

Echtzeit-Darstellung von extrem großen Landschaften mit Realtime-Ray-Tracing

Prof. Dr.-Ing. Philipp Slusallek

Lehrstuhl Computergraphik

Universität des Saarlandes

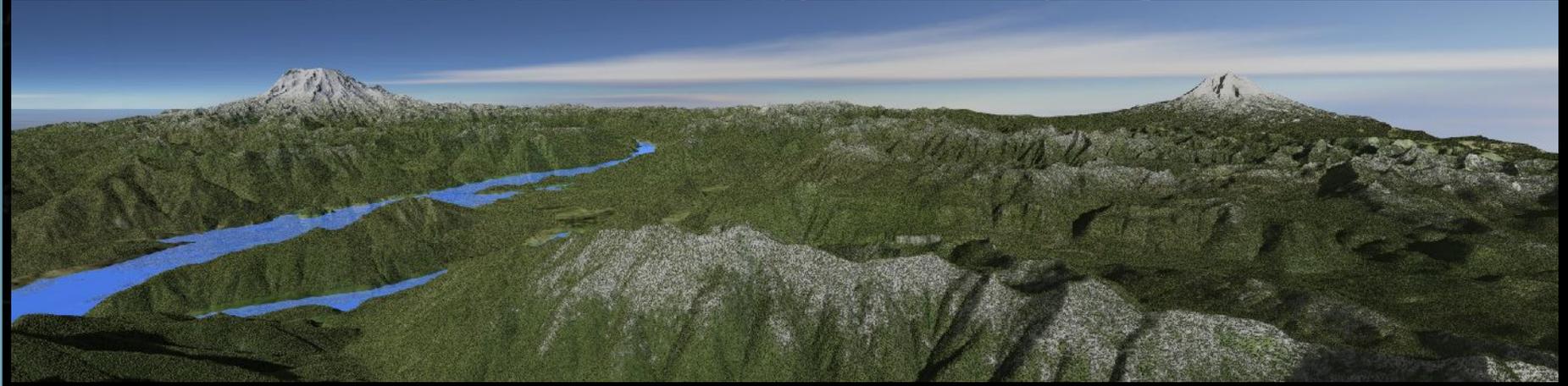
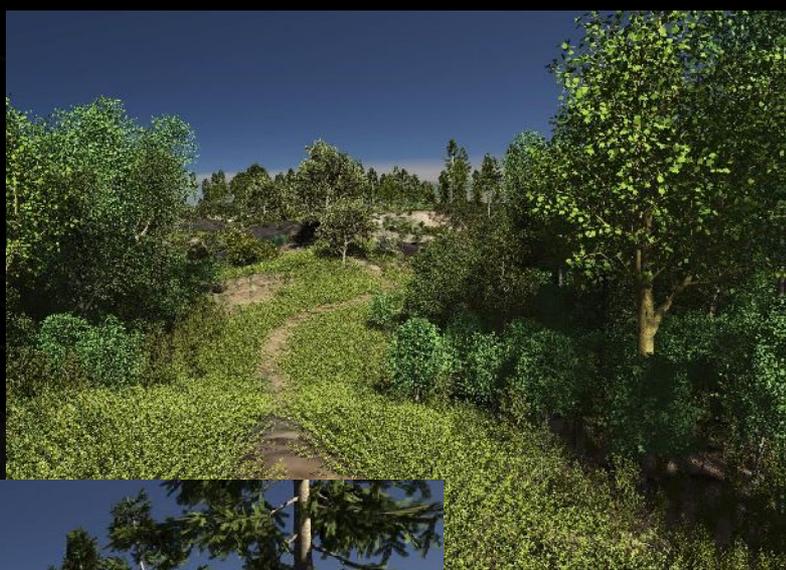


UNIVERSITÄT
DES
SAARLANDES

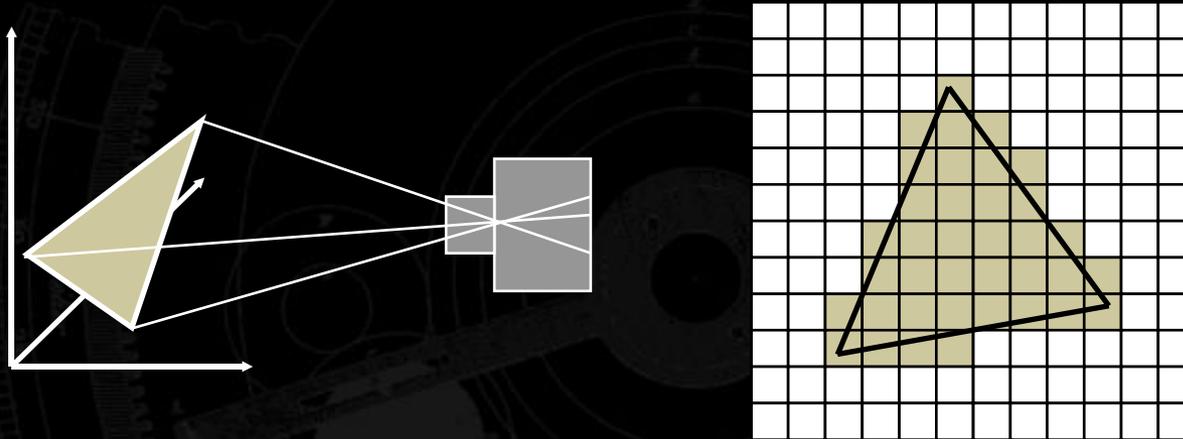
informatik
saarland.



COMPUTER GRAPHIK
UNIVERSITÄT
DES SAARLANDES

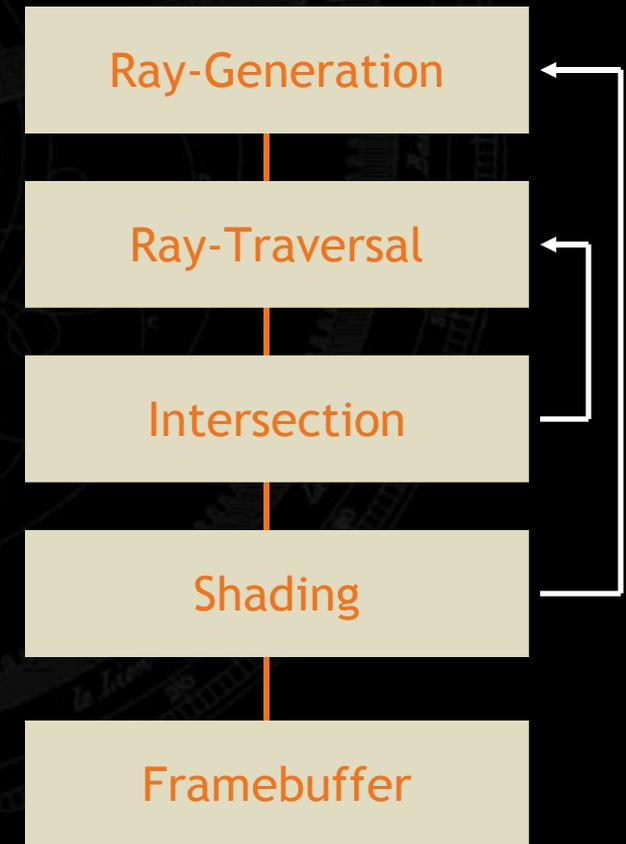
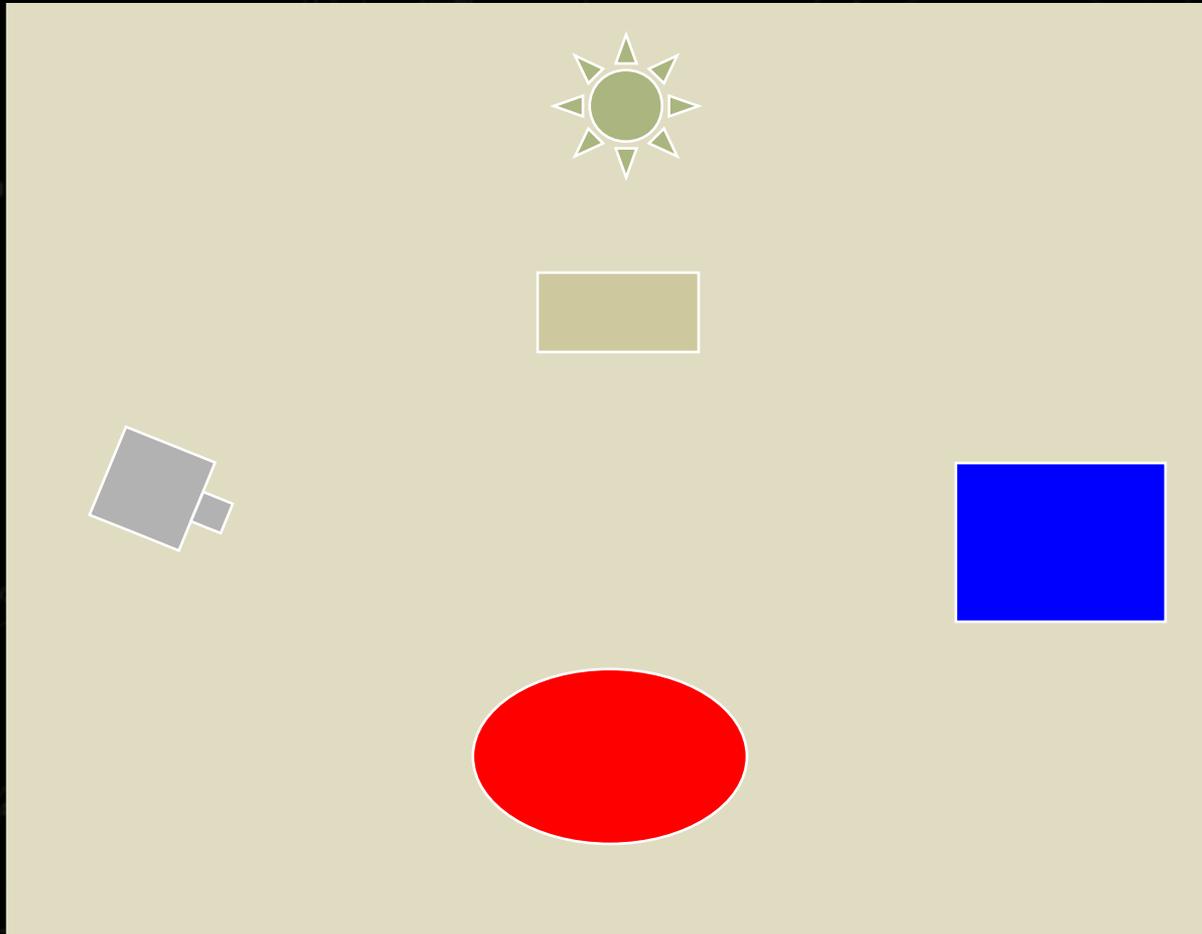


