Details of Ray/OBB Intersection

Wildfire Games – `http://wildfiregames.com/0ad/`

July 1, 2012

1 Definitions

A ray is defined as a collection of points $\mathbf{r}(t)$ such that:

$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$$

where $\mathbf{o}$ denotes the origin of the ray, $\mathbf{d}$ is a unit vector indicating the direction of the ray, and t is an independent scalar parameter that specifies the distance of $\mathbf{r}(t)$ from the origin of the ray. Points that are generated by negative values of t are said to lie behind the ray origin, and are not considered part of the ray.

An OBB, or Oriented Bounding Box, is an arbitarily-rotated three-dimensional rectangular cuboid defined by a center point $\mathbf{a}_c$, a set of mutually orthonormal basis vectors $(\mathbf{a}_u, \mathbf{a}_v, \mathbf{a}_w)$, and the half-distances (h_u, h_v, h_w) of each face from the origin along their respective axes.

2 Method

In 0 A.D., rays are tested for intersection with OBBs using the slab method as outlined in [1]. In brief, the goal of the algorithm is to compute the distances from the ray's origin to its intersection points with each of three slabs (one for each dimension). By then performing a clever comparison of these intersection point distances, it is able to determine whether the ray hits or misses the shape.

This document is concerned with providing some details of how these distances are computed, in order to allow the reader to more thoroughly comprehend the algorithm. Before continuing, the reader should have the algorithm as it appears in [1] at hand. For the most part, the same variable names will be used here.

Figure 1 presents a visual illustration of a sample 2D case, for one particular slab. The points $\mathbf{t}_1$ and $\mathbf{t}_2$ are the intersection points of the ray with the slab. The goal is to determine the distances between $\mathbf{o}$ and these points in the general case. Observe that each basis vector $\mathbf{a}_i$ corresponds to a slab of which it is the normal vector.

As in the algorithm, in what follows, let:

$$\begin{aligned} f &\overset{\Delta}{=} \mathbf{a}_i \cdot \mathbf{d} = \cos\theta \\ e &\overset{\Delta}{=} \mathbf{a}_i \cdot \mathbf{p} \end{aligned}$$

Both $\mathbf{a}_i$ and $\mathbf{d}$ are unit vectors; therefore by definition of the dot product, $f = \cos\theta$ where θ is the angle between the two; i.e., the ray direction and the slab normal. The scalar e is the distance between $\mathbf{o}$ and the center of the box $\mathbf{a_c}$, projected onto the slab normal $\mathbf{a}_i$.

In a right-angled triangle, per definition of the cosine, the hypotenuse's length is $\cos\theta$ times the length of the adjacent side:

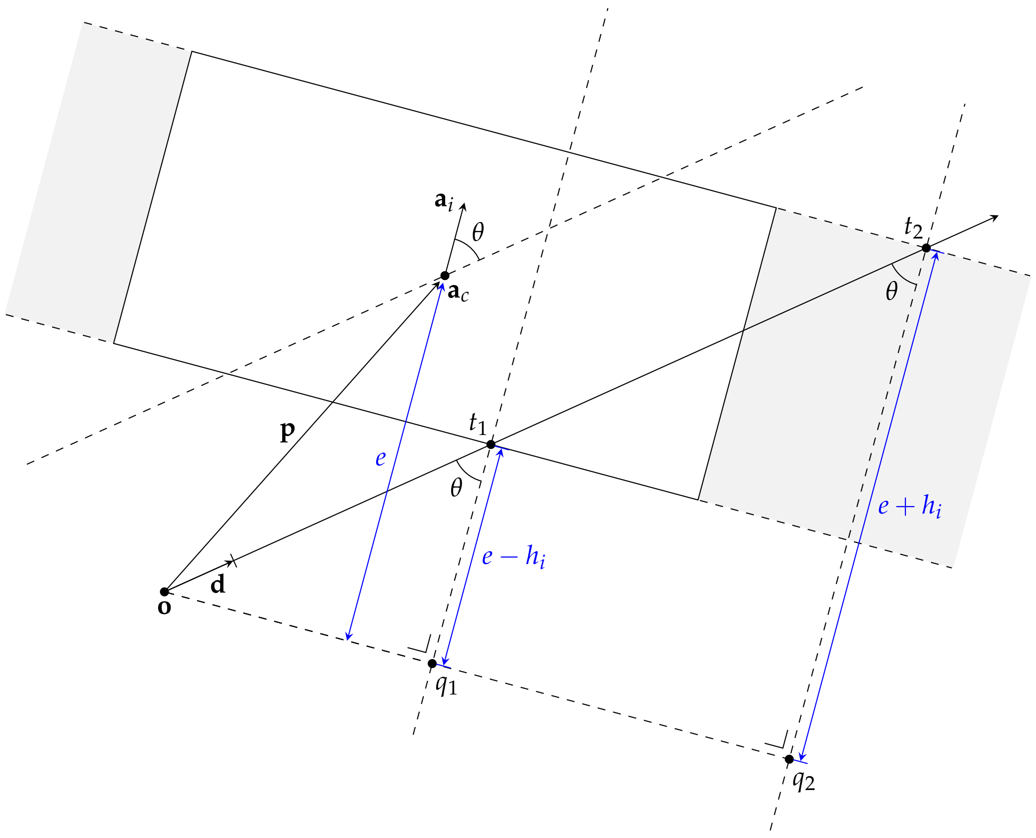
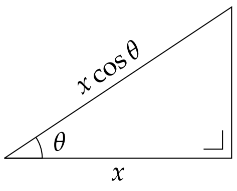


