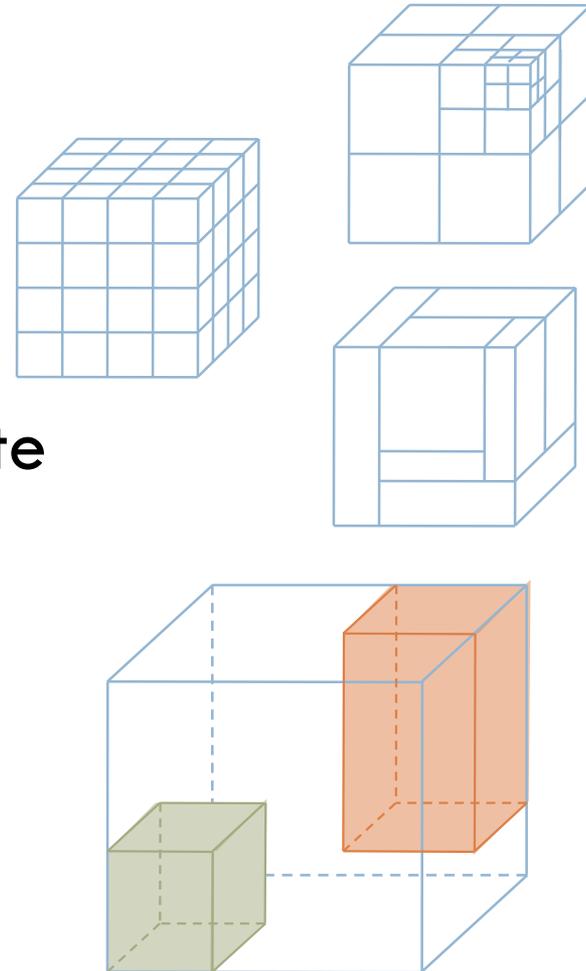


SIMD-RAYTRACING MITTELS SINGLE SLAB HIERARCHY

Vortrag zur Bachelorarbeit
Christian Woizschke
05. September 2008

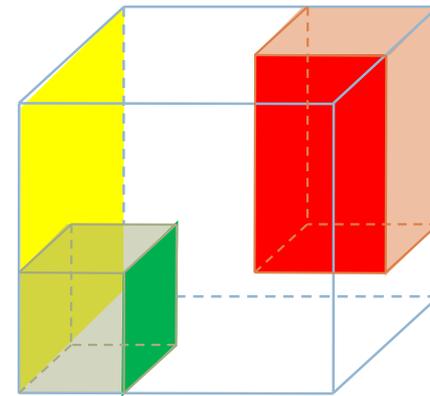
Beschleunigungsstrukturen

- **Aufteilung des Raums**
 - ▣ Uniform Grid
 - ▣ Octree
 - ▣ BSP / kd-Tree
- **Aufteilung der Geometrieliste**
 - ▣ Bounding Volume Hierarchy
 - ▣ Single Slab Hierarchy



Single Slab Hierarchy

- Beschleunigungsstruktur
- Abgeleitet von BVH
- Binärbaum
- Idee: AABB Seiten redundant
- Eine Seite der AABB
 - ▣ Als Ebene
- Schlechtere Approximation der Geometrie
 - ▣ Mehr Schnitttests zwischen Strahl und Knoten/Geometrie
- Schnellerer Schnitttest pro Knoten als bei BVH
- Traversierung mithilfe eines „aktiven Strahlintervalls“



Aufgaben

- Iterative Traversierung
- Walkthrough-Modus
- Beleuchtungseffekte
- Praktischer Vergleich der SSH mit einer BVH

