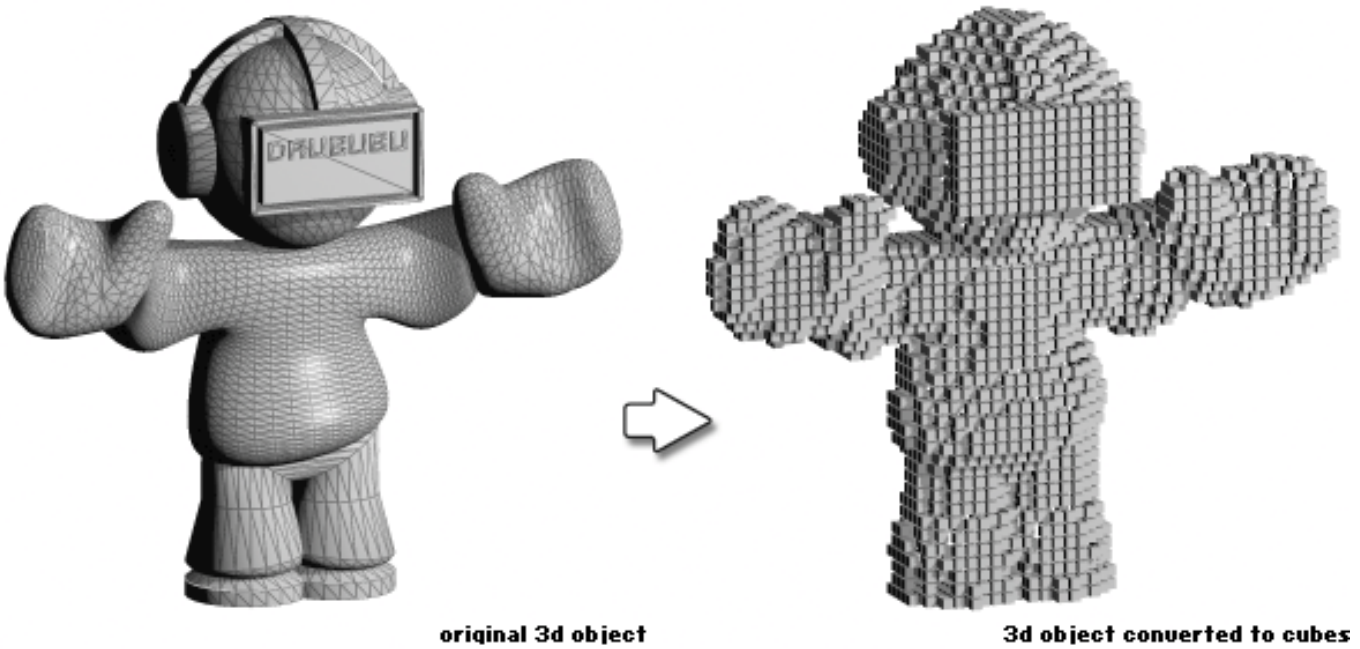
# GigaVoxels : Ray-Guided Streaming for Efficient and Detailed Voxel Rendering

