

Aufgabe k-d-Baum

Die Anzahl Schnittpunkttests ist aufgrund des Brute-Force-Vorgehens zu hoch. Sie kann mit Hilfe räumlicher Unterteilungen reduziert werden.

Es soll wie folgt ein k-d-Baum aufgebaut werden:

- Der Baum wird für alle Dreiecke auf einmal erstellt, statt jedes Dreieck einzeln einzufügen.
- Die achsen-parallele Bounding-box der Szene wird am Anfang berechnet.
- Der k-d-Baum wird rekursiv aufgebaut. Die Bounding-box entlang der längsten Achse in der Mitte in eine "linke" und "rechte" Bounding-box aufgeteilt.
- Wenn ein Dreieck in beiden Boxen ist, wird es am Ende der Liste des aktuell betrachteten Knotens angehängt, ansonsten wird es zu einer "linken" oder "rechten" Liste am Ende angehängt. Diese beiden Listen werden anschließend rekursiv für den Aufbau des linken und rechten Teilbaums verwendet.
- Die Rekursion bricht ab, wenn die Anzahl einzufügender Dreiecke eine vorgegebene Schranke unterschreitet.

Die Suche nach einem Schnittpunkt soll ebenfalls rekursiv implementiert werden:

- Die Suche wird abgebrochen, wenn der Sehstrahl die Bounding-box des Knotens nicht schneidet.
- Falls ein linker oder rechter Teilbaum existiert, wird in diesen Teilbäumen rekursiv ein Schnittpunkt gesucht.
- Nach der Rekursion wird für den Sehstrahl ein Schnittpunkt für die Liste der Dreiecke beim Knoten gesucht.

Dabei soll die optimierte Variante des Schnittpunkttests verwendet werden. Der Parameter `minimum_t` für den bisher gefundenen Schnittpunkt muss genutzt werden, um die Schnittpunkttests frühzeitig abubrechen.

Ein Dreieck ist in einer Bounding-box enthalten, wenn einer der drei Eckpunkte enthalten ist. Ein Punkt ist in einem achsenparallelen Quader enthalten, wenn die Koordinaten des Punkts innerhalb des Intervalls der min-max-Koordinaten der zugehörigen Achse enthalten sind.

Teilaufgabe 1

Implementieren Sie die mit TODO gekennzeichneten Methoden in `kdtree.cc`. Fügen Sie keine weiteren Methoden oder Funktionen hinzu.

Teilaufgabe 2

Für die Zeitmessung soll immer der optimierte Schnittpunkttest verwendet werden.

Übersetzen Sie das Programm ohne k-d-Baum auf den Rechnern in E203 mit

```
g++ -Wall -pedantic -std=c++11 -march=native -mfpmath=sse -msse -O3  
-D OPTIMIZED_INTERSECTS raytracer.cc statistics.cc kdtree.cc
```

Messen Sie für `teapot.obj` mit 256 mal 256 Auflösung die Ausführungszeiten 5 mal mit dem übersetzten Programm und bilden Sie den Durchschnittswert.

Übersetzen Sie das Programm für den k-d-Baum auf den Rechnern in E203 mit

```
g++ -Wall -pedantic -std=c++11 -march=native -mfpmath=sse -msse -O3  
-D OPTIMIZED_INTERSECTS -D USE_KDTREE raytracer.cc statistics.cc  
kdtree.cc
```

und bilden Sie wieder den Durchschnittswert.

Teilaufgabe 3

Erweitern Sie Ihren Bericht mit einem Abschnitt zu dieser Aufgabe mit dem Quelltext Ihrer Funktionen und einer tabellarischen Gegenüberstellung der Ausführungszeiten sowie der Anzahl durchgeführter Schnittpunkttests und gefundener Schnittpunkte.

Bemerkungen

Beim Aufbau eines k-d-Baum für das Raytracing wird normalerweise nicht in der Mitte einer Achse aufgespalten. Auch nicht beim Median. Statt eines balancierten Baumes zur Erzeugen, wählt man die Trennebenen so, das ein großer leerer Raum (mit wenigen Dreiecken) und kleiner voller Raum (mit vielen Dreiecken) entsteht (surface area heuristic).

Die Grenzen der resultierenden zwei Bounding-Boxes können oft noch auf die Bounding-box der enthaltenen Dreiecke verkleinert werden.

Beim Suchen nach einem Schnittpunkt, kann mit Hilfe der im slab test gefundenen Parametern der Teilbaum, der dem Auge des Sehstrahls am nächsten ist, zuerst durchwandert werden. Falls dort ein Schnittpunkt gefunden wurde, braucht im anderen Teilbaum nicht mehr gesucht werden.

Dreiecke, die in beiden Teilbäumen liegen, werden nicht dem inneren Knoten zugeordnet, sondern im linken und rechten Teilbaum eingefügt. Um eine mehrfache Schnittpunktberechnung zu vermeiden, wird für ein Dreieck ein Flag verwendet, um eine zweite Berechnung zu verhindern. Alternativ wird das Dreieck in zwei oder drei Dreiecke an der Trennebene aufgeteilt.

Man kann mehrere Raytracingstrahlen zu Paketen zusammenfassen und gemeinsam durch den k-d-Baum propagieren (nicht rekursiv).