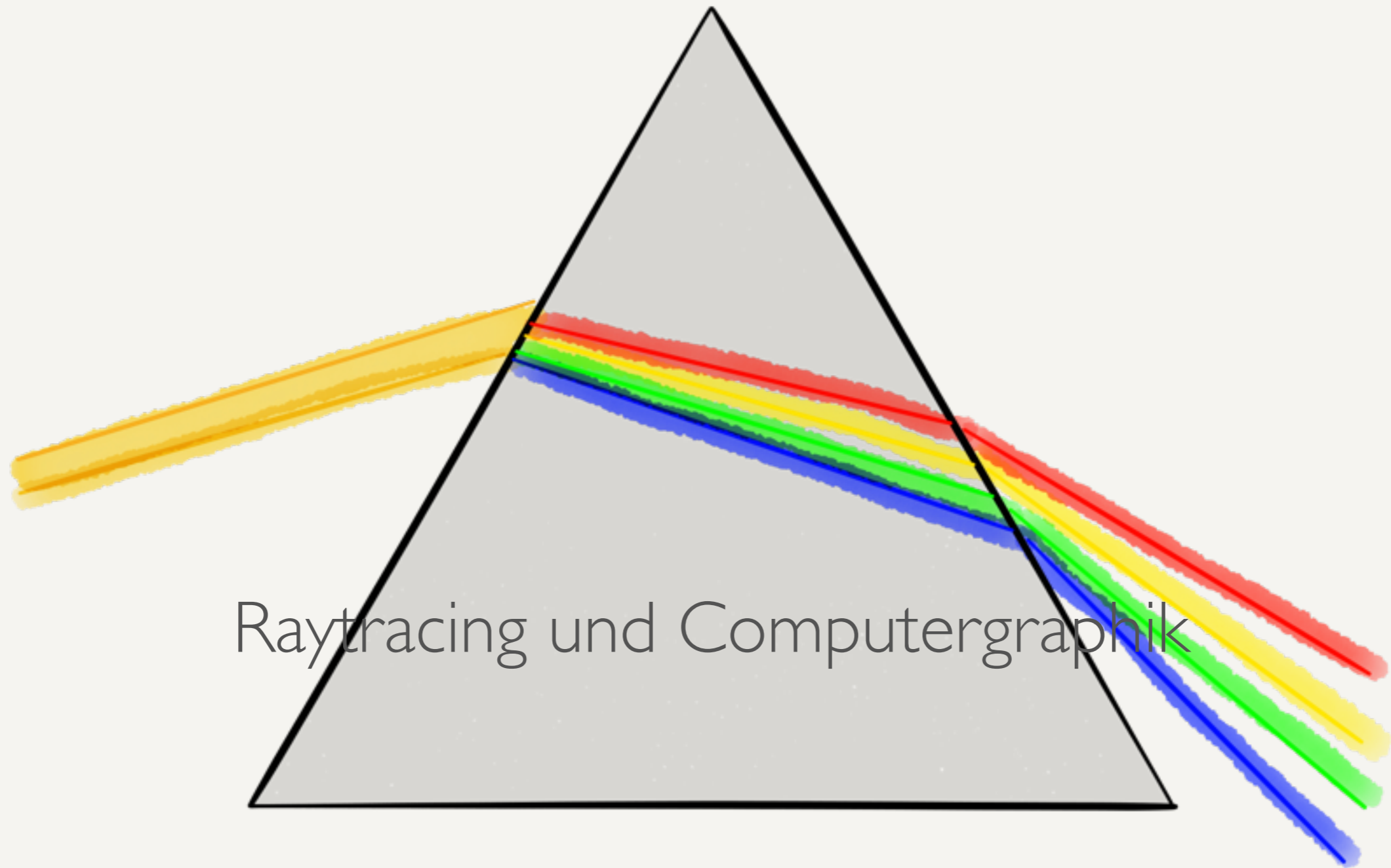


Photonik

Technische Nutzung von Licht



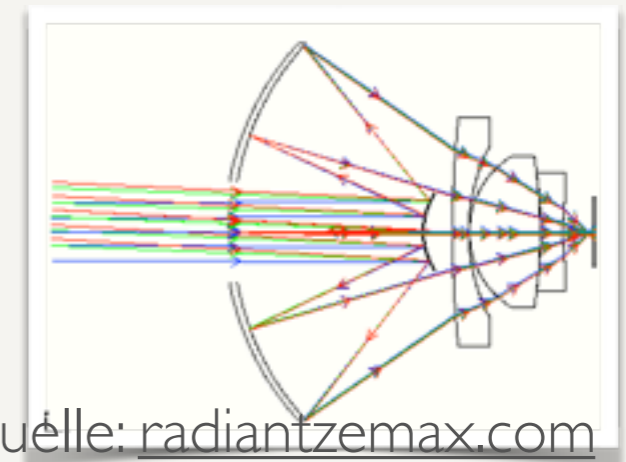
Raytracing und Computergraphik

Überblick

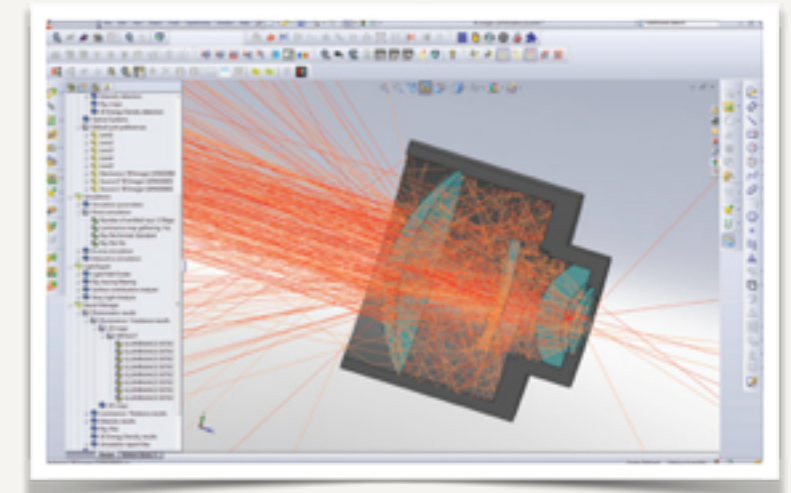
- Raytracing
- Typen von Raytracern
- z-Buffer

Raytracing

- „Lichtstrahlen-Verfolgung“ (*engl. ray tracing*): Berechnung von Lichtstrahlen nach geometrischer Optik.
- Drei große Bereiche:
 - ▶ **Graphisches Raytracing:** Berechnung, wie ein Bild aussieht.
 - ▶ **Sequentielles Raytracing:** Reihenfolge der brechenden Flächen vorgegeben. Für Optik-Design (Objektive, Mikroskope, Ferngläser, ...).
 - ▶ **Nicht-sequentielles Raytracing:** Reihenfolge der Flächen nicht vorgegeben. Für Aus- und Beleuchtung.



Quelle: radiantzemax.com



Quelle: Laserfocusworld

Raytracing

- „Lichtstrahlen-Verfolgung“ (*engl. ray tracing*): Berechnung von Lichtstrahlen nach geometrischer Optik.
- Drei große Bereiche:
 - ▶ **Graphisches Raytracing:** Berechnung, wie ein Bild aussieht.
 - ▶ **Sequentielles Raytracing:** Reihenfolge der brechenden Flächen vorgegeben. Für Optik-Design (Objektive, Mikroskope, Ferngläser, ...).
 - ▶ **Nicht-sequentielles Raytracing:** Reihenfolge der Flächen nicht vorgegeben. Für Aus- und Beleuchtung.

Gutes Aussehen und schnelle
Berechnung wichtiger als
physikalische Konsistenz



Wissenschaftliche und
technische
Anwendungen mit
physikalisch korrekten
Modellen

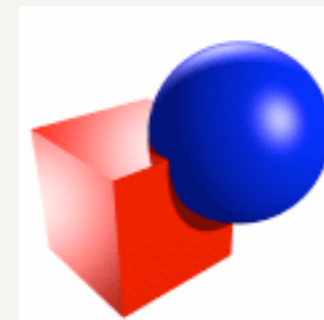


Objekte

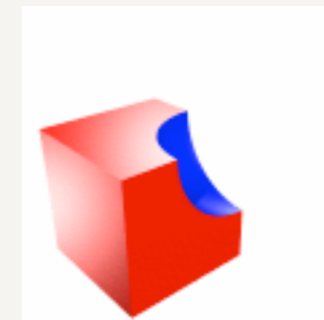
- Modellierung der Objekte im Raum.
- Aufgebaut aus sogenannten graphischen Primitiven: Kreise (Kugeln), Rechtecke (Würfel), Polygone, Dreiecke, Zylinder, Rotationskörper, ...
- Beispiel: CSG - Constructive Solid Geometry
- Szenenerstellung typischerweise mit einem spezialisierten 3d-Modellierungswerkzeug
- Komplexe Objekte werden heutzutage aus Dreiecken zusammengesetzt.

