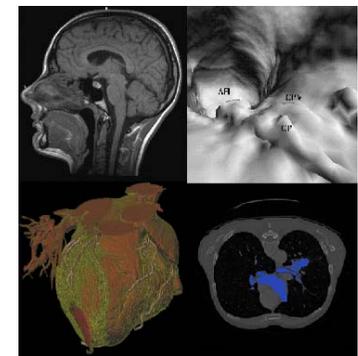
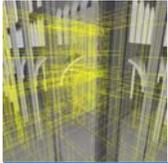


# Computergraphik

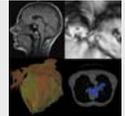
Thema: 6. Rendering  
Dozent: Prof. Dr. Dirk Bartz  
ICCAS  
dirk.bartz@medizin.uni-leipzig.de  
Sprechstunde: Do 17 Uhr (nach Vorlesung)  
Umfang: 2  
Prüfungsfach: Medizininformatik, Angewandte Informatik





# Inhaltsverzeichnis - Themen

Visual  
Computing (VCM)



- §1 Hardwaregrundlagen
- §2 Transformationen und Projektionen
- §3 Repräsentation und Modellierung von Objekten
- §4 Rasterung
- §5 Visibilität und Verdeckung

## §6 Rendering

...

### 6.6 Globale Beleuchtung

#### 6.6.1 Raytracing

#### 6.6.2 Radiosity-Verfahren

### 6.7 Renderpipeline

§7 Abbildungsverfahren (Texturen, etc.)

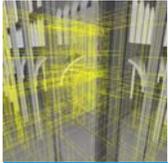
§8 Freiformmodellierung

Anhang: Graphiksprachen und  
Graphikstandards

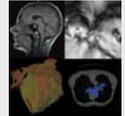
Anhang: Einführung in OpenGL

Weitere Themen:

Netze, Fraktale, Animation, ...

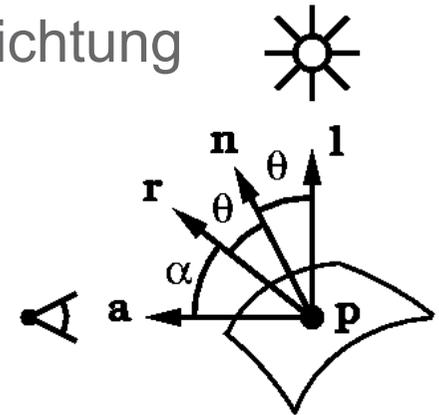


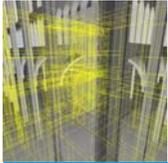
## 6.6 Globale Beleuchtung



### Erinnerung: **Lokale Beleuchtung**

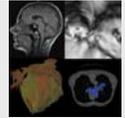
- Beachtet wird nur die Wechselwirkung der Oberfläche mit **direkt aus Lichtquellen** einfallendem Licht.
- Dabei wichtige Faktoren:
  - Geometrie: Oberflächennormale, Licht-, Blickrichtung
  - Reflexionsverhalten der Oberflächen
- Für photorealistische(re) Darstellung ist **globale Beleuchtung** notwendig
  - Physikalisch basierte Modellierung der Lichtausbreitung
  - Beachtet auch Wechselwirkung zwischen Objekten



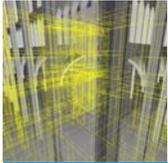


## 6.6 Globale Beleuchtung

Visual  
Computing (VCM)

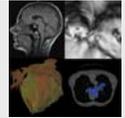


Lokale Beleuchtung:  
*Diffuse Reflexion*



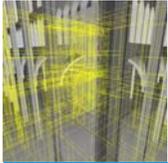
## 6.6 Globale Beleuchtung

Visual  
Computing (VCM)



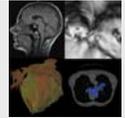
Lokale Beleuchtung:  
*Diffuse Reflexion*

Lokale Beleuchtung:  
*Spiegelnde Reflexion*



## 6.6 Globale Beleuchtung

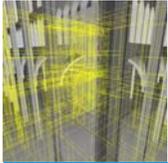
Visual  
Computing (VCM)



Lokale Beleuchtung:  
*Diffuse Reflexion*

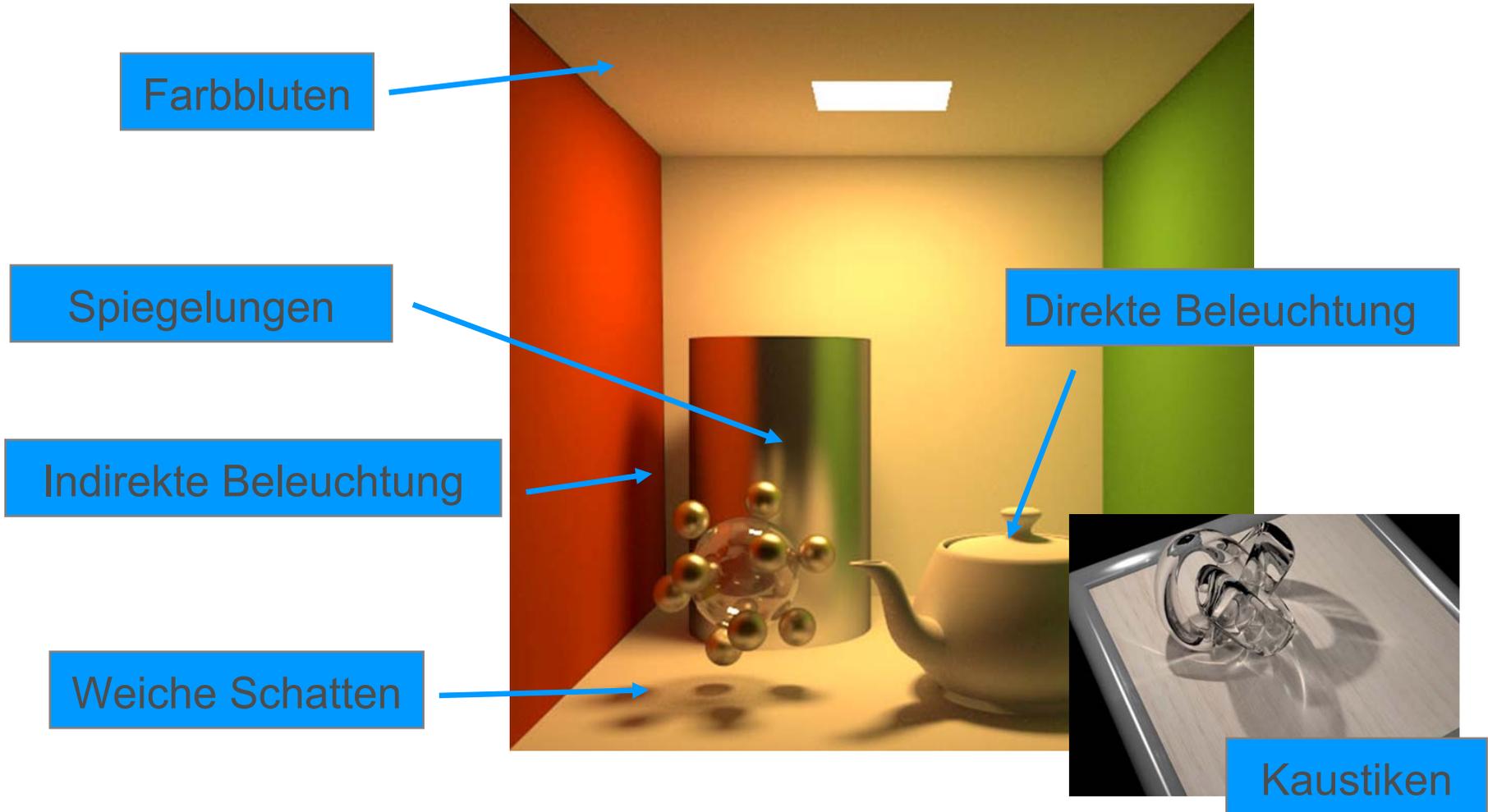
Lokale Beleuchtung:  
*Spiegelnde Reflexion*

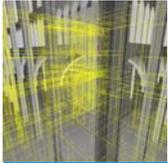
Globale Beleuchtung:  
*Mehrfachreflexion*  
*(Weiche) Schatten*



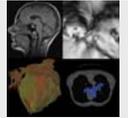
# 6.6 Globale Beleuchtung

Beleuchtungsphänomene:



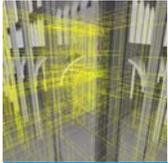


## 6.6 Globale Beleuchtung

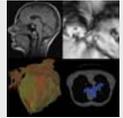


### Zwei Hauptansätze

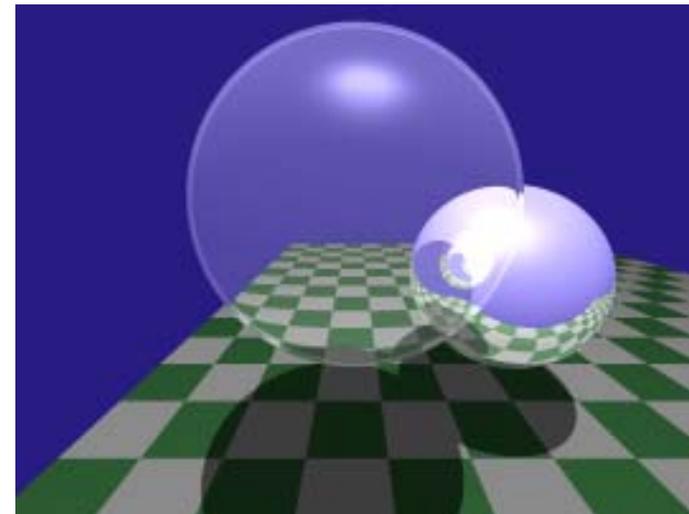
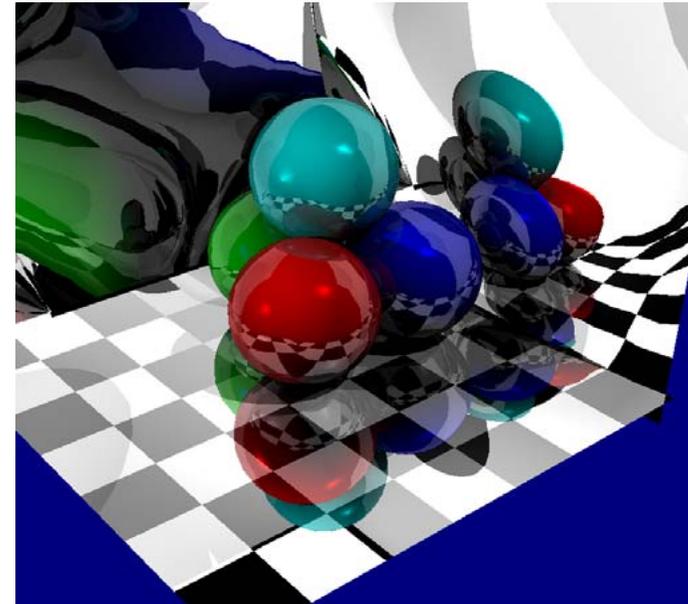
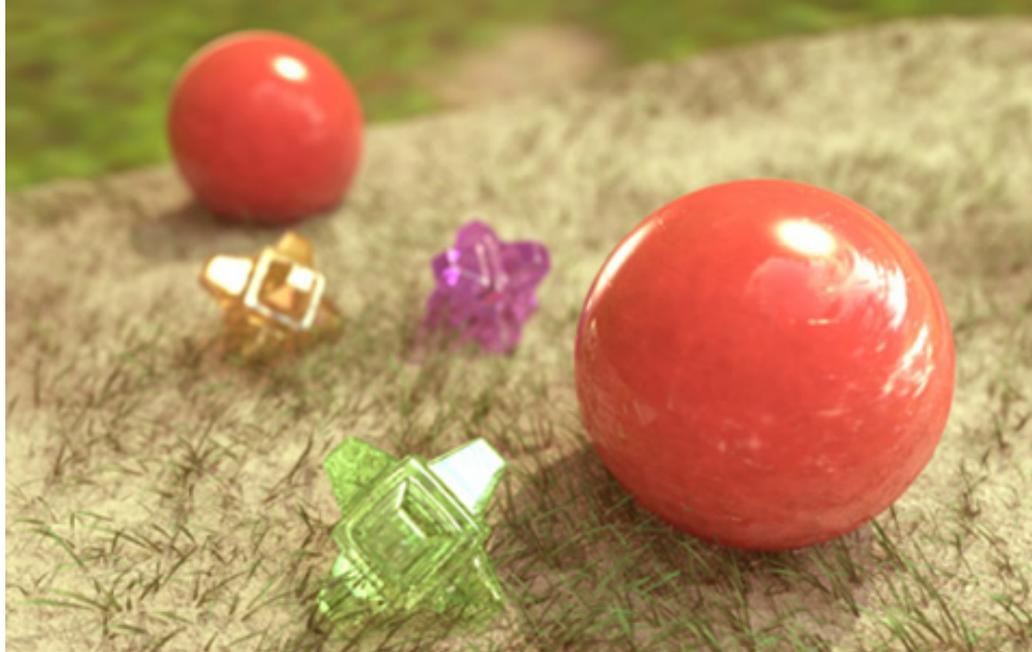
- **Raytracing** (Whitted, 1979)
  - Modelliert Lichtausbreitung nach Gesetzen der idealen Spiegelung und Brechung
  - Für spiegelnde und transparente Oberflächen geeignet
  - Blickrichtungsabhängig
- **Radiosity-Verfahren** (Goral & Greenberg, 1984)
  - Lichtausbreitung basierend auf Energieerhaltung
  - Vorberechnung aller Interaktionen des Lichts mit Objekten der Szene
  - Diffuse Reflexion
  - Unabhängig von der Blickrichtung



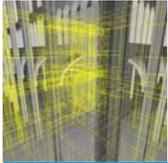
# 6.6.1 Raytracing



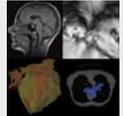
## Beispiele



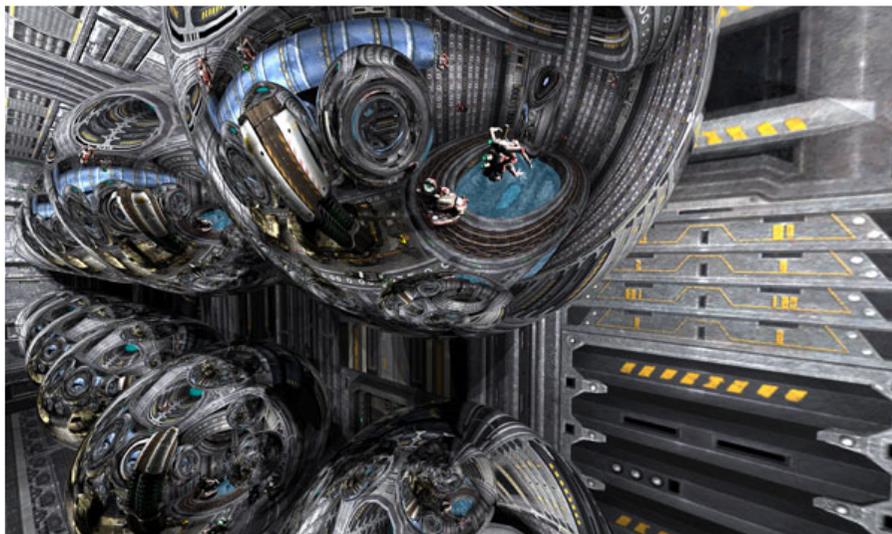
Besonders geeignet für Szenen mit hohem spiegelndem und transparentem Flächenanteil

