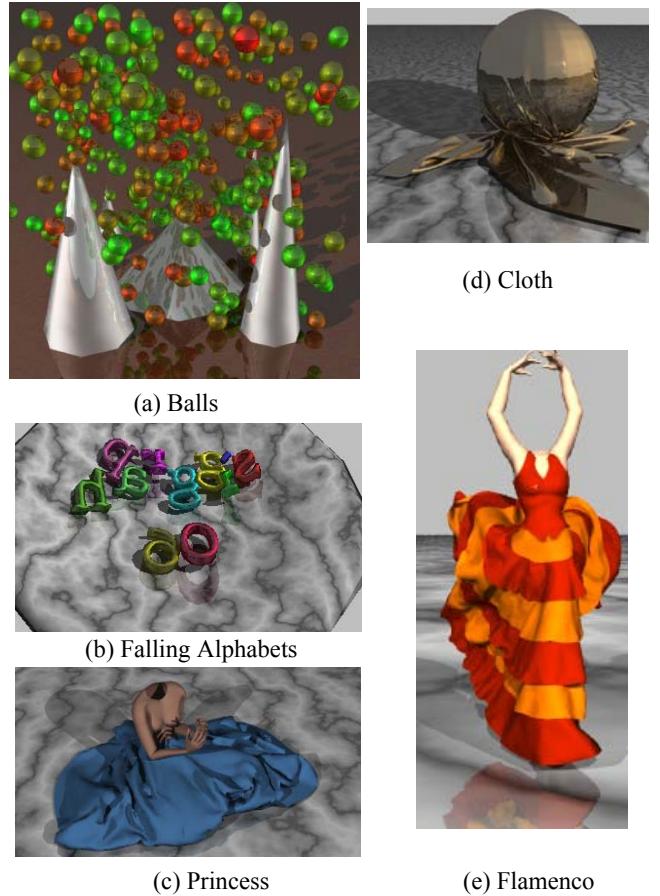

(a) Balls

(d) Cloth

(b) Falling Alphabets

(c) Princess

(e) Flamenco

Figure 7: Benchmarks: *All the benchmarks have multiple simulation steps. We perform CCD, including self-collisions, between discrete steps of the simulation and compute the first time-of-contact.*

efficiency in many of them. The basic filter test proposed in the non-coplanar theorems is conservative. Intuitively, our non-coplanar theorem eliminates the need to perform exact tests on pairs where no 'penetrations' occur along the continuous trajectory. In practice, due to the temporal coherence between subsequent frames, the 'penetrations' only occur infrequently. Fig. 10 shows the changing of ratios of primitive pairs in which no 'penetration' occurs during the time interval. In the figure, the ratios stay above 80%, and are changing smoothly. Due to coherence, our deforming non-penetration filter can be quite effective.

We use the Interval-Newton method to solve the cubic equations. An elementary test takes about roughly 155 additions, 217 multiplications, and 6 divisions on average (all for float point values). On the other hand, the deforming non-penetration filter only need to perform 29 additions and 40 multiplications on average. In practice, the speed of a deforming non-penetration filter is about 5.5-10.2x faster than an exact test.

5.2 Comparison

In this section, we compare our novel algorithm with prior culling algorithms.

Bounding volume based culling: Bounding volumes (e.g. spheres [Hubbard 1993; Palmer and Grimsdale 1995; Bradshaw and O'Sullivan 2004], AABBs [van den Bergen 1997], k-
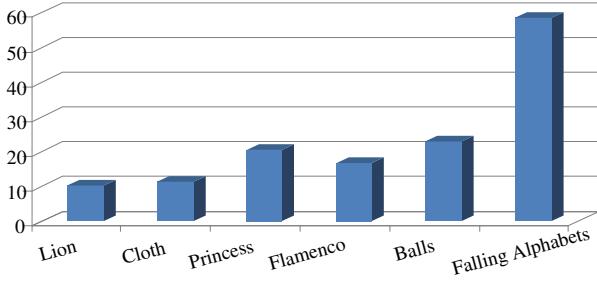
Figure 8: Culling efficiency: *This figure shows the culling efficiency of our algorithm by comparing the number of elementary tests performed. For the benchmarks, our algorithm is capable of reducing the number of elementary tests by $10 - 58.7x$.*
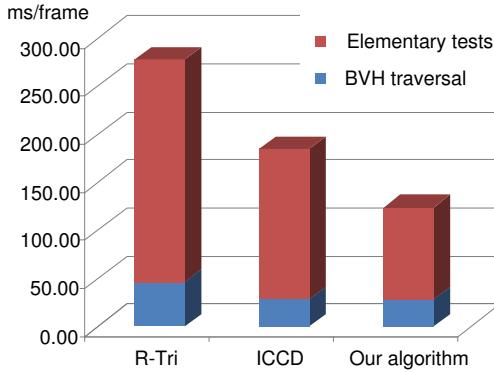