Making of a Rose

Vereinigung



Differenz



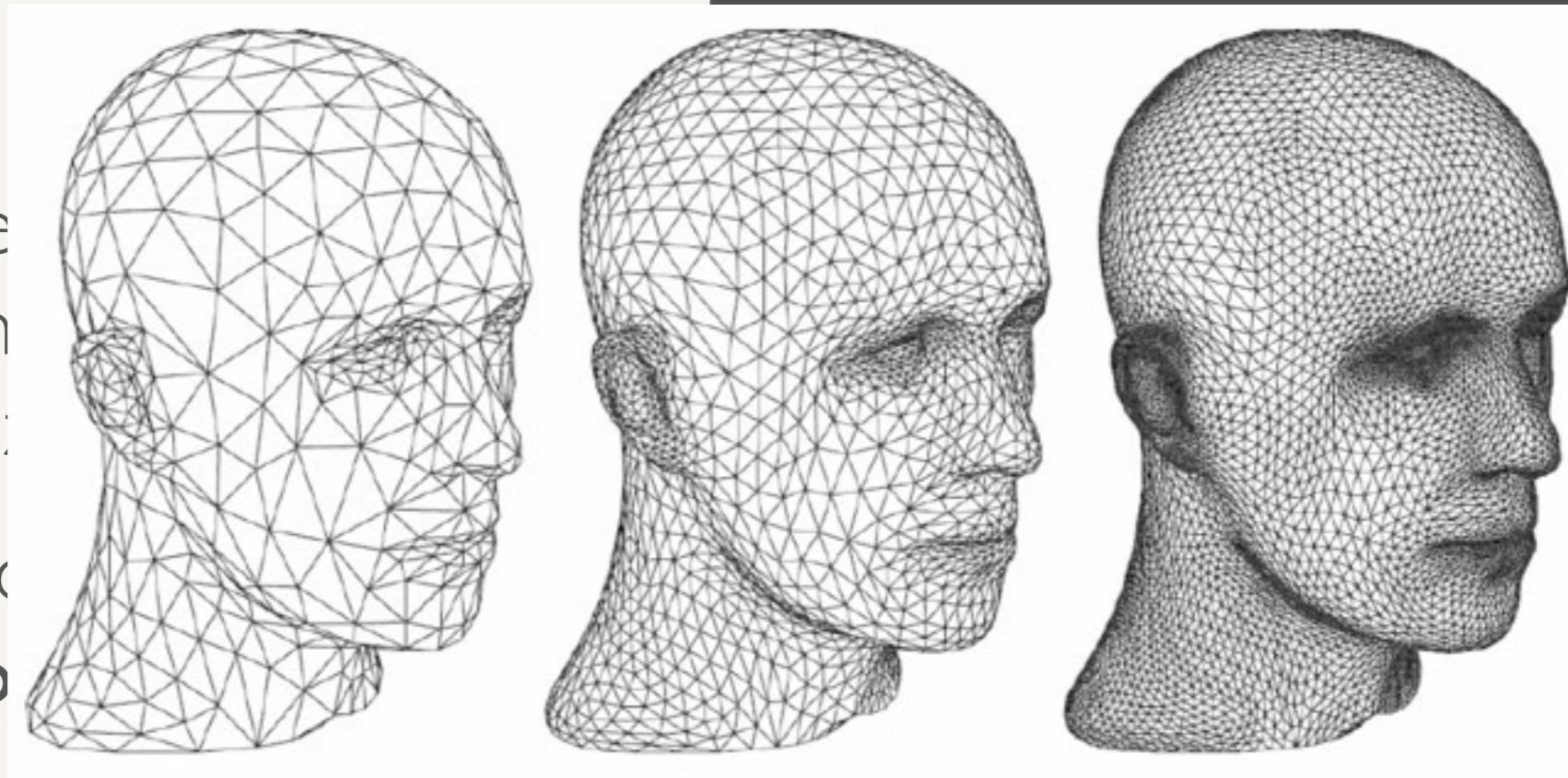
Schnitt



http://de.wikipedia.org/wiki/Constructive_Solid_Geometry

Tessellation

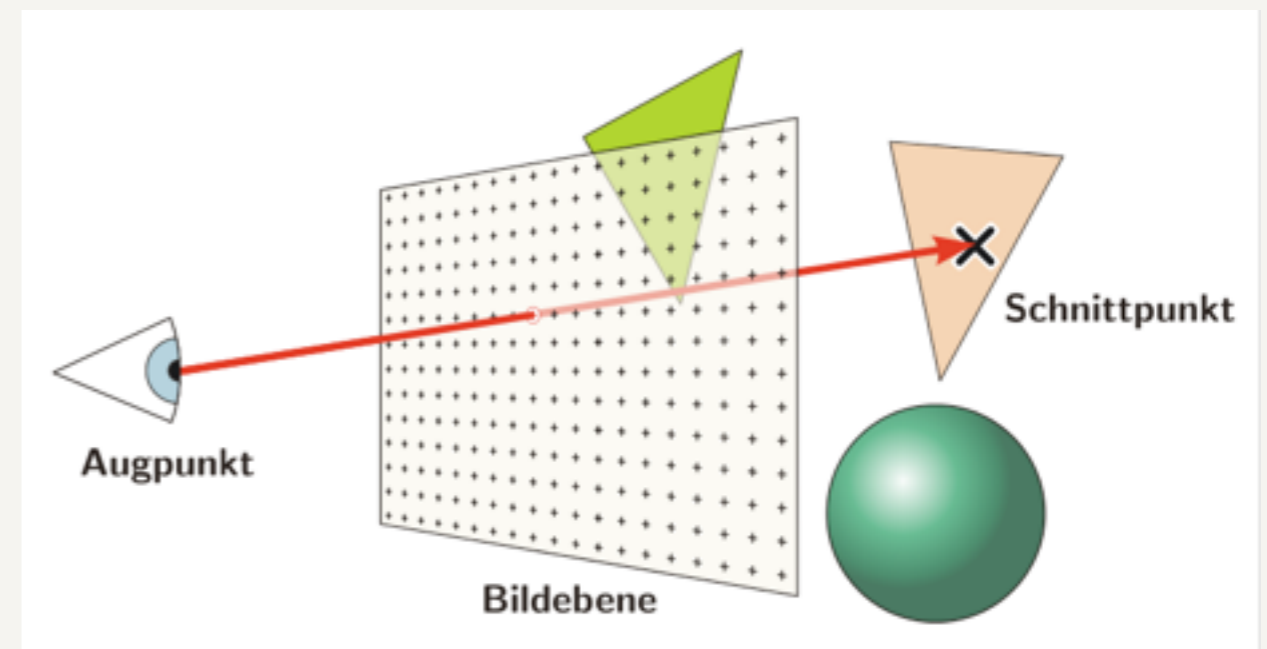
- Komplexe
werden in
Dreiecke
- Dieser Pro
Tessellatio



