Thursday, March 25, 2004

Prashant Thakare <thakare@cc.gatech.edu>
Christopher Henke <bigqtip@cc.gatech.edu>

Project 3: Due on paper in class on April 20
Web page can be used as an add-on to show extra images/results.

Interference:

Boxes A and B intersect if and only if:

If such test is used over and over, an optimized version is required. With such optimized test, interference tests of complicated objects can be approximated with boxes around the surface.

Note that the boxes can overlap. With such box tests, we have a tradeoff between size of boxes used and the speed of tests. While smaller boxes approximate the surfaces better, too many of them are required. Larger boxes might omit details.

Fast ways to quickly reject/test box intersections:
- projection on planes
- project on lines
- Edge/Face intersections (all pairs?)
- Separating surfaces. Normal
Point in parallelogram (axis aligned)
• Stencil projections

Point in a box test:
Box Representation: Origin, 3 vectors

I, J, K are pair wise orthogonal.
Alternatively, with nine degrees of freedom, same box can be represented using an origin and three scalars viz: width, height and length

Is point P inside the box? Which of the following is a good choice?
dot product
cross product

Inbox(P) is defined as
\[ 0 < OP.I < I^2 \quad \text{AND} \quad 0 < OP.J < J^2 \quad \text{AND} \quad 0 < OP.K < K^2 \]

If a vertex of one box is inside the other box, then there is interference.

Edge/Face intersections
For each edge of A do
For each face of B do
    If edge intersects face then return hit
Reverse roles of A and B

Projection on planes

Idea 1
Project both boxes on 3 planes (xy, yz, zx). If the two projections are disjoint on any plane, then the boxes are disjoint. Otherwise, the two boxes intersect.

If B's shadow is found on 3 differently oriented faces of A, then do we have an intersection of A and B? NO!

Counter-example
Have the corner of A really close to the center of a face of B. The three projections show hits but the two solids do not interfere.
Idea 2

P(A) is defined as projection of A
Let S = set of directions

If P(A) intersect P(B) is NULL
  For any direction in S, A intersect B is NULL
  If for all directions the two projections intersect then A intersects B

S' = {I, J, K, I', J', K'} is insufficient for boxes

What's a sufficient set S for boxes?

S = {
  I, J, K,
  I', J', K',
  IxI', IxJ', IxK',
  JxI', JxJ', JxK',
  KxI', KxJ', KxK'
}

Testing interference with directions and projections

A projection interval on direction D can be calculated as $SV_D$. Two boxes can be said to intersect if the intervals of their projections intersect.

Using Graphics hardware
  update the z-buffer using $>$, $<$, $>=$, or $<=$ tests. Additionally use callbacks if the tests succeed.
If for a pixel $p$, $A$ intersect $B$ have the same value then they intersect. Using this, an intersection can be found by rendering $A$ and $B$ multiple times with following combination.

render front $A$ with front $B$
render front $A$ with back $B$
render back $A$ with front $B$
render back $A$ with back $B$

Parity Test

- Place a point $P$
- Shoot a ray from $P$ in any direction
- The parity of the number of times that the state toggles tells whether $p$ is in or out

Above tests can be used for intersections by rendering front faces of $A$ and then testing how many times $B$ is behind $A$, repeating for back faces of $A$.

Does this technique work for objects more complex than boxes (concave objects)? In general, no. There is an extension that does work though.

Number of surfaces are need to be compared as the front and back corresponds to frontmost and backmost surfaces. This is related to the depth complexity of the object. Depth complexity can be obtained looking at the object from several directions and finding the visible surfaces.