Rasterisierung

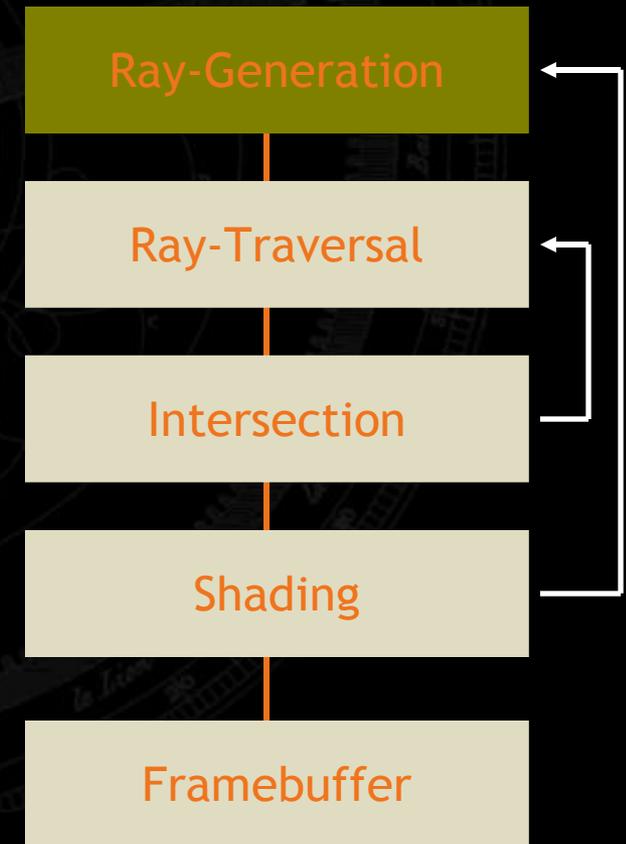
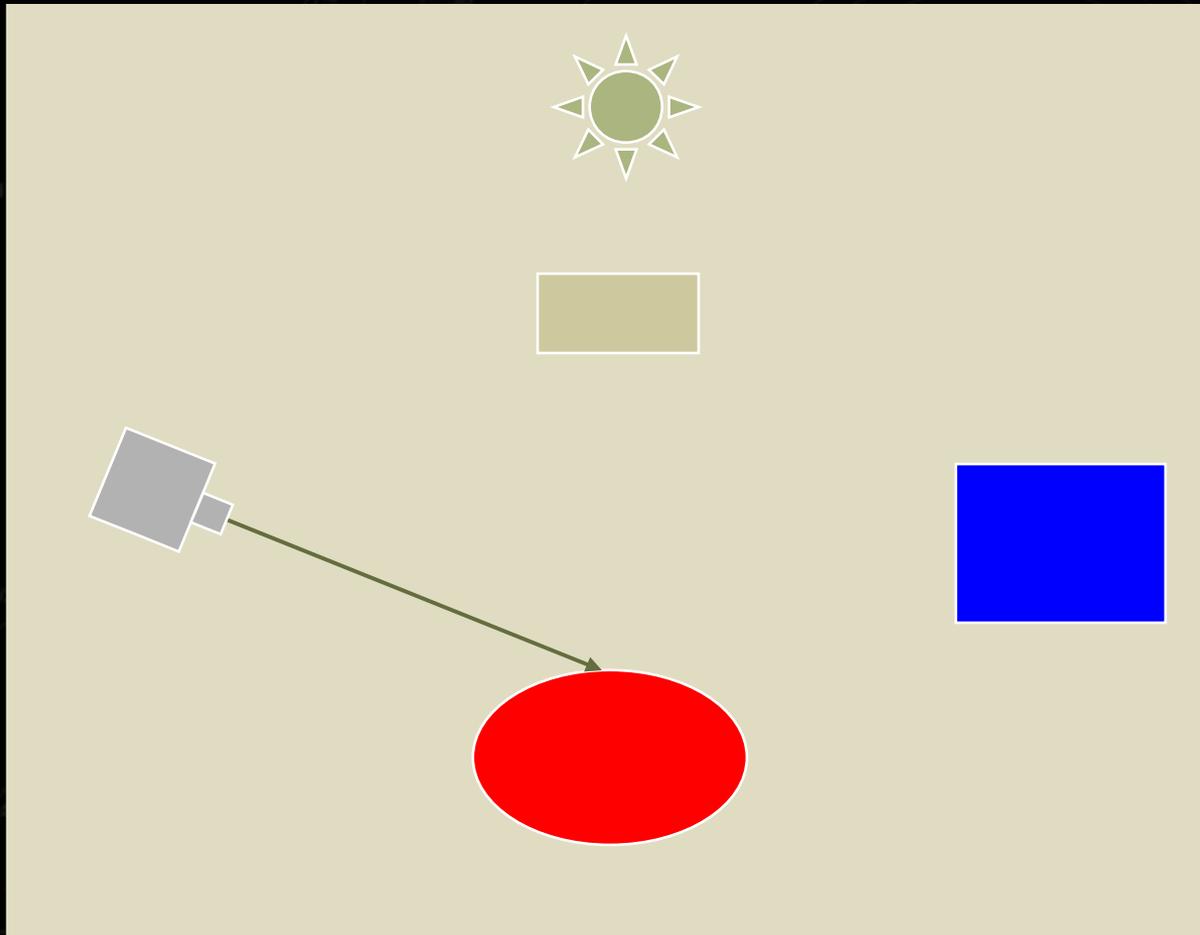


- **Hauptoperation der interaktiven Graphik !!**
 - Projektion und einfärben der bedeckten Pixel eines Dreiecks
- **Jedes Dreieck einzeln & nacheinander**
 - Kein Zugriff auf andere Dreiecke (spart Speicher in HW)
 - Aber fast alle Effekte brauchen Zugriff auf mehr Dreiecke: Schatten, Reflektion, indirekte Beleuchtung, ...
- **Geht das nicht besser?**

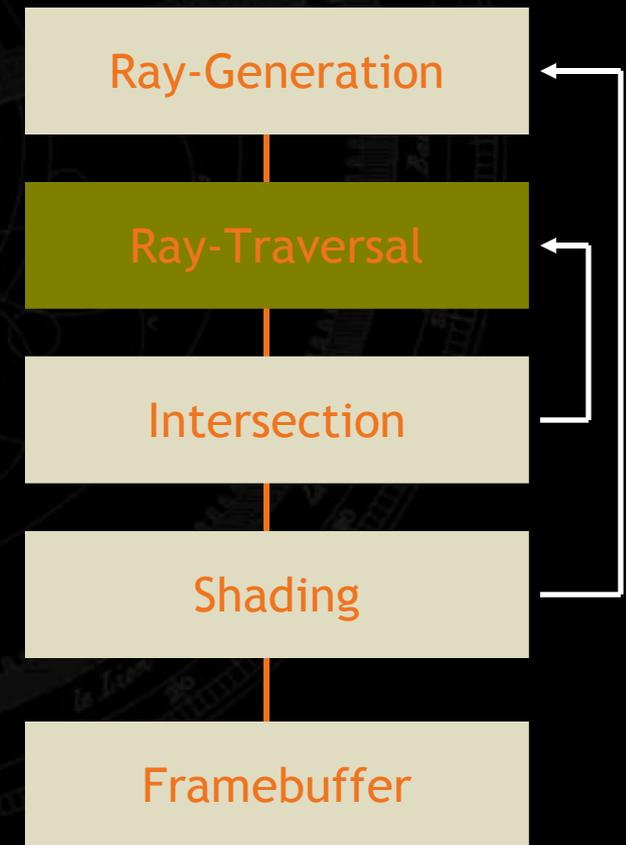
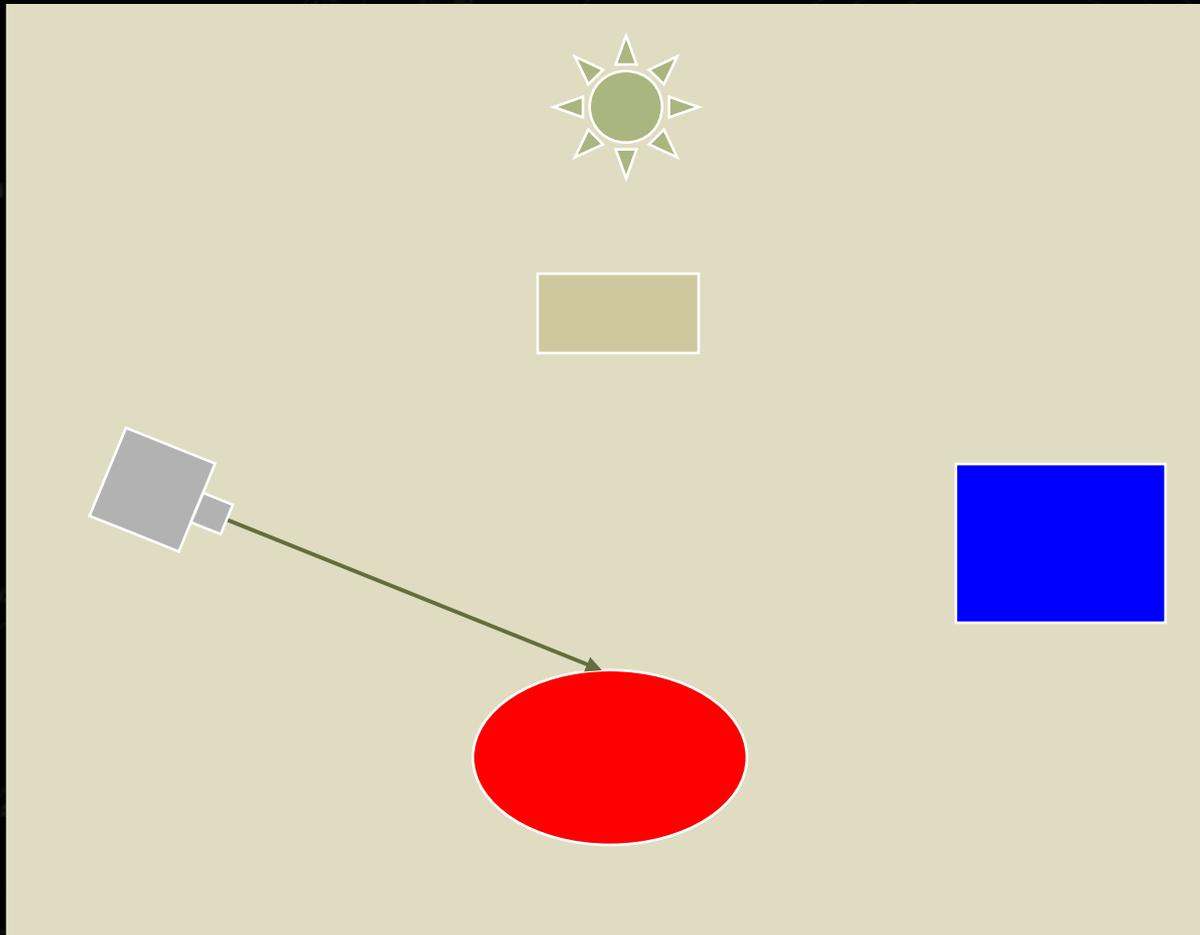
Ray-Tracing-Pipeline



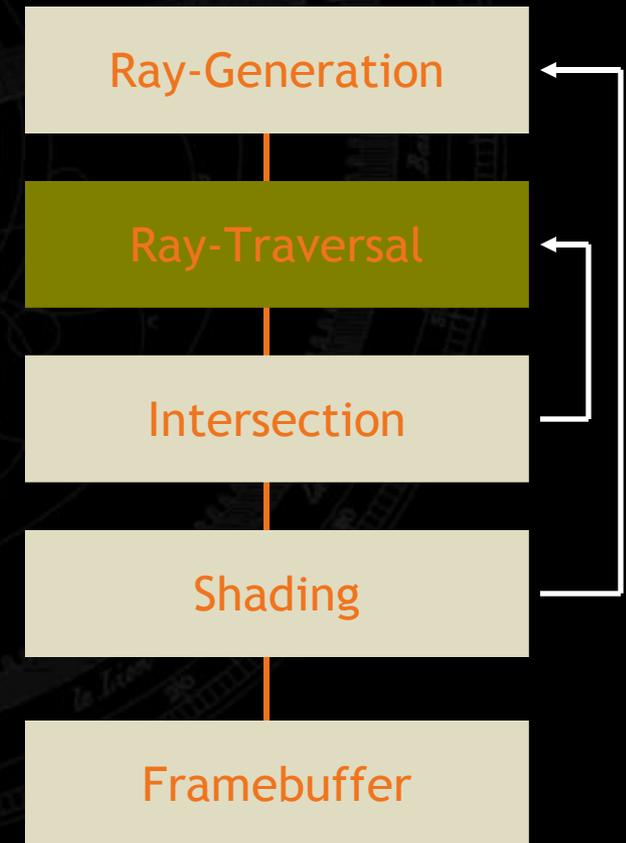
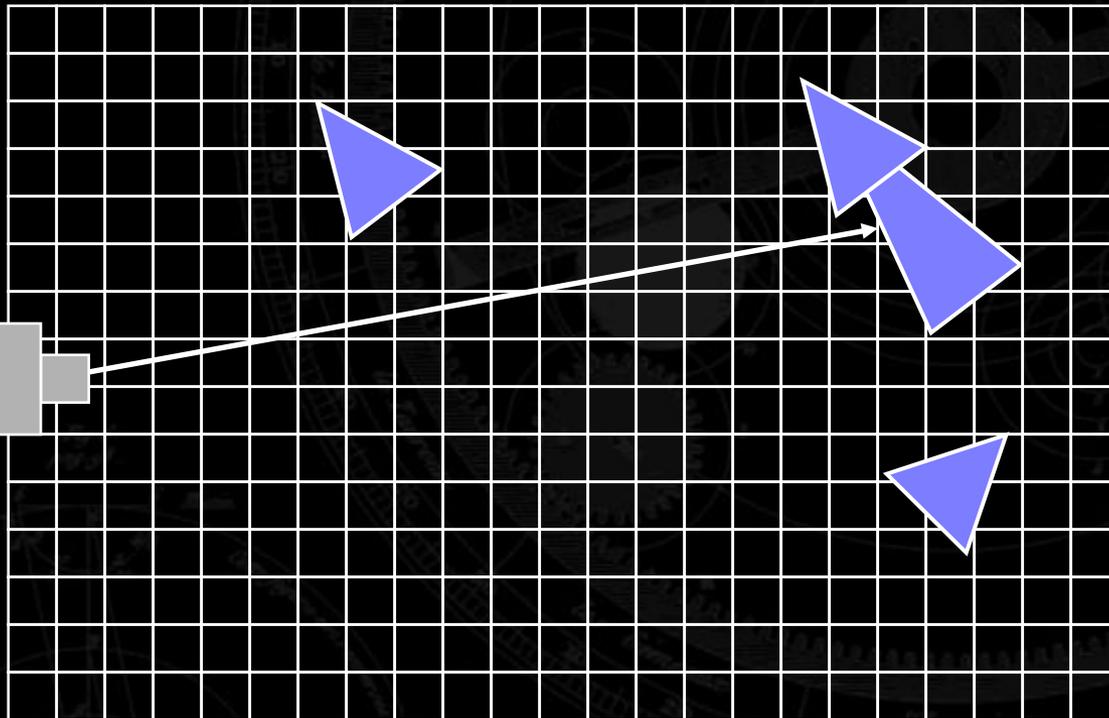
Ray-Tracing-Pipeline