http://www.guru3d.com/articles_pages/radeon_hd_5870_review_test,7.html

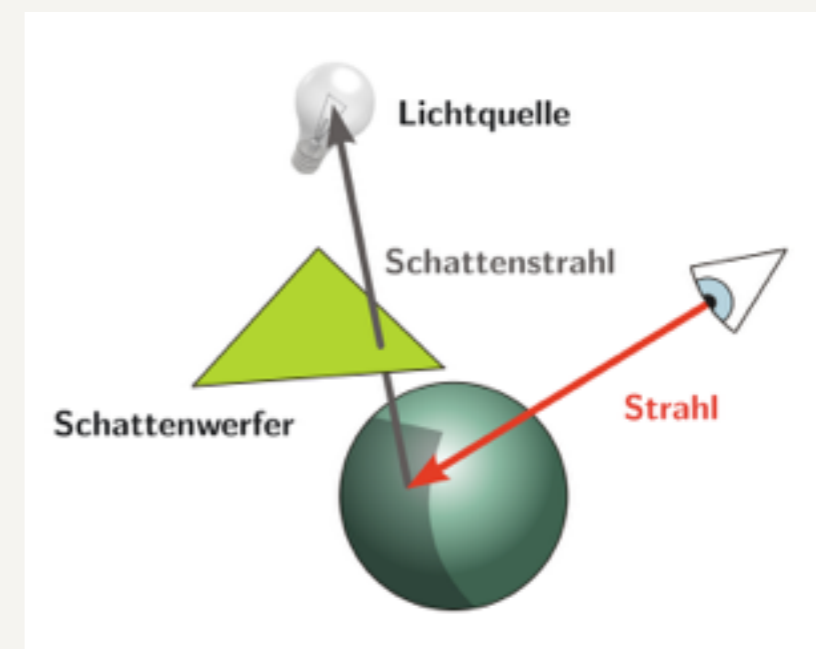
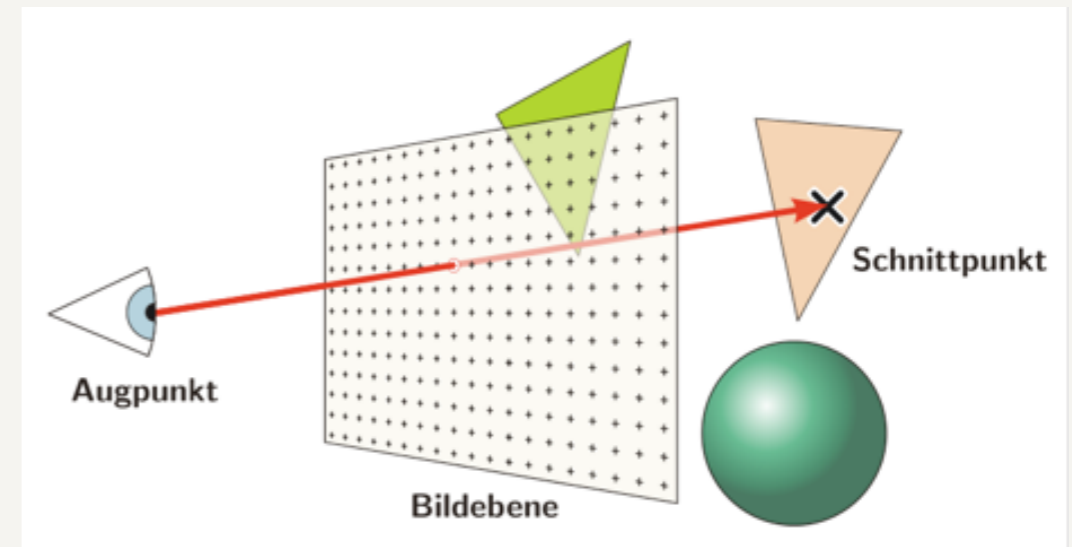
Graphisches Raytracing

- Auge ‚schießt‘ Lichtstrahlen durch die Bildebene.
- Wie Lochkamera, nur dass Apertur und Bildfläche vertauscht sind.
- Es wird berechnet, auf welches Objekt der Lichtstrahl zuerst trifft.
- Das Pixel bekommt die Farbe des Objektes an der getroffenen Stelle.



Algorithmus

- Berechne möglichen Schnittpunkt mit allen Primitiven der Szene
- Das Objekt mit der geringsten Entfernung ,gewinnt‘.
- Bestimmung der Farbe des getroffenen Objekts
- Berücksichtige Lichtquellen
- Berücksichtige Schatten



Schnittpunkt-Berechnung

- Herzstück eines Raytracers
- Analytische Darstellung für viele sogenannte Primitive (Kugeln, Zylinder, Polygone, ...)
- Abschätzung der Kosten (Rechenzeit) möglich.
- Bei der Schnittpunktberechnung wird die meiste Rechenzeit verbraucht.
- Kosten werden nach Grad des Lösungspolynoms sortiert: Wurzel aus einem Polynom 2. Grades, oder Polynom 6. Grades.
- Erinnerung: s. Schnittpunkt Grade-Kreis aus der Excel-Übung

Schnittpunkt Gerade - Kreis

$$K : (\mathbf{x} - \mathbf{m})^2 = r^2$$

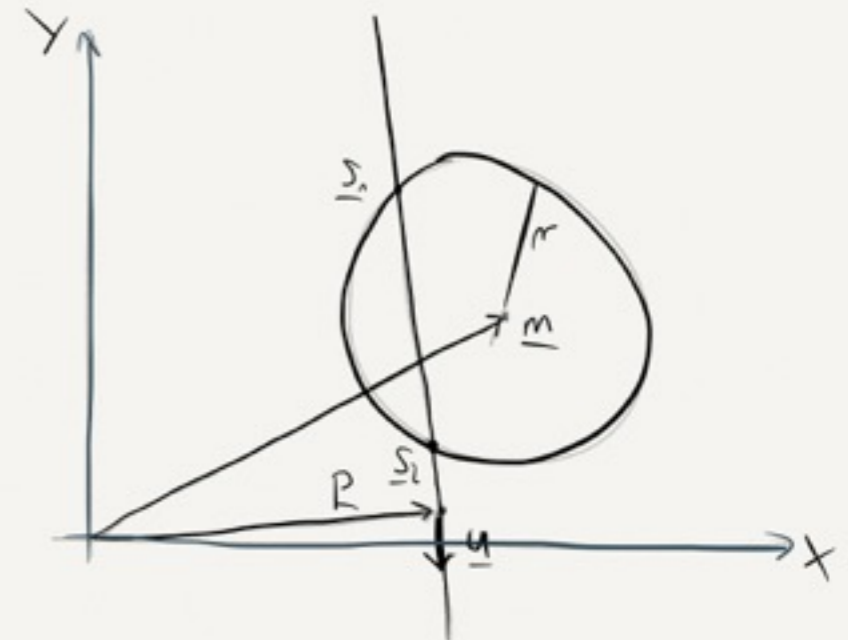
$$g : \mathbf{p} + t \cdot \mathbf{u}$$

Einsetzen:

$$r^2 = (\mathbf{p} + t \cdot \mathbf{u} - \mathbf{m})^2$$

$$0 = t^2 \cdot \mathbf{u}^2 + t \cdot (2\mathbf{u} \cdot \mathbf{p} - 2\mathbf{u} \cdot \mathbf{m}) - 2\mathbf{p} \cdot \mathbf{m} + \mathbf{p}^2 + \mathbf{m}^2 - r^2$$

Bei uns normiert, also = 1



Lösung:

$$t_{1,2} = \mathbf{u} \cdot \mathbf{m} - \mathbf{u} \cdot \mathbf{p} \pm \sqrt{(\mathbf{u} \cdot \mathbf{p} - \mathbf{u} \cdot \mathbf{m})^2 + 2\mathbf{p} \cdot \mathbf{m} + r^2 - \mathbf{p}^2 - \mathbf{m}^2}$$

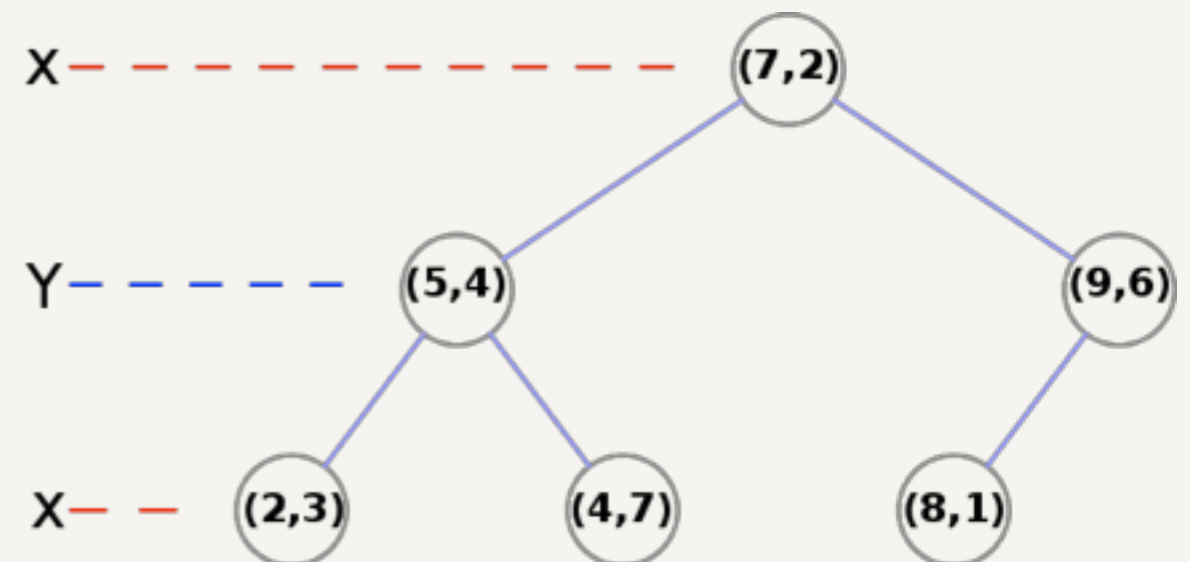
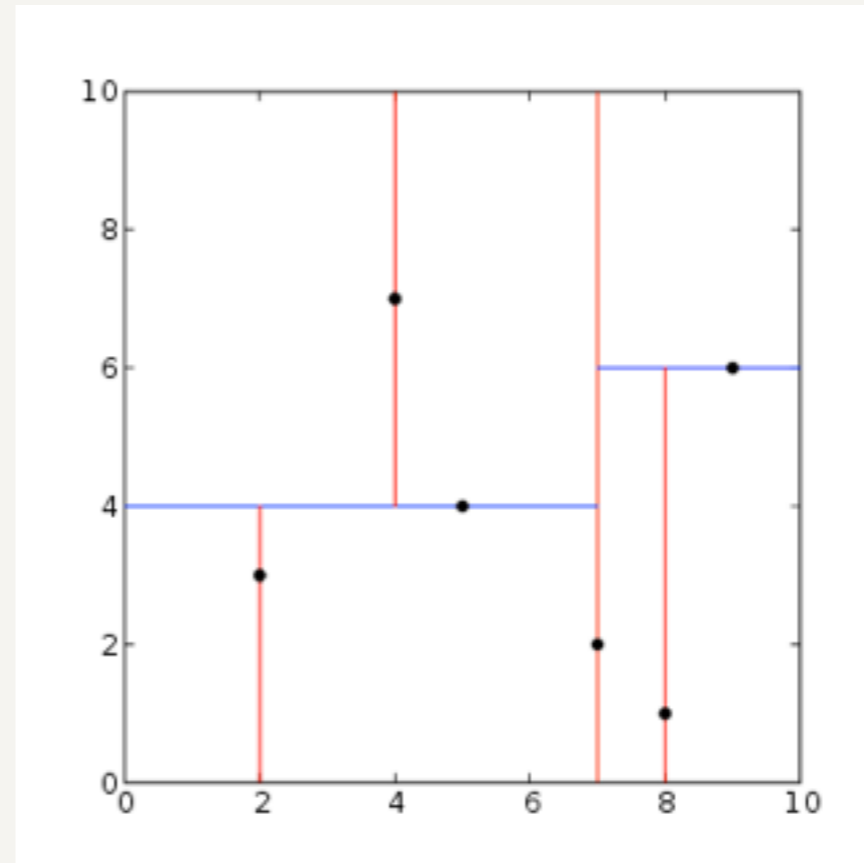
Radikant > 0 ?

