Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2010)
M. Otaduy and Z. Popovic (Editors)

Supplementary Report

FASTCD: Fracturing-Aware Stable Collision Detection

Jae-Pil Heo[1], Joon-Kyung Seong[1], DukSu Kim[1], Miguel A. Otaduy[2], Jeong-Mo Hong[3], Min Tang[4], Sung-Eui Yoon[1]

[1] KAIST [2] URJC Madrid [3] Dongguk Univ. [4] Zhejiang Univ.

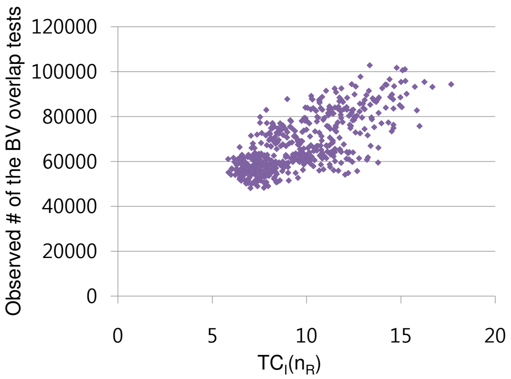

Figure 10: *This graph shows the relationship between TC_I values of the root nodes of BVHs of the Stanford bunny and the observed numbers of the BV overlap tests to perform inter-collision queries between the two Stanford bunny models that are randomly placed. The linear correlation between them is 0.73.*

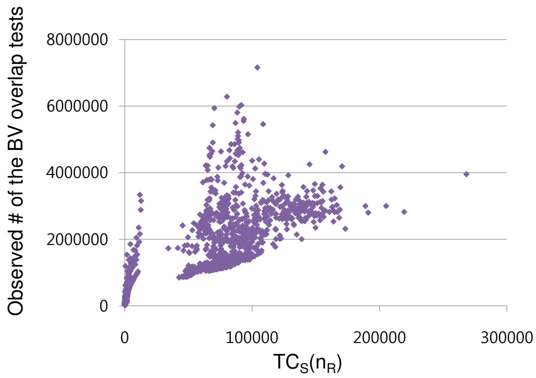

Figure 11: *This graph shows the relationship between TC_S values of the root nodes of BVHs of the cloth simulation benchmark model and the observed number of the BV overlap tests to perform a self-collision query to the same benchmark. The linear correlation between them is 0.72.*

Appendix A: Extension to CCD

We extend our dual-cone culling algorithm to CCD. CCD computes intersecting primitives at the first time-of-contact during a time interval between two discrete time steps. CCD methods model continuous motions of primitives. Typically, a simple linear motion is one of the widely used motions [Pro97]. In the CCD case, collisions arise in two contact configurations, vertex-face (VF) case and edge-edge (EE) cases. These two cases are detected by performing VF and EE elementary tests, which reduce to solving cubic equations given the linear continuous motion between two discrete time steps [Pro97].

To extend our culling method for CCD, we need to compute the surface normal and binormal cones with the deforming triangles in the time interval between two discrete time steps. Tang et al. [TCYM08] computed the surface normal cone in such time interval. Therefore, we only focus on computing a binormal cone for CCD.

Before presenting a theorem that computes a *continuous binormal cone*, we define terms for the theorem. Given a triangle T that deforms during a time interval $[0,1]$, we let a_t, b_t, and c_t to be positions of three vertices of a triangle T, which are linearly interpolated in the time interval with respect to the time variable $t \in [0,1]$ (see Fig. 13). Also, $\vec{N}_0$ and $\vec{N}_1$ denote normals of the triangle T at time 0 and 1 respectively, i.e. $\vec{N}_0 = (b_0 - a_0) \times (c_0 - a_0)$ and $\vec{N}_1 = (b_1 - a_1) \times (c_1 - a_1)$. The triangle T has three different edges, $\{(a_t, b_t), (b_t, c_t), (c_t, a_t)\}$. We use (p_t, q_t) to represent one of such edges. Given an edge (p_t, q_t), $\vec{V}_p$ represents the vector that starts from p_0 and ends at p_1, i.e. $\vec{V}_p = p_1 - p_0$; $\vec{V}_q$, $\vec{V}_a$, $\vec{V}_b$, and $\vec{V}_c$ are defined similarly with q_t, a_t, b_t, and c_t respectively.

Theorem A.1 (Continuous Binormal Cone) Given a triangle T that deforms during a time interval $[0,1]$, the binormal vector $\vec{B}_t$ for an edge (p_t, q_t) of the triangle T at time t, is given by the following cubic Bezier curve:

$$\vec{B}_t = \alpha B_0^3(t) + (3\alpha+\beta)/3 B_1^3(t) + (3\alpha+2\beta+\gamma)/3 B_2^3(t) + (\alpha+\beta+\gamma+\omega) B_3^3(t),$$

where $B_i^3(t)$ is the i^{th} Bernstein polynomials of the degree three, $\alpha = (q_0 - p_0) \times \vec{N}_0, \beta = (q_0 - p_0) \times \vec{D} + (\vec{V}_q - \vec{V}_p) \times \vec{N}_0, \gamma = (q_0 - p_0) \times \vec{L} + (\vec{V}_q - \vec{V}_p) \times \vec{D}, \omega = (\vec{V}_q - \vec{V}_p) \times \vec{L}$, $\vec{L} = (\vec{V}_b - \vec{V}_a) \times (\vec{V}_c - \vec{V}_a)$, and $\vec{D} = \vec{N}_1 - \vec{N}_0 - \vec{L}$.