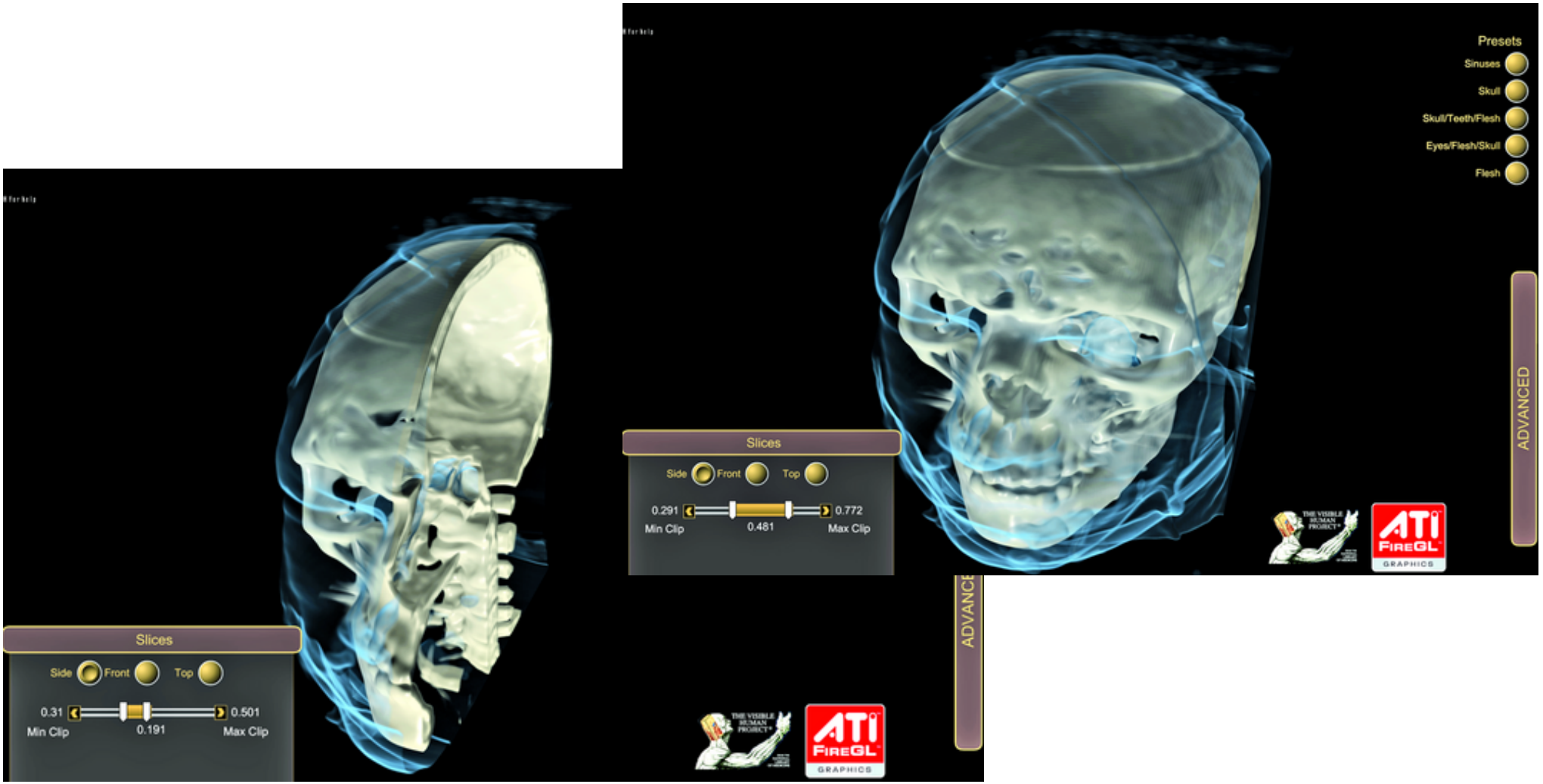
Jens Künemund
17.11.2009

# Voxel




Quelle: http://www.drububu.com/tutorial/images/voxel_pixelart.gif

# Einsatzgebiet von Voxeln



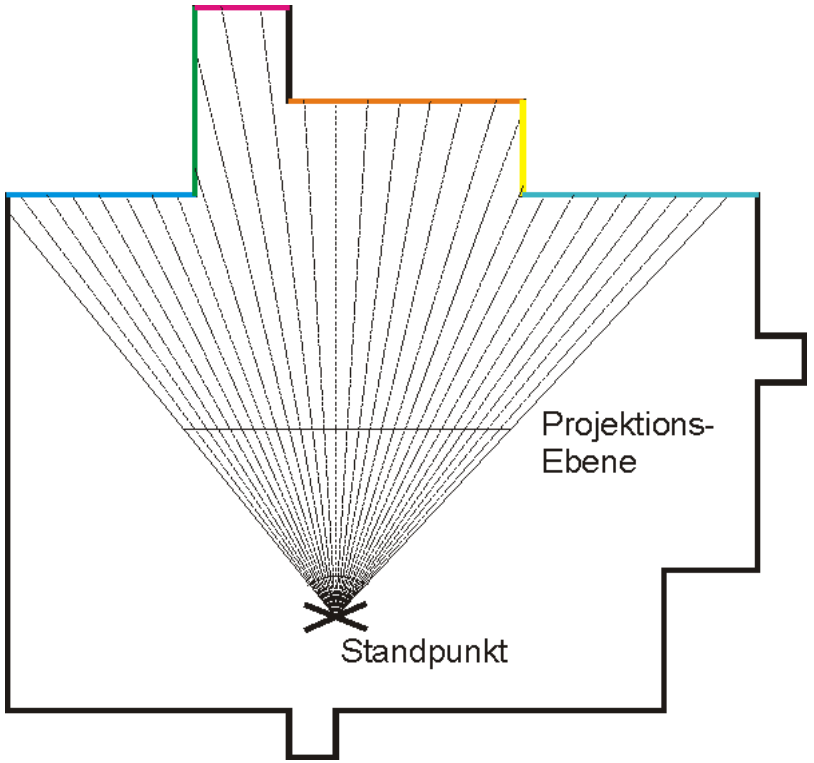