$$\mathbf{s}_{1,2} = \mathbf{p} + t_{1,2} \cdot \mathbf{u}$$

Aufteilung des Raumes

- Baumstrukturen zur Unterteilung des Raumes und der Objekte darin.
- Eigenständiges großes mathematisches Forschungsgebiet
- Beispiel kd-Bäume, hier: 2d-Baum

http://en.wikipedia.org/wiki/K-d_tree



Graphisches Raytracing

Eigenschaften

- Physikalisch meist nicht korrekt oder ungenau.
- Beispiel: Brechung nur mit einem Brechungsindex, keine Dispersion.
- Für viele Eigenschaften eigene Techniken: Brechung, Kaustik, Dispersion, diffuse Schatten, globale Beleuchtung, ...
- Großes eigenständiges Forschungsgebiet.

<http://www.povray.org/>

POVray





<http://www.oyonale.com/image.php?code=464>

Spectral rendering

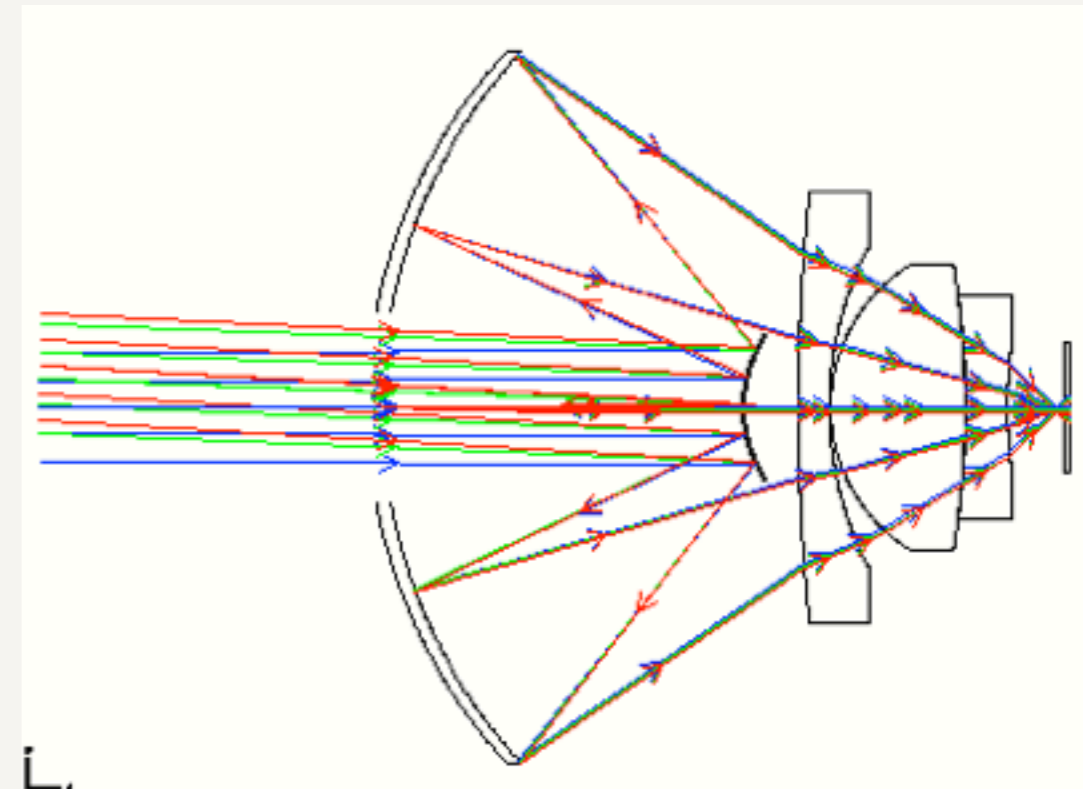
POVray



Sequentielles Raytracing

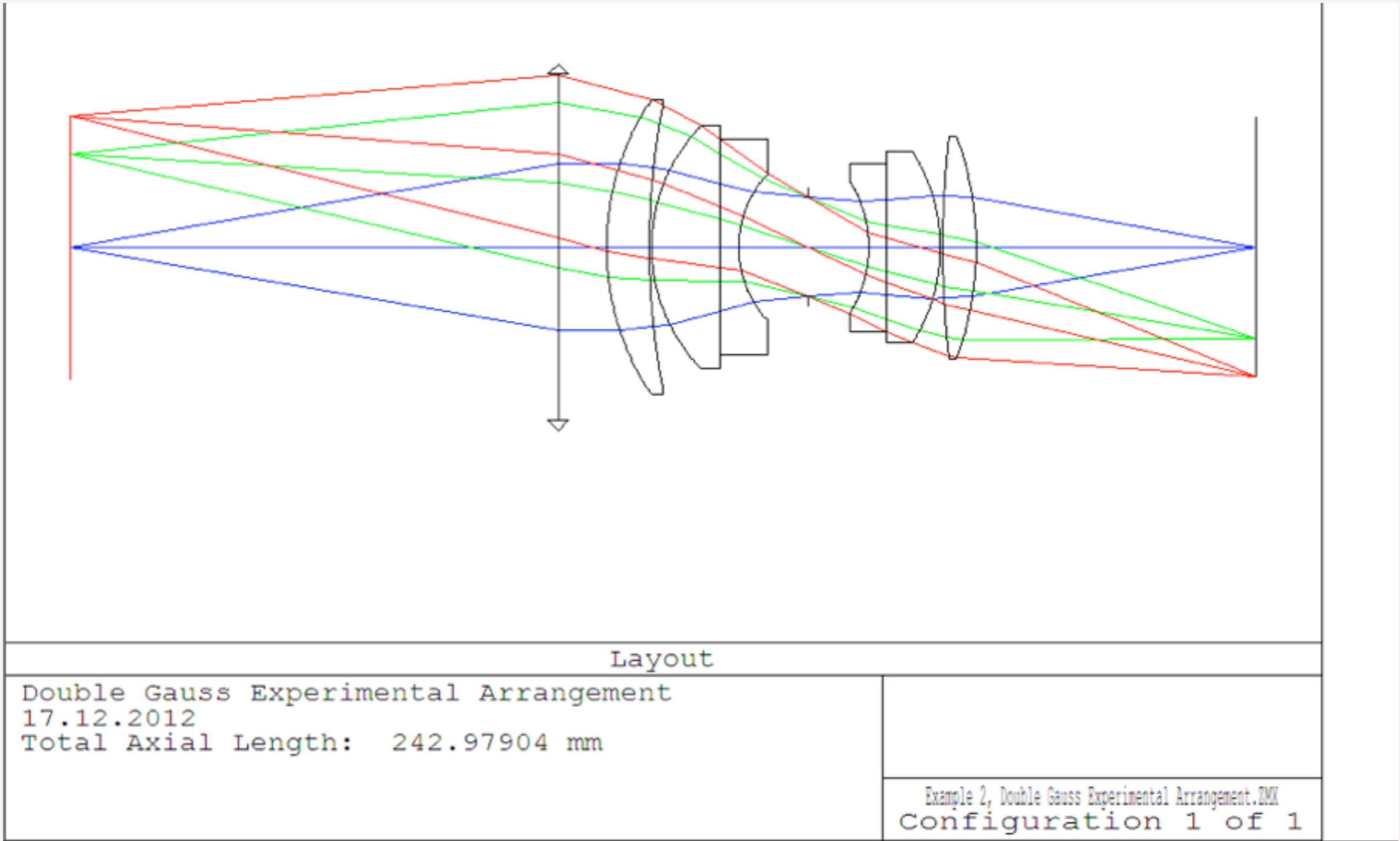
Quelle: radiantzemax.com

- Fest vorgegebene Reihe von optischen Flächen (Reflexion, Brechung)
- Raytracing von einem *Objektpunkt* zu einer *Bildebene*, keine Objekte.
- Schnittpunktberechnung einfach immer mit der nächsten Fläche
- Physikalisch korrekte Modellierung der Fresnel'schen Formeln, Snellius'sche Brechungsgesetz und Dispersion
- Zielsetzung: Entwurf und Optimierung von optischen Geräten (Objektive, Prismen, ...)