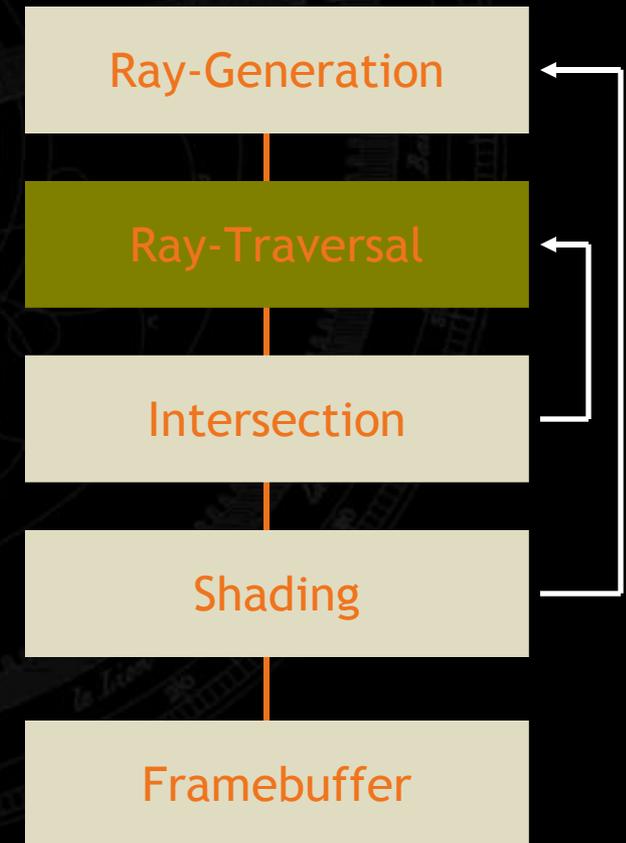
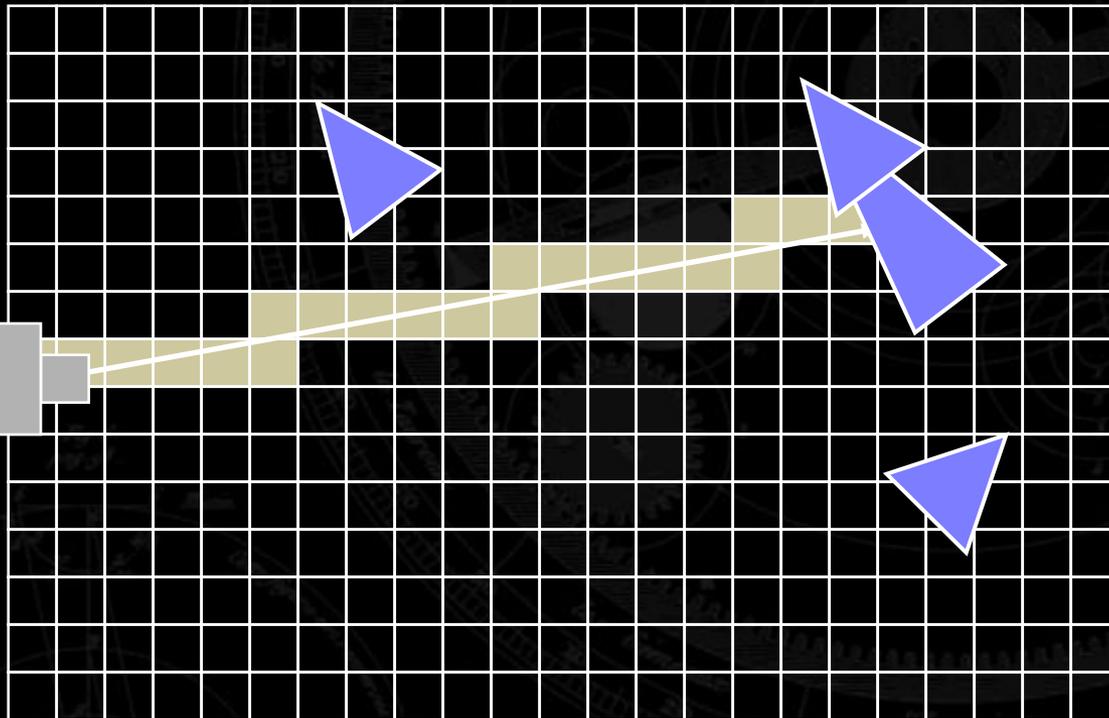
Ray-Tracing-Pipeline