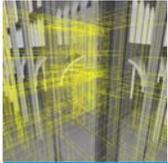


# 6.6.1 Raytracing

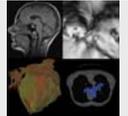


## Beispiele



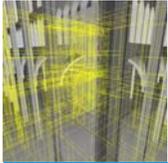


## 6.6.1 Raytracing

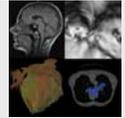


### Grundprinzip des **rekursiven Raytracing**

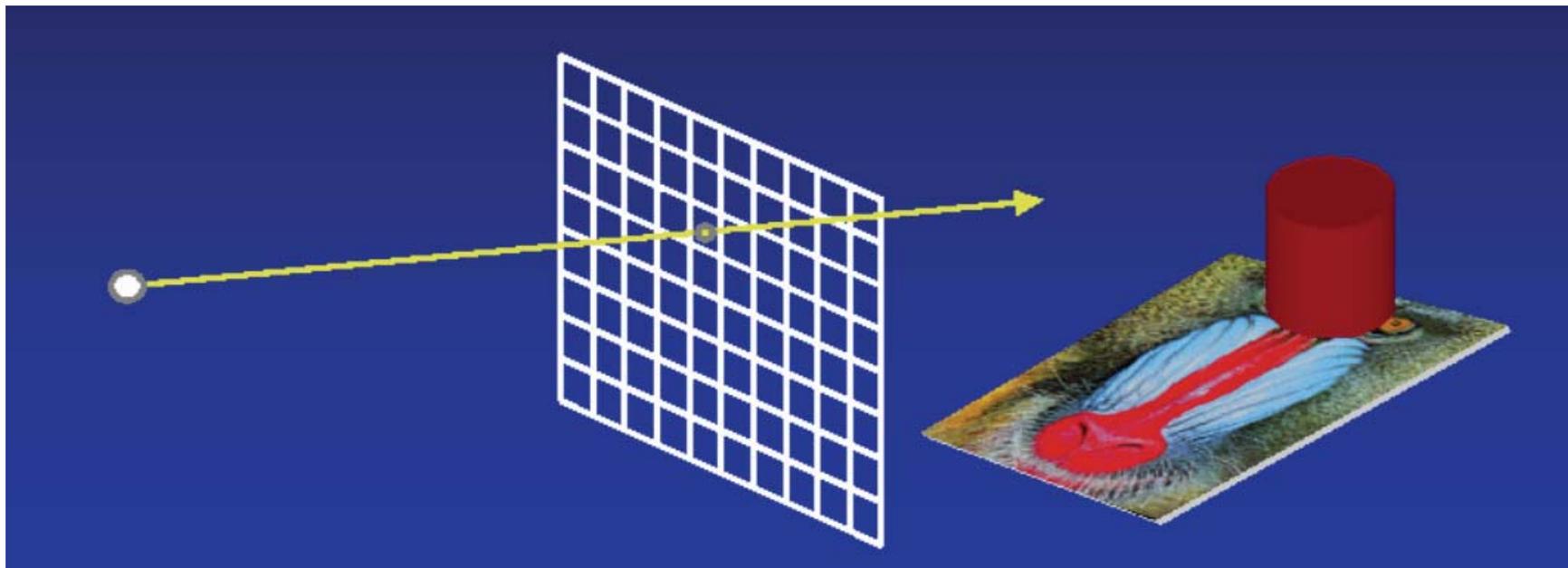
- Verfolgung der Lichtstrahlen von der Quelle bis zum Auge
- Nur wenige Strahlen erreichen das Auge, daher **Umkehrung** des Verfahrens (Reziprozität der Reflexion)
- Vom Augpunkt wird durch jedes Pixel des Bildschirms ein Strahl in die Szene gesendet.
- Bei Schnittpunkten mit Objekten werden ideal reflektierte und ideal gebrochene Strahlen weiterverfolgt.
- **Rekursive** Auswertung des lokalen Beleuchtungsmodells

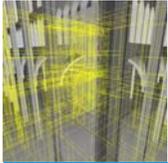


## 6.6.1 Raytracing

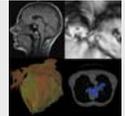


- Vom Augpunkt werden Strahlen durch die Pixel in die Szene geschickt.

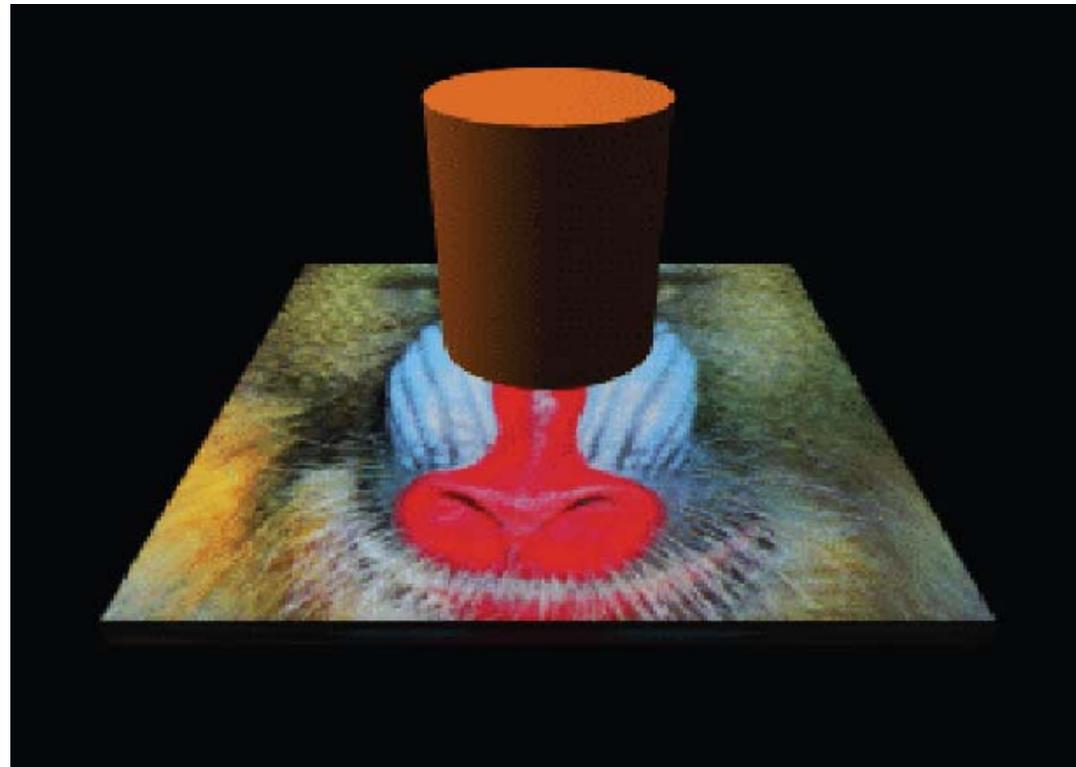


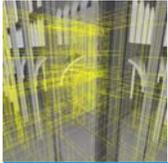


## 6.6.1 Raytracing

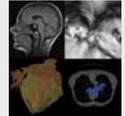


- Schneidet ein Strahl mehr als ein Objekt, wird das nächstliegende gezeichnet.
- Sonst erscheint die Hintergrundfarbe.

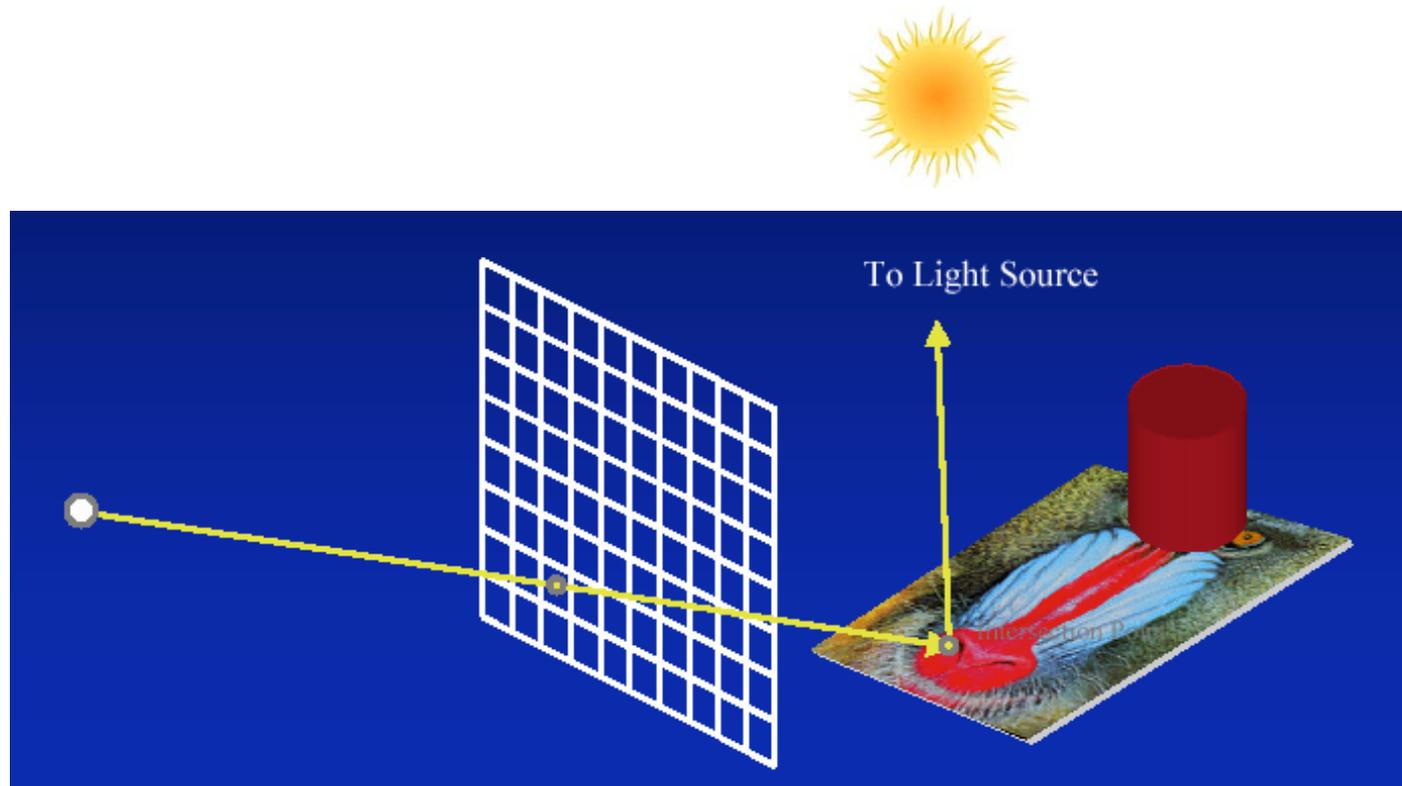


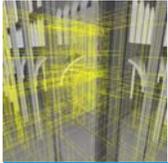


## 6.6.1 Raytracing

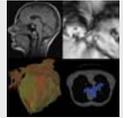


- Schneidet ein Strahl ein Objekt, so werden **Schattenstrahlen** vom Schnittpunkt zu allen Lichtquellen gesendet.

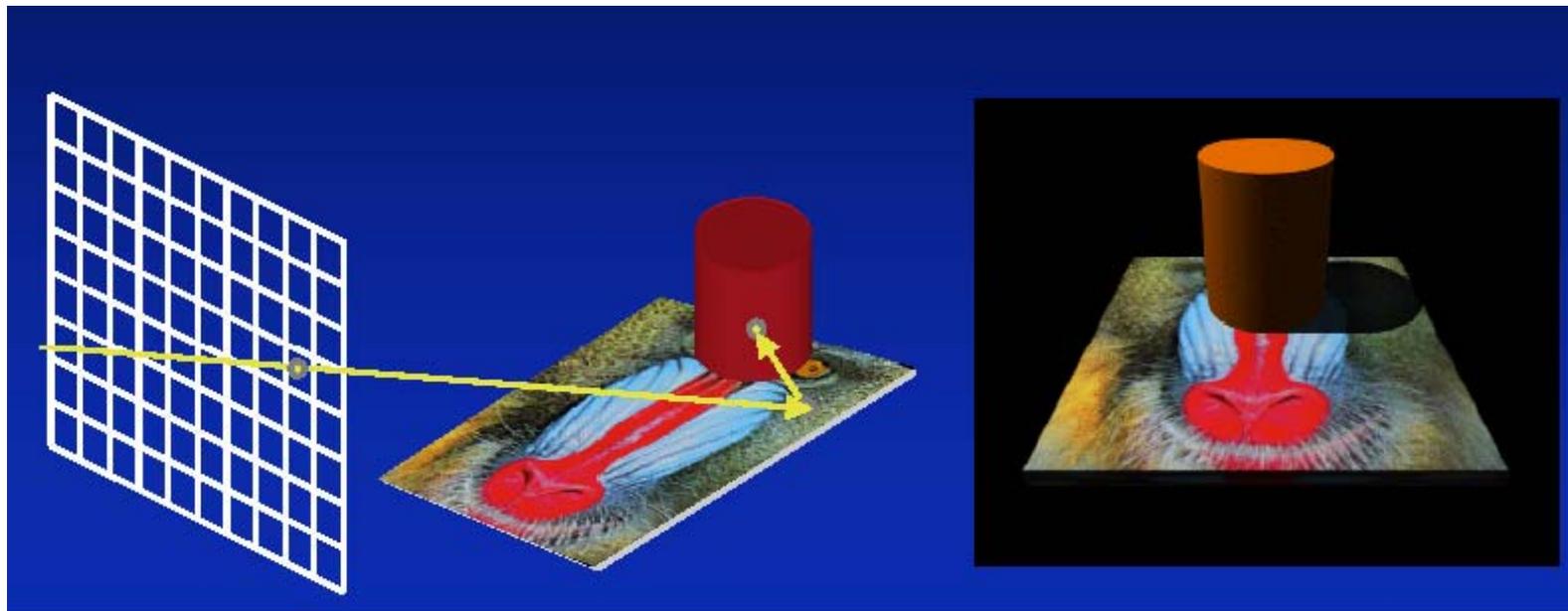


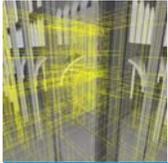


## 6.6.1 Raytracing

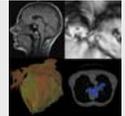


- Schneidet ein Strahl ein Objekt, so werden **Schattenstrahlen** vom Schnittpunkt zu allen Lichtquellen gesendet.
- Treffen die Schattenstrahlen auf ein Objekt, so liegt der Punkt im Schatten. ☀