Zemax: keine Lichtquellen, sondern Objektpunkte werden abgebildet.

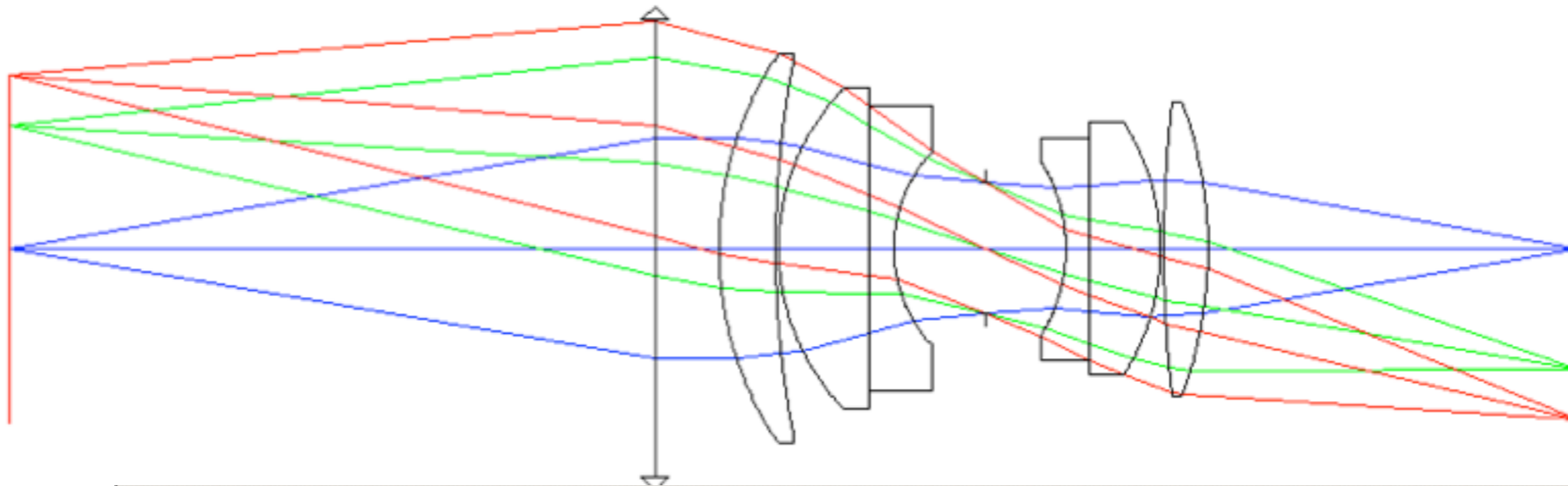
Zemax: Beispiel



Zemax: Beispiel

Edit Solves View Help											
Surf:Type		Comment	Radius		Thickness		Glass		Semi-Diameter	Conic	Par O (unu
OBJ	Standard		Infinity		100.000				27.000	0.000	
1	Paraxial				10.000				35.451		
2	Standard		54.153	V	8.747	N-SK2		30.256	0.000		
3	Standard		152.522	V	0.500			29.289	0.000		
4	Standard		35.951	V	14.000	N-SK16		24.916	0.000		
5	Standard		Infinity		3.777	F5		22.123	0.000		
6	Standard		22.270	V	14.253			15.174	0.000		
STO	Standard		Infinity		12.428			10.229	U	0.000	
8	Standard		-25.685	V	3.777	F5 P		13.587	0.000		
9	Standard		Infinity		10.834	N-SK16 P		17.239	0.000		
10	Standard		-36.980	V	0.500			19.676	0.000		
11	Standard		196.417	V	6.858	N-SK16 P		22.525	0.000		
12	Standard		-67.148	V	57.305	V		22.799	0.000		
IMA	Standard		Infinity		-			26.549	0.000		

Zemax: Beispiel



Edit Solves View Help								
Surf	Type	Comment	Radius	Thickness	Glass	Semi-Diameter	Conic	Par O (unu)
OBJ	Standard		Infinity	100.000		27.000	0.000	
1	Paraxial			10.000		35.451		
2	Standard		54.153 V	8.747	N-SK2	30.256	0.000	
3	Standard		152.522 V	0.500		29.289	0.000	
4	Standard		35.951 V	14.000	N-SK16	24.916	0.000	
5	Standard		Infinity	3.777	F5	22.123	0.000	
6	Standard		22.270 V	14.253		15.174	0.000	
STO	Standard		Infinity	12.428		10.229 U	0.000	
8	Standard		-25.685 V	3.777	F5 P	13.587	0.000	
9	Standard		Infinity	10.834	N-SK16 P	17.239	0.000	
10	Standard		-36.980 V	0.500		19.676	0.000	
11	Standard		196.417 V	6.858	N-SK16 P	22.525	0.000	
12	Standard		-67.148 V	57.305 V		22.799	0.000	
IMA	Standard		Infinity	-		26.549	0.000	

Double
17.12.20
Total A

.MX
f 1

Sequentielles Raytracing

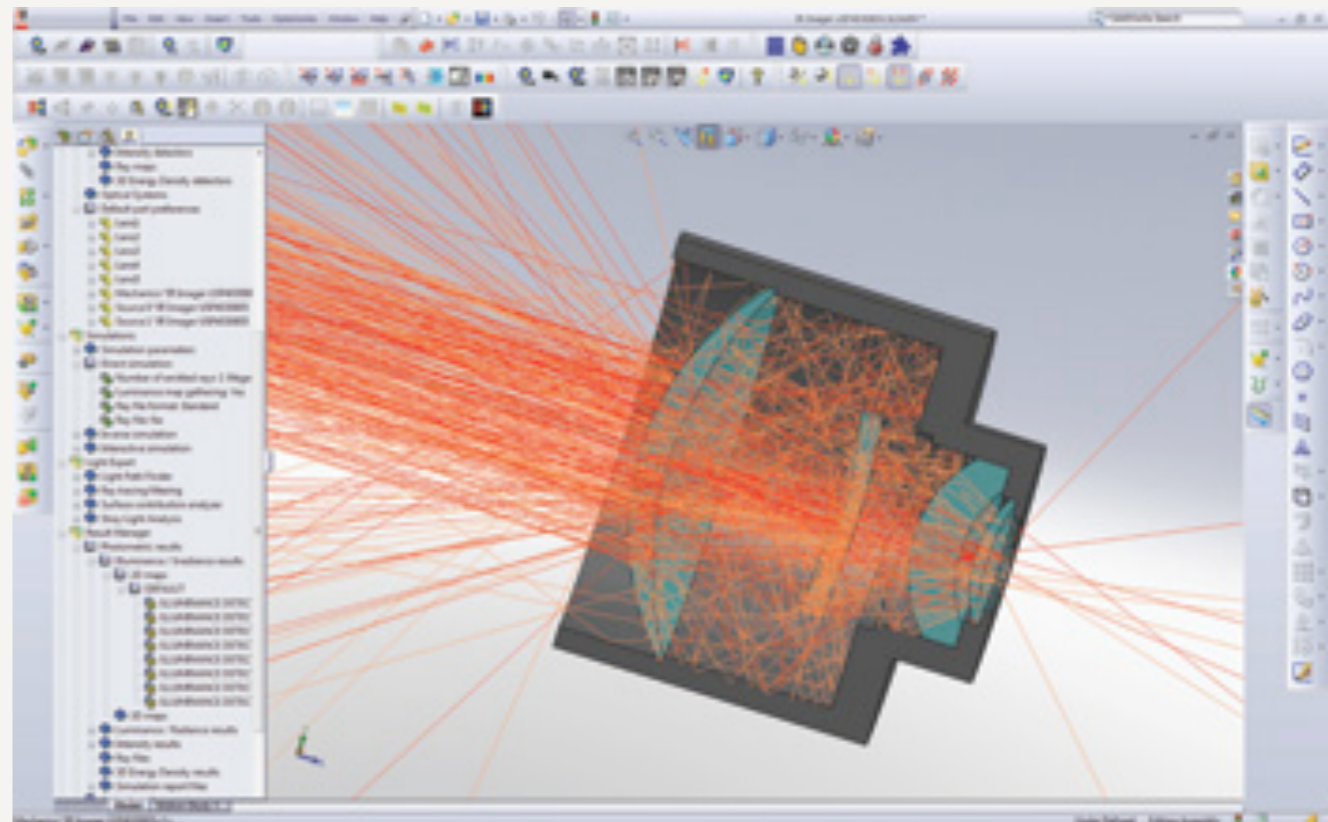
Eigenschaften

- Viele verschiedene Analysemöglichkeiten: MTF, Bild-Geometrie, Spektren, mechanische und thermische Toleranzrechnungen
- Output entsprechend eher analytischer Natur: MTF-Kurven, Spektren, Toleranzgraphen, ...