Quelle: http://www.pcgameshardware.de/aid,681315/
Voxel-Rendering-am-Beispiel-id-Tech-6-erlaeutert/
Technologie/Wissen/bildergalerie/?iid=1004200

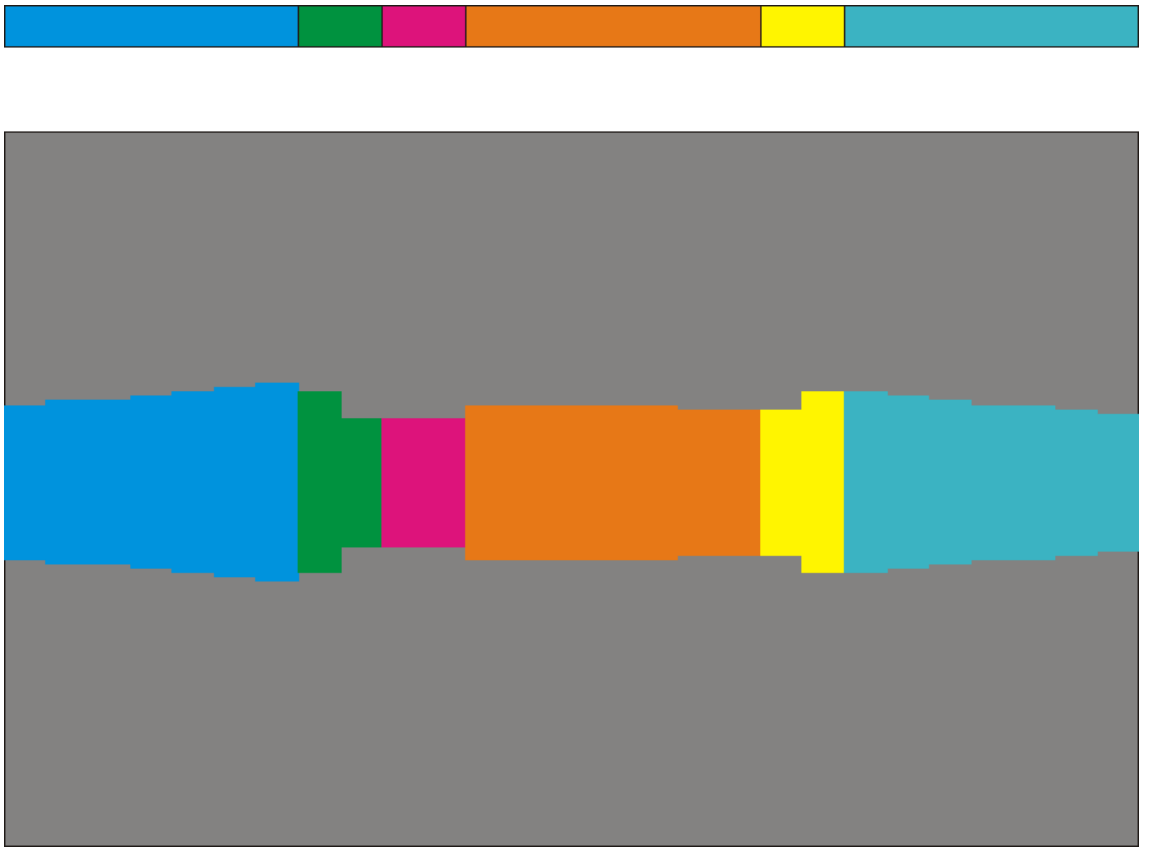
- Probleme bei voxelbasiertem Volumenrendern
  - Sehr aufwendiges Rendern