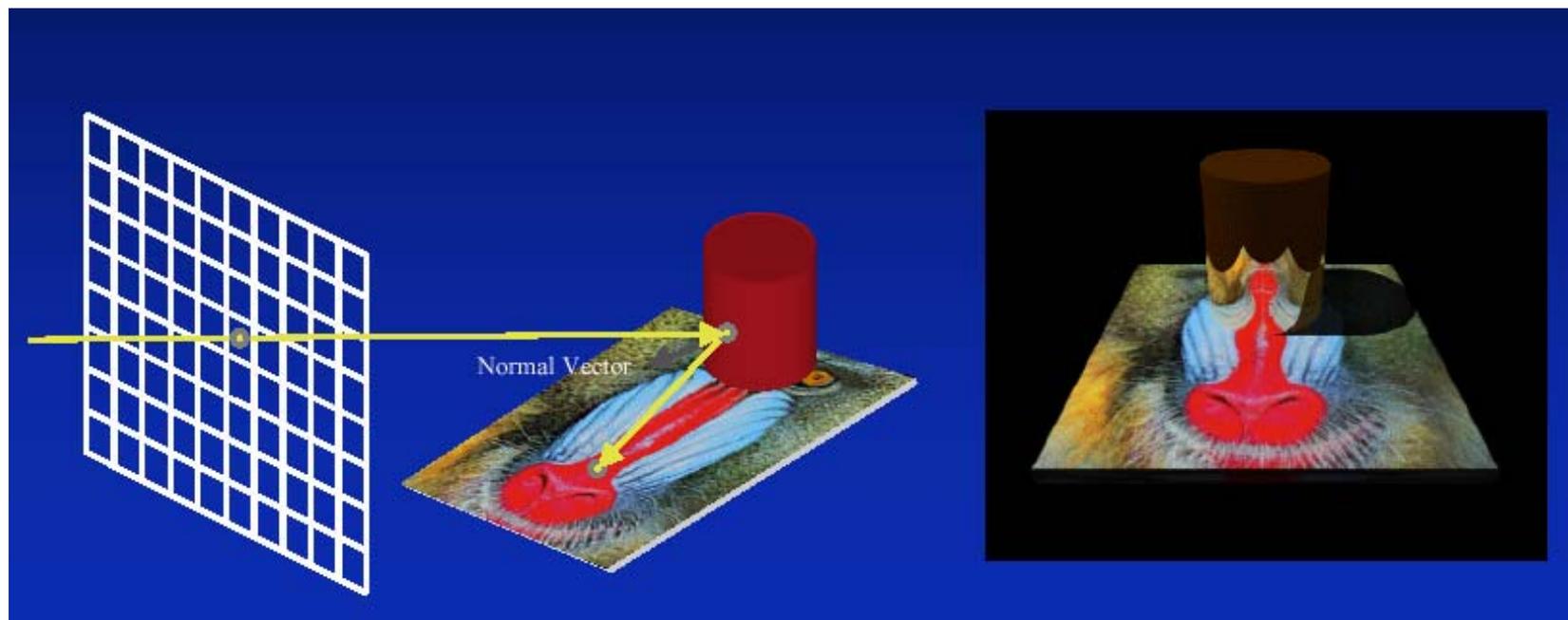


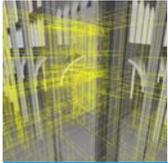


## 6.6.1 Raytracing

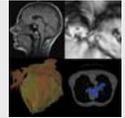


- Ist die Oberfläche **spiegelnd**, wird ein **reflektierter** Strahl verfolgt.

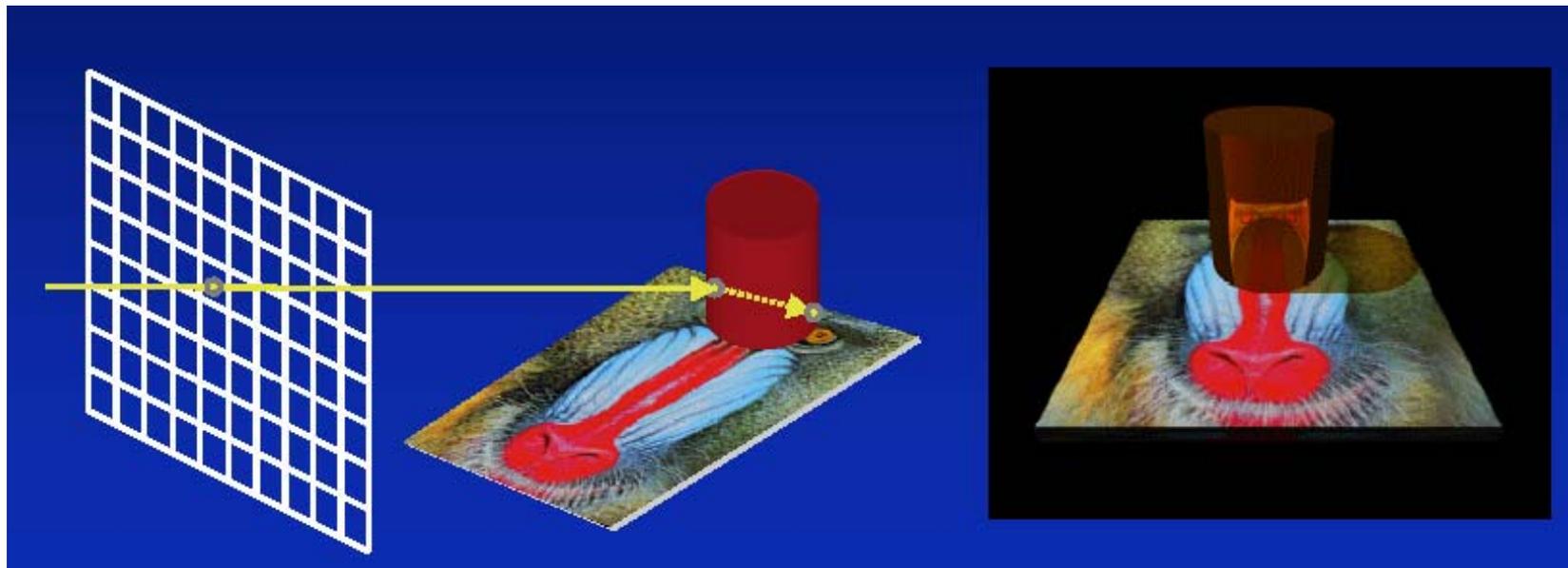


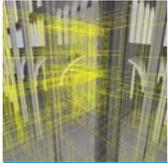


## 6.6.1 Raytracing

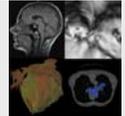


- Ist die Oberfläche **transparent**, wird **zusätzlich** ein **gebrochener** Strahl in die Szene geschickt.





## 6.6.1 Raytracing



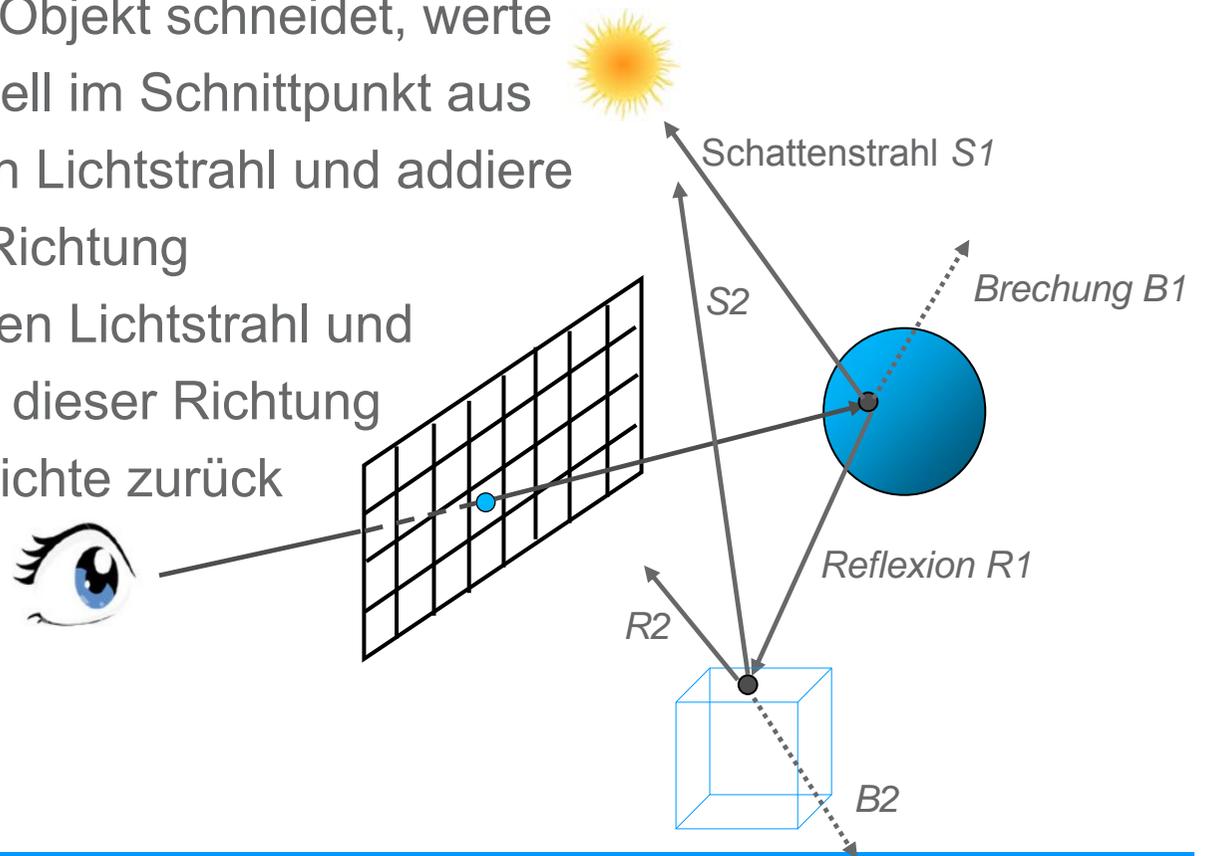
### Algorithmus in Pseudocode

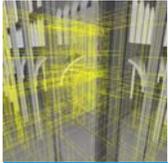
Für jeden Strahl {

1. Bestimme Schnittpunkt mit nächstliegendem Objekt
2. Falls Schattenstrahl kein Objekt schneidet, werte Phong-Beleuchtungsmodell im Schnittpunkt aus
3. Verfolge ideal reflektierten Lichtstrahl und addiere Leuchtdichte aus dieser Richtung
4. Verfolge ideal gebrochenen Lichtstrahl und addiere Leuchtdichte aus dieser Richtung
5. Gebe summierte Leuchtdichte zurück

}

**Rekursion!**

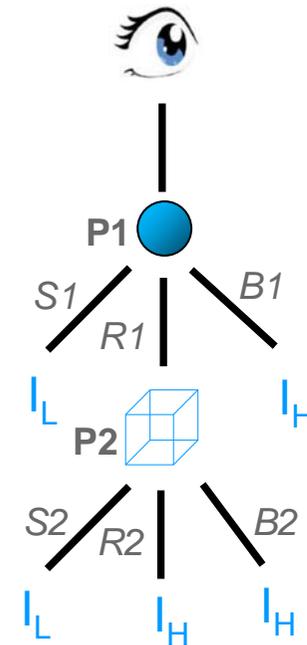




# 6.6.1 Raytracing

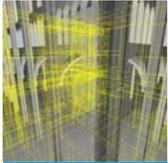
## Strahlenbaum

- Beschreibung der rekursiven Strahlverfolgung
- Traversierung zur Bestimmung der Pixel-Intensität, berücksichtigt:
  - Lichtquellenintensität  $I_L$
  - Hintergrundintensität  $I_H$
  - Materialkoeffizienten  $r$
  - Verdeckung von Lichtquellen  $V$

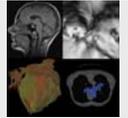


$$I_{ges} = V(P_1, L) \cdot I_{phong, P_1} + r_{r, P_1} I_{R_1} + r_{b, P_1} I_{B_1}$$

$$I_{R_1} = V(P_2, L) \cdot I_{phong, P_2} + r_{r, P_2} I_{R_2} + r_{b, P_2} I_{B_2}$$

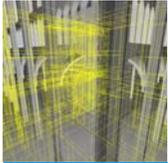


## 6.6.1 Raytracing

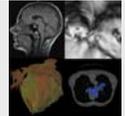


### Abbruchbedingungen

- Verfolgte Strahlen schneiden keine Objekte mehr
- Vorgegebene maximale Baumtiefe ist erreicht
- Farbanteil an der Pixelfarbe eines zu verfolgenden Teilstrahls ist zu gering (*adaptive Rekursionstiefenkontrolle*)

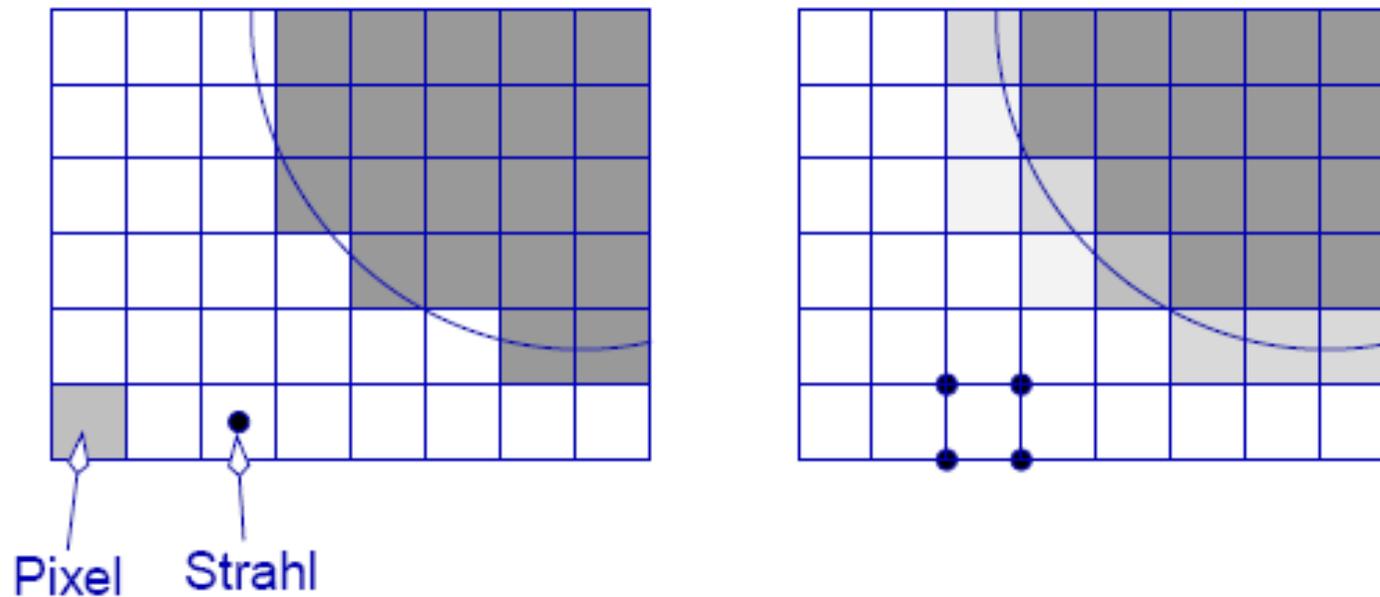


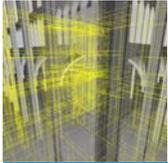
## 6.6.1 Raytracing



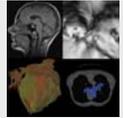
### Anti-Aliasing

- Strahlen ohne Ausdehnung verursachen Abtastfehler



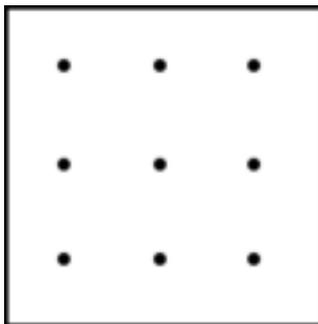


## 6.6.1 Raytracing

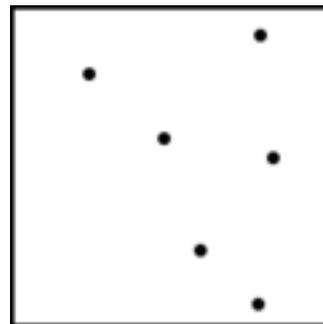


### Anti-Aliasing: Lösungen

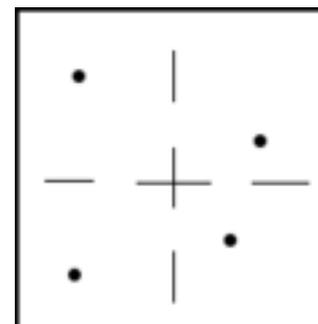
- Strahlen ohne Ausdehnung verursachen Abtastfehler
- **Supersampling**
  - Mehrere Strahlen pro Pixel und Mittelung der Farbwerte