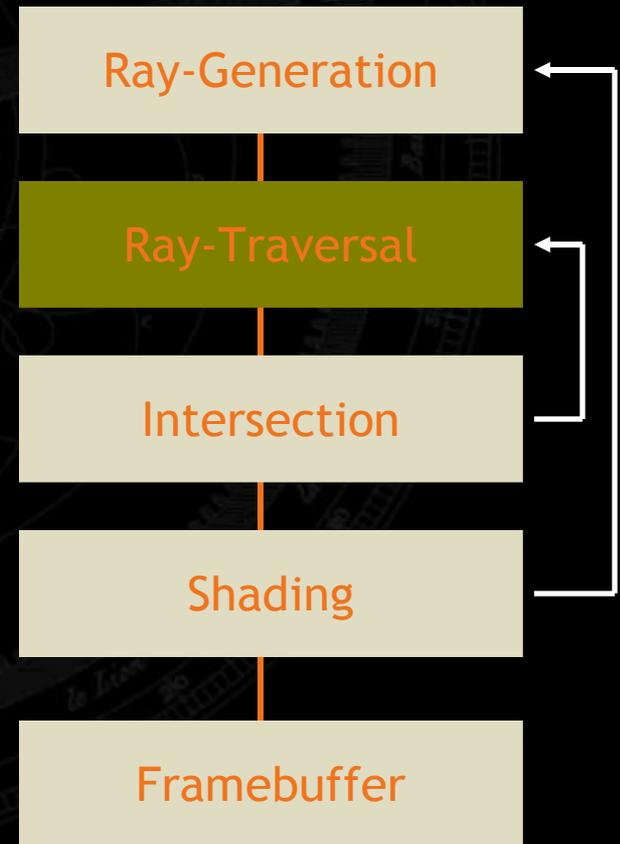
Ray-Tracing-Pipeline: Traversierung



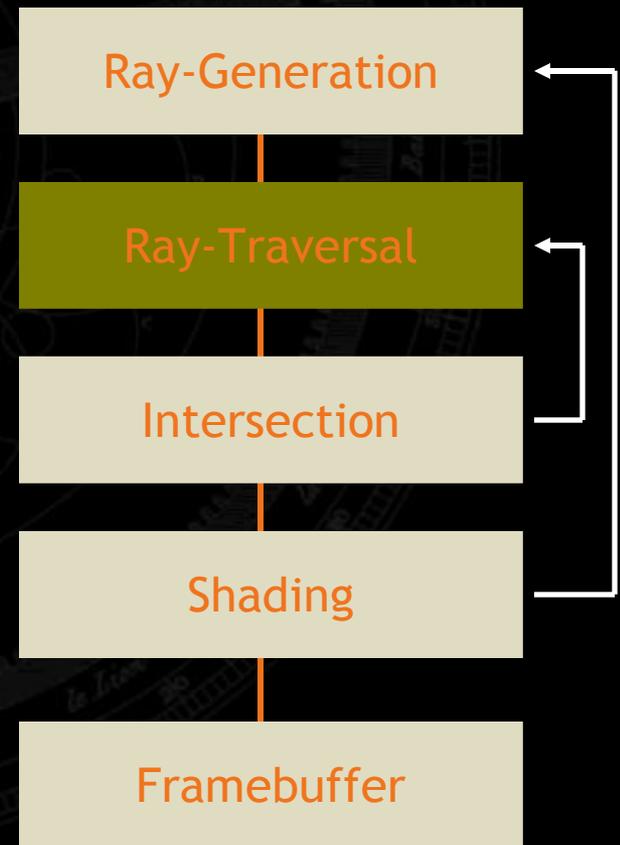
Ray-Tracing-Pipeline: Traversierung



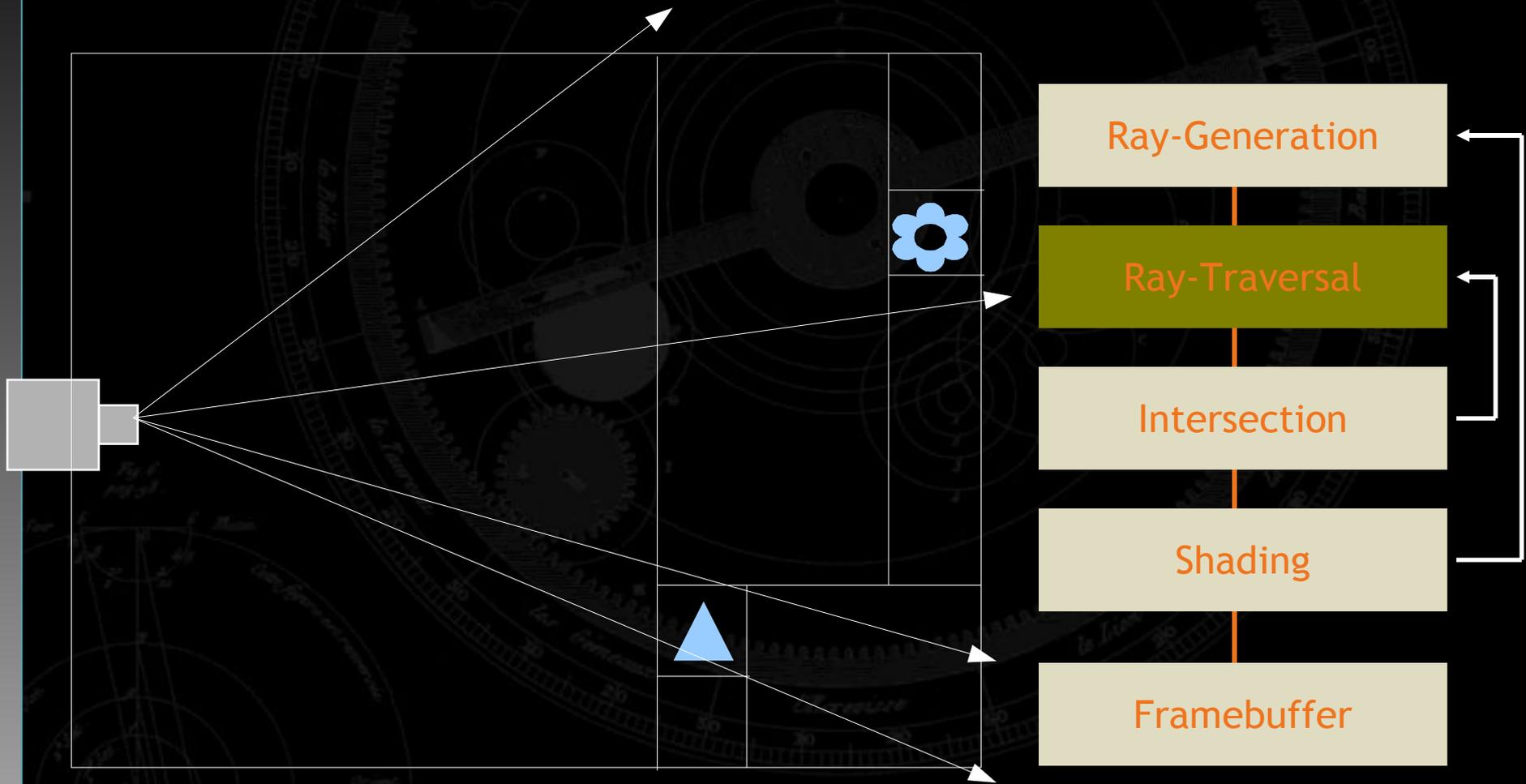
Ray-Tracing-Pipeline: kd-Baum



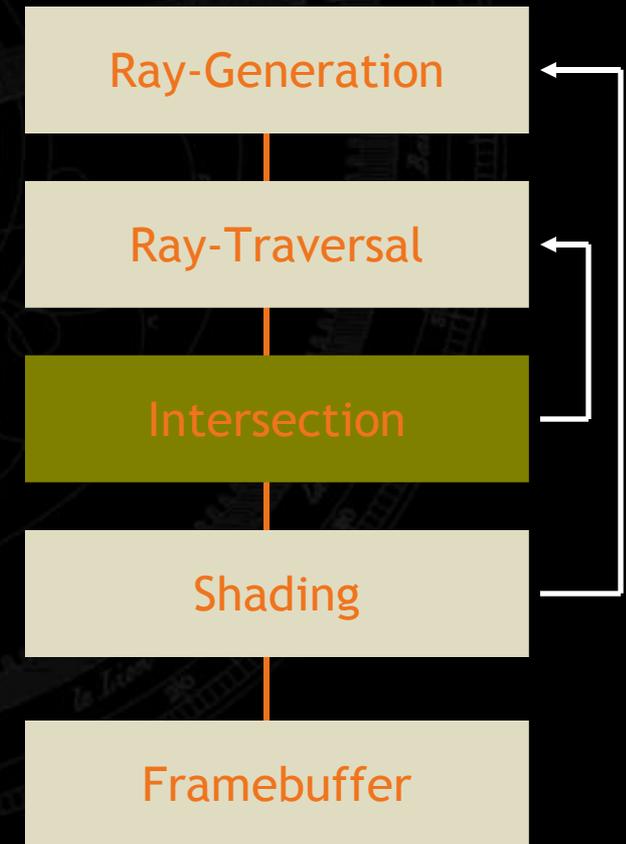
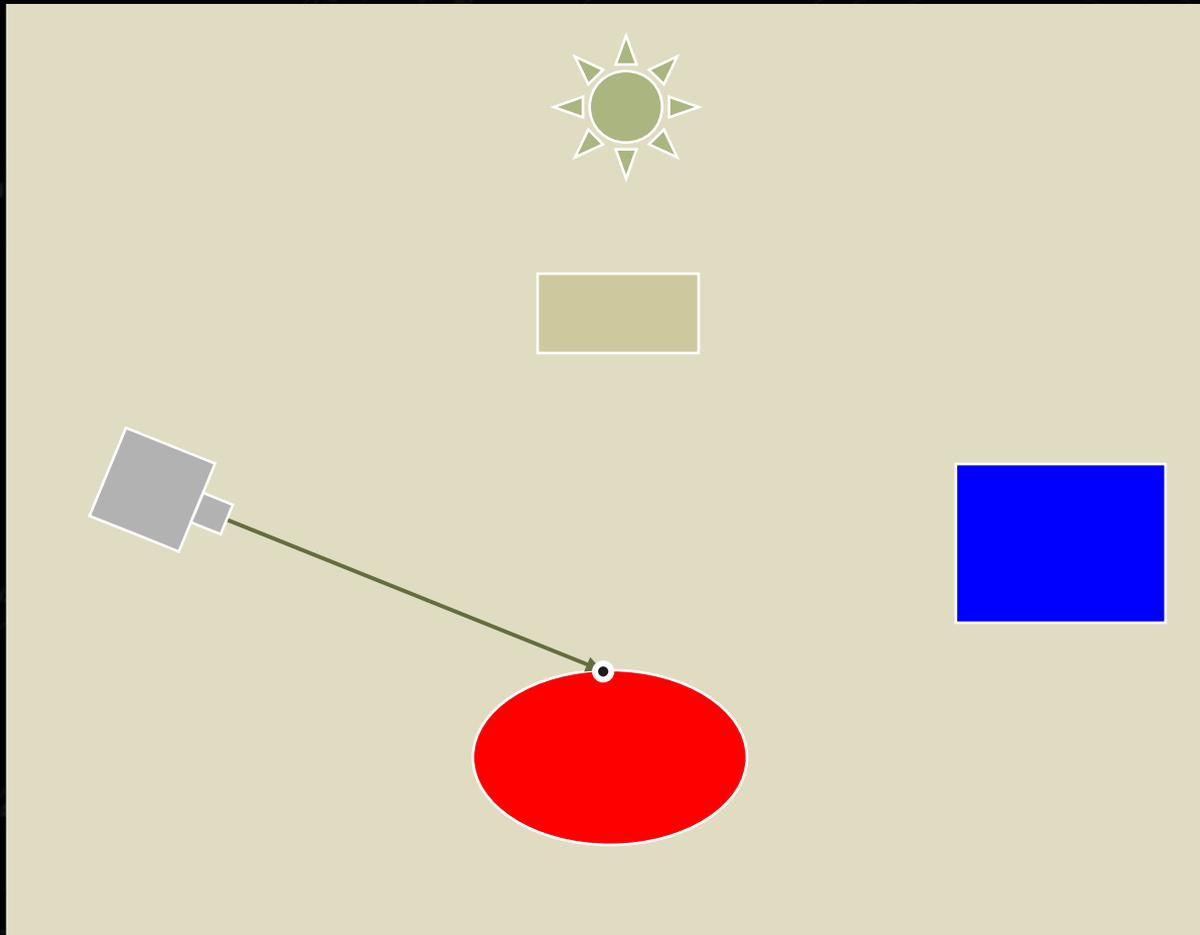
Ray-Tracing-Pipeline: kd-Baum



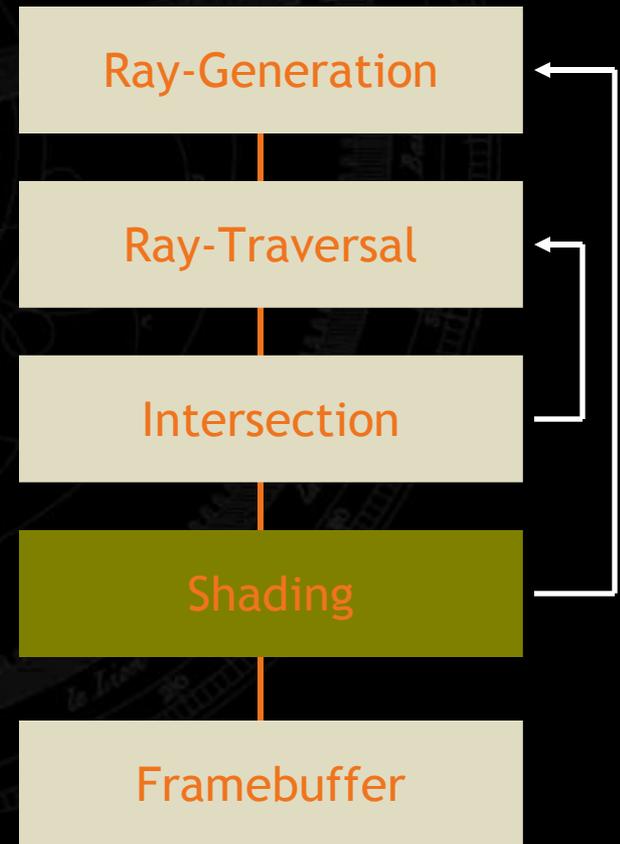
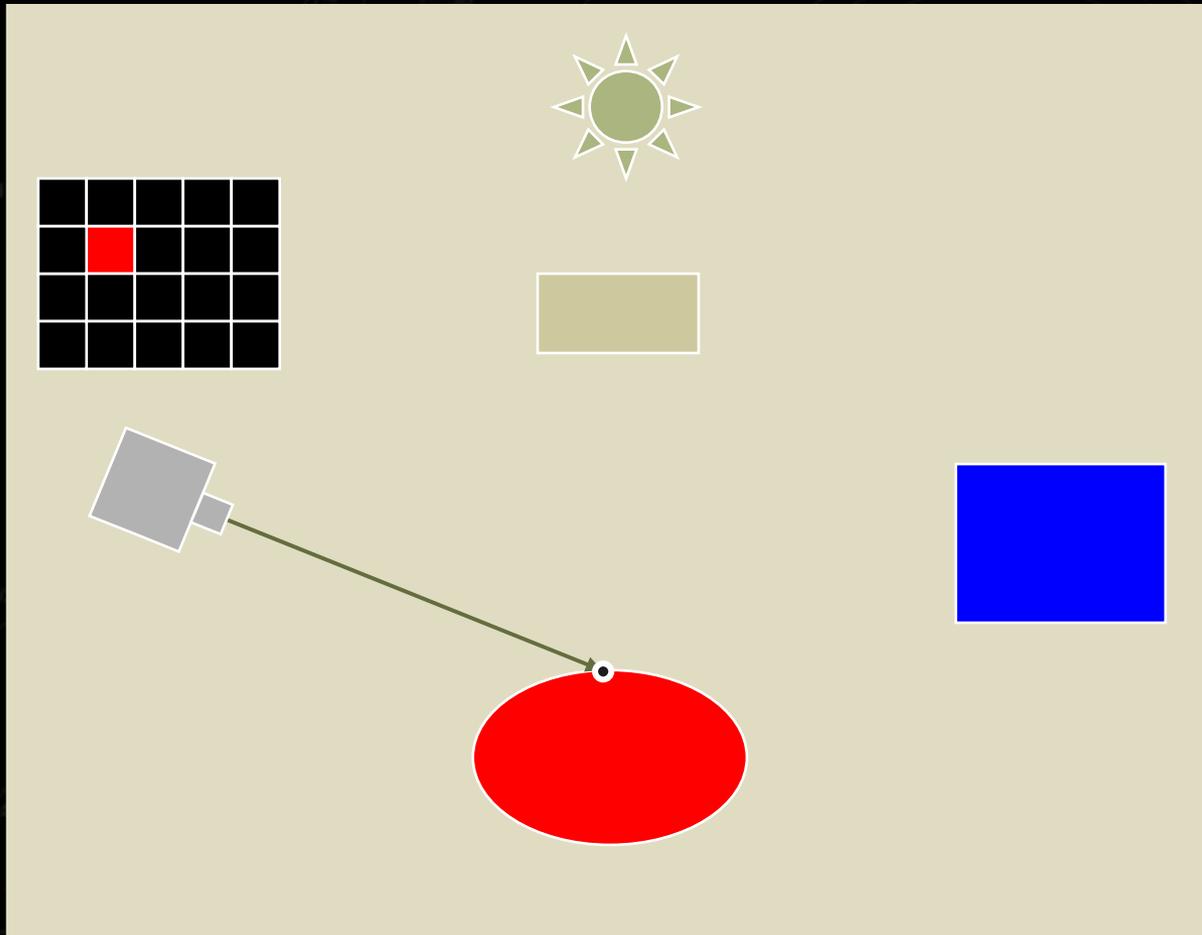
Ray-Tracing-Pipeline: kd-Baum



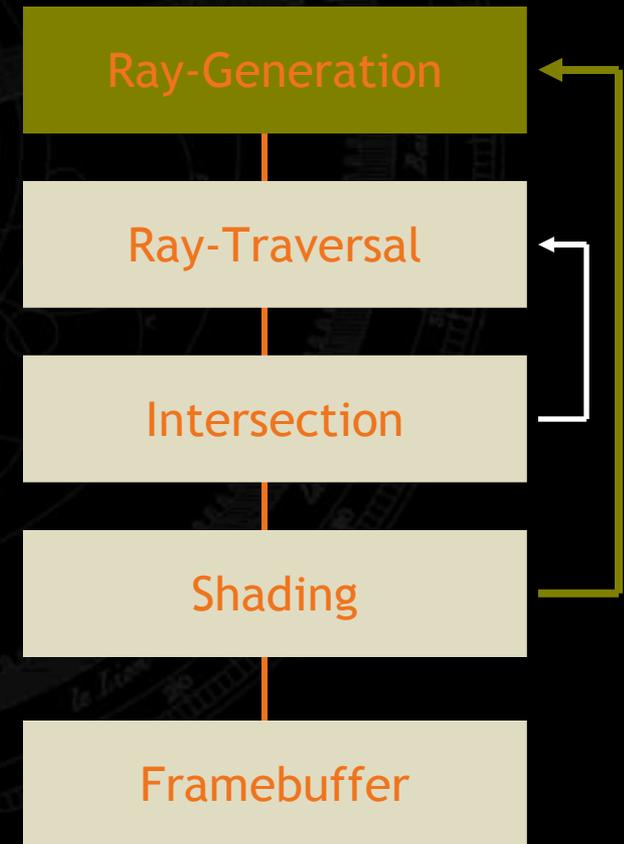
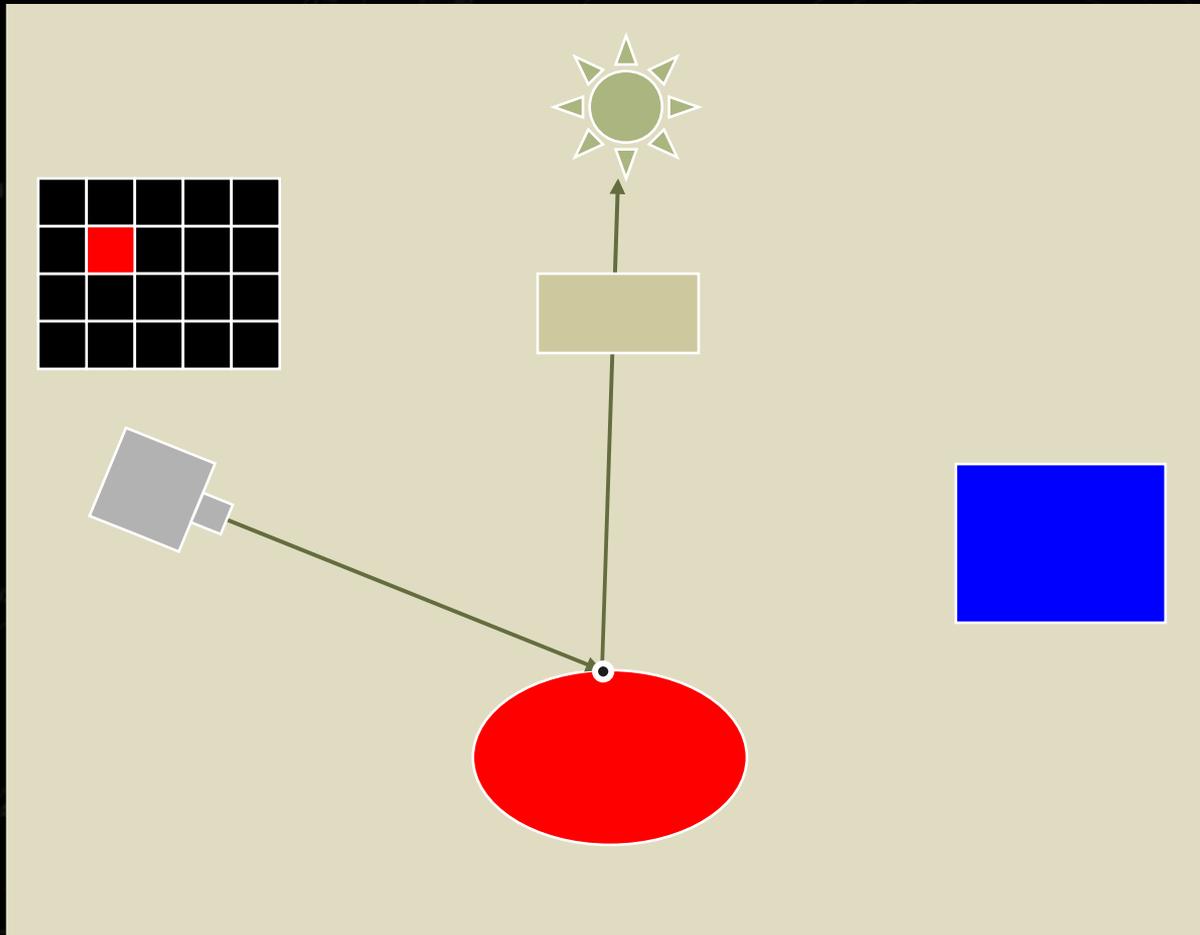
Ray-Tracing-Pipeline



