Fast Proximity Computation Among Deformable Models Using Discrete Voronoi Diagrams: Implementation Details

Avneesh Sud Naga Govindaraju Russell Gayle Ilknur Kabul Dinesh Manocha
Department of Computer Science
University of North Carolina at Chapel Hill
http://gamma.cs.unc.edu/DVD

1 Introduction
We provide implementation details of the paper “Fast Proximity Computation Among Deformable Models using Discrete Voronoi Diagrams” [Sud et al. 2006b]. We describe the hierarchical data structures used in scene representation. We also discuss optimizations for efficient computation of the second order discrete Voronoi diagram (DVD) on the GPU. Our algorithm is implemented using Visual C++, with OpenGL graphics API and Cg language for GPU programming.

2 Scene Representation
The scene is represented as a list of objects. Each object stores the mesh geometry along with upper and lower bounds on the separation distance. Each object is decomposed into a set of features (prisms, triangles, edges and vertices). A feature is the basic primitive used for computing the distance field (section 3). We maintain sorted lists of intervals corresponding to the projection of an object’s AABB along each axis. The intervals are used for computation of conservative bounds on the separation distance in the first stage of our algorithm. For each object we also maintain an AABB hierarchy of the triangles. The AABB hierarchy is updated in parallel to GPU based DVD computation, and is used to efficiently prune exact triangle-triangle distance tests. We do not explicitly store the feature level potential neighbor set for each object. Instead the minimum separation distance and pairs of closest features are updated during hierarchy traversal.

Penetration Depth queries: The N-body distance query is restricted to features corresponding to set of overlapping triangles. First we compute intersecting triangles using AABB hierarchies, then perform a local walk to compute overlapping triangles.

Continuous Collision Detection queries: We compute tight bounding prisms enclosing the swept volumes of each primitive. We then use the bounding prisms as features in performing the N-body distance queries.

3 GPU-Based DVD Computation
The second order DVD is computed on a uniform 3D grid, represented as a flat 2D render texture. The precision of the render texture is determined by maximum number of sites. The DVD is computed only at pixels that lie on an object using 2 rendering passes. In the first pass, we scan convert the triangles into the red channel of the grid, giving the 1st order DVD. In the second pass, we perform distance field computations [Sud et al. 2006a]. The 2nd order DVD is concurrently computed in the green channel.

Inter-object queries: During scan conversion, the id of each object is stored in the stencil buffer. During distance field computation, the reference value and function for the stencil test are set to discard the fragment if the current object id is equal to value in stencil buffer.

Intra-object queries: The list of adjacent feature ids is stored in an adjacency texture. During distance field computation, a dependent texture lookup is performed to query this list, and the fragment is discarded if the current feature id is present in the adjacency list.

References