Nicht-sequentielles Raytracing

<http://www.youtube.com/watch?v=gjKDkGYpC7s>

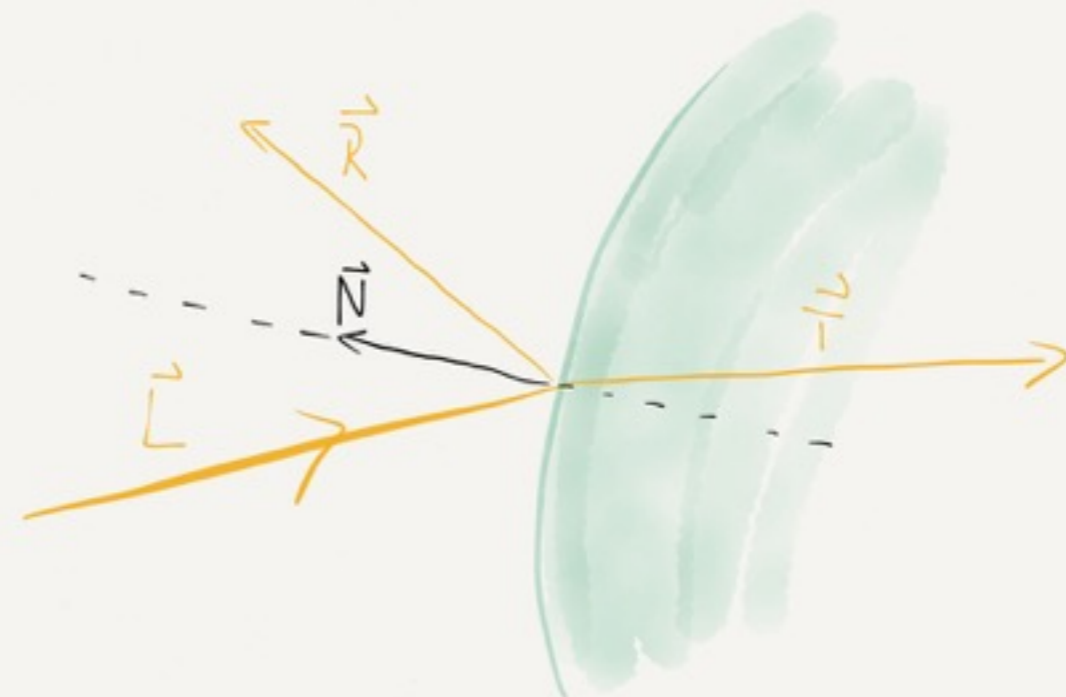
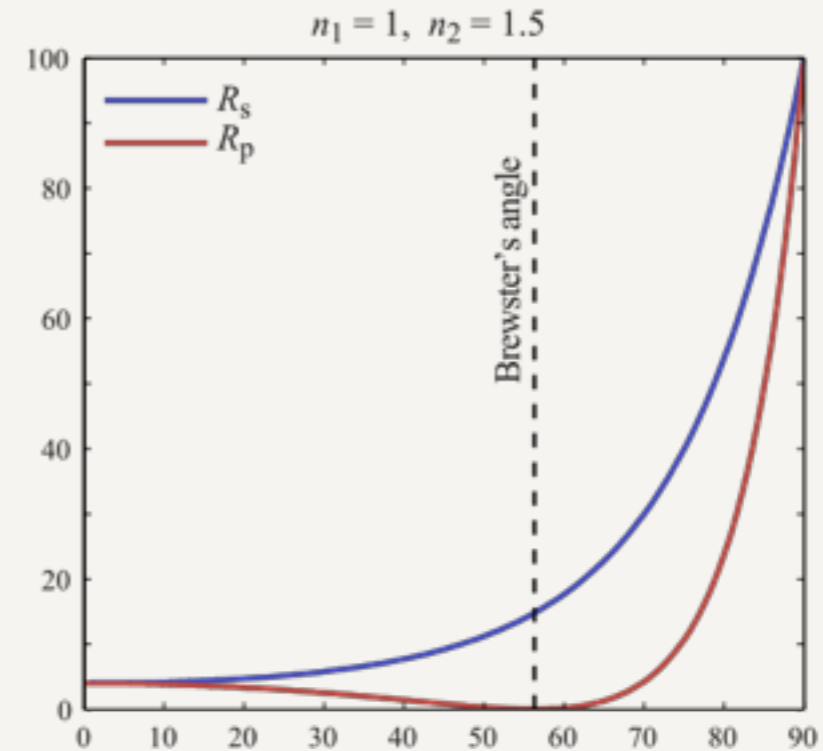
- Strahlen werden durch eine Lichtquelle erzeugt und bis zu einem Detektor verfolgt.
- Die Reihenfolge der beteiligten Flächen ist nicht vorgegeben.
- Eignet sich vor allem für die Aus- und Beleuchtung
- Benötigt wesentlich mehr Strahlen als sequentielles Raytracing.
- Vielen Strahlen landen im ‚Nichts‘.



Quelle: [Laserfocusworld](http://www.laserfocusworld.com)