    → aufgrund der Datenmenge
  - GPU Speicherlimit

    → nur ein begrenzter Teil der Daten kann in der GPU vorgehalten werden

# Raycasting

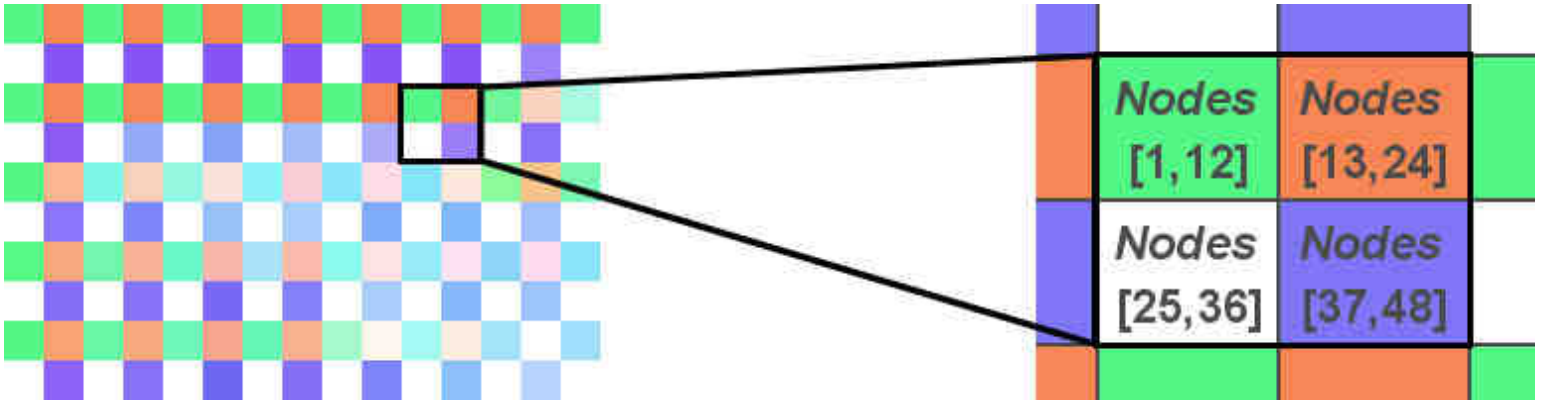


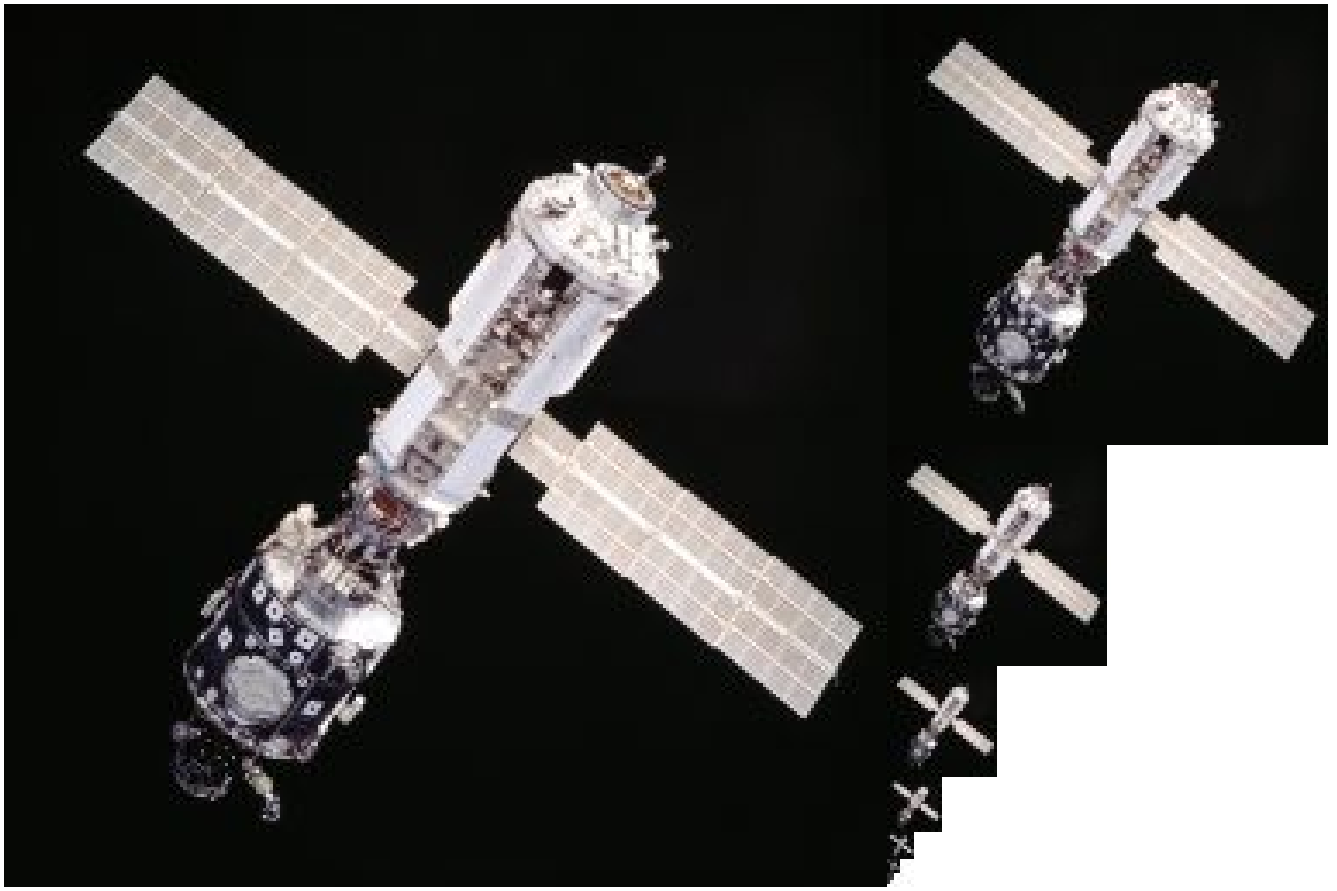
Quelle:
http://de.wikipedia.org/wiki/Raycasting