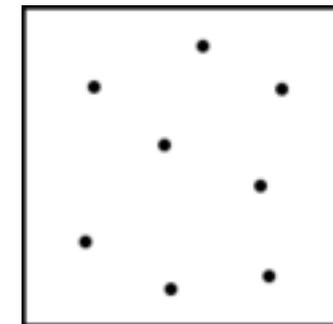
grid



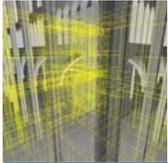
random



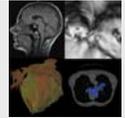
jitter



stochastisch  
(z.B. Poisson)

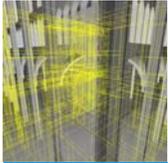


## 6.6.1 Raytracing

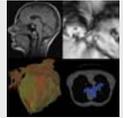


### Anti-Aliasing: Lösungen

- Strahlen ohne Ausdehnung verursachen Abtastfehler
- Supersampling
  - Mehrere Strahlen pro Pixel und Mittelung der Farbwerte
- **Adaptives Supersampling**
  - An Grenzen mehr Strahlen als auf homogenen Flächen
  - Vorgehen:
    - Senden einiger Strahlen pro Pixel
    - Unterscheiden sich die Werte, werden weitere gesendet
  - Stochastische Verteilung möglich

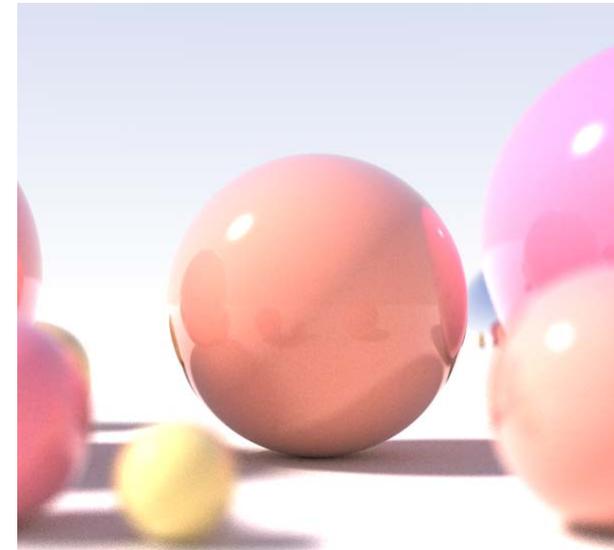


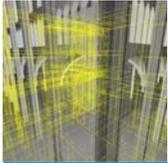
## 6.6.1 Raytracing



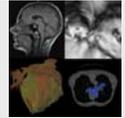
### Distributed Raytracing

- Ideale Reflexion und Brechung entspricht nicht der Realität und lässt gerenderte Szenen unecht erscheinen.
- Ziel: **Realistisch unscharfe Effekte**
  - Verschwommene Reflexionen und Brechungen
  - Weiche Schatten
  - Tiefenunschärfe
  - Bewegungsunschärfe



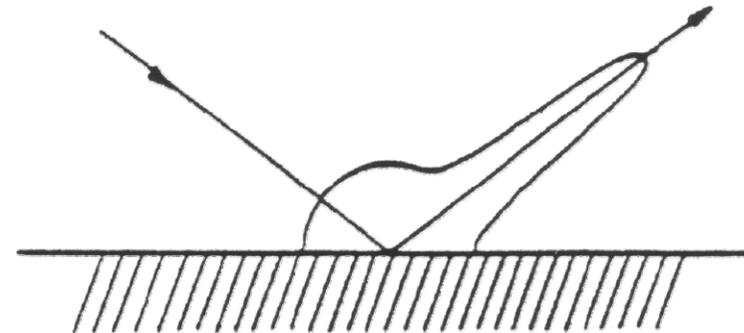
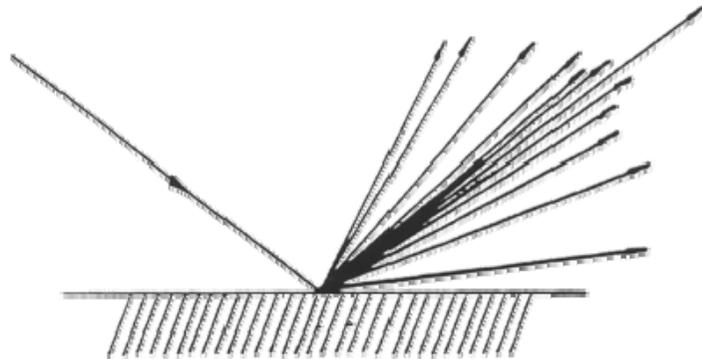


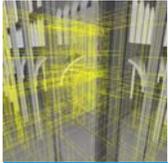
## 6.6.1 Raytracing



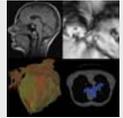
### Distributed Raytracing: **Verschwommene Reflexionen**

- Vervielfachung der reflektierten Strahlen
- Stochastische Verteilung der Strahlrichtungen
- Anschließende Mittelung der Farbwerte
- Auch für Brechung anzuwenden

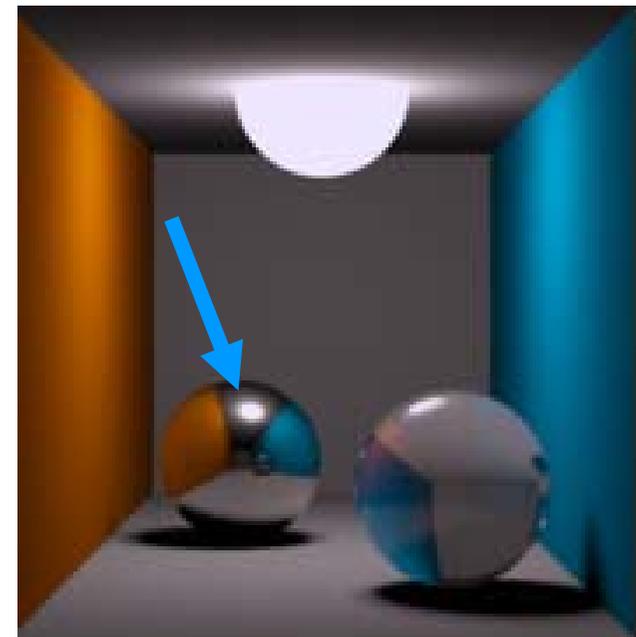
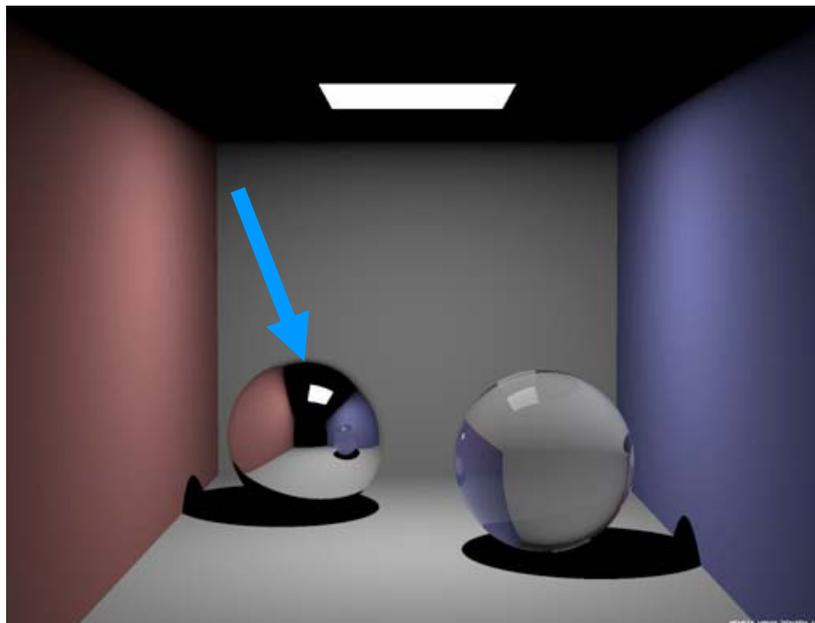


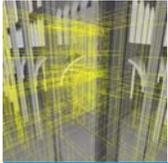


## 6.6.1 Raytracing

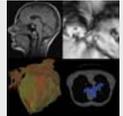


Beispiel: Verschwommene Lichtreflexionen



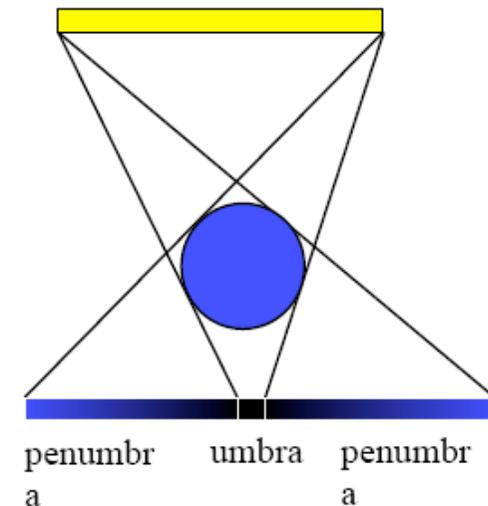
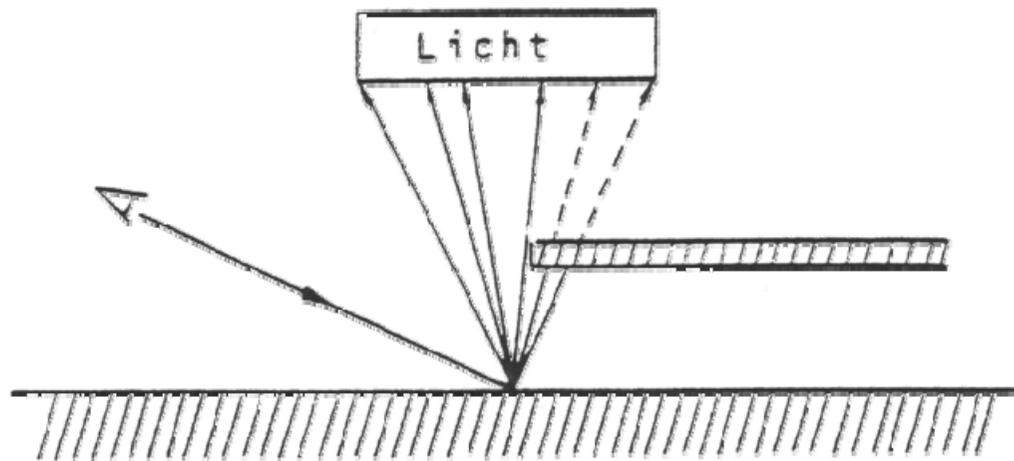


## 6.6.1 Raytracing

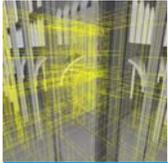


### Distributed Raytracing: Halbschatten

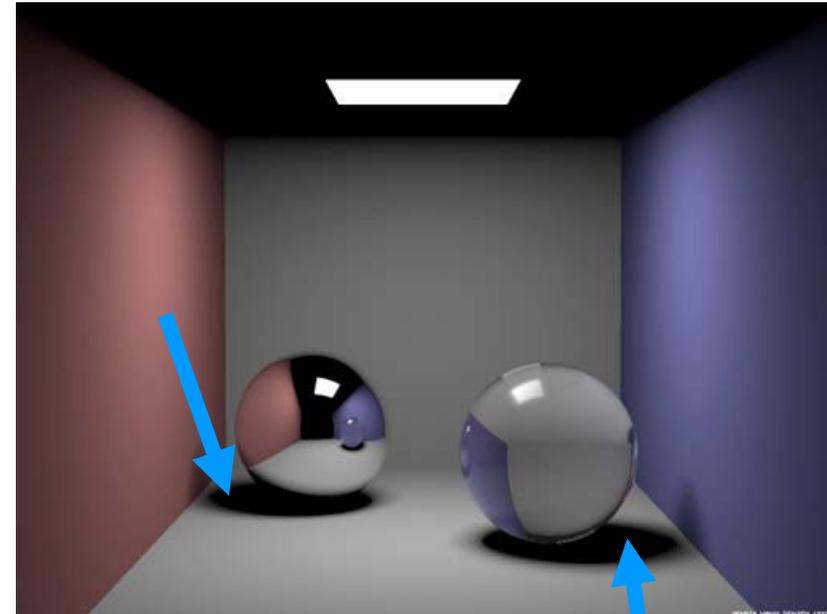
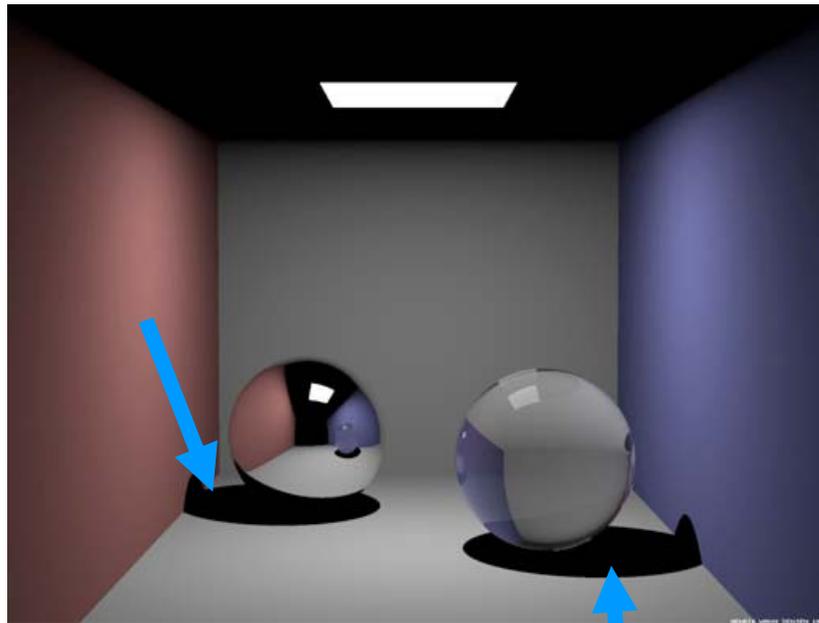
- Verwendung von **Flächenlichtquellen**
- Simuliert durch Vielzahl punktförmiger Lichtquellen