Iterative Traversierung

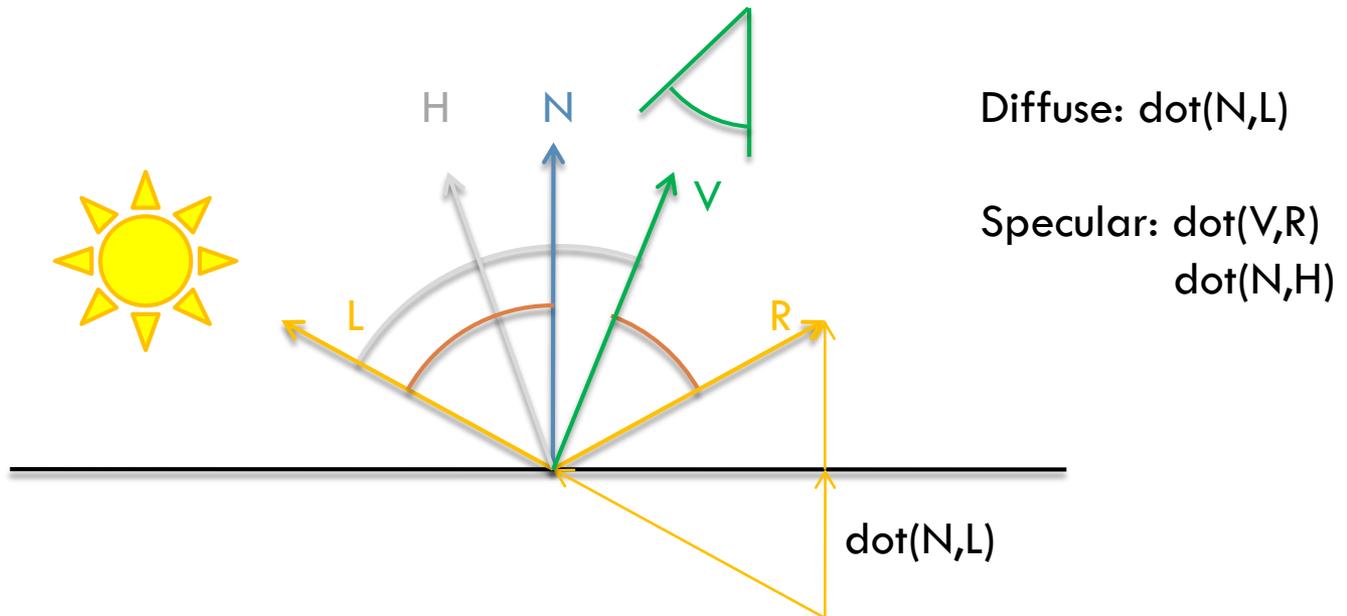
- Schleife
 - ▣ Schneide Strahl mit aktuellem Knoten
 - ▣ Durchlaufe Kindknoten 1 im nächsten Durchlauf
 - ▣ Setze Kindknoten 2 inkl. Strahlintervall auf einen Stack
 - ▣ Hole Knoten und Intervall vom Stack wenn
 - Aktueller Knoten nicht vom Strahl getroffen
 - Blattknoten erreicht und Dreiecke getestet
- Vorteile
 - ▣ Kein Overhead durch Funktionsaufrufe
 - ▣ Szenenkomplexität bzw. Höhe des Baums
 - Nicht mehr durch Größe des Call-Stacks begrenzt
 - durch Hauptspeicher begrenzt
- Nachteil
 - ▣ Software-Stack langsamer als Call-Stack

Walkthrough-Modus

- GLUT
- Verfahren zur Bildübertragung an Grafikkarte
 - ▣ glDrawPixels
 - ▣ glTexImage2D
 - Texture Quad
 - ▣ Pixel Buffer Object
 - glMapBuffer, glTexSubImage2D
 - Texture Quad
 - Schnellstes Verfahren