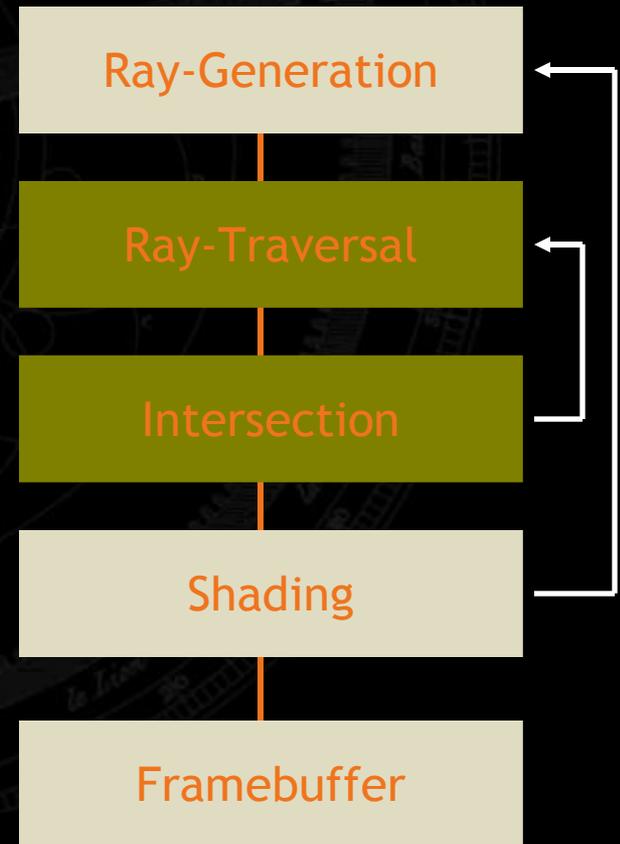
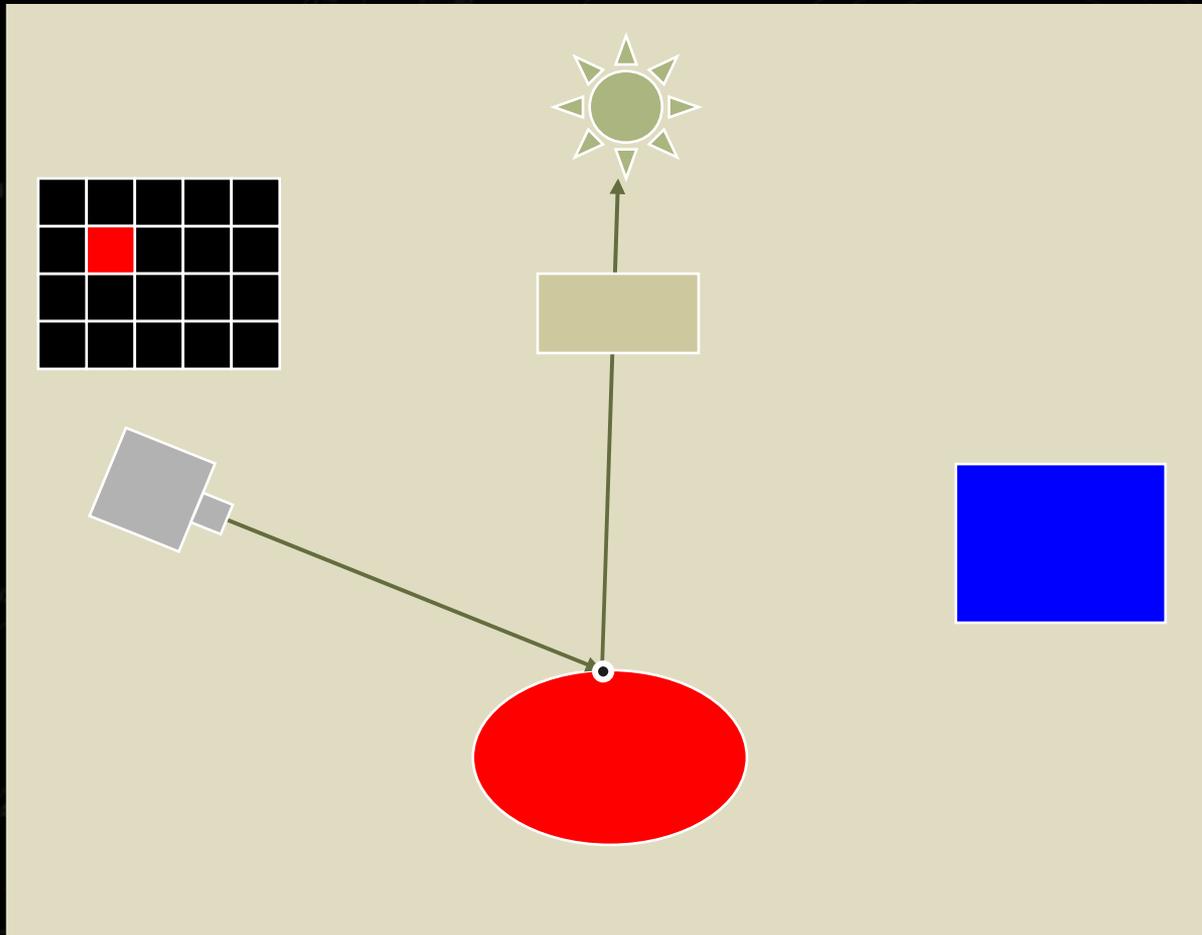
Ray-Tracing-Pipeline



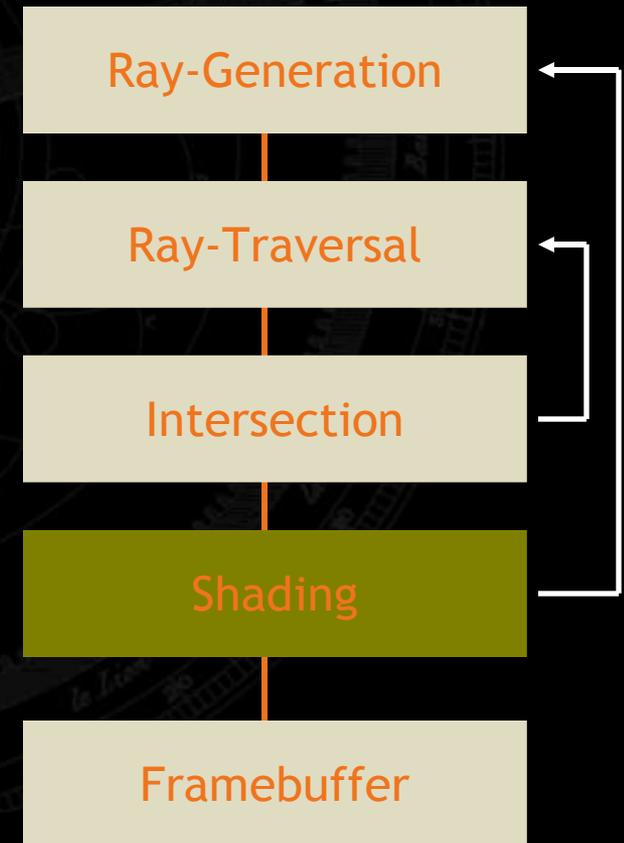
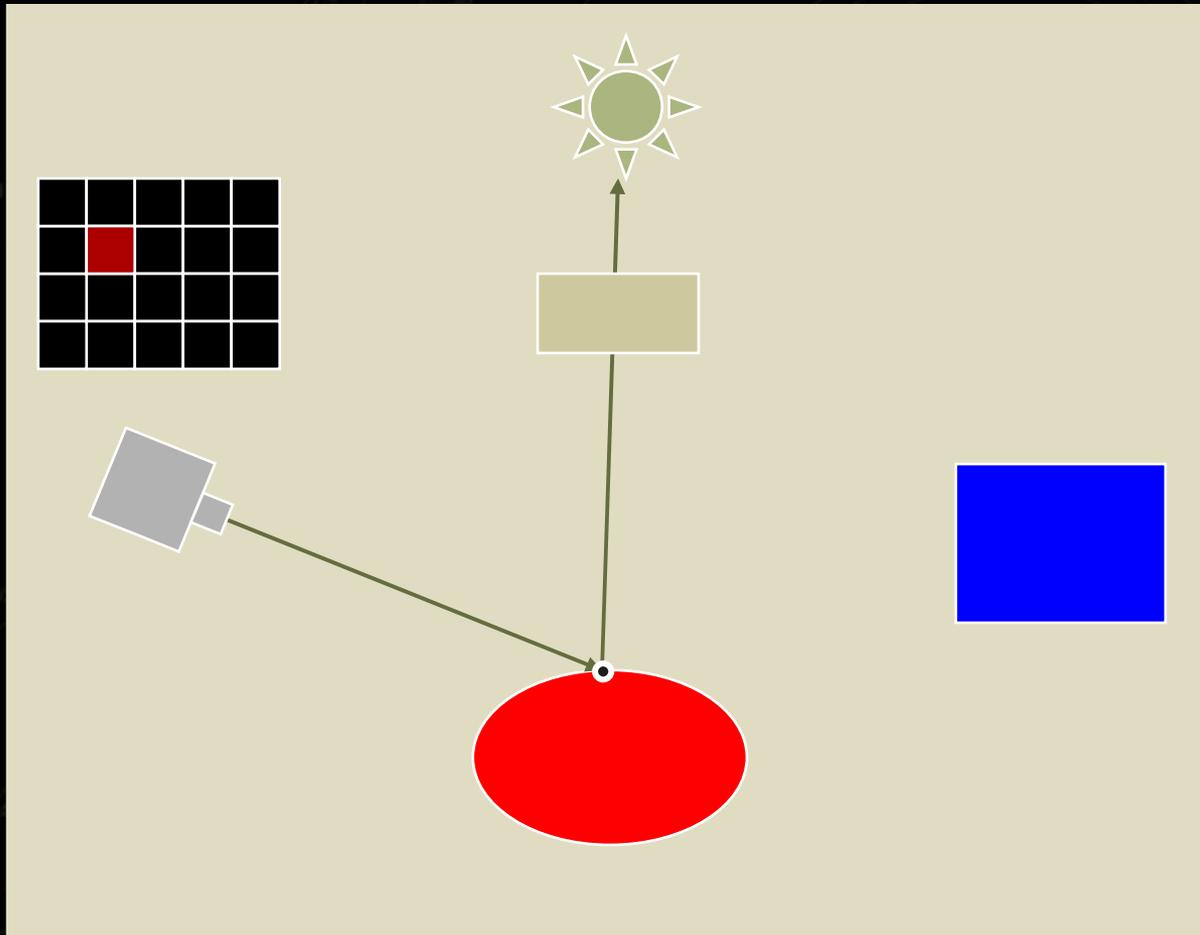
Ray-Tracing-Pipeline