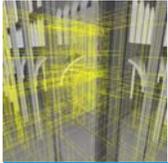


- Stochastische Verteilung der Schattenstrahlen
- Stärke des Schattens entspricht Verhältnis von verdeckten zur Gesamtzahl an Schattenstrahlen

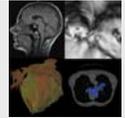


# 6.6.1 Raytracing



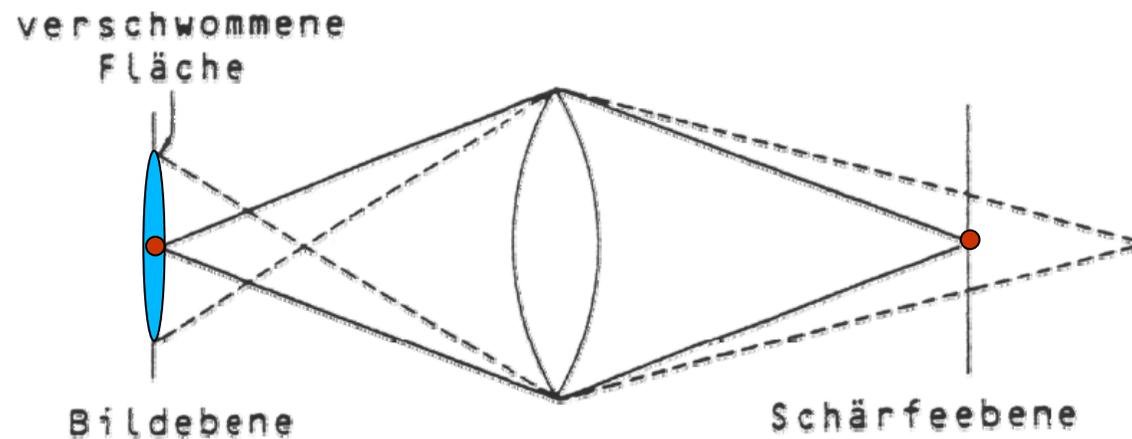


## 6.6.1 Raytracing

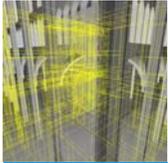


### Distributed Raytracing: **Tiefenunschärfe**

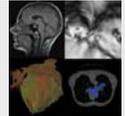
- Simulation der Blendenöffnung einer Kamera



- Punkte außerhalb der Schärfeebene erscheinen verschwommen (wenn größer als *circle of confusion*).
- Korrekte Berechnung der Linsenbrechung mit stochastischer Strahlverteilung über Linsenoberfläche.

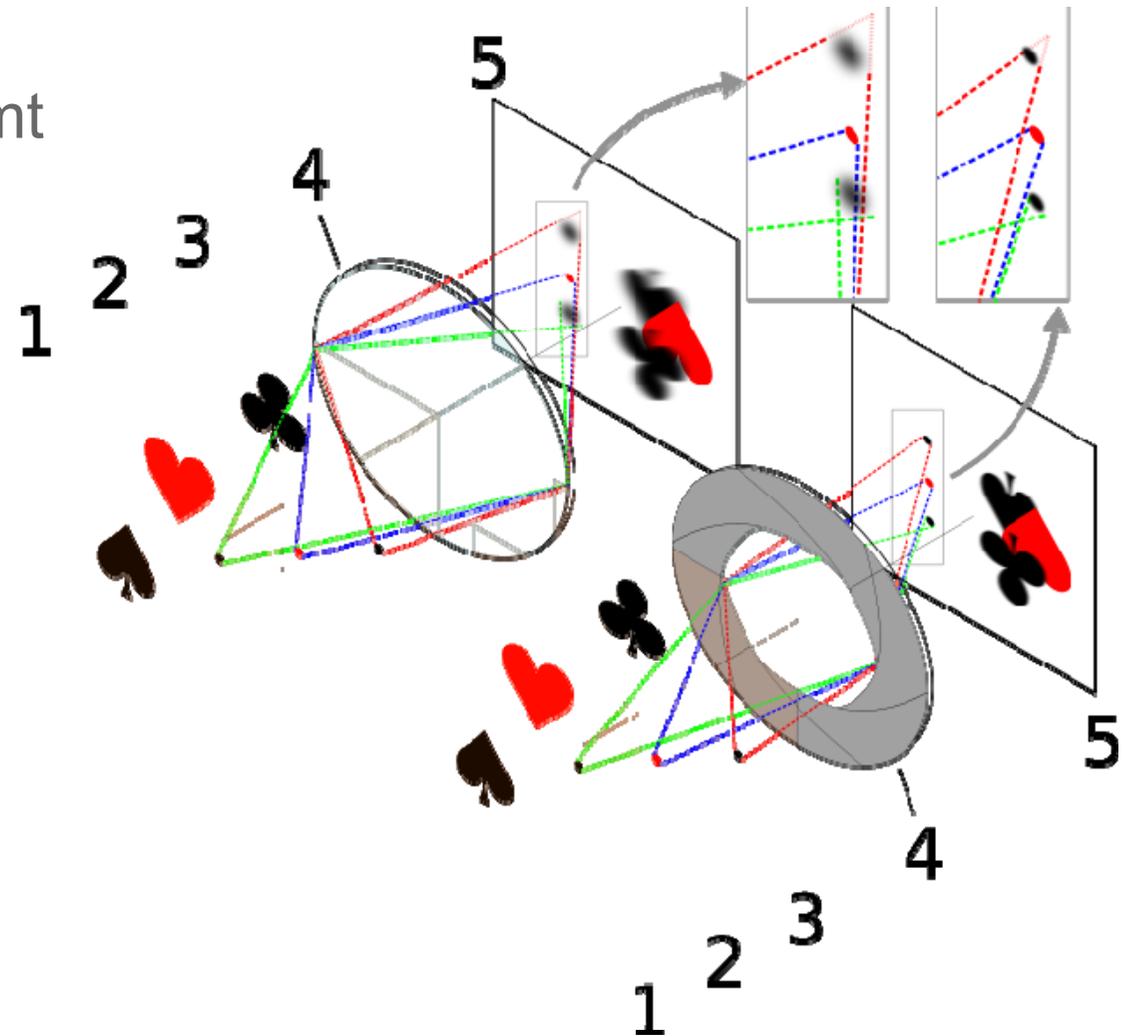


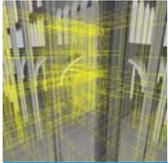
## 6.6.1 Raytracing



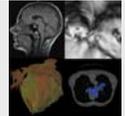
### Tiefenunschärfe

Blendenöffnung bestimmt Bereich der Szene, der scharf abgebildet wird (depth of field).





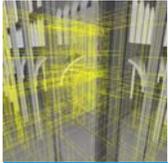
## 6.6.1 Raytracing



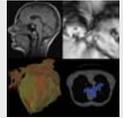
### Beispiel für **Tiefenunschärfe**



Chess (c) ROUX . Y (LERESQUATOR) OCTOBRE 2001

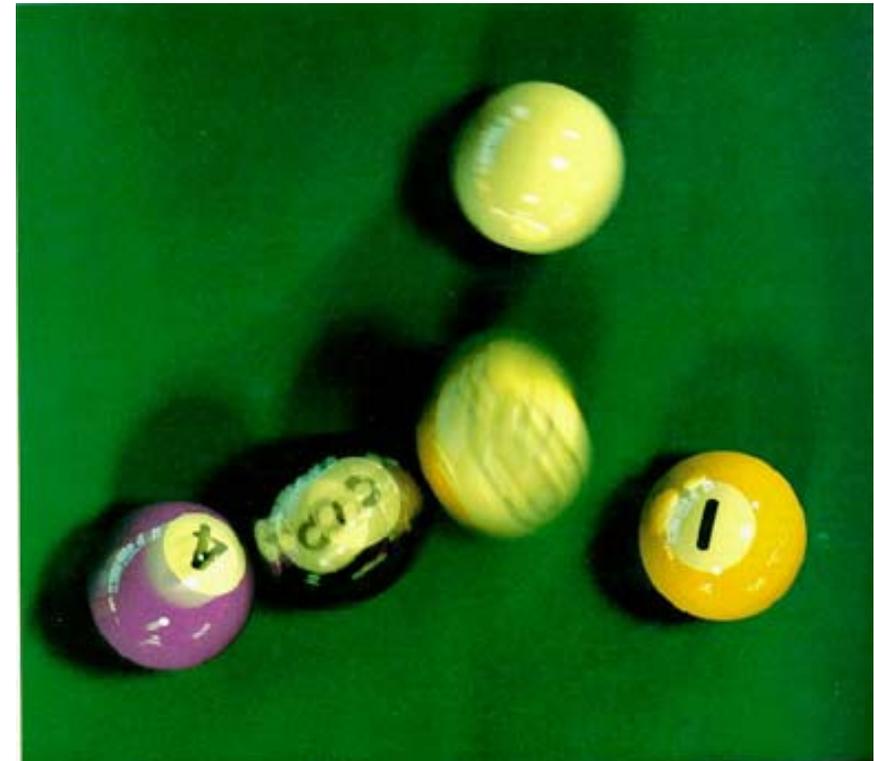


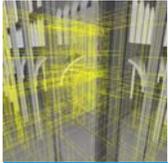
## 6.6.1 Raytracing



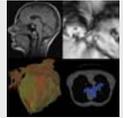
### Distributed Raytracing: **Bewegungsunschärfe**

- **Zeitliche** Verteilung der Augstrahlen
- Senden von Strahlen durch gleichen Punkt mit zeitlicher Verzögerung
- Filter bestimmt Verschußzeit und damit Stärke des Motion Blur





## 6.6.1 Raytracing



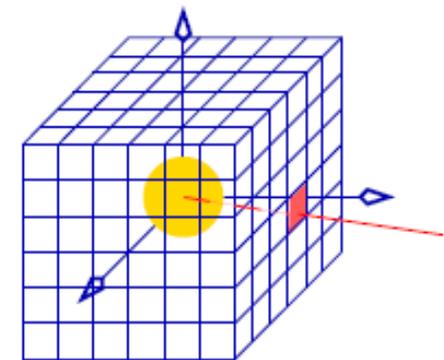
### Warum ist Raytracing so langsam?

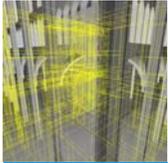
- Aufwändige Schnittpunktberechnungen (Strahl-Objekt)
- Viele Berechnungen  $O(n \cdot m)$  ← Anzahl der Strahlen

### Beschleunigung

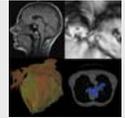
- Schnellere Schnittpunktberechnungen
  - Bounding Boxes
- Weniger Strahlen
  - Adaptive Rekursionstiefenkontrolle
  - Adaptives Anti-Aliasing
- Weniger Schnittpunktberechnungen
  - Hüllkörperhierarchien (z.B. Octrees)
  - Light Buffer

← Anzahl der Objekte



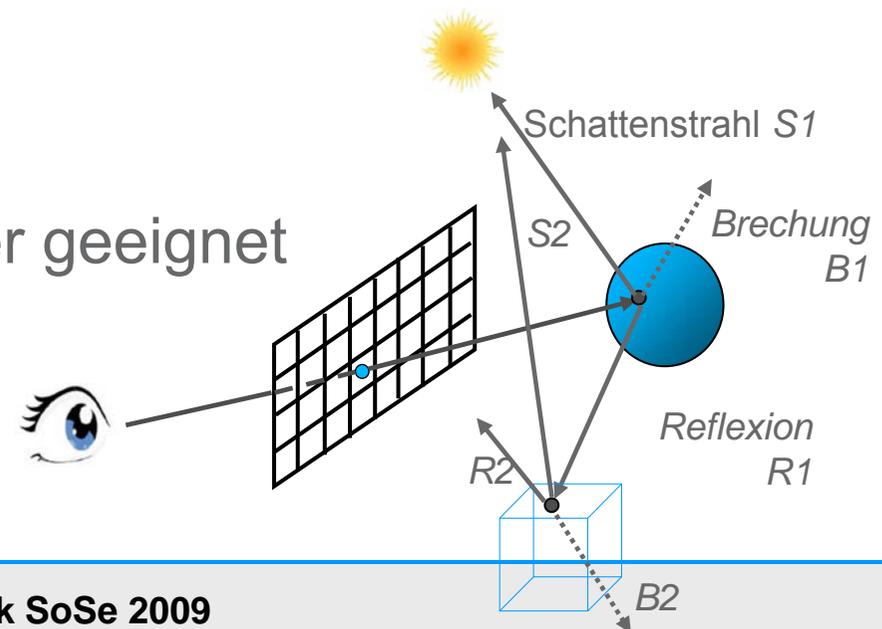


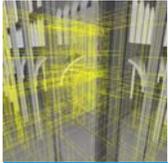
## 6.6.1 Raytracing



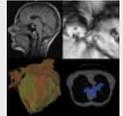
### Eigenschaften

- + Gute Simulation des physikalischen Beleuchtungsvorgangs (Strahlengang)
- + Hervorragend für Spiegelungen geeignet
- + Sichtbarkeitsproblem automatisch gelöst
- + Implizite Schattenberechnung
- + Große Wirklichkeitsnähe
- Für diffuse Reflektion weniger geeignet
- Sehr großer Rechenaufwand



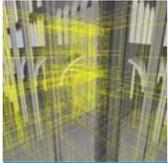


## 6.6.2 Radiosity-Verfahren

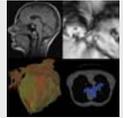


### Erste Fakten

- Modelliert rein diffuse Reflexion
- Ausbreitung des Lichts unter Beachtung des **Energiegleichgewichts** in einem geschlossenenem System
- Gleichgewicht des durch Lichtquellen zugeführten und durch Oberflächen absorbierten **Lichtstroms**
- Modellierung von unterschiedlichem Einfall des ambienten (indirekten) Lichts besser als beim klassischen Raytracing
- Gut geeignet für Lichtverteilung in Gebäuden



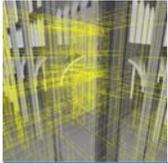
## 6.6.2 Radiosity-Verfahren



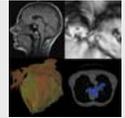
### Radiosity B (spezifische Ausstrahlung)