Phong

- Ambient, Diffuse, Specular
- SIMD
- Blinn



Reflexion und Transmission

- Snelliussches Brechungsgesetz

$$n_1 \cdot \sin(\alpha) = n_2 \cdot \sin(\beta)$$

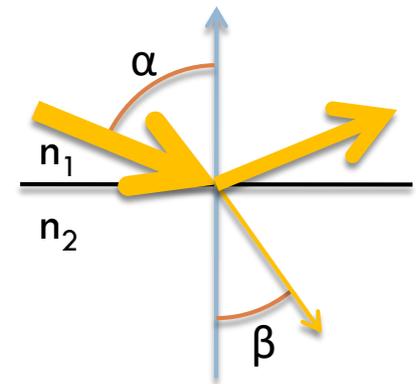
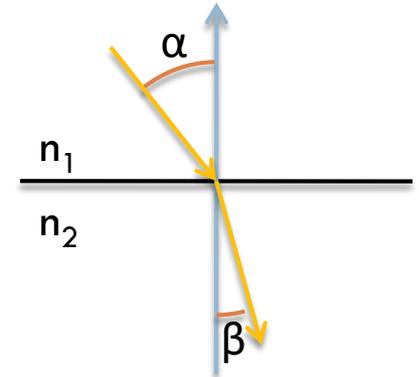