# Raycasting

- Level of Detail: Voxel pro Pixel
- Top-Down Suche innerhalb eines $N^3$ Baumes
- Unabhängige Strahlen
- Ausnutzen von Abhängigkeiten



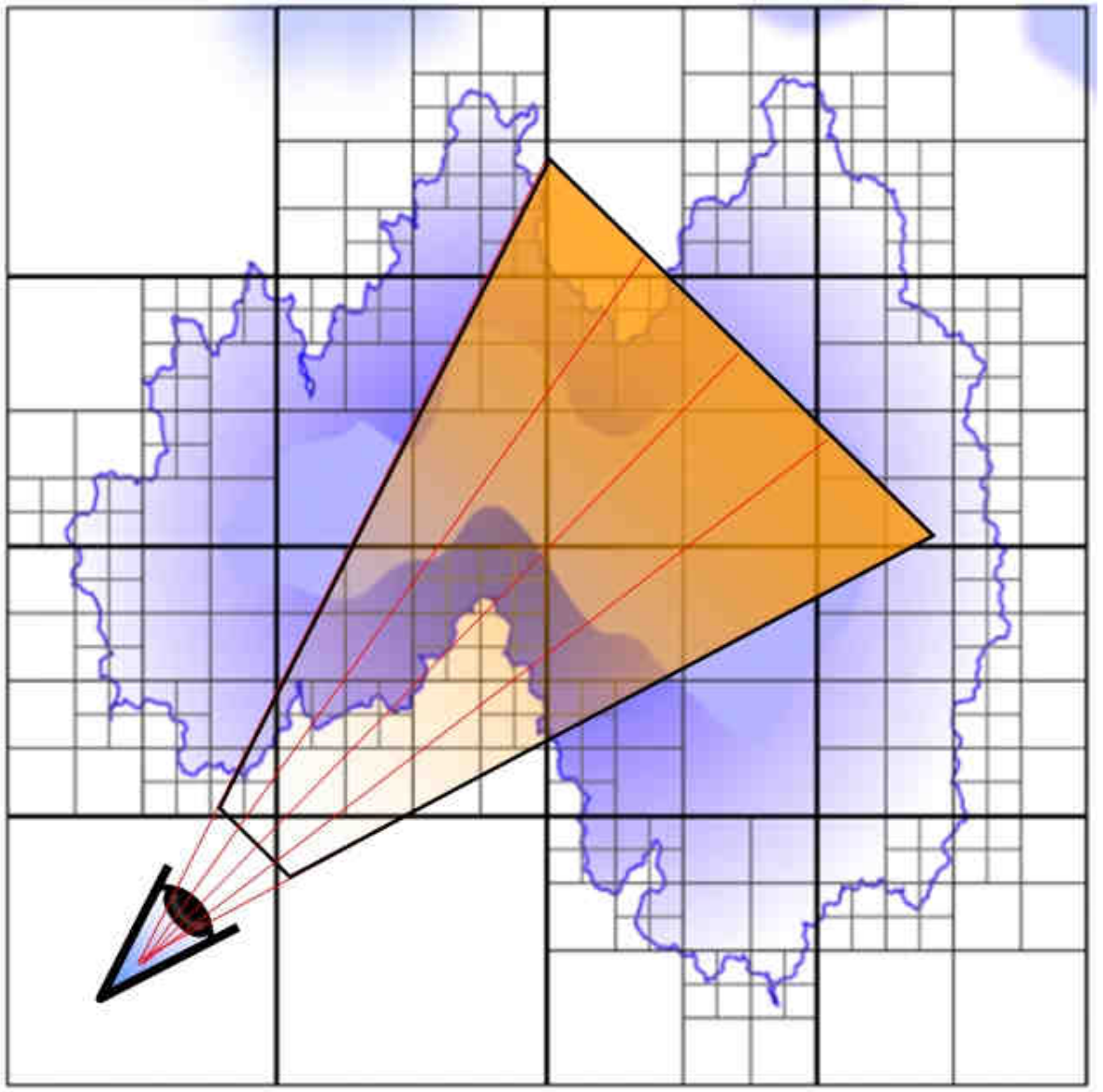
# Mipmapping

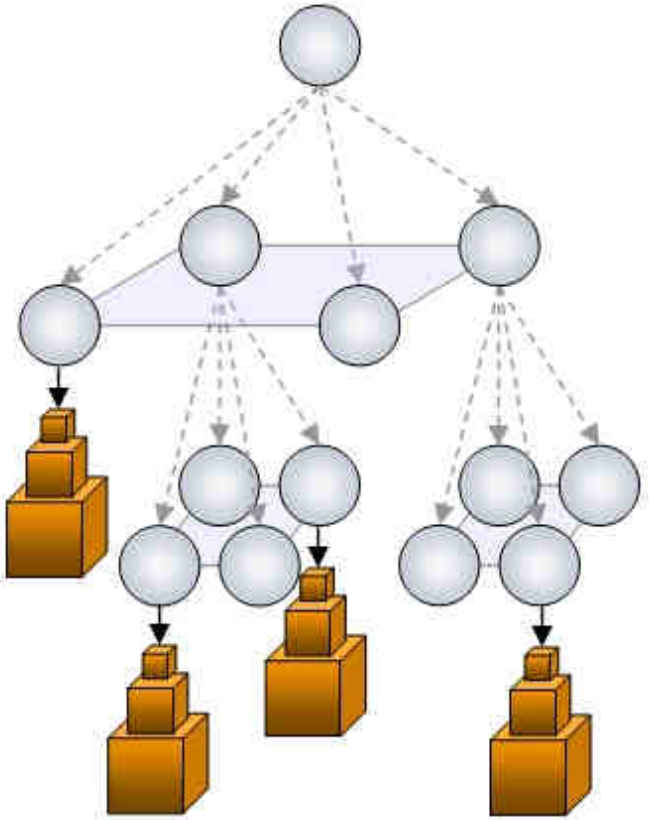


$$\sum_{i=0}^{\infty} \left(\frac{1}{4}\right)^i = \frac{1}{1-\frac{1}{4}} = \frac{4}{3} = 1 + \frac{1}{3} \text{ (geometrische Reihe)}$$