- Lichtmenge  $Q$ , die ein Flächenelement  $dA$  pro Zeiteinheit  $dt$  *emittiert*
- Oder: Lichtstrom  $\Phi$ , der von Flächenelement  $dA$  ausgesendet wird

$$B = \frac{dQ}{dt \cdot dA} = \frac{d\phi}{dA}$$

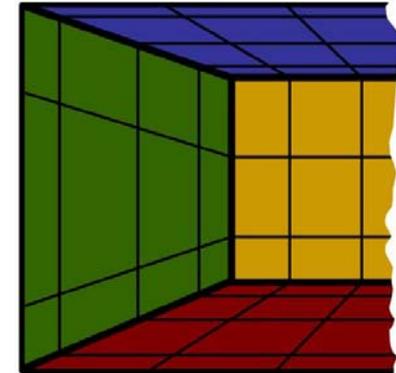


## 6.6.2 Radiosity-Verfahren



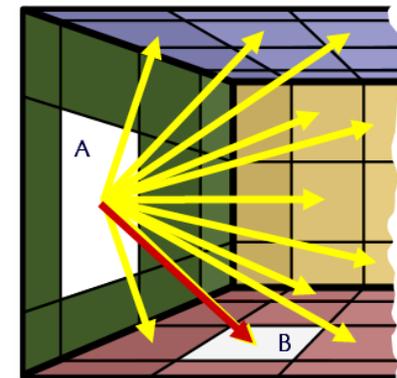
### Gegeben:

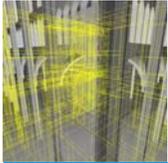
- Szene in Flächensegmente unterteilt
- Geometrische Informationen über die gegenseitige Lage aller Objekte
- Lichttechnische Kenngrößen der Objekte



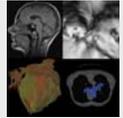
### Gesucht:

- Radiosity für jede Fläche
- Dient dann zur Farbgebung beim Rendern der Szene





## 6.6.2 Radiosity-Verfahren



### Radiosity-Gleichung

- Lichtstrom einer infinitesimalen Fläche  $dA_j$

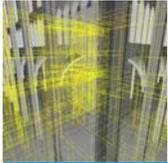
$$\phi_j = \phi_{Ej} + \rho_j \phi_{Ij}$$

Emittiert  $\nearrow$   $\phi_{Ej}$       Reflexionsgrad  $\nearrow$   $\rho_j$       Auftreffend  $\nearrow$   $\phi_{Ij}$

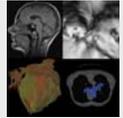
- Ausgedrückt durch Radiosités

$$B = \frac{d\phi}{dA}$$

$$B_{dA_j} dA_j = E_{dA_j} dA_j + \frac{1}{dA_j} \int_{dA_j \cup A_i, i \neq j} F_{dA_i \rightarrow dA_j} B_{dA_i} dA_i$$



## 6.6.2 Radiosity-Verfahren

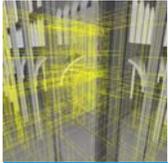


### Vereinfachung der Radiosity-Gleichung

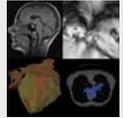
- Statt infinitesimaler Flächenelemente  $dA_i$ :
  - Kleine, ebene Patches  $A_i$
  - Reflexionsgrad und Radiosity einheitlich

$$B_j = E_j + \rho_j \sum_{i=1}^N F_{ij} B_i \frac{A_i}{A_j}$$

- Aus Integralgleichung wurde lineares Gleichungssystem

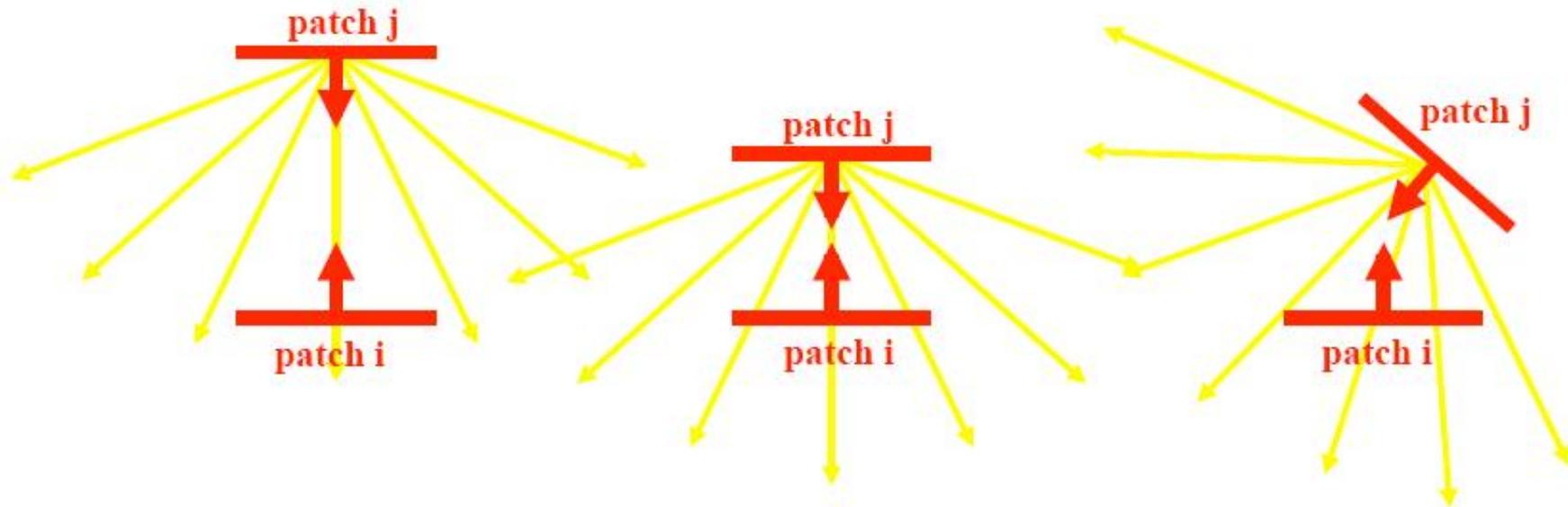


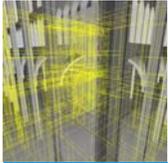
## 6.6.2 Radiosity-Verfahren



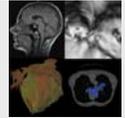
### Formfaktoren

- $F_{ij}$  : Anteil des Lichtstroms von  $A_i$ , der  $A_j$  erreicht
- Nur abhängig von
  - Geometrie (Abstand, Orientierung, Größe)
  - Sichtbarkeit





## 6.6.2 Radiosity-Verfahren



### Berechnung der Formfaktoren $F_{ij}$

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi \cdot r^2} V_{ij} dA_j dA_i$$

$$F_{ij} = \frac{\cos \theta_i \cos \theta_j}{\pi \cdot r^2} V_{ij} A_j$$

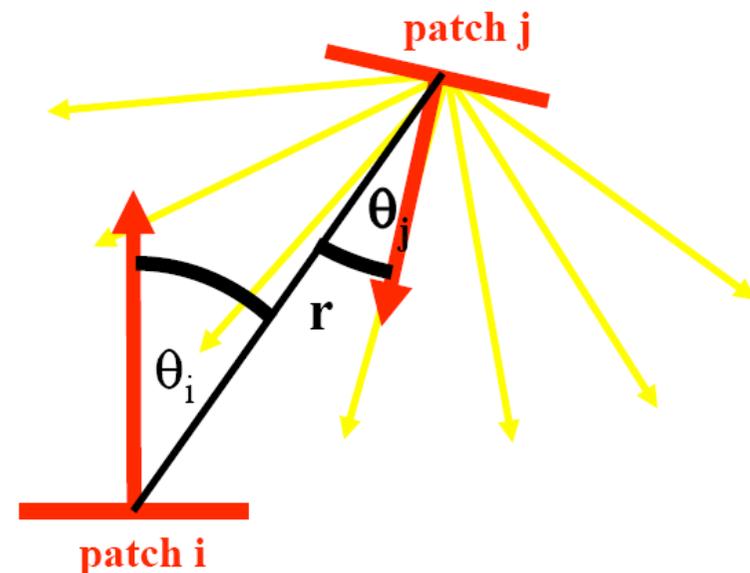
Sichtbarkeit:

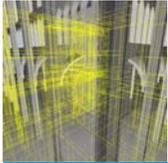
$$V = \begin{cases} 1, & \text{wenn sichtbar} \\ 0, & \text{sonst} \end{cases}$$

Anmerkung:

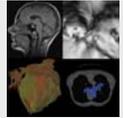
Herleitung über Grundgesetz der Strahlungsübertragung

$$F_{dA_i \rightarrow dA_j} = \frac{d\phi_{dA_i \rightarrow dA_j}}{d\phi_{dA_i}}$$



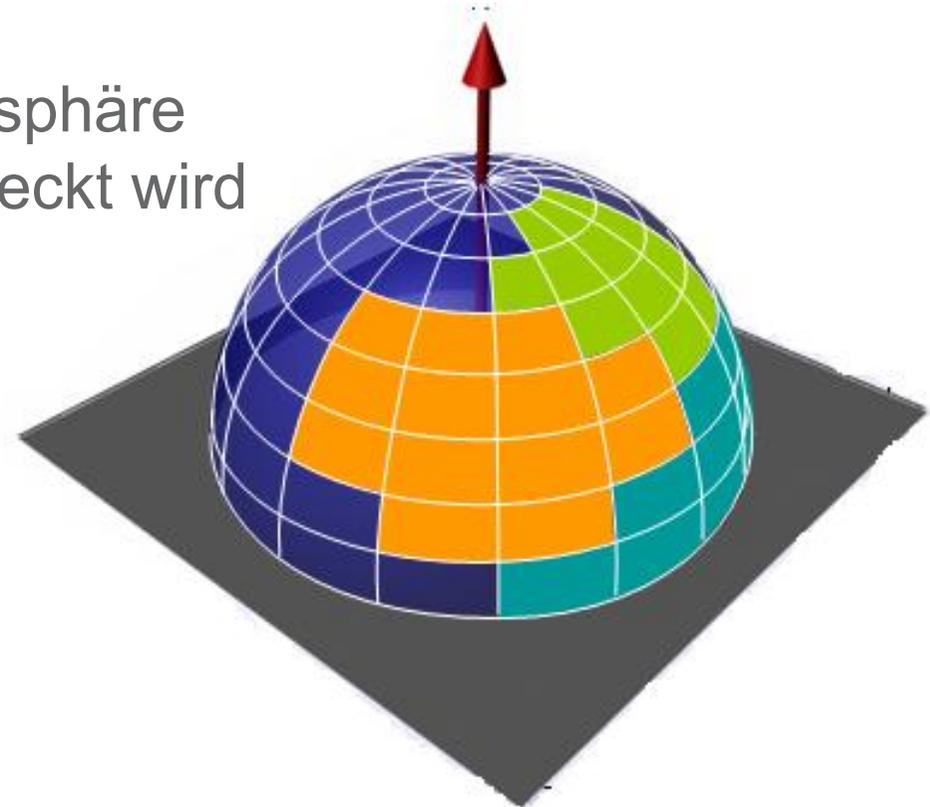
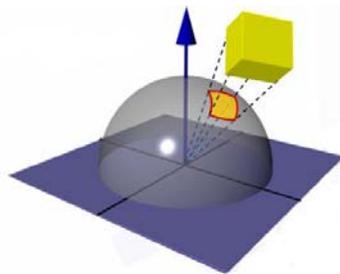


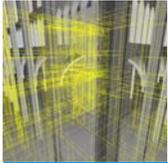
## 6.6.2 Radiosity-Verfahren



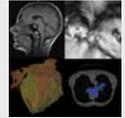
### Approximation der Formfaktoren $F_{SE_i}$

1. Senderfläche  $S$  in Patches  $S_i$  unterteilen
2. Hemisphäre für jedes Patch in Normalenrichtung
3. Empfängerflächen  $E_j$  auf Hemisphäre projizieren (Verdeckung beachten)
4. Für jedes  $E_j$  Anteil an Hemisphäre bestimmen, der von ihr bedeckt wird
5. Beiträge von  $E_j$  über alle Patches  $S_i$  mitteln



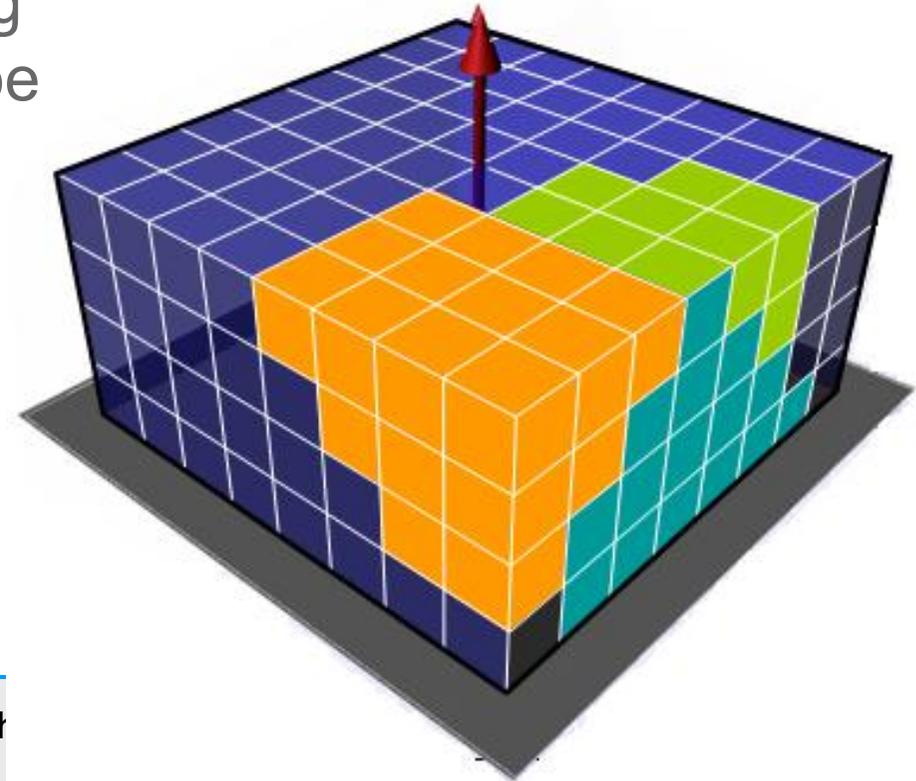
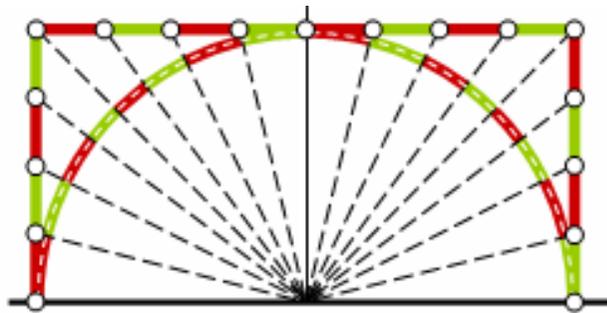