- Fresnel

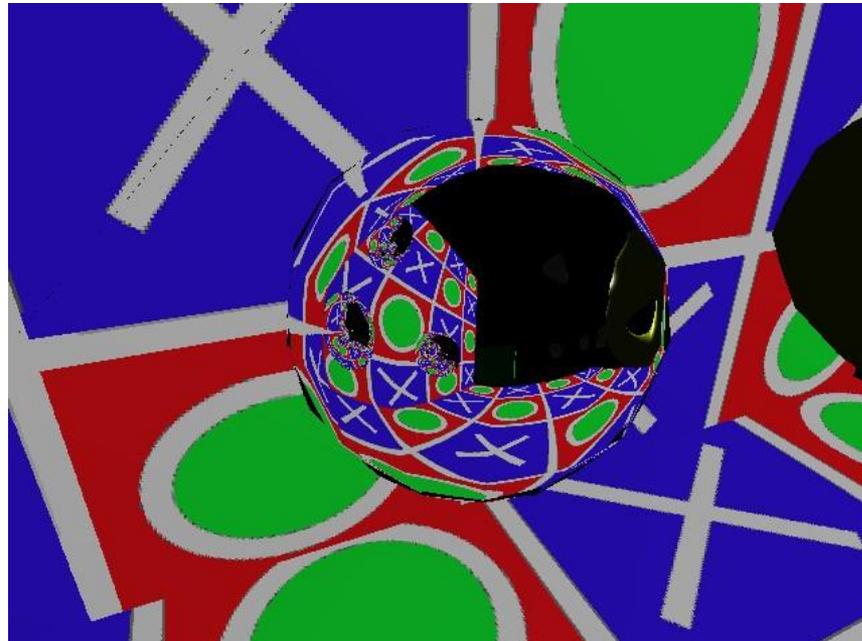
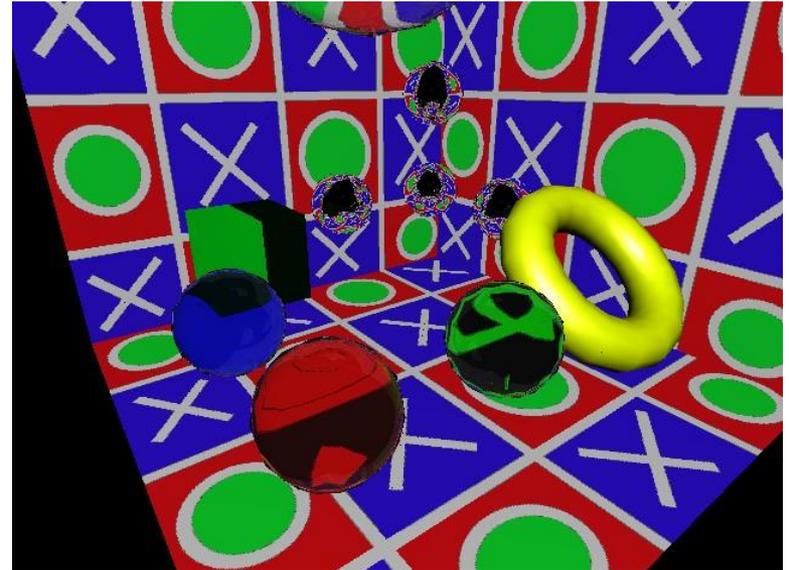
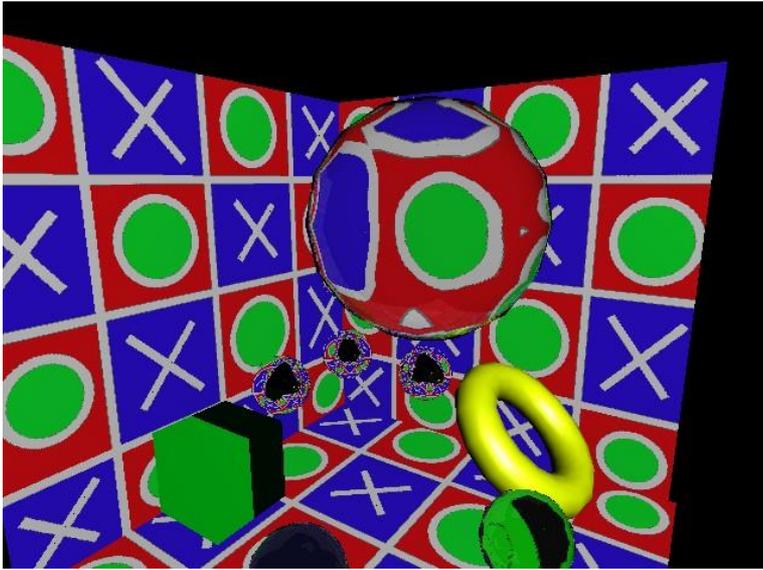
$$r_p = \frac{n_2 \cdot \cos(\alpha) - n_1 \cdot \cos(\beta)}{n_2 \cdot \cos(\alpha) + n_1 \cdot \cos(\beta)}$$

