

Summer Semester 2024

Assignment on Advanced Computer Graphics - Sheet 3

Due Date 22.05.2024

Exercise 1 (Acceleration Data Structures, 10+1 Credits)

In this assignment you should implement **one** of the following data structures to accelerate the raytracer provided in `AccelerationFramework`:

1. Uniform Grid (slides 12-16)
2. kD-Tree (slide 41-46, 49-66)
3. Octree (slide 37)
4. AABB-Tree (slide 70-95)
5. Sphere-Tree (slide 70-95)

(slide references are for lecture 03 - *Acceleration Data Structures*)

Each acceleration data structure (ADS) has a prototype class under `scene/accelerators/`, which you can extend.

All ADS are derived from the common base class `Accelerator`. You have to implement the constructor, the destructor, and the `intersect()` function, respectively, in order to make the ADS accessible for the Raytracer.

Inside the constructor, you have to construct the ADS. To do that, the scene will be passed to it in form of a `SurfaceList` object. The `SurfaceList` object contains a list with all geometric objects in the scene (accessed by `getSurfaces()`). Each `Surface` represents a geometric object, which all have a function `getAabb()` that returns an axis-aligned bounding box of the object. Probably, this feature will be helpful. But pay attention to objects of infinite size, like the `Plane` and `Checkerboards`. It might be wise to handle these separately.

The `intersect()` function of the acceleration data structures is called for the whole scene instead of the currently used `intersect()` from the `SurfaceList` class. As expected, it should compute the first intersection between the input ray and the closest object. In order to get the intersection with the actual geometric object in your ADS, you can use the existing `intersect()` for the geometric objects.

Finally, the destructor should free potentially allocated memory.

If you want to activate your ADS, you can simply choose the respective ADS in the dialogue of the raytracer's GUI.

You can load 3D triangle mesh files in `obj` format by using `meshfile` in the XML scene description. The scenes `Mesh:xyz` use various mesh files in the `obj` format.

Your solution is expected to have a speed up of around **10** for the **Mesh:Monkies**. The creation time of the ADS is not considered for the speed up, but it should still be in the range of interactive feedback (aim for less than 1 second).

Bonus: We will honor the fastest ADS (in terms of speedup) with 1 extra point, for the first and second place. The solution with the fastest render times for the **Mesh:Monkies** scene wins.

For reference, you could expect the following times for the **Mesh:Monkies** scene (with the default settings):

- Unaccelerated: 84.42s
- Uniform Grid: 2.53s (speedup: 33.37)
- AABB-Tree: 4.31s (speedup: 19.59)
- Sphere-Tree: 3.57s (speedup: 23.65)
- kD-Tree: 5.60s (speedup: 15.08)