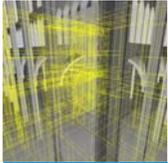
## 6.6.2 Radiosity-Verfahren



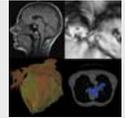
### Hemicube-Verfahren

- Einfachere Approximation der Formfaktoren
- Hemisphäre durch Hemicube ersetzen
- Einfachere Abbildung der Empfänger
- Unterschiedliche Gewichtung der Teilflächen des Hemicube





## 6.6.2 Radiosity-Verfahren



### Lösung des Radiosity-Gleichungssystems

$$B_j = E_j + \rho_j \sum_{i=1}^N F_{ij} B_i \frac{A_i}{A_j}$$

$$B_j = E_j + \rho_j \sum_{i=1}^N F_{ji} B_i$$



wegen Reziprozität:

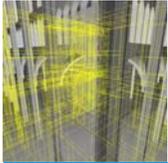
$$A_i F_{ij} = A_j F_{ji}$$

- Matrixschreibweise:

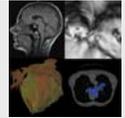
$$\vec{B} = \vec{E} + \rho F \vec{B} \quad \Rightarrow \quad (\mathbf{I} - \rho F) \vec{B} = \vec{E}$$

N-Vektoren

NxN-Matrizen



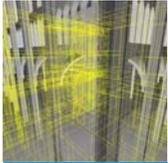
## 6.6.2 Radiosity-Verfahren



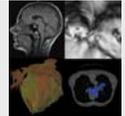
### Lösung des Gleichungssystems

$$\begin{bmatrix} 1 & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 & \cdots & -\rho_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

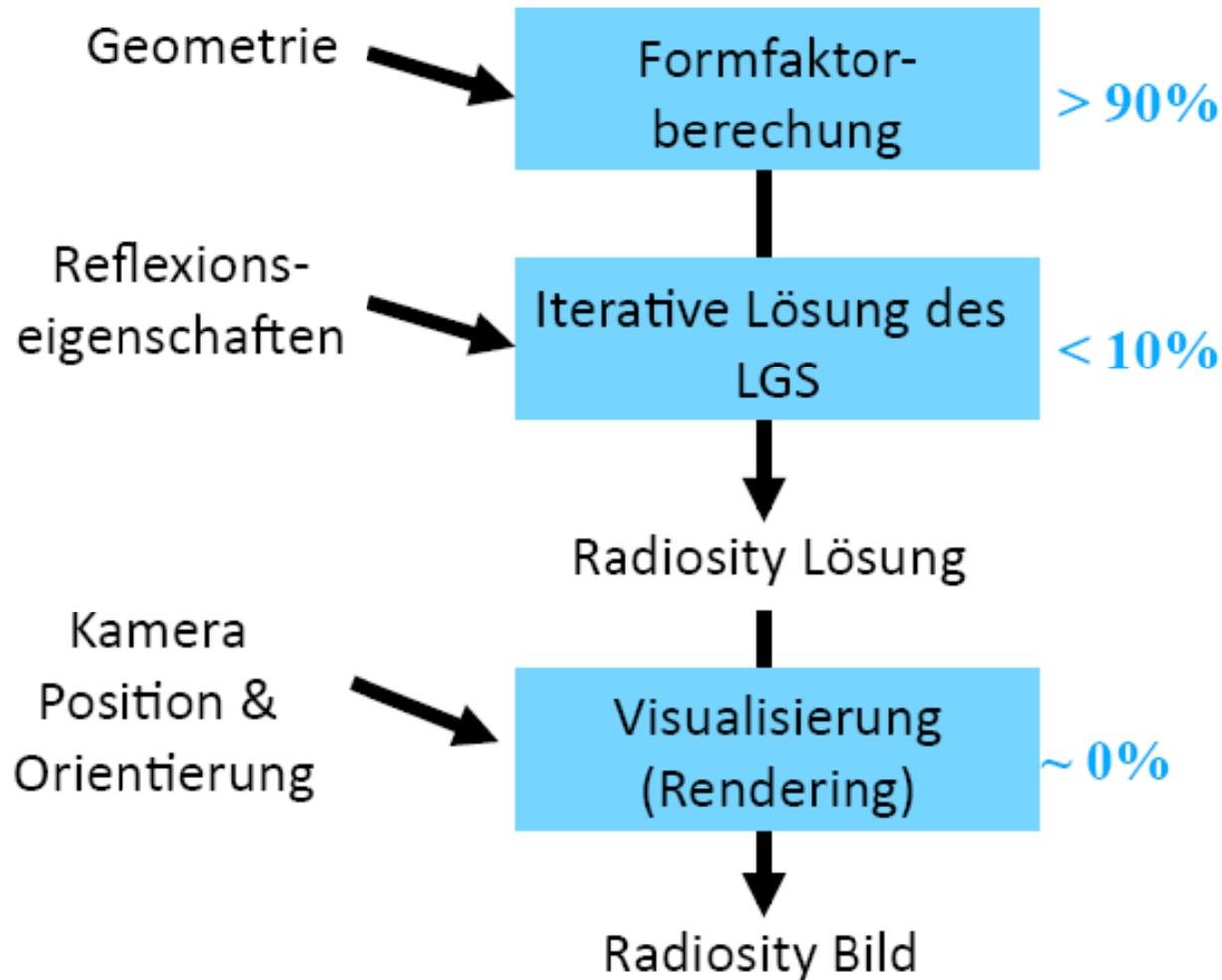
- Dieses System wird getrennt für die verschiedenen Frequenzen des Lichts (z.B. **RGB**) gelöst.
- Koeffizientenmatrix diagonaldominant, daher lösbar mit **iterativem Verfahren**, wie Gauss-Seidel
- „*Gathering*“ bzw. „*Sammeln*“

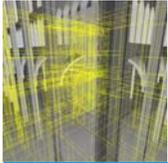


## 6.6.2 Radiosity-Verfahren

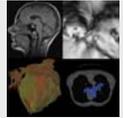


### Darstellung einer Szene



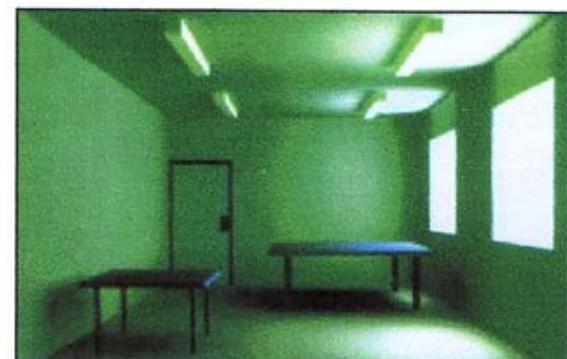
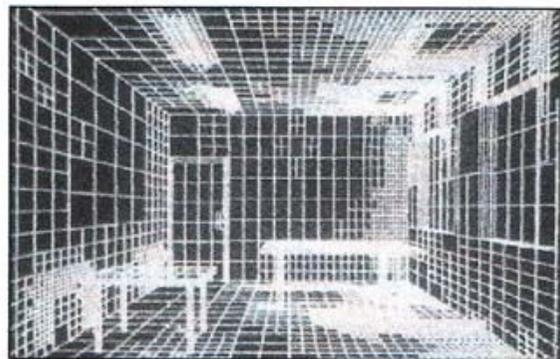
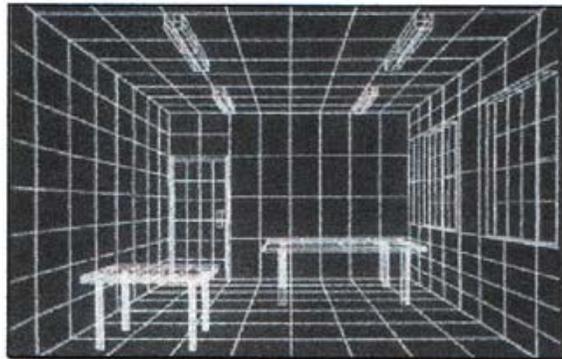


## 6.6.2 Radiosity-Verfahren

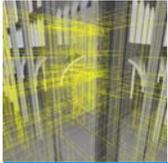


### Aufteilung der Szene

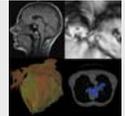
- Je feiner die Aufteilung der Szene in Patches, desto besser das Renderingergebnis
- Aber Aufwand steigt mit Anzahl  $N$  der Patches, z.B. Anzahl der Formfaktoren  $O(N^2)$



- Alternative zu Gauss-Seidel: **Progressive Refinement**

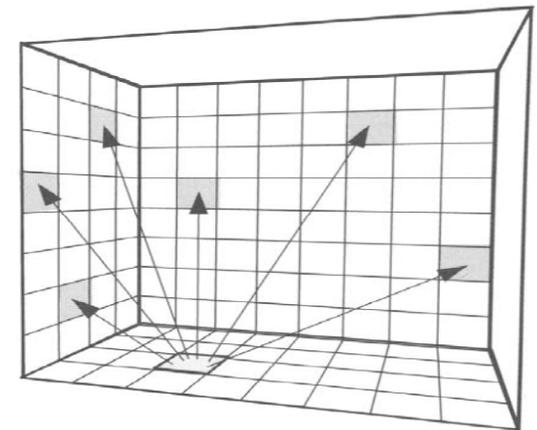
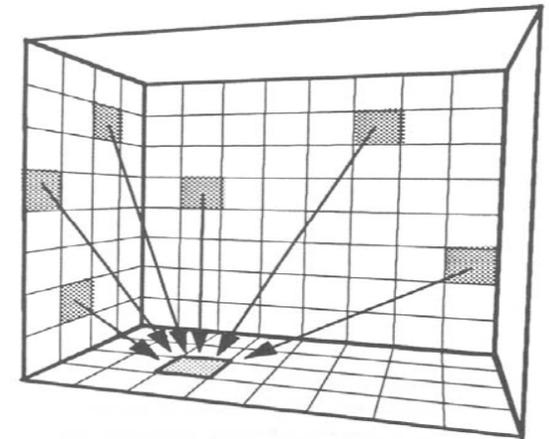


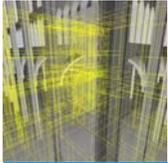
## 6.6.2 Radiosity-Verfahren



### Progressive Refinement

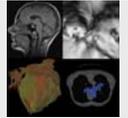
- „Verteilen“ statt „Sammeln“
- Initialisierung:  $B_i = E_i$  und  $\Delta B_i = E_i$
- Pro Iterationsschritt:  
Verteilen der Radiosity der Fläche mit  
größten unverteilter Radiosity  $\Delta B_i$   
(beginnend bei stärkster Lichtquelle)
- Vorteile:
  - **Schnelle Konvergenz** (nach  
wenigen Iterationen)
  - Pro Iterationsschritt nur Formfaktoren  
der aktuellen Fläche benötigt



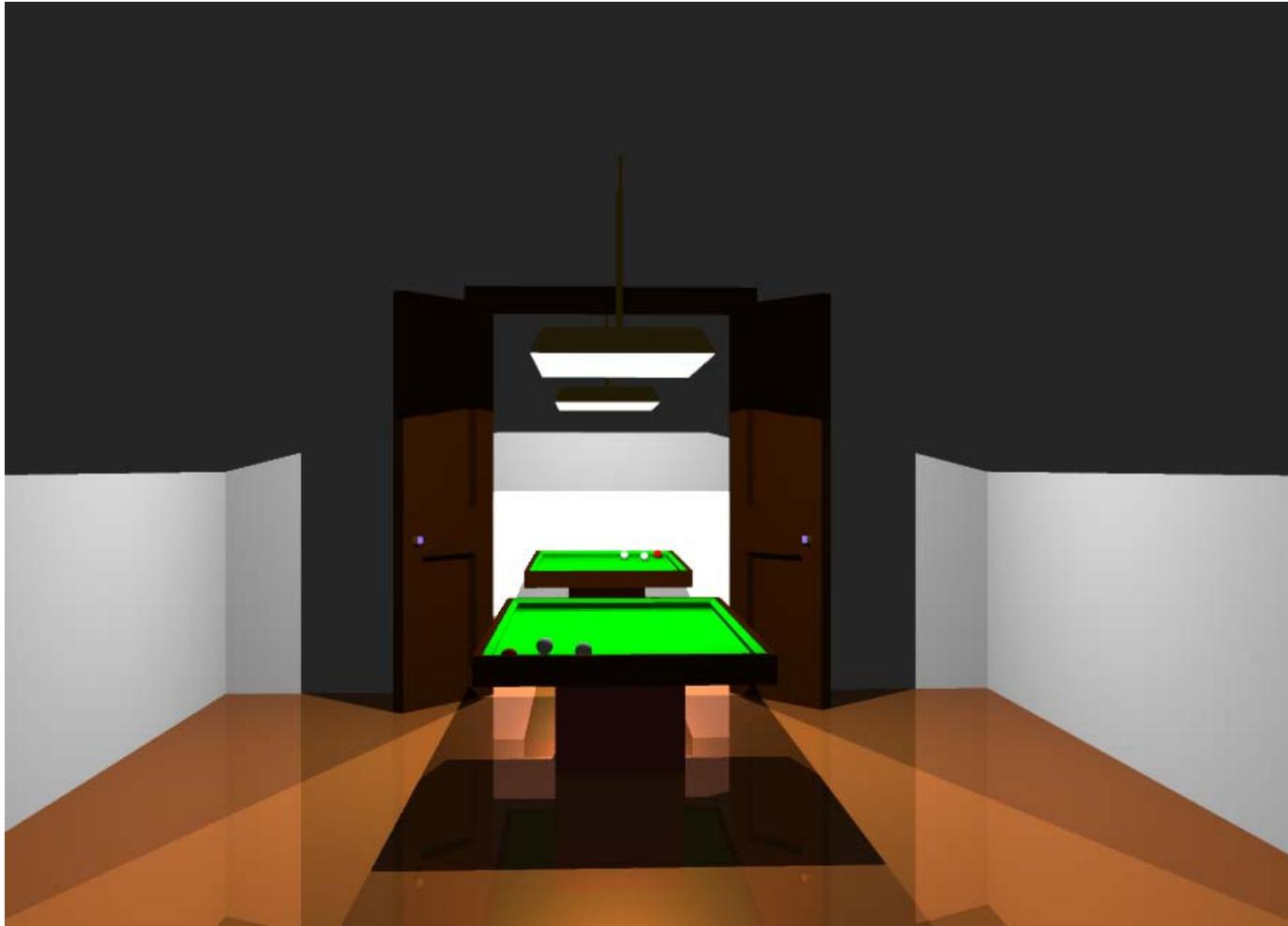


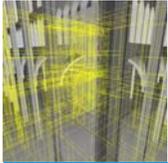
## 6.6 Globale Beleuchtung

Visual  
Computing (VCM)



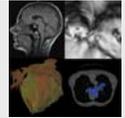
### Nur Raytracing





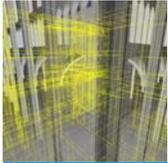
## 6.6 Globale Beleuchtung

Visual  
Computing (VCM)