Quelle: http://de.wikipedia.org/wiki/Mipmapping

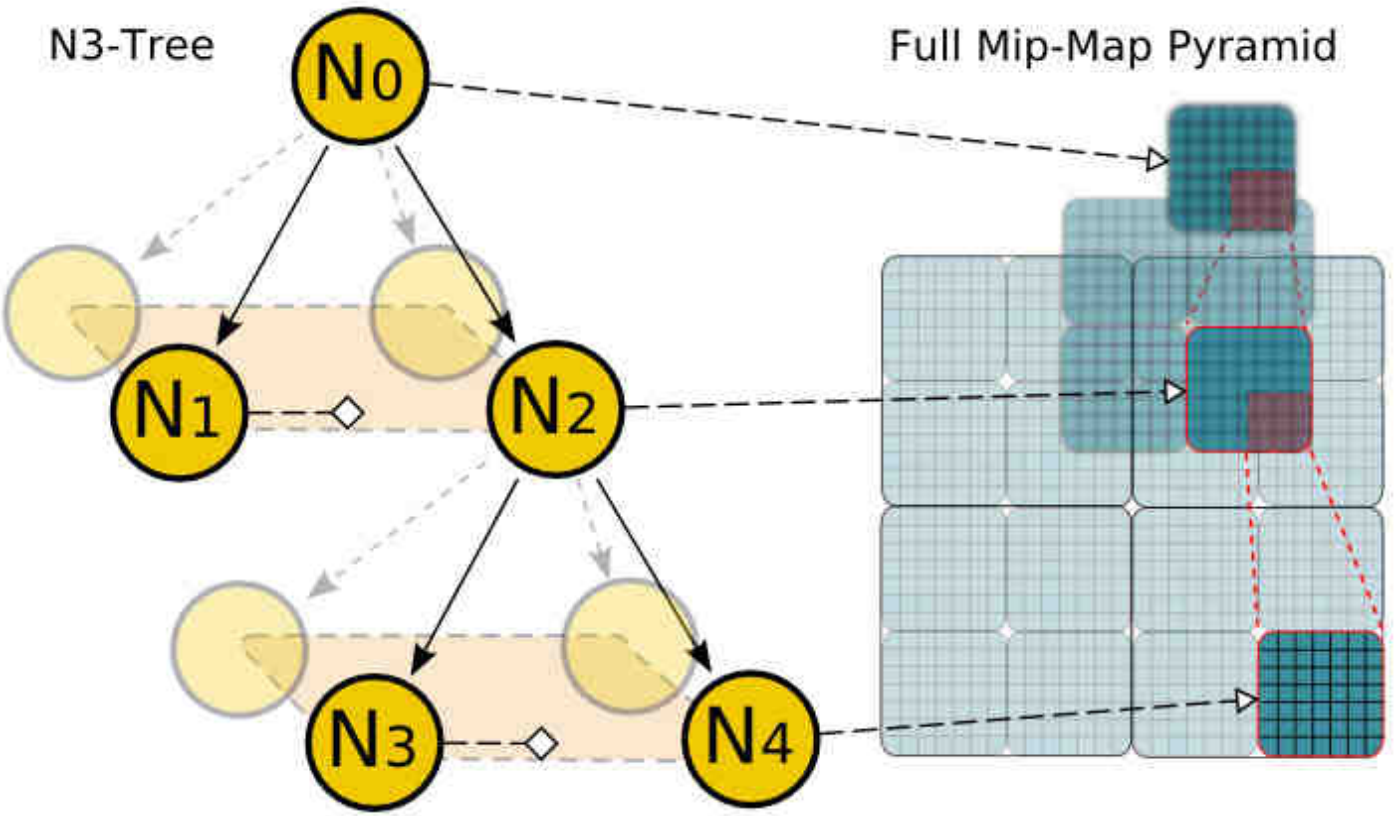
# N³ - Baum



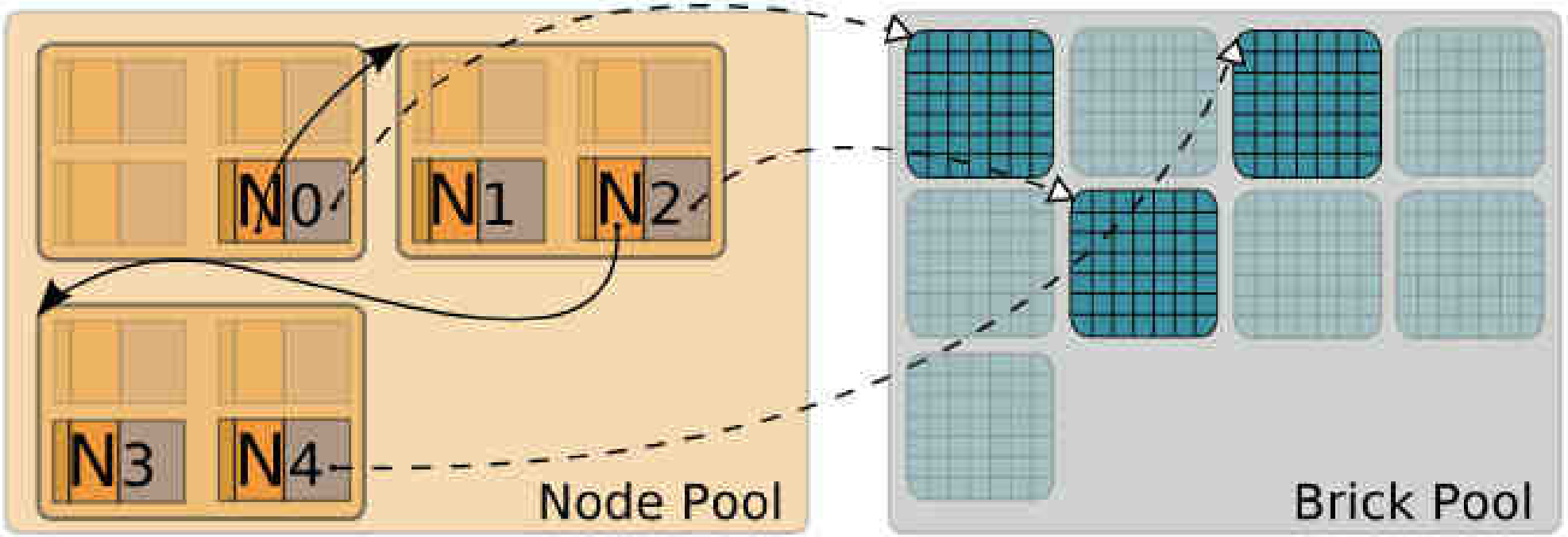


- „Brick“-Pointer
- konstanter Wert

# Verbindung N³ Baum und Mipmap



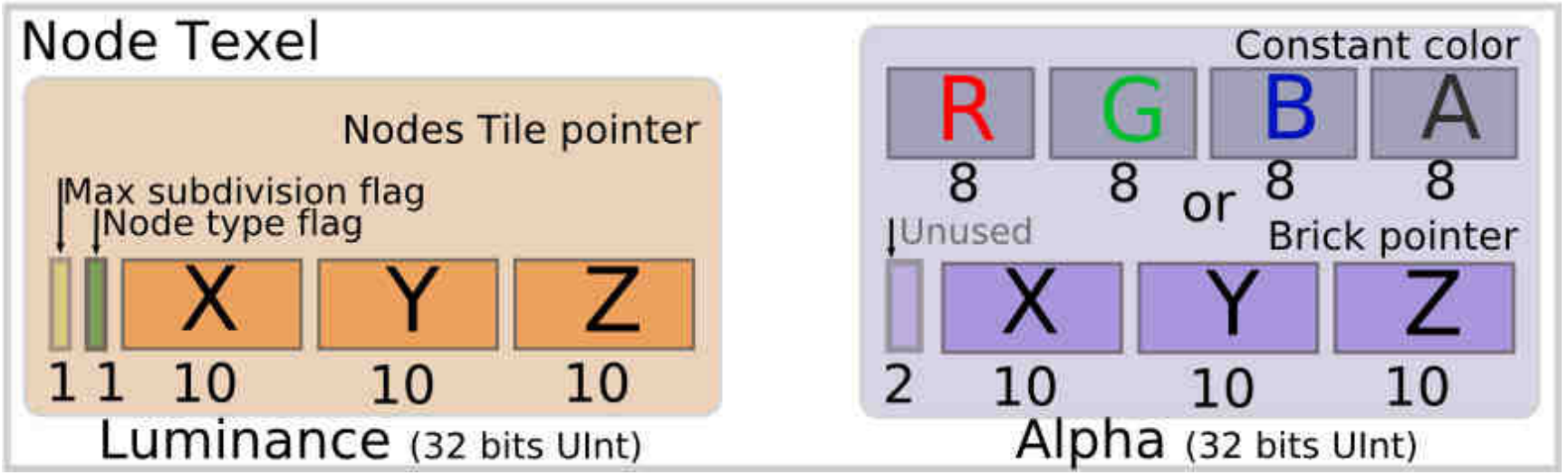
# Knoten und „Brick“ Pool




Voxel Netz mit $M^3$

# Speichern von Knoten

- 30 Bit: Pointer zum nächsten Knoten
- 2 Bit: Verwaltung
- 30/32 Bit: RGBA8 bzw. „Brick“-Pointer



# Baum Updates

- Zu viele Daten für die GPU
- Nur die gerade sichtbaren Daten sollten auf der GPU vorhanden sein

- Fehlende Daten sofort nachholen
- Gröberes „Brick“ Level

# Tests

Knochen

- 8192 ³ Voxel N = 2 und M = 16 → 60Fps

- Wolken
  - 2048 ³ Voxel N = 2 und M = 32

- Node Pool 4MB → 64³ Einträge
- Brick Pool 430MB → 42³ Einträge

# Ausblick

- Andere hierarchische Strukturen
- Animationen