To appear in ACM SIGGRAPH, Conference Abstracts and Applications, page 213, 2001.

k-DOPs as Tighter Bounding Volumes for Better Occlusion Performance

Dirk Bartz, WSI/GRIS, University of Tübingen, bartz@gris.uni-tuebingen.de
James T. Klosowski, IBM T. J. Watson Research Center, jklosow@us.ibm.com
Dirk Staneker, WSI/GRIS, University of Tübingen, dstaneke@gris.uni-tuebingen.de

Introduction

Bounding volumes are used in computer graphics to approximate the actual geometric shape of an object in a scene. The main intention is to reduce the costs associated with interference tests such as intersection tests in ray tracing and collision detection, and to determine the visibility of an object. The bounding volumes most commonly used for these purposes are axis-aligned bounding boxes (AABB). However, in many cases this approximation fills a much larger volume in object space, and a much larger screen area - once rasterized into screen space - than the actual geometry. This results in *false positive* interference results which can increase the computational load significantly. Alternatively, oriented bounding boxes (OBB) were proposed, where the spanning axes of the bounding box are oriented according to the shape of the object, thus generating a tighter approximation of the original shape than AABBs. While OBBs perform better for collision detection than AABBs, the benefits for occlusion culling are significantly smaller. This is mainly due to the fact that the rasterized screen-area of an OBB is almost the same as for an AABB, and that the corners of an OBB still protrude through exterior hull elements, which are occluding the actual geometry. Another commonly used bounding volume primitive are spheres, which have also used for ray tracing and collision detection, since the intersection with a sphere is very easy to compute. However, a sufficiently tessellated sphere requires many polygons which increase the costs for an image-space occlusion culling interference test. Furthermore, spheres tend only to approximate compact objects well. Convex hulls are also good bounding primitives, but they are significantly more expensive to compute than other bounding volumes[1] and quickly become impractical in design tasks, where model objects are modified frequently. In 1996, Klosowski et al. [1] proposed a collision detection scheme using discrete orientation polytopes (*k*-dops) which enabled faster collision tests than OBBs. Essentially, *k*-dops are an approximation of an object by computing bounding planes of an object along *k*/2 directions [2]. An AABB is one example of a 6-dop, whose bounding planes correspond to the coordinate axes. Another common *k*-dop is the 26-dop, which is an AABB with the twelve edges and eight corners cut to the object's surface (6 + 12 + 8 = 26 bounding planes).

Experiments

In our experiments, we employed an image-space occlusion culling test using the Hewlett-Packard occlusion culling flag, implemented on the HP fx-series of graphics subsystems. This flag determines if geometry rendered during a special occlusion mode would modify the depth buffer, which indicates potential visibility. In other words, if a bounding volume is rendered but the HP occlusion culling flag indicates that the depth buffer would not have changed, then we need not render any of the geometry contained within that bounding volume. We use this flag on a depth-sorted list of objects of the tested models, which are located in the view frustum.

We tested a variety of "real world" MCAD datasets using AABBs and *k*-dops. On average, we achieved a 50% improvement of the culling rate using *k*-dops, instead of AABBs. The interior objects of MCAD datasets are frequently occluded by exterior hood or cover objects, like the hull of the servo screwdriver in Figure 1. However, AABB do not provide a very tight approximation for rounded shapes. Hence, they frequently extend through the hull objects and generate false positive visibility test results. In contrast, *k*-dops provide a much tighter approximation, where corners of the respective AABB have been cut off.

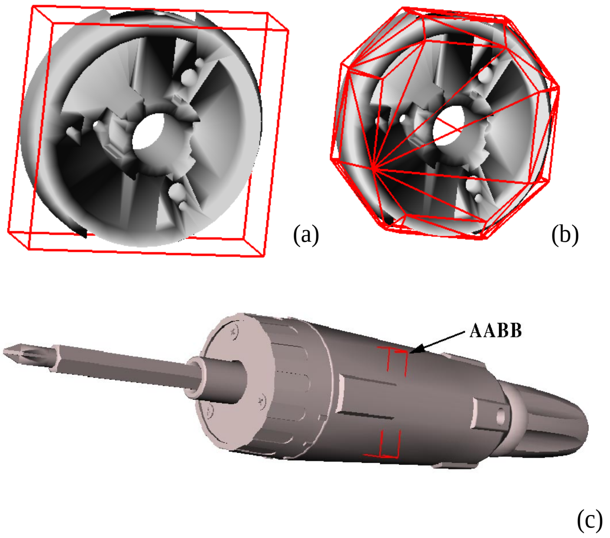

Figure 1: Servo screwdriver: (a) motor part and AABB; (b) motor part and 26-dop; (c) AABB of motor part is visible through hull, while 26-dop is completely occluded by hull.

The polygonal complexity of a *k*-dop is naturally larger than the complexity of an AABB; if *k*=26, up to 26 polygons are used for a *k*-dop, while only six polygons are needed for an AABB. However, an occlusion culling query requires an update or synchronization of the visibility information which is a pipeline flush for HP occlusion culling flag based approaches. The latency of the pipeline flush is equivalent to the rendering of approximately 190 triangles of an average size [3]. If the graphics subsystem does not provide hardware support for such queries, this latency is even larger. Experiments with specific polygonal models where *k*-dops do not facilitate a higher culling rate provide evidence for this statement, since the higher overhead of rendering three times more polygons for the occlusion test is not reflected in a lower framerate. In fact, the framerate did not change much beyond the limits of measurement noise.

Overall, *k*-dops provide tight bounding volumes for polygonal objects. They significantly reduce false positive visibility queries in comparison to AABBs. The increased rendering costs due to the higher polygonal complexity of the *k*-dops are overshadowed by the latency of the required synchronization step of state-of-the-art graphics subsystems.

References

1 M. Held, J. Klosowski and J.Mitchell. Real-time Collision Detection for Motion Simulation Within Complex Environments. In *Visual Proc. of ACM SIGGRAPH*, page 151, 1996.

2 T. Kay and J. Kajiya. Ray Tracing Complex Scenes. In *Proc. of ACM SIGGRAPH*, pages 269-278, 1986.

3 K. Severson. VISUALIZE fx Graphics Accelerator Hardware, Hewlett-Packard Company Whitepaper, 1999.

---

[1] Note that bounding volumes do not need to be convex, they only need to completely contain the objects.