Prof. G. Zachmann
P. Lange

University of Bremen
School of Computer Science
CGVR Group
May 15, 2014

Summer Semester 2014

Assignment on Advanced Computer Graphics - Sheet 3

Due Date 29. 05. 2014

Exercise 1 (Ray Casting Fan Shot, *5 Credits*)

Imagine a ray casting "fan shot", defined by a start vector (with starting point S and direction d) and angle α:

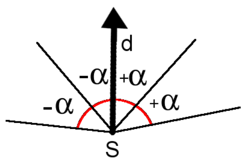

Figure 1: Fan shot view for one axis.

The fan shot generates an equally distributed field of rays (see Figure 2), starting from the start vector, for the x and y axis (applying the above illustrated scheme), whereas z shall be the pointing direction:

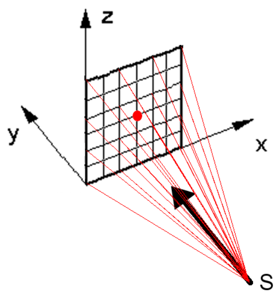

Figure 2: Complete fan shot.

The fan shot shall be limited for each axis by a total of 90 degree.

1. Give a pseudo code, which describes the generation of the fan shot rays
2. Implement your algorithm in C++. You can use the math template classes from the previous framework
3. Test your implementation with the following test cases and document your results:
   Start vector with $S = (0, 0, 0)$, $d = (0, 0, 1)$, $\alpha = 45$
   Start vector with $S = (0, 0, 0)$, $d = (0, 0, 1)$, $\alpha = 30$

Exercise 2 (BVHs for Ray Tracing, *5 Credits*)

You implemented a simple raytracer with a recursive ray trace function, a camera, a phong lightning and intersection test. The scene objects are managed in a list and you test every ray against every object in the list.

Your tasks:

1. To accelerate the ray intersection test, we want to use a hierarchical data structure called bounding volume hierarchy (BVH). Propose a (good) algorithm (in your own words or pseudo code) which creates a BVH for ray tracing applications.
2. Consider the following two-dimensional scene:
   Triangle $A = (-3, 3)$ $B = (-1, 3)$ $C = (-3, 1)$
   Rectangle $A = (-2, 3)$ $B = (-1, 3)$ $C = (-2, 1)$ $D = (-1, 1)$
   Circle $r = 1$ $C = (2, 2)$
   Triangle $A = (0, 2)$ $B = (0, 0)$ $C = (2, 0)$
   Triangle $A = (1, 0)$ $B = (1, -1)$ $C = (0, -1)$
   Triangle $A = (0, -2)$ $B = (0, -3)$ $C = (-3, -3)$
   Quad $A = (2, -2)$ $B = (3, -2)$ $C = (2, -3)$ $D = (3, -3)$
3. Make a suitable two-dimensional sketch of the scene with bounding boxes
4. Build a BVH with your proposed algorithm (by hand - do not implement your algorithm) and give the resulting bounding volume tree. Argue why your bounding volume tree (respectively your proposed algorithm) is a good solution. (Remark: A good proposed algorithm has more advantages than a bad one)