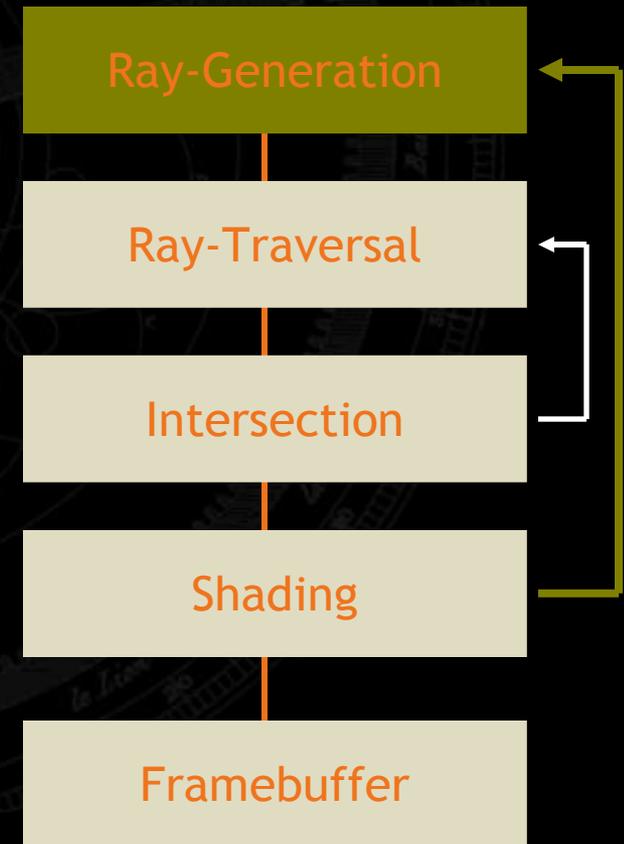
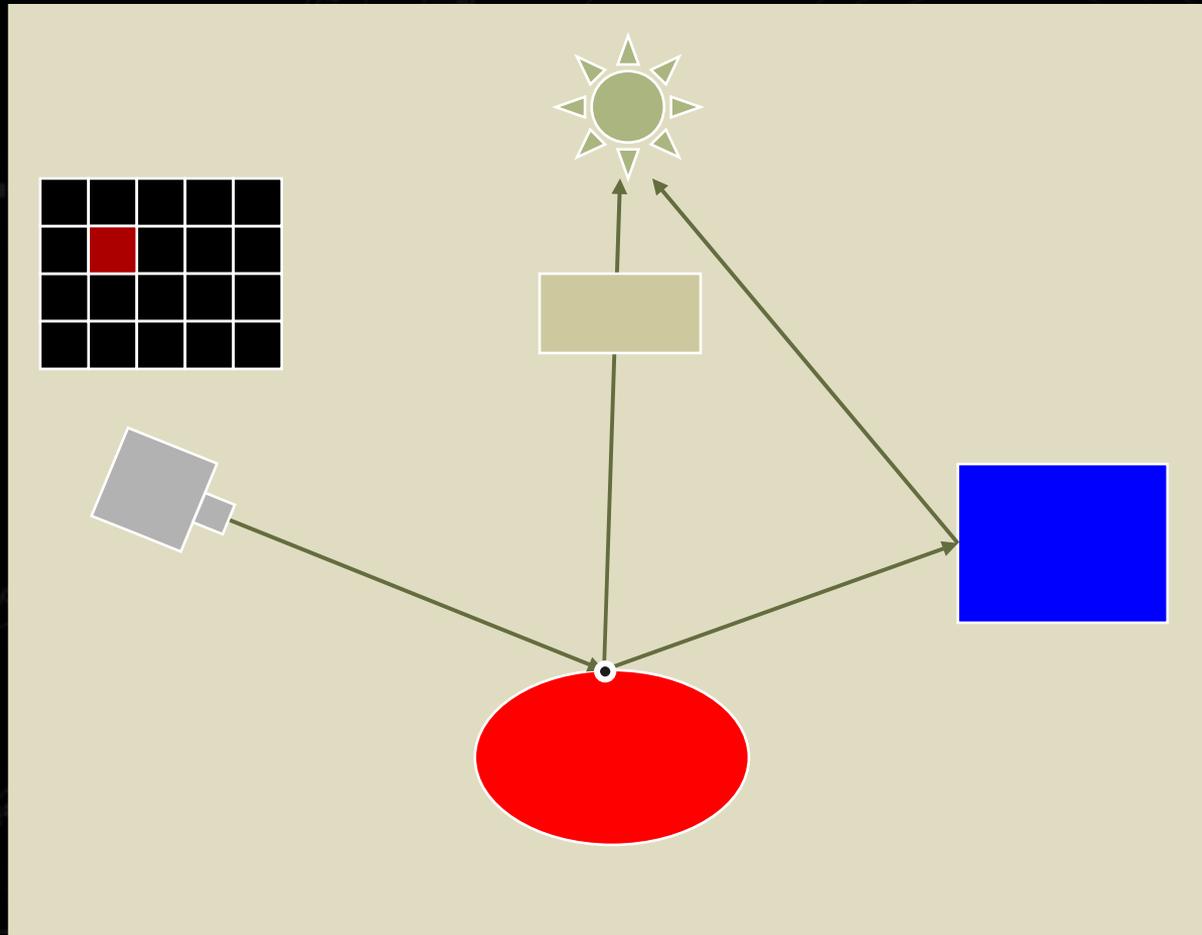
Ray-Tracing-Pipeline



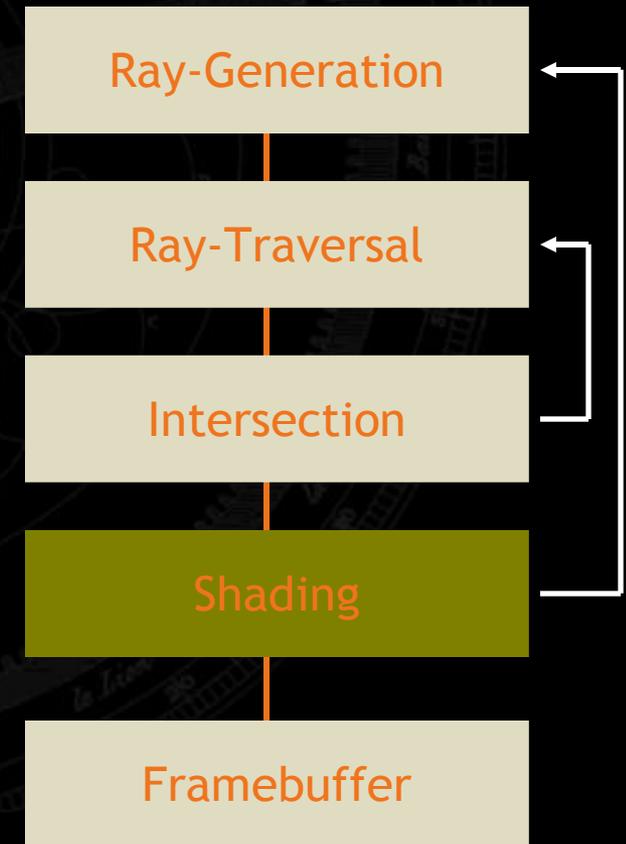
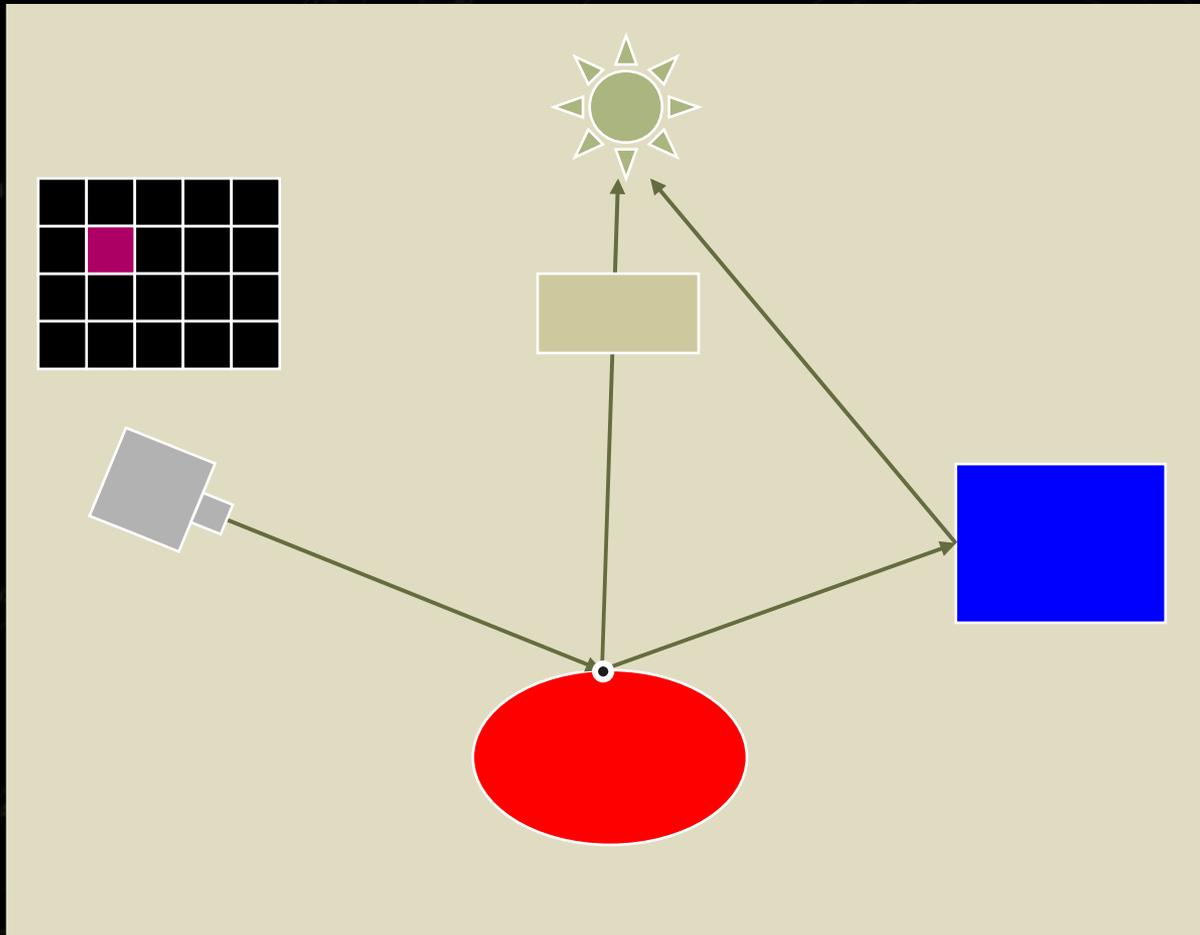
Ray-Tracing-Pipeline



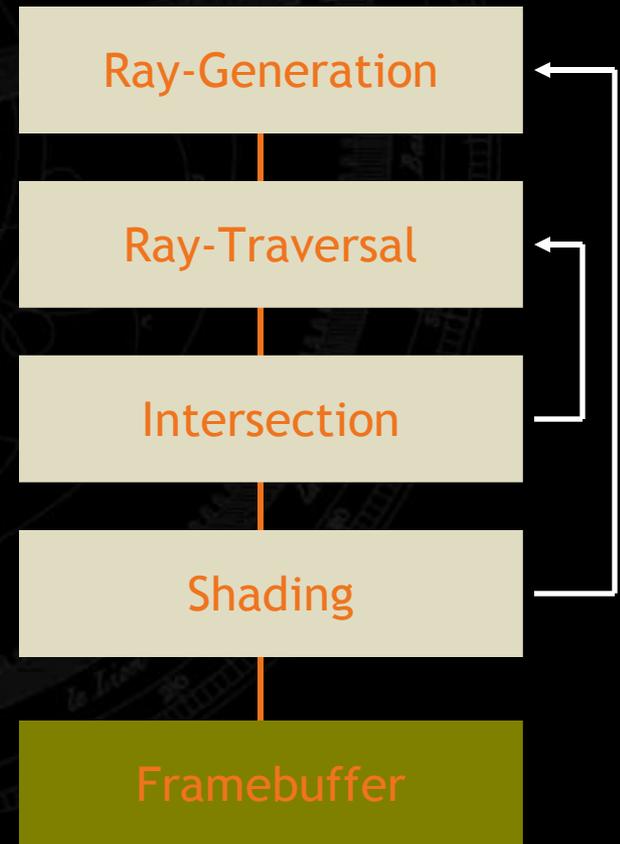
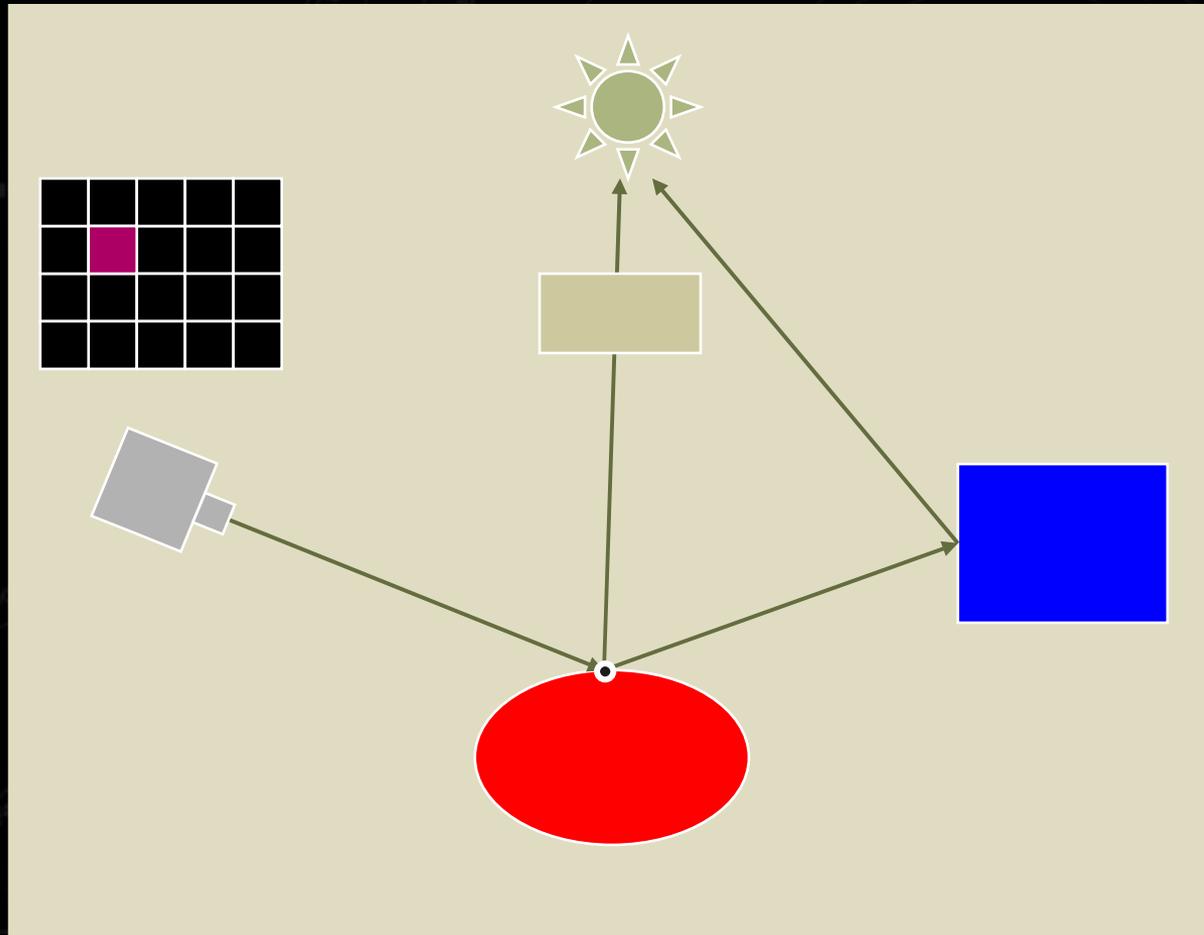
Ray-Tracing-Pipeline



Ray-Tracing-Pipeline



Ray-Tracing-Pipeline



Was ist Ray-Tracing?