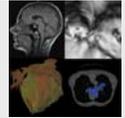
### Nur Radiosity



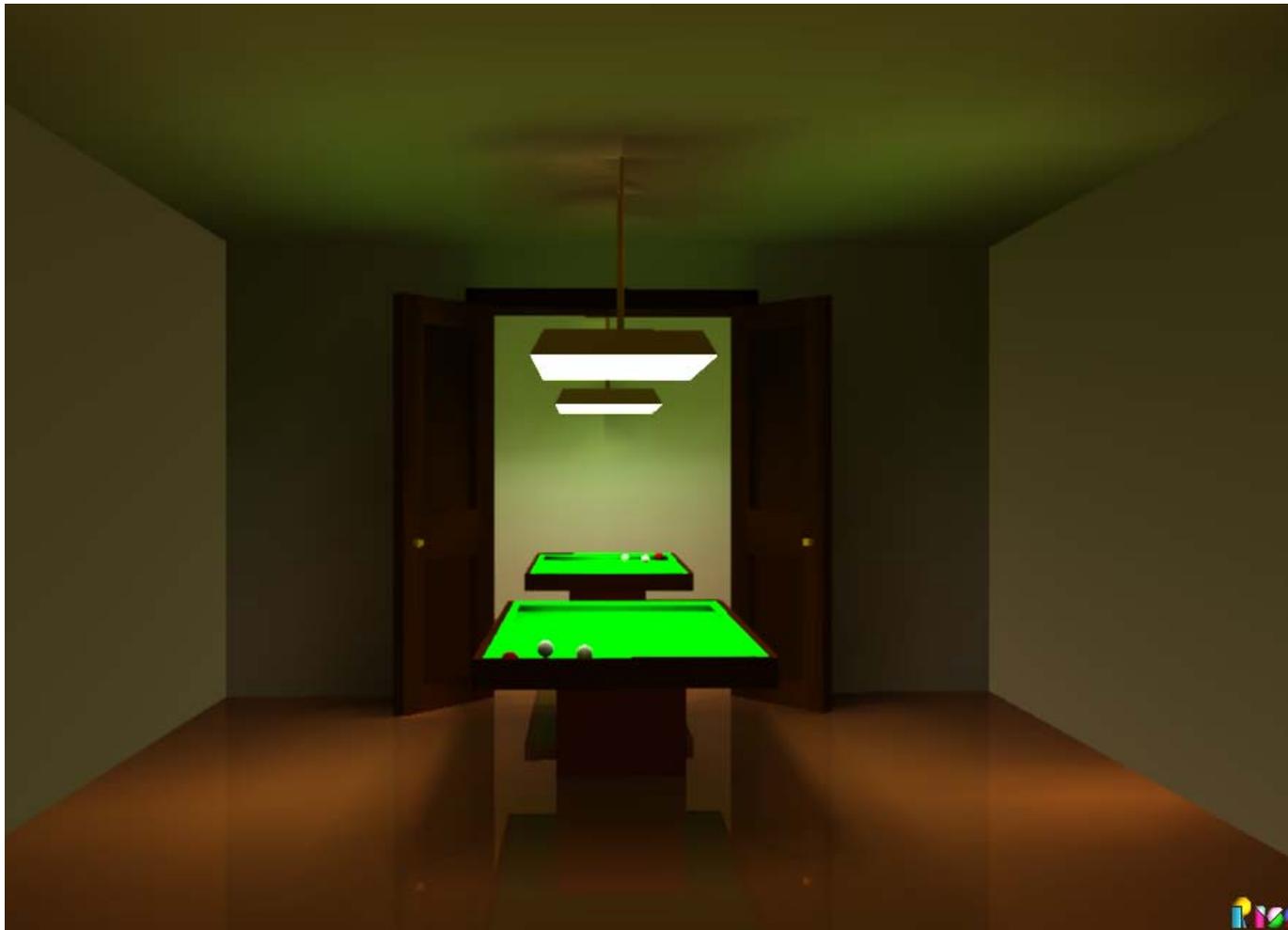


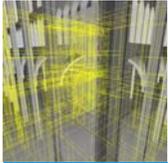
## 6.6 Globale Beleuchtung

Visual  
Computing (VCM)



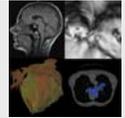
### Kombination von Raytracing und Radiosity



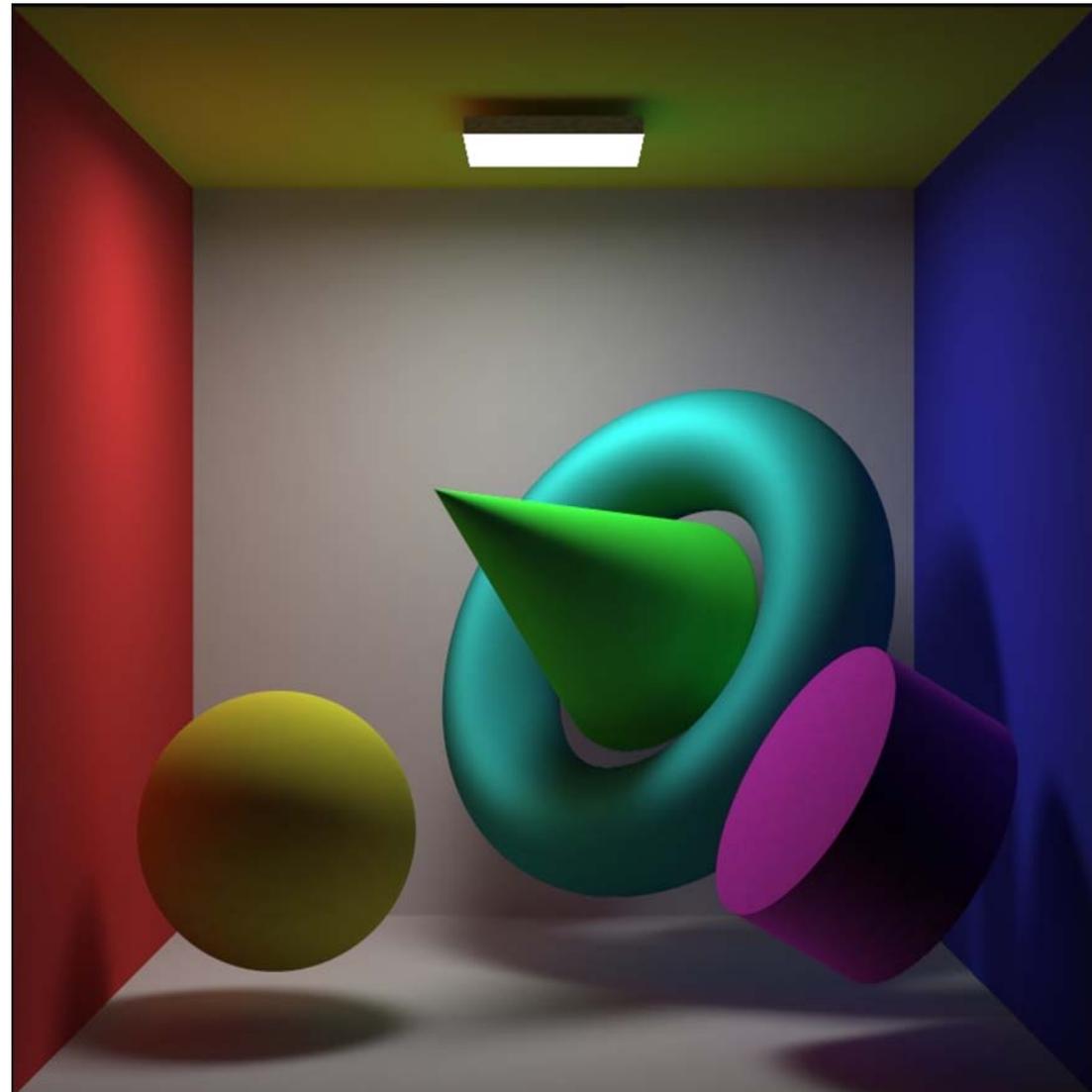


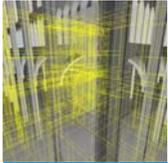
## 6.6 Globale Beleuchtung

Visual  
Computing (VCM)



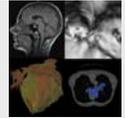
Nur Radiosity



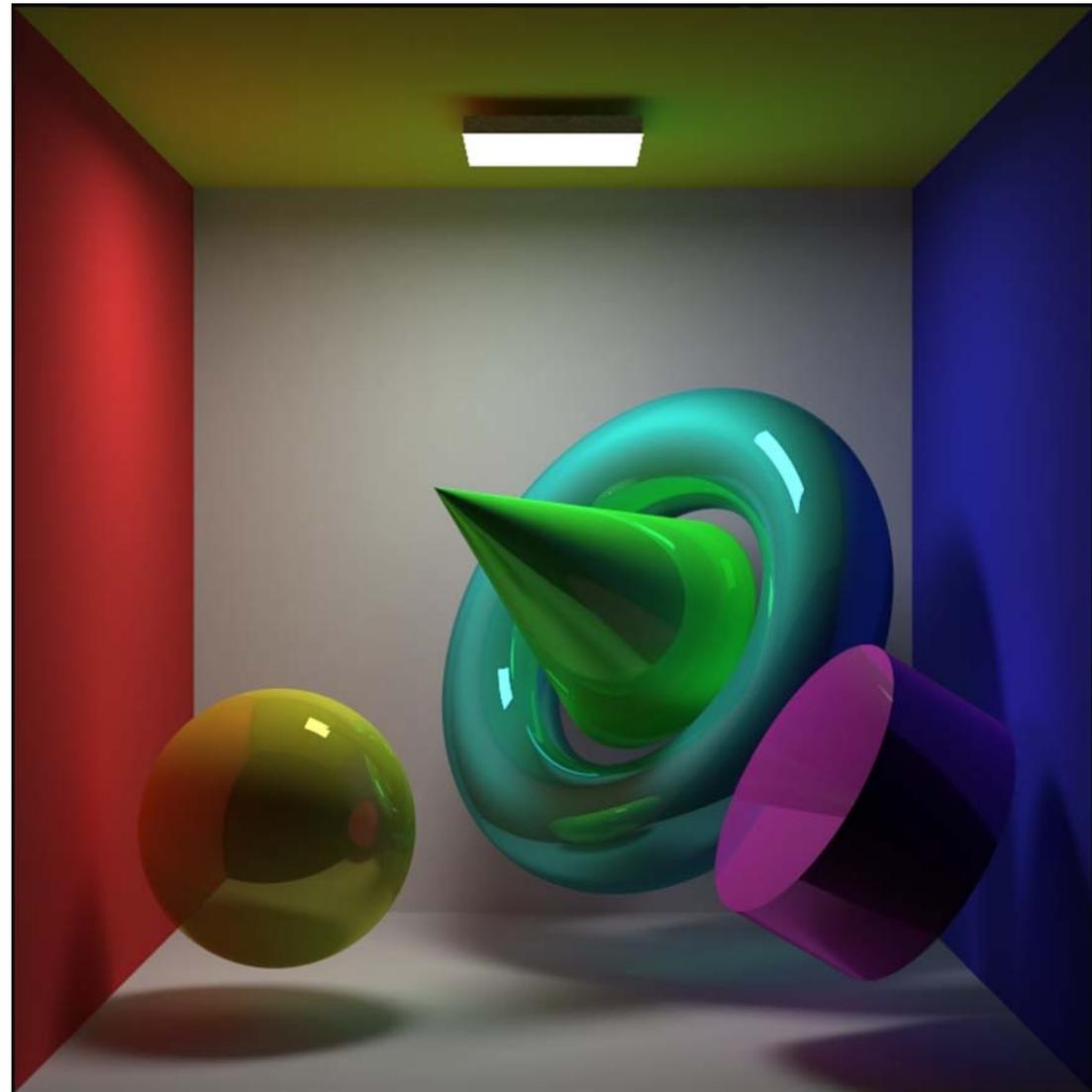


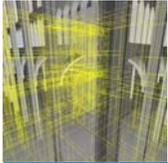
## 6.6 Globale Beleuchtung

Visual  
Computing (VCM)

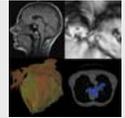


Radiosity und  
Raytracing



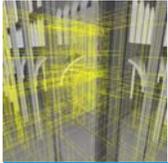


## 6.7 Renderpipeline



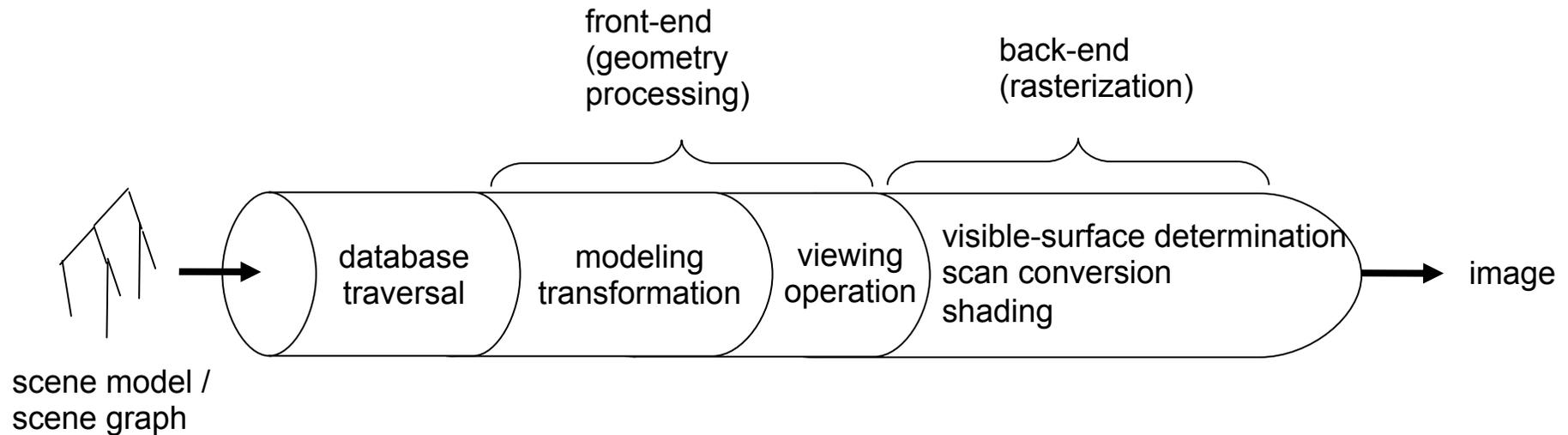
### Begriffe

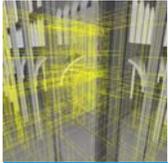
- **Rendering:** Der Prozess der Bildsynthese, also die Abbildung eines Modells bzw. einer Szene auf ein digitales Bild (auf dem Bildschirm).
- **Renderpipeline:** Eine *konkrete Implementierung* dieses mehrstufigen Prozesses in Soft- bzw. Hardware.
- Die einzelnen Stufen der Pipeline setzen die vorgestellten **Basis-Algorithmen der Computergraphik** um.
- Der Aufbau von Renderpipelines kann sich je nach Art und Umsetzung des Renderings stark unterscheiden.