Figure 9: Average running time per frame: *This figure shows the average running time per frame of Benchmark Flamenco with R-Triangles [Curtis et al. 2008], ICCD [Tang et al. 2009a], and our algorithm, respectively.*

DOPs [Klosowski et al. 1998], OBBs [Gottschalk et al. 1996], etc.) can be used to remove primitive pairs whose primitives are not in close proximity to each other. These methods work for inter-object and intra-object collisions. But for CCD, the bounding volumes of primitives are quite conservative and make false positive rates very high. Our deforming non-penetration filter can improve the culling efficiency.

Orphan sets: Orphan sets [Tang et al. 2009a] are quite effective at removing redundant elementary tests between adjacent triangles. However, they do not reduce the false positives between non-adjacent primitives, whereas our filter can be used for these cases.

Continuous normal cones: CNCs [Tang et al. 2009a] are applied to remove false positives on large areas where are relatively flat in a mesh. They only work for self-collisions, and are quite conservative on models with high curvatures.

Representative triangles: Representative triangles [Curtis et al. 2008] can remove all the redundant elementary tests that are caused by shared features between the adjacent triangles. Procedural representative triangles [Tang et al. 2009a] combine it with CNC culling. Our deforming non-penetration filter complements these methods.

GPU based culling: GPU algorithms that use occlusion queries [Govindaraju et al. 2005] and rasterization based distance field computation [Sud et al. 2006] are complementary to our culling algorithm.

Table 1: Performance and Speedup: *This table shows the average query time of our method and performance improvement over ICCD [Tang et al. 2009a] and R-Triangles [Curtis et al. 2008].*

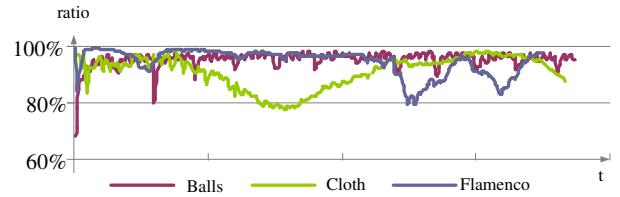| Model	Query (time ms)	Speedup over ICCD	Speedup over R-Triangles
Cloth	144	2x	2.4x
Princess	18.7	2.4x	3.1x
Flamenco	110	1.5x	2.8x
Balls	52.8	1.7x	3.4x
Falling Alphabets	4.6	3.5x	8.2x
Lion	4432	2.4x	4.1x

Figure 10: Temporal coherency: *(a), (b), and (c) show the changing ratios of primitive pairs in which no penetration occurs during the time interval for the benchmarks Balls, Cloth, and Flamenco, respectively. The ratios tend to be above 80%, and are changing smoothly.*

5.3 Limitations

Our approach only provides a filter at the feature level. As a result, it needs to be applied for each VF or EE test. Moreover, our test can be conservative due to the underlying formulation. If a high-level culling algorithm is able to cull away a high percentage of false positives, then we may obtain a relatively small improvement using our deforming non-penetration filter.

6 Conclusion and Future Work

We have presented a novel culling algorithm for CCD between complex deformable models by proposing a deformable non-penetration filter. By checking the coplanarity of the vertices of the primitives, our algorithm can significantly reduce the number of false positives, and subsequently improve the overall performance of CCD algorithms. Moreover, our approach is complementary to and can be combined with most prior methods. We test the performance on different benchmarks and observed considerable improvement in terms of reducing the number of false positives.

In our future work, it is interesting to develop enhanced filters that would further improve the culling efficiency. It may also be possible to further improve the performance of the filters by further utilizing temporal coherence and/or using SSE instructions. Currently, our CCD formulation is based on linear movements of vertices, it is possible to derive similar bounds for other motions of the vertices, e.g. arbitrary interplaiting motion [Redon et al. June, 2004]. Finally, we would like to use our CCD algorithm with some well known game Physics engines (e.g., Bullet, Hovak, PhysX, etc.), and also apply our CCD algorithm to more complex benchmarks.

Acknowledgments

References

BRADSHAW, G., AND O'SULLIVAN, C. 2004. Adaptive medial-axis approximation for sphere-tree construction. *ACM Trans. on Graphics 23*, 1, 1–26.

BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treament for collisions, contact and friction for cloth animation. *Proc. of ACM SIGGRAPH*, 594–603.