- **Simulation der Physik des Lichts**

- Simulation durch das Verfolgen von “Photonen”
- Vollständig parallelisierbar: Wie in der Natur
- Nur nach Bedarf: Von der Kamera aus
- Berechnung in der richtigen Reihenfolge
- Kann alle *globalen Effekte* berechnen
- Alle Effekte sind voneinander unabhängig
- **Damit wird 3D-Graphik wieder intuitiv!!**

Warum macht das nicht jeder?

- **Geschichte**

- Gleich alt wie Rasterisierung
- Sehr hoher Rechenaufwand
- Keine Hardware-Unterstützung

- **Sehr geringe Leistung**

- Bilder brauchen oft Stunden
- Bisher nur Off-line-Benutzung



Internet Ray Tracing Competition:

1. Preis: > 6 Stunden

2. Preis: > 24 Stunden

80% > 1 Stunde, 15% < 15 Min.

Was kann man besser machen?

- **Algorithmische Änderungen**

- Berechnung von Strahlpaketen
- Bessere Algorithmen um kd-Bäume zu bauen
- Neue räumliche Indexstrukturen

- **Optimierte Implementierungen**

- Nutzung neuer CPU-Features (SIMD Berechnung)
- Cache-freundliches Design (Daten-Layout, ...)
- Allgemeine Codeoptimierung

➔ **Verbesserung der Performance: >30x**

Was geht heute schon?

• Paralleles Ray-Tracing

- Hoch-optimierte Software [EG'01, STAR'02, STAR'03]
- Verteilte Berechnung [RW'01, EuroPar'03]
- Neue OpenRT-Schnittstelle [OpenSG'03]
- Extrem große Modelle [RW'01, EGSR'04, VMV'05]
- Dynamische Szenen [PVG'03, GH'04]
- Echtzeit-Beleuchtungsberechnung [RW'02, EG'03]
- Echtzeit-Photon-Mapping [EGSR'04, EG'04]
- Hoch-komplexe Beleuchtungsszenarien [EGSR'03]
- Direkte Berechnung von Freiformflächen [AG'04]
- Volumendaten und Punktdaten [VMV'04/'05, TVCG'05]
- Ray-Tracing-Hardware [GH'02, GH'04, Siggraph'05]

Was kann man damit machen?



Headlight: 800K triangles, complex reflections and refractions

Anwendung (Spin-Off inTrace GmbH): VW Visualisierungs-Zentrum



inTrace Release 1.5: Area lights, glossy reflection, transparency, ...

Anwendung (Spin-Off inTrace GmbH): VW Visualisierungs-Zentrum



Global Beleuchtungs- Simulation

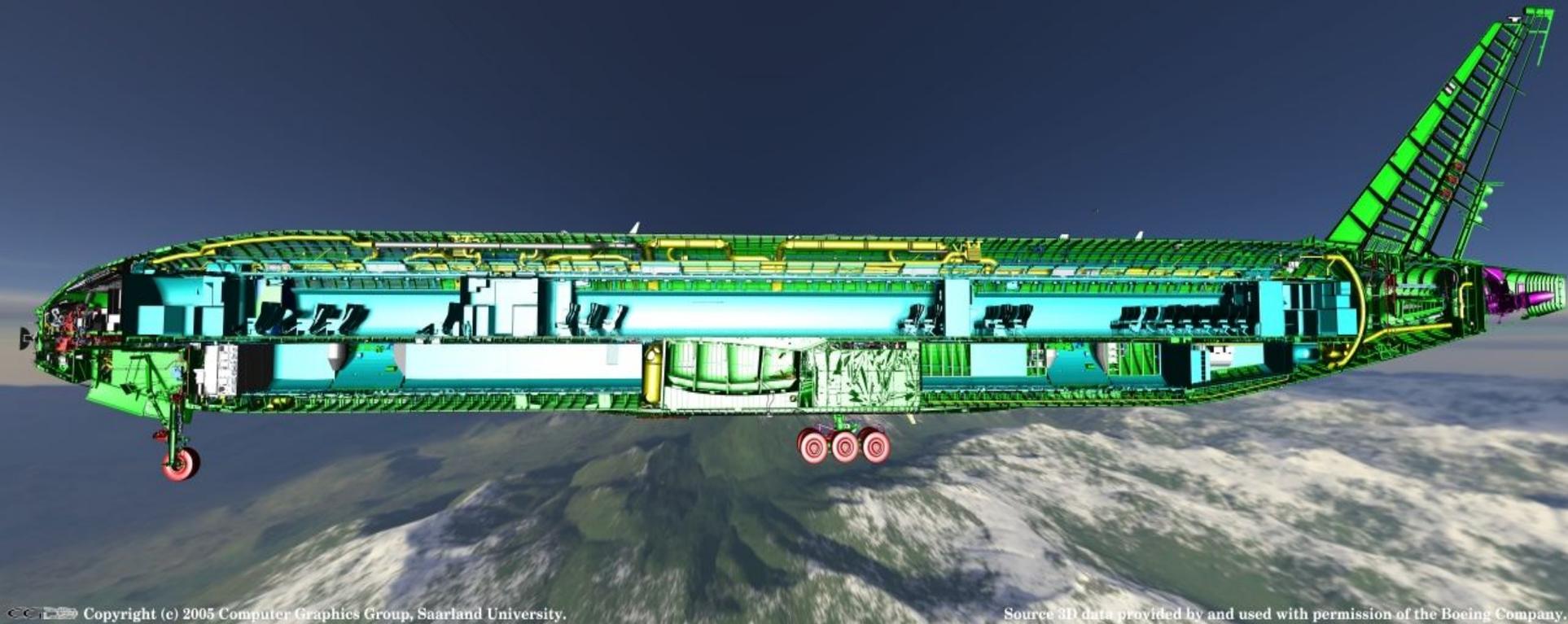


Conference: 280K triangles, 104 area lights
Wine glass: physically accurate caustics

Computerspiele mit Ray-Tracing



Extrem komplexe Modelle: Komplette CAD-Daten von Boeing 777



Copyright (c) 2005 Computer Graphics Group, Saarland University.

Source 3D data provided by and used with permission of the Boeing Company.

CATIA Model: 350 Millionen Dreiecke, 30 GB on Platte, 2-3 fps auf Dual-Opteron

Landschaftsvisualisierung



Landschaftsvisualisierung



Wie geht das?



- **Trivial einfach:**

- Man nehme die 3D-Modelle der Szene (Blume/Baum)
- Man baue ein kd-Baum für jedes Objekt
- Man platziere Kopien der Objekte (Box) in der Szene
 - Braucht nicht viel Speicher (eine Transformation)
- Viele Boxen: Baue neuen kd-Baum über Boxen
- Wenn ein Strahl eine Box/Objekt trifft
 - Transformiert man den Strahl zu dem Objekt
 - Verfolgt man sie in dem kd-Baum des Objektes weiter

Und wenn es komplizierter wird?

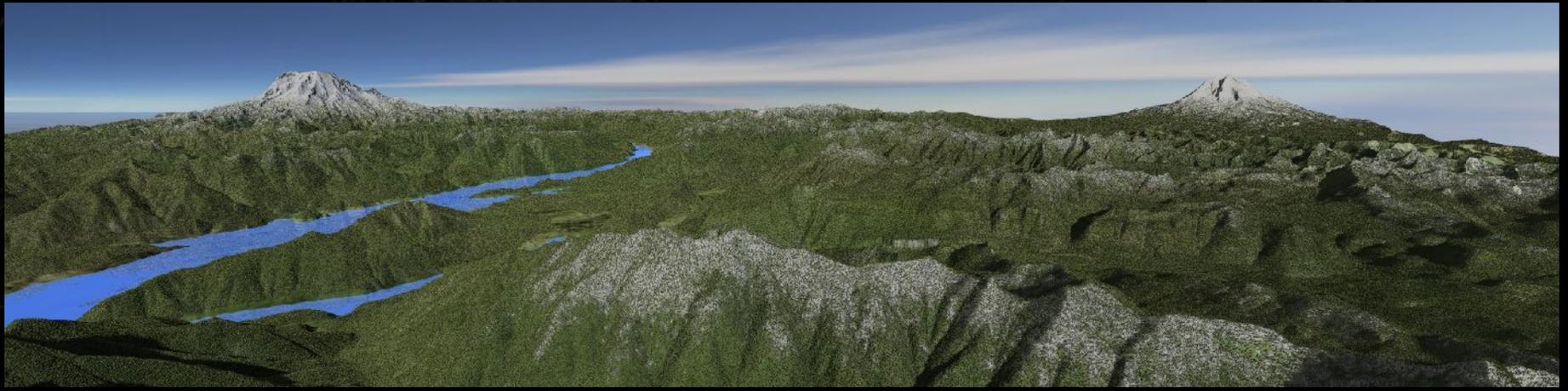
- **Landschaften mit mehreren Milliarden Dreiecken?**
 - Geht das immer noch!!!



Layer	Species	Instances	average Complexity	total Triangles
layer 3 (higher haulm)	6	204,750	1,000	204,750,000
layer 4 (lower shrub)	19	144,091	5,000	720,455,000
layer 5 (higher shrub)	12	10,464	25,000	261,600,000
layer 6 (lower tree)	10	5,209	50,000	260,450,000
layer 7 (higher tree)	9	290	75,000	21,750,000
layer 8 (overlap tree)	12	237	100,000	23,700,000
Total	68	365.041		≈1.500.000.000

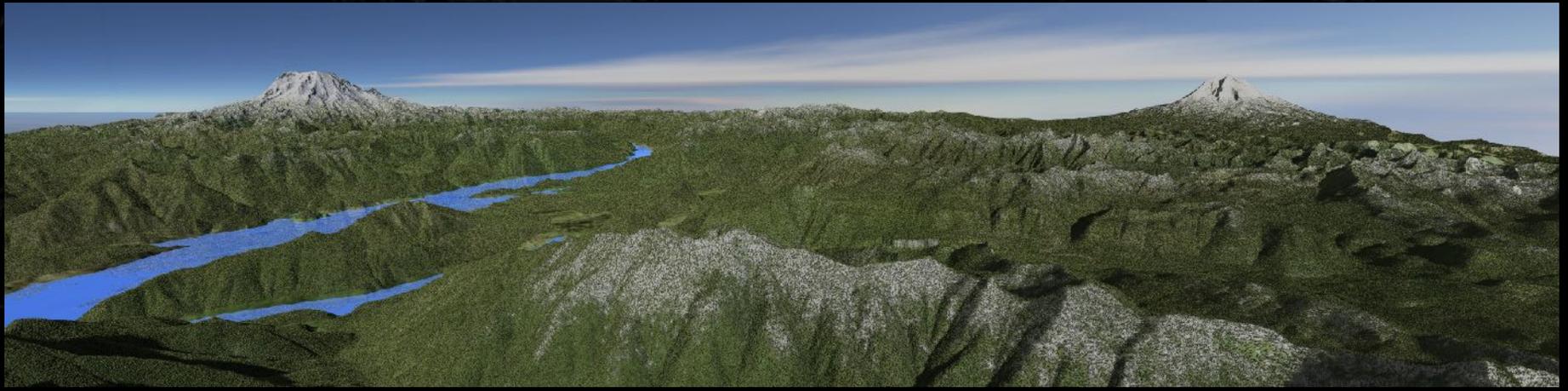
Und wenn es noch viel komplizierter wird?

- **Landschaften mit Milliarden von Bäumen**
 - 80km x 80km im Nordwesten der USA
 - Das Grüne sind ist keine Farbe



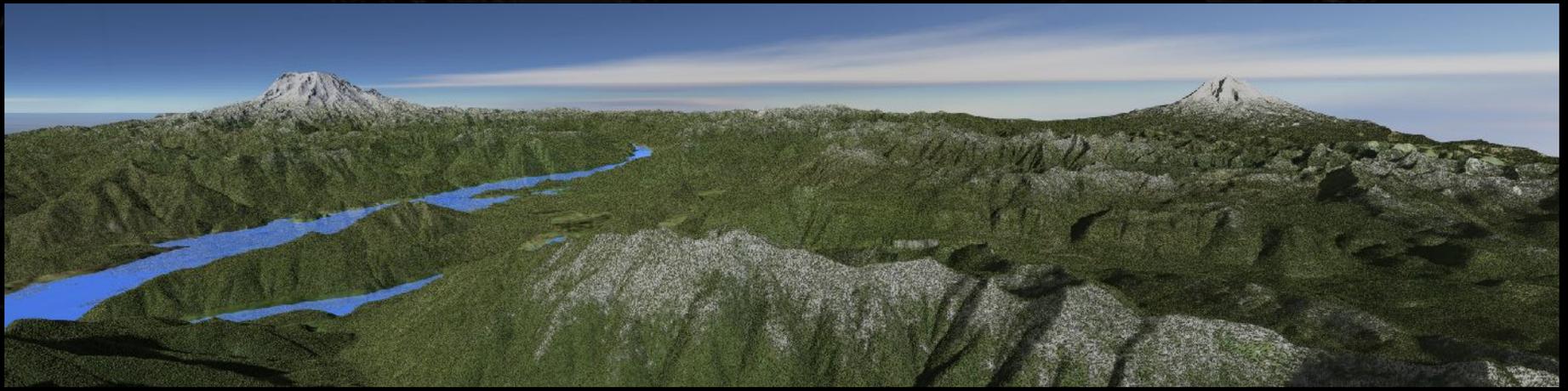
Und wenn es noch viel komplizierter wird?