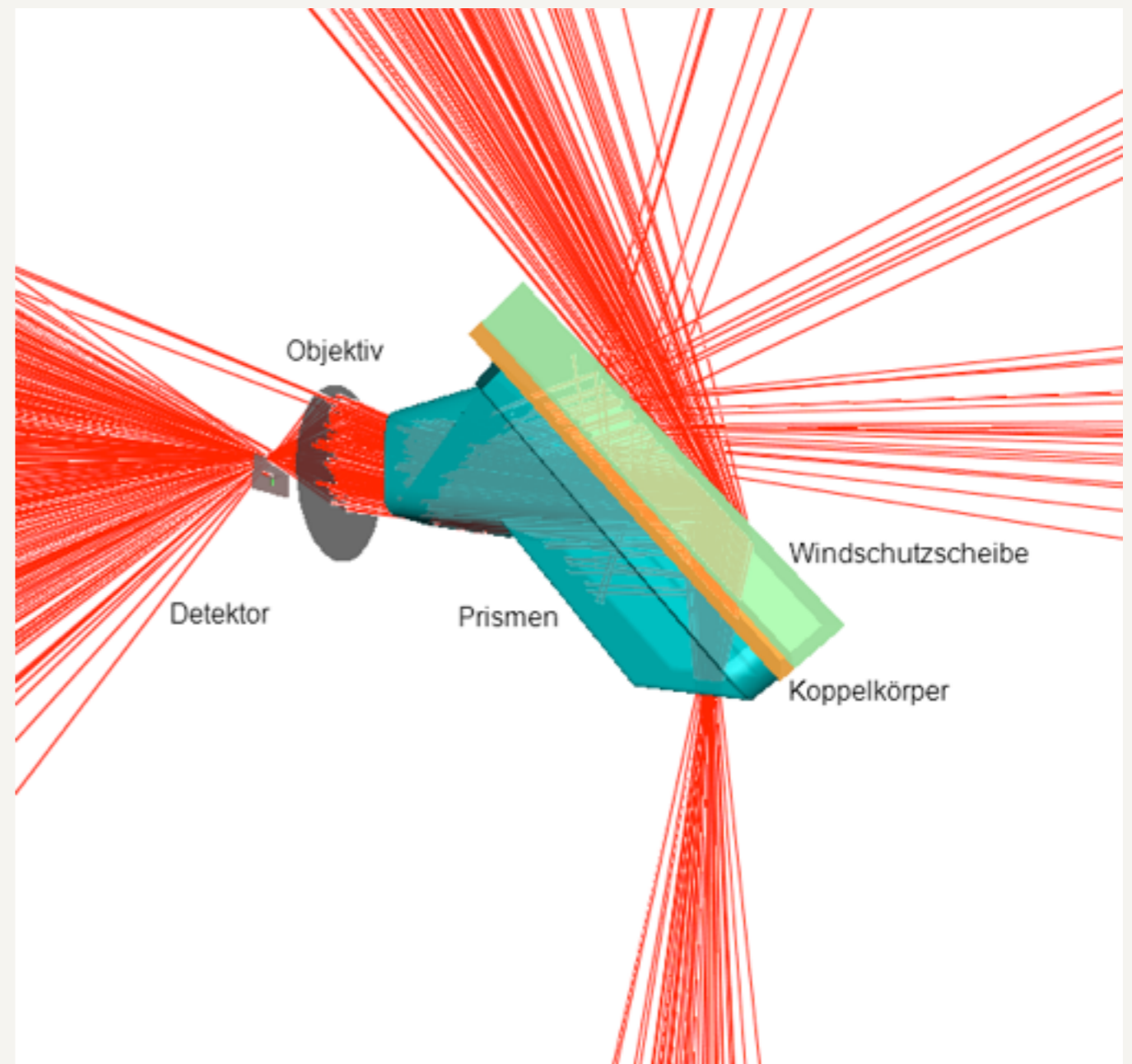
Fresnel'schen Formeln im Raytracing

- Strahlaufspaltung nach Fresnel:
 - ▶ Anzahl
 - ▶ Intensität
- Bei Anzahl wird zufällig bestimmt, ob reflektiert oder gebrochen wird.
- Bei Intensität wird ein neuer Strahl erzeugt und die Lichtintensität auf die beiden Strahlen laut Fresnel'scher Formeln aufgeteilt.



Speos

- Beispiel für nicht-sequentielles Raytracing mit echten 3d-Daten - Konstruktion aus CATIA
- Viele Reflexe
- Viele Strahlen enden im ‚Nichts‘



Blender

- Sehr mächtiges, schwierig zu bedienendes 3d-Tool
- Geeignet für Raytracing, Animationen, Spiele und Filme



<http://www.blender.org/features-gallery/features/>

<http://www.bigbuckbunny.org/>

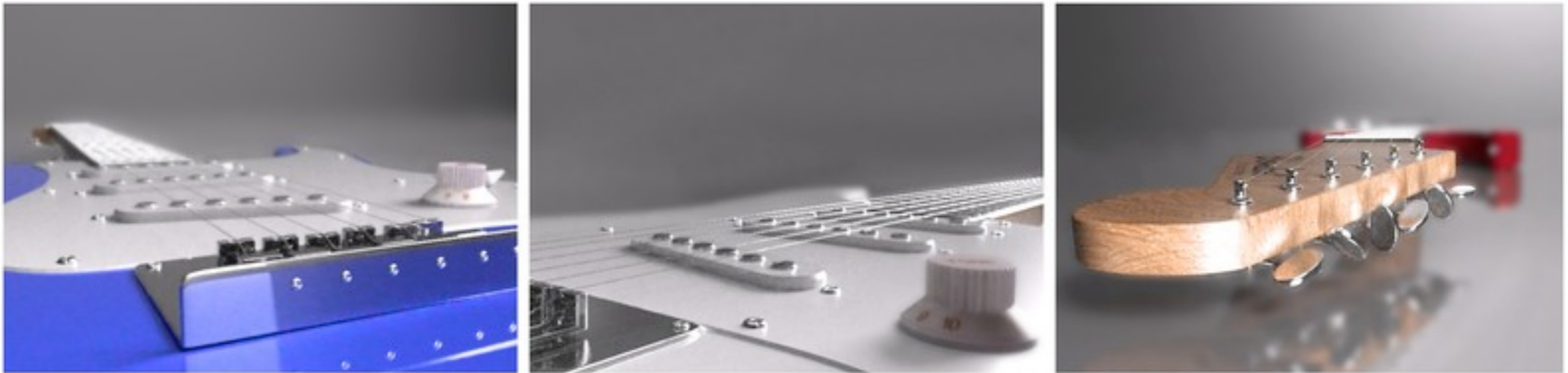
Blender



Blender



Blender

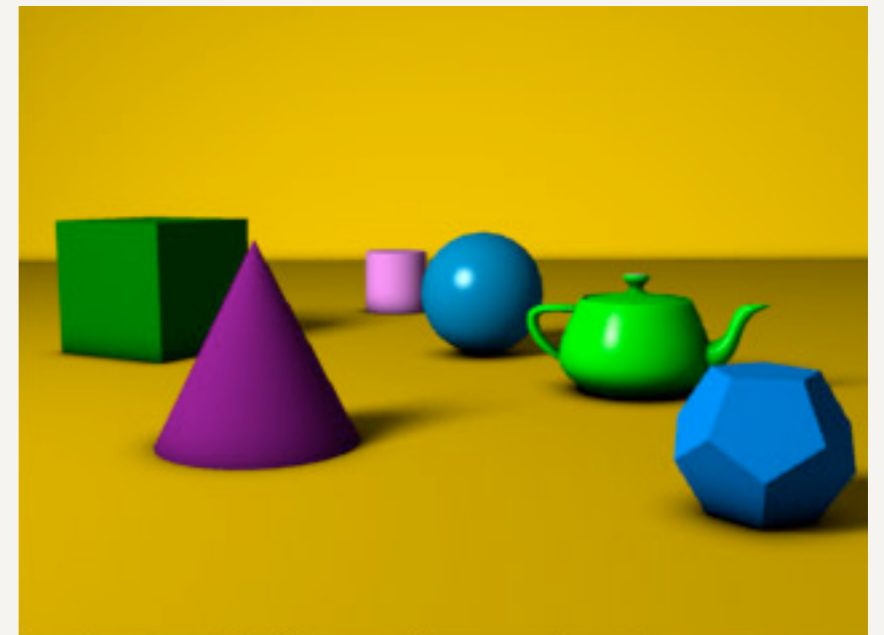


Blender[®]
Stratocaster

z-Buffer

- Standard Methode der graphischen 3d-Berechnung ohne Raytracing.
- Wird in allen Computerspielen eingesetzt.
- Die Tiefeninformation der Objekte (in Blickrichtung der Kamera) wird geeignet in einem eigenen Arbeitsspeicher abgelegt: dem z-Buffer.
- Beim Zeichnen des Bildes wird überprüft, welches Objekt ganz vorne liegt und somit zu sehen ist.

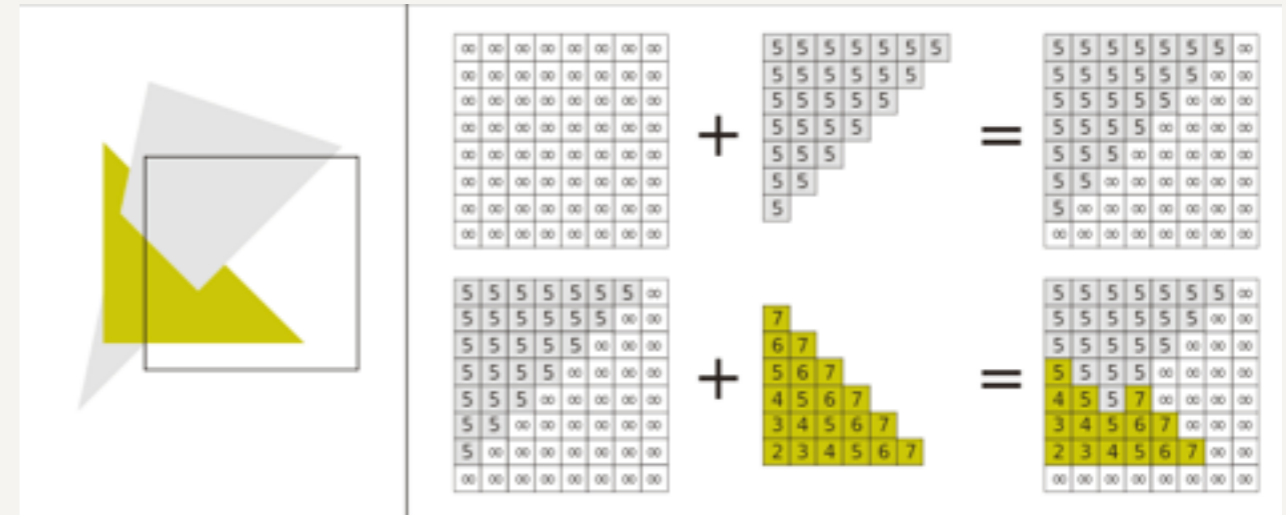
3d-Szene



z-Buffer

Algorithmus

- Projektion aller Primitive auf die Bildfläche (perspektivische (Zentral-) oder orthogonale Projektion)
- Für jedes Primitiv und für dessen projizierte Pixel:
 - ▶ Tiefenwert berechnen
 - ▶ Prüfe ob der z-Buffer-Wert größer ist als der aktuelle Eintrag.
 - ▶ Wenn ja: ersetzen.