$$r_s = \frac{n_1 \cdot \cos(\alpha) - n_2 \cdot \cos(\beta)}{n_1 \cdot \cos(\alpha) + n_2 \cdot \cos(\beta)}$$

$$r = \frac{r_s + r_p}{2} \quad t = 1 - r$$

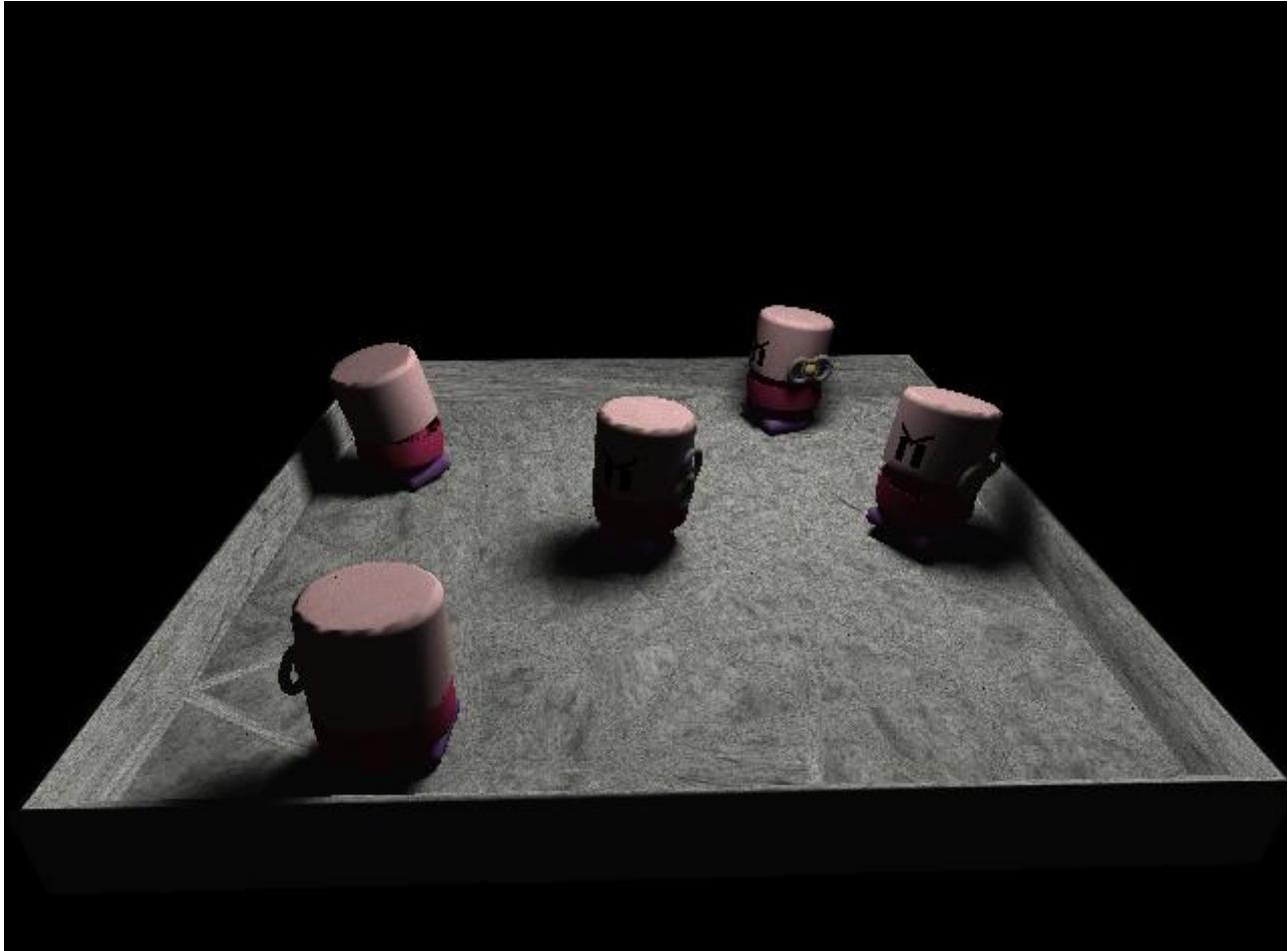
- SIMD





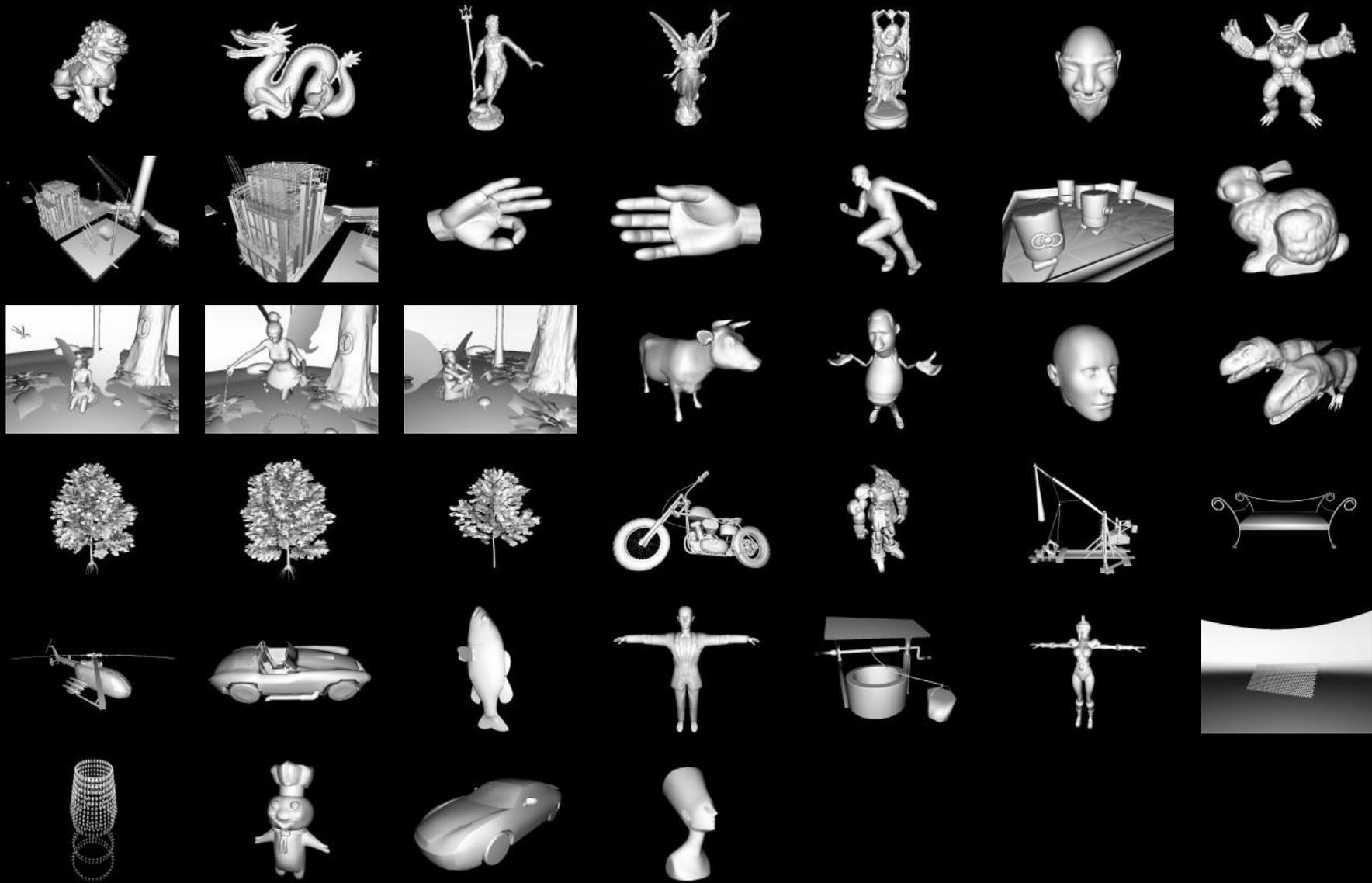
Licht

- Lichtquellen
 - Punktlicht
 - Paralleles Licht
 - Flächenlicht
 - Scheibe mit random sampling
 - Rechteck mit uniform sampling



Tests

- 27 Testszenen ausgewertet
 - 3500 bis 28mio Dreiecke
- Geordnete Traversierung
 - Knoten nahe der Strahlquelle (z.B. Kamera) zuerst
 - Abbruch wenn Strahlintervall hinter Schnittpunkt beginnt
- Optimiert durch den Compiler
 - O3
 - SSE
 - Prozessorarchitektur
- Kein Shading, nur Primärstrahlen
- Aktuelles System (Quad Core CPU, 4GB RAM)



Testergebnisse

- Geschwindigkeitsunterschied $\pm 30\%$, $\emptyset 0\%$
- Speicherverbrauch 28,6% bzw. 37,5%

```
struct SshNode
{
    float slab;
    unsigned long flagsAndChilds;
}
```

64bit:
4 Byte + 8 Byte = 12 Byte

32bit:
4 Byte + 4 Byte = 8 Byte

```
struct BvhNode
{
    float min[3];
    float max[3];
    unsigned long flagsAndChilds;
}
```

64bit:
6*4 Byte + 8 Byte = 32 Byte

32bit:
6*4 Byte + 4 Byte = 28 Byte

Testergebnisse

- Bei der SSH werden 1,5-2,7 Mal so viele Schnitttests (Strahl vs. Knoten) durchgeführt wie bei der BVH
 - ▣ Nicht 6 Mal so viele, obwohl nur $1/6$ der Informationen pro Knoten
 - ▣ Gründe:
 - Mindestens die Hälfte der AABB Seiten redundant
 - Oft wird der Baum bis zu den Blattknoten durchlaufen
 - Bei gleicher Anzahl an Knoten und gleicher Baumhöhe
- Bei der SSH werden in gleicher Zeit doppelt so viele Schnitttests (Strahl vs. Knoten) durchgeführt wie bei der BVH
 - ▣ Grund: Schnitttest besteht nicht nur aus Test gegen Ebene/Box

Fragen?

Vielen Dank für die Aufmerksamkeit