Figure 1: A visual representation of how the distances between **o** and $\mathbf{t}_1$ and $\mathbf{t}_2$ are computed in the two-dimensional case, for a generic slab $i \in \{u, v\}$ with normal vector $\mathbf{a}_i$ and half-size h_i.



Thus, looking at the right-angled triangles $\Delta\mathbf{o}q_1t_1$ and $\Delta\mathbf{o}q_2t_2$ in Figure 1, we find that:

$$\begin{aligned} ||\mathbf{o} - \mathbf{t_1}|| &= \frac{e - h_i}{\cos\theta} \\ ||\mathbf{o} - \mathbf{t_2}|| &= \frac{e + h_i}{\cos\theta} \end{aligned}$$

3 Degenerate cases

It's prudent to consider what happens for the degenerate case of testing a ray for intersections against an OBB that has no size in one or more dimensions. Figure 2 depicts the degenerate variant of the 2D case seen earlier, where the box now has a null size in the u dimension.

In the method seen above, the only prerequisites to computing the distances between the ray origin and its intersection points with the slab are $\cos\theta$ and $e \pm h_i$. In the degenerate case, h_u is 0, and so both intersection points will be correctly found at $e/\cos\theta$. None of the other terms depend upon the size of the box along the dimension at hand, and hence remain unaffected. However, they do depend on the degenerate dimension's normal vector remaining well-defined and non-null.

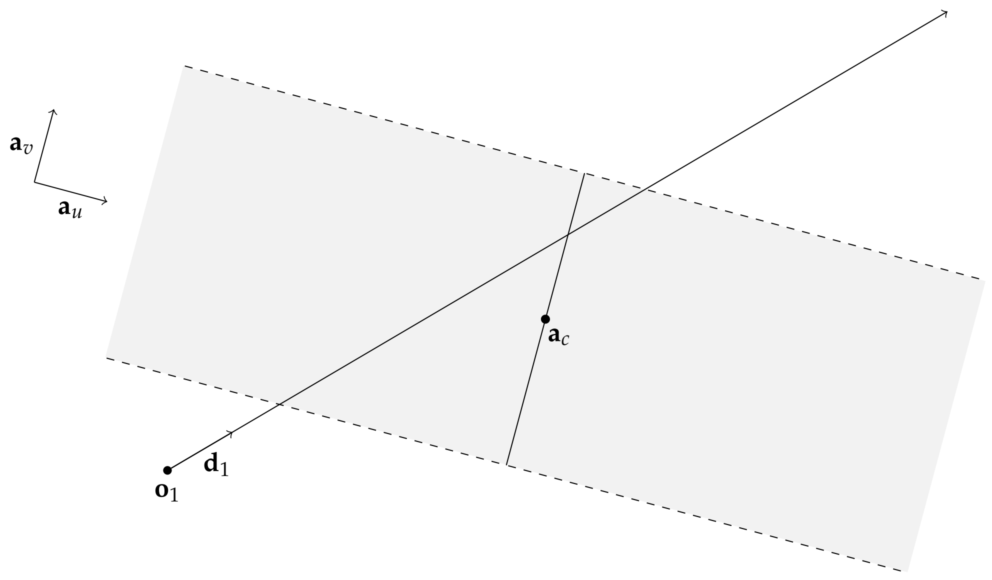


Figure 2: Degenerate 2D case with null size in the u dimension.

References

[1] Tomas Akenine-Möller, Eric Haines, and Nathy Hoffman. *Real-Time Rendering, Third Edition*. A K Peters, Ltd., 2008. ISBN 978-1-56881-424-7.