Proof:

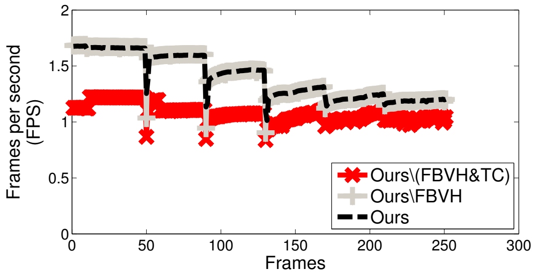

Figure 12: *This figure shows frame rate graphs of CCD for the dragon benchmark with our method (**Ours**), and **Ours****FBVH** that does not use our fast BVH construction method, but uses the median-based partitioning. It also shows another graph of CCD with **Ours****(FBVH&TC)** that does not use our traversal cost metric, but uses the LM metric. By using our fast BVH construction method, we further ameliorate the performance degradations at fracturing events.*

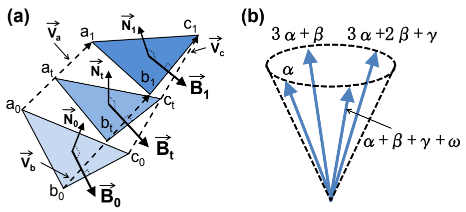

Figure 13: ***Continuous Binormal Cone:*** *This figure shows (a) a deforming triangle in a time interval $[0,1]$ and (b) one of three binormal cones for edges of the triangle computed from four coefficients of a cubic Bezier curve.*

The surface normal vector $\vec{N}_t$ of the triangle T at time t is given by following equation [TCYM08] :

$$\begin{aligned}\vec{N}_t &= \vec{N}_0 B_0^2(t) + (\vec{N}_0 + \vec{N}_1 - \vec{L})/2 B_1^2(t) + \vec{N}_1 B_2^2(t)\\ &= (1-t)^2 \vec{N}_0 + t(1-t)(\vec{N}_0 + \vec{N}_1 - \vec{L}) + t^2 \vec{N}_1\\ &= \vec{N}_0 + t(\vec{N}_1 - \vec{N}_0 - \vec{L}) + t^2 \vec{L}\end{aligned}$$

The binormal vector $\vec{B}_t$ for an edge (p,q) at time t is

$$\begin{aligned}\vec{B}_t &= (q_t - p_t) \times \vec{N}_t\\ &= [(q_0 + t\vec{V}_q) - (p_0 + t\vec{V}_p)] \times \vec{N}_t\\ &= [(q_0 - p_0) + t(\vec{V}_q - \vec{V}_p)] \times \vec{N}_t\\ &= (q_0 - p_0) \times \vec{N}_t + t(\vec{V}_q - \vec{V}_p) \times \vec{N}_t\end{aligned}$$

By replacing $\vec{N}_t$ with $\vec{N}_0 + t(\vec{N}_1 - \vec{N}_0 - \vec{L}) + t^2\vec{L}$, we obtain:

$$\begin{aligned}\vec{B}_t &= (q_0 - p_0) \times [\vec{N}_0 + t(\vec{N}_1 - \vec{N}_0 - \vec{L}) + t^2 \vec{L}]\\ &\quad + t(\vec{V}_q - \vec{V}_p) \times [\vec{N}_0 + t(\vec{N}_1 - \vec{N}_0 - \vec{L}) + t^2 \vec{L}]\\ &= (q_0 - p_0) \times \vec{N}_0\\ &\quad + t[(q_0 - p_0) \times (\vec{N}_1 - \vec{N}_0 - \vec{L}) + (\vec{V}_q - \vec{V}_p) \times \vec{N}_0]\\ &\quad + t^2[(q_0 - p_0) \times \vec{L} + (\vec{V}_q - \vec{V}_p) \times (\vec{N}_1 - \vec{N}_0 - \vec{L})]\\ &\quad + t^3(\vec{V}_q - \vec{V}_p) \times \vec{L}\end{aligned}$$

We define $\vec{D}$ as $\vec{D} = (\vec{N}_1 - \vec{N}_0 - \vec{L})$

By plugging $\vec{D}$, we obtain:

$$\begin{aligned}\vec{B}_t &= (q_0 - p_0) \times \vec{N}_0\\ &\quad + t[(q_0 - p_0) \times \vec{D} + (\vec{V}_q - \vec{V}_p) \times \vec{N}_0]\\ &\quad + t^2[(q_0 - p_0) \times \vec{L} + (\vec{V}_q - \vec{V}_p) \times \vec{D}]\\ &\quad + t^3(\vec{V}_q - \vec{V}_p) \times \vec{L}\end{aligned}$$

To bound $\vec{B}_t$ using Bernstein polynomials of degree three, we form following equation:

$$\vec{B}_t = x_0 B_0^3(t) + x_1 B_1^3(t) + x_2 B_2^3(t) + x_3 B_3^3(t)$$

By solving above equation, we obtain:

$x_0 = \alpha$

$x_1 = (3\alpha + \beta)/3$

$x_2 = (3\alpha + 2\beta + \gamma)/3$

$x_3 = \alpha + \beta + \gamma + \omega$

where,

$\alpha = (q_0 - p_0) \times \vec{N}_0$

$\beta = (q_0 - p_0) \times \vec{D} + (\vec{V}_q - \vec{V}_p) \times \vec{N}_0$

$\gamma = (q_0 - p_0) \times \vec{L} + (\vec{V}_q - \vec{V}_p) \times \vec{D}$

$\omega = (\vec{V}_q - \vec{V}_p) \times \vec{L}$

$\square$

References

[Pro97] Provot X.: Collision and self-collision handling in cloth model dedicated to design garment. *Graphics Interface* (1997), 177–189. 1

[TCYM08] Tang M., Curtis S., Yoon S.-E., Manocha D.: Interactive continuous collision detection between deformable models using connectivity-based culling. *ACM Symp. on Solid and Physical Modeling* (2008), 25 – 36. 1, 2