BROCHU, T., AND BRIDSON, R. 2009. Numerically robust continuous collision detection fro dynamic explict surfaces. Tech. rep., TR-2009-03, University of British Columbia, Vancouver.

CURTIS, S., TAMSTORF, R., AND MANOCHA, D. 2008. Fast collision detection for deformable models using representative-triangles. In *SI3D '08: Proceedings of the 2008 Symposium on Interactive 3D graphics and games*, 61–69.

GOTTSCHALK, S., LIN, M. C., AND MANOCHA, D. 1996. Obb-tree: a hierarchical structure for rapid interference detection. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 171–180.

GOVINDARAJU, N., KNOTT, D., JAIN, N., KABUL, I., TAMSTORF, R., GAYLE, R., LIN, M., AND MANOCHA, D. 2005. Interactive collision detection between deformable models using chromatic decomposition. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH) 24*, 3, 991–999.

HUBBARD, P. M. 1993. Interactive collision detection. In *Proceedings of IEEE Symposium on Research Frontiers in Virtual Reality*.

HUTTER, M., AND FUHRMANN, A. 2007. Optimized continuous collision detection for deformable triangle meshes. In *Proc. WSCG '07*, 25–32.

KIM, B., AND ROSSIGNAC, J. 2003. Collision prediction for polyhedra under screw motions. In *ACM Symposium in Solid Modeling and Applications*, ACM Press, 4–10.

KLOSOWSKI, J., HELD, M., MITCHELL, J., SOWIZRAL, H., AND ZIKAN, K. 1998. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Trans. on Visualization and Computer Graphics 4*, 1, 21–37.

MANOCHA, D., AND DEMMEL, J. 1995. Algorithms for intersecting parametric and algebraic curves ii: multiple intersections. *Graph. Models Image Process. 57*, 2, 81–100.

MEZGER, J., KIMMERLE, S., AND ETZMUβ, O. 2003. Hierarchical techniques in cloth detection for cloth animation. *Journal of WSCG 11*, 1, 322–329.

PALMER, I. J., AND GRIMSDALE, R. L. 1995. Collision detection for animation using sphere-trees. *Computer Graphics Forum 14*, 2, 105–116.

PROVOT, X. 1997. Collision and self-collision handling in cloth model dedicated to design garment. *Graphics Interface*, 177–189.

REDON, S., KHEDDAR, A., AND COQUILLART, S. 2002. Fast continuous collision detection between rigid bodies. *Proc. of Eurographics (Computer Graphics Forum) 21*, 3, 279–288.

REDON, S., KIM, Y. J., LIN, M. C., AND MANOCHA, D. June, 2004. Fast continuous collision detection for articulated models. In *SM '04: Proceedings of the ninth ACM symposium on Solid modeling and applications*, 145–156.

SANNA, A., AND MILANI, M. 2004. CDFast: an algorithm combining different bounding volume strategies for real time collision detection. *SCI Proceedings 2*, 144–149.

SEDERBERG, T. W., AND NISHITA, T. 1990. Curve intersection using be´zier clipping. *Comput. Aided Des. 22*, 9, 538–549.

SUD, A., GOVINDARAJU, N., GAYLE, R., KABUL, I., AND MANOCHA, D. 2006. Fast proximity computation among deformable models using discrete voronoi diagrams. *Proc. of ACM SIGGRAPH*, 1144–1153.

TANG, M., CURTIS, S., YOON, S.-E., AND MANOCHA, D. 2009. ICCD: Interactive continuous collision detection between deformable models using connectivity-based culling. *IEEE Transactions on Visualization and Computer Graphics 15*, 4, 544–557.

TANG, M., KIM, Y. J., AND MANOCHA, D. 2009. C2a: Controlled conservative advancement for continuous collision detection of polygonal models. *Proceedings of International Conference on Robotics and Automation*, 849–854.

TAUBIN, G. 1994. Rasterizing algebraic curves and surfaces. *IEEE Comput. Graph. Appl. 14*, 2, 14–23.

VAN DEN BERGEN, G. 1997. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools 2*, 4, 1–14.

VOLINO, P., AND THALMANN, N. M. 1994. Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity. *Computer Graphics Forum (EuroGraphics Proc.) 13*, 3, 155–166.

WONG, W. S.-K., AND BACIU, G. 2006. A randomized marking scheme for continuous collision detection in simulation of deformable surfaces. *Proc. of ACM VRCIA*, 181–188.

ZHANG, X., REDON, S., LEE, M., AND KIM, Y. J. 2007. Continuous collision detection for articulated models using taylor models and temporal culling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 3, 15.