- **Landschaften mit Milliarden von Bäumen**
 - 80km x 80km im Nordwesten der USA
 - Das Grüne sind ist keine Farbe, keine Textur



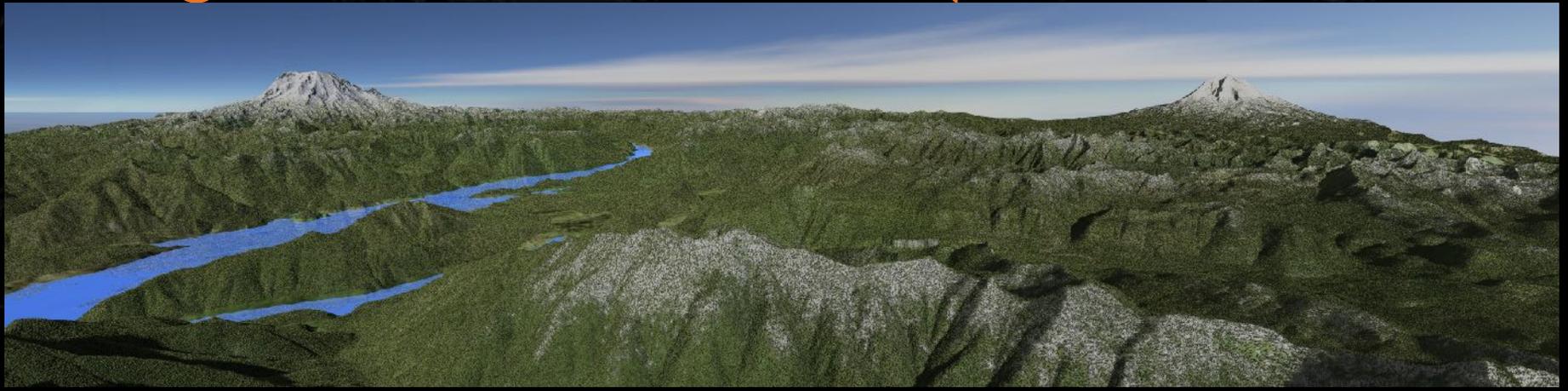
Und wenn es noch viel komplizierter wird?

- **Landschaften mit Milliarden von Bäumen**
 - 80km x 80km im Nordwesten der USA
 - Das Grüne sind ist keine Farbe, keine Textur
 - Das Grüne sind alles richtige Bäume !!



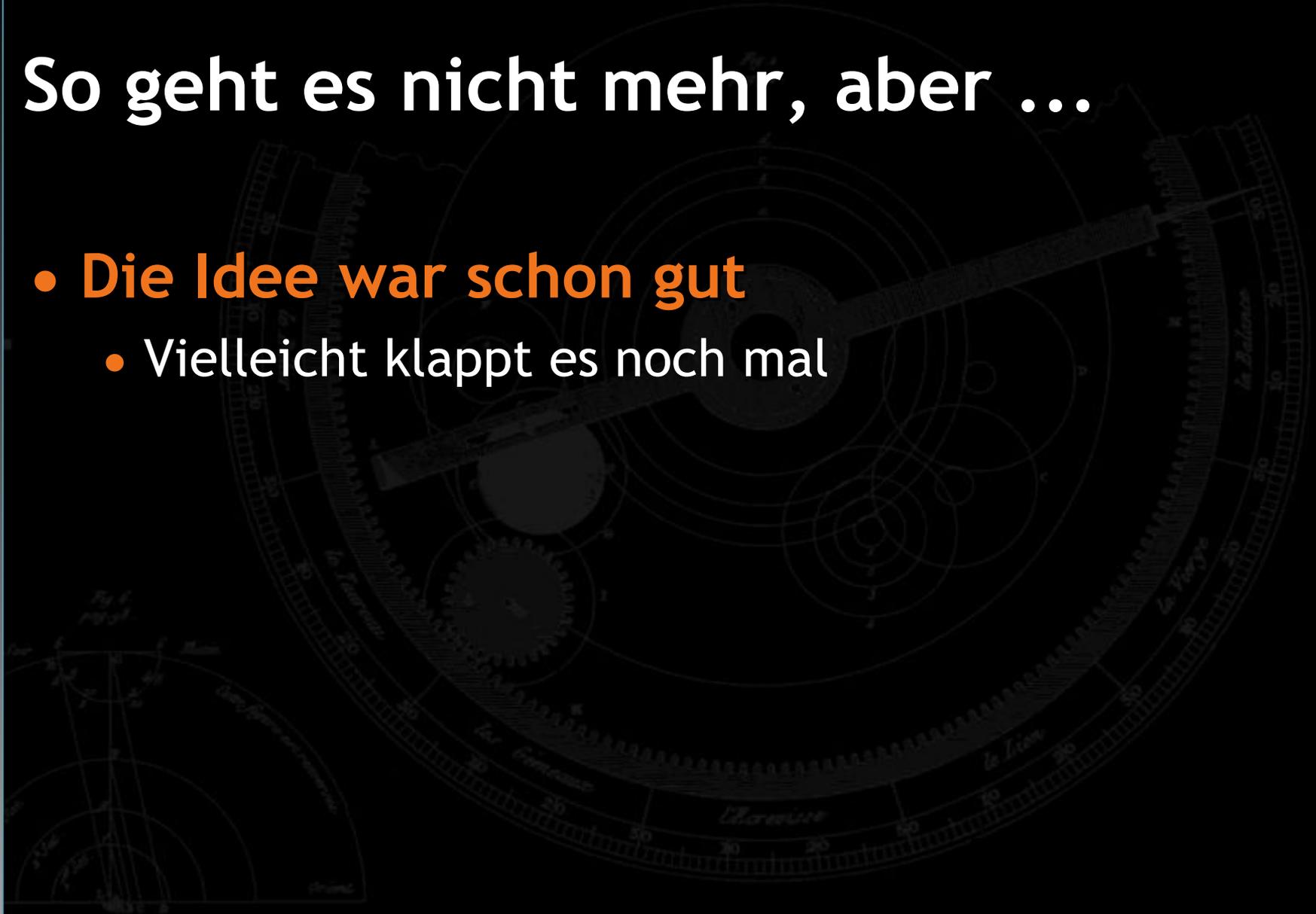
Und wenn es noch viel komplizierter wird?

- **Landschaften mit Milliarden von Bäumen**
 - 80km x 80km im Nordwesten der USA
 - Das Grüne sind ist keine Farbe, keine Textur
 - Das Grüne sind alles richtige Bäume !!
- **So geht das nicht mehr :-)**



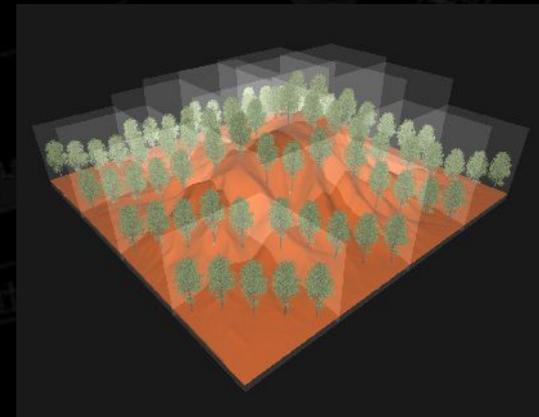
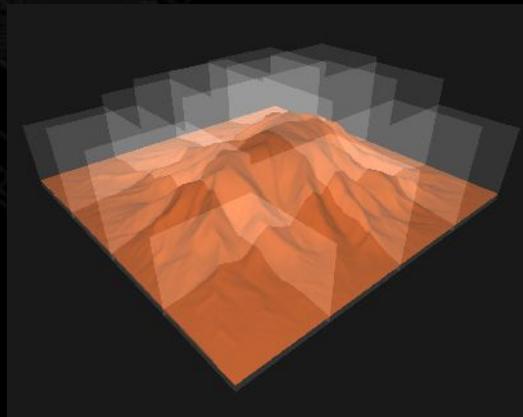
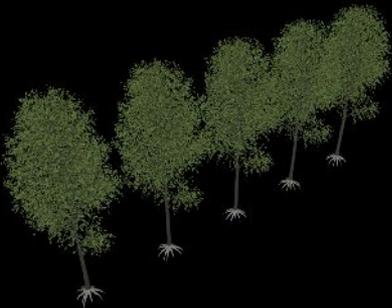
So geht es nicht mehr, aber ...

- **Die Idee war schon gut**
 - Vielleicht klappt es noch mal



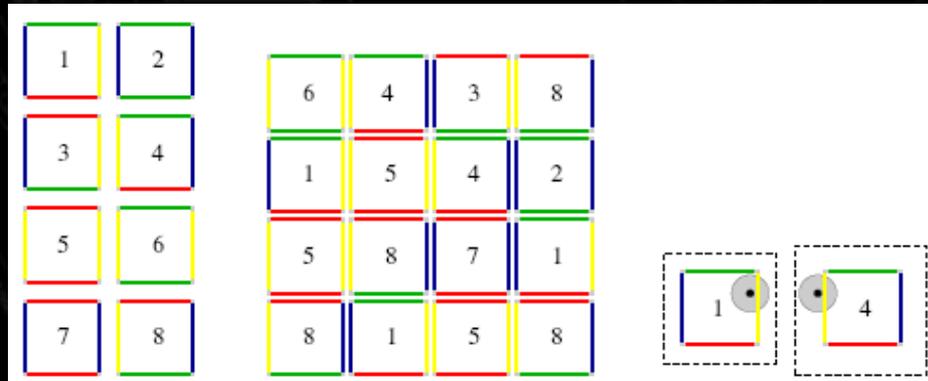
So geht es nicht mehr, aber ...

- **Die Idee war schon gut**
 - Vielleicht klappt es noch mal
- **Neue Idee: Packe Boxen in größere Boxen**
 - Und diese in noch größere Boxen,



Aber das wäre zu einfach ...

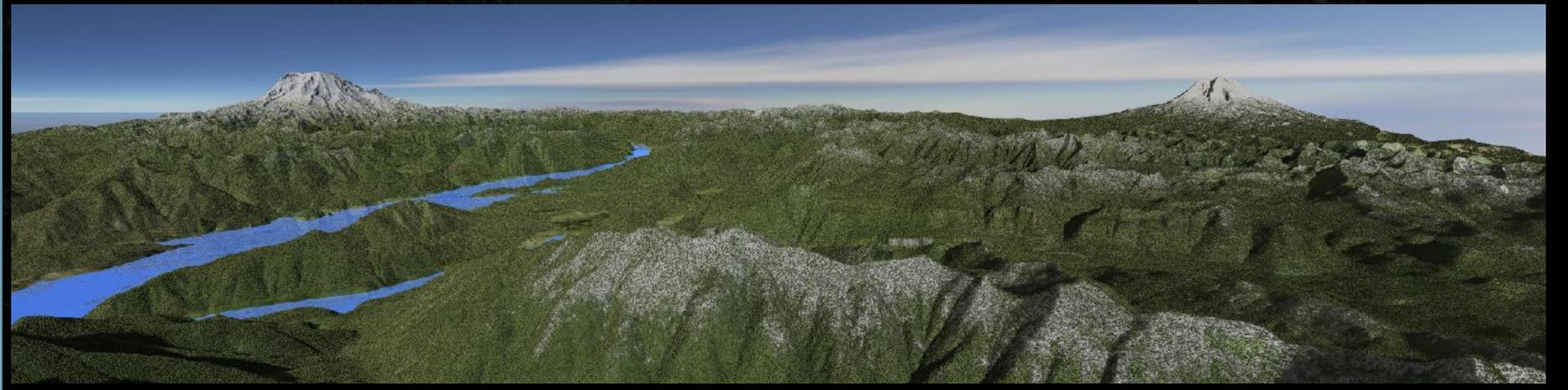
- **Nicht überall Bäume**
 - Boxen einfach weglassen (dynamische Bedingungen)
- **Nicht immer die gleichen Bäume**
 - Baue verschiedene Boxen und kombiniere sie



Aperiodische Kacheln mit Wand Tiles

Aber das wäre zu einfach ...

- **Bäume sind unterschiedlich hoch**
 - Nimm einfach die größte Box
- **Das Terrain is uneben (Berge)**
 - Verschiebe die Bäume in Abhängigkeit von der Position





Ziel:

Die Welt in virtuelle Realität

- **Nimm GoogleEarth ...**
 - Erdoberfläche mit Texturen
- **... und stelle alles drauf, was drauf gehört**
 - Bäume, Häuser, Menschen, Auto, ...
 - Simulation der Bewegung, ...
 - Erlaube beliebige Ansichten
- **Unzahl an Anwendungen**
 - Unter anderem ein realistisches „Second Life“

Zusammenfassung

- **Realtime-Ray-Tracing**
 - Schnelle, physikalisch-korrekte Visualisierung
- **Räumliche Indexstrukturen (kd-Baum)**
 - Erlaubt fast beliebige Skalierung der Szene
- **Akkurate Simulation der Beleuchtung**
 - Bisher fehlender Realismus in virtuellen Welten