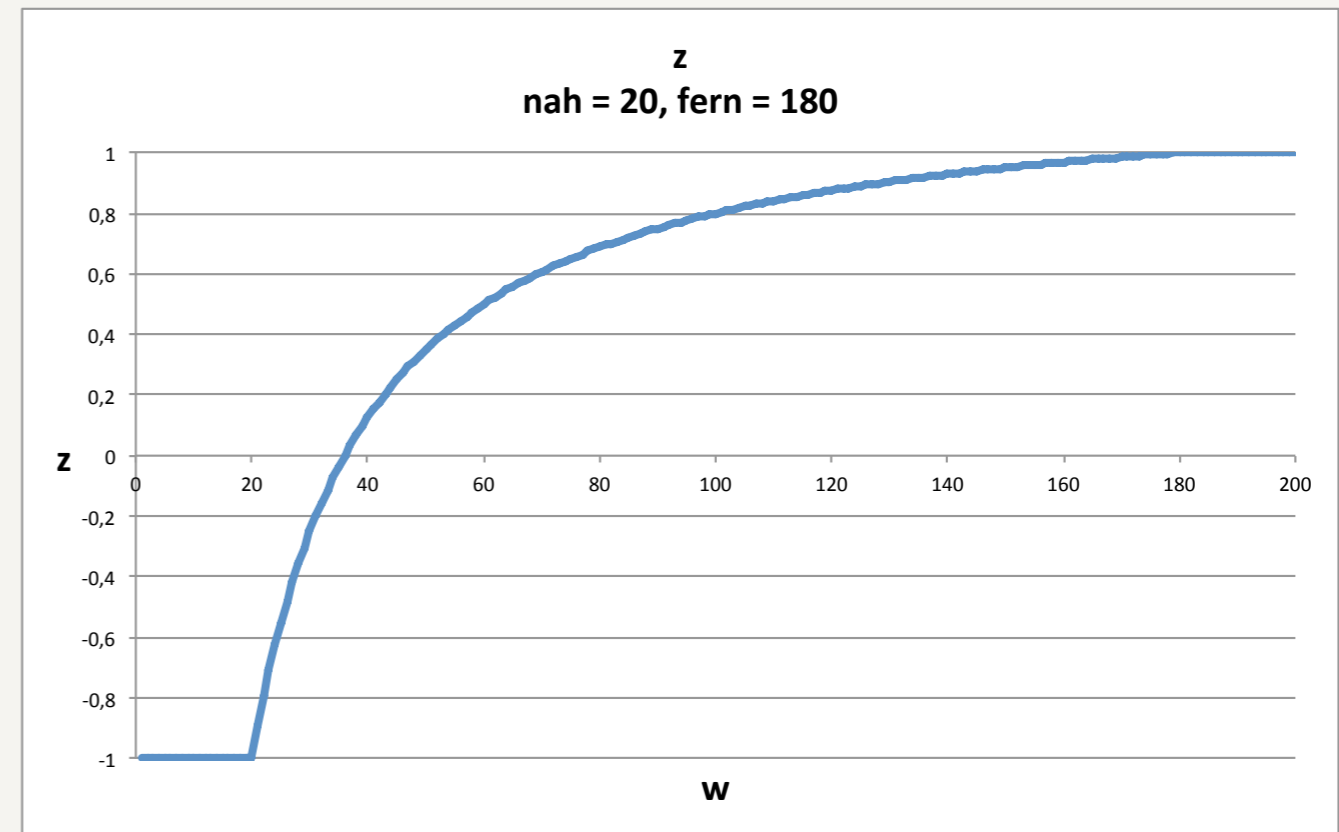


```

for(each polygon P in the polygon list) do{
  for(each pixel(x,y) that intersects P) do{
    Calculate z-depth of P at (x,y)
    If (z-depth < z-buffer[x,y]) then{
      z-buffer[x,y]=z-depth;
      COLOR(x,y)=Intensity of P at(x,y);
    }
  }
}
    
```

Eigenschaften

- Heutzutage entweder 24bit oder 32bit Tiefenauflösung
- Blickgrenzen **nah** und **fern**.
- Werte außerhalb **nah** oder **fern** werden nicht abgebildet.
- Tiefenwerte w sind nicht linear auf die z -Werte verteilt: deutlich höhere Auflösung nahe der Kamera.

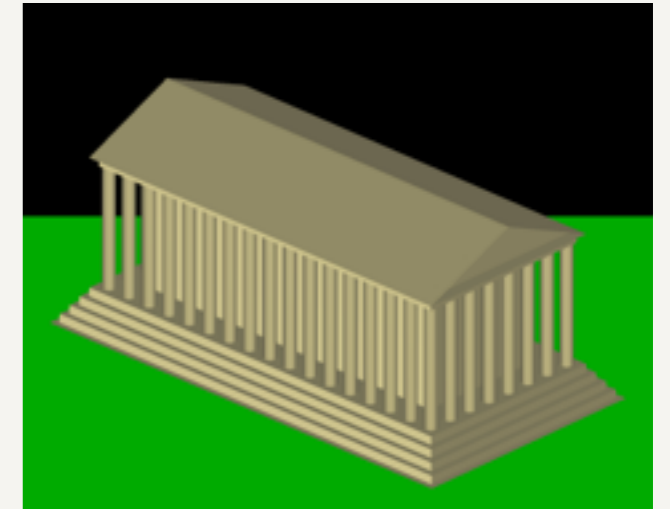


$$z = \frac{\text{fern} + \text{nah}}{\text{fern} - \text{nah}} + \frac{1}{w} \left(\frac{-2 \cdot \text{fern} \cdot \text{nah}}{\text{fern} - \text{nah}} \right)$$

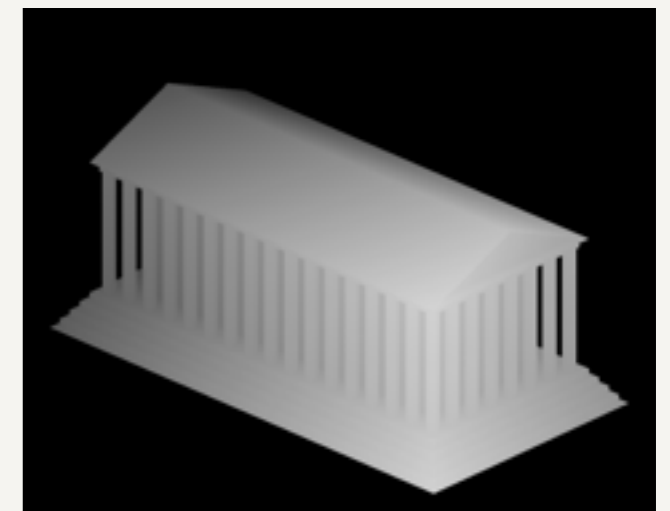
Shadow Mapping

- Auch ein z-Buffer, nur aus Sicht der Lichtquelle(n).
- Kleine Lichtquellen arbeiten mit perspektivischer Projektion, ausgedehnte mit orthogonaler (z.B. Sonne).
- Textur und Schattierung legen die endgültige Farbe des Pixels fest.

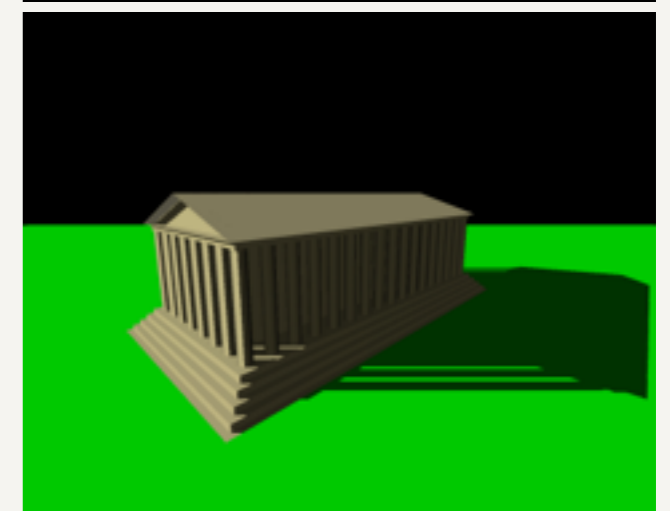
Objekt aus Sicht der Lichtquelle



z-Buffer aus Sicht der Lichtquelle



Objekt aus Sicht der Kamera



Beispiele für z-Buffer



<http://www.port-royale3.com/shared/images/screenshots/06.jpg>

Crysis 3



<http://www.ea.com/de/crysis-3/images/crysis-3-urbaner-regenwald>

Blender



Spectral rendering

POVray



Cloud-basiertes Raytracing

- Aktueller Rechenaufwand für Einzelplatzrechner zu hoch
- Erste Vorschläge für Raytracing in der Cloud

<http://www.wolfrt.de/>

<http://software.intel.com/en-us/articles/cloud-based-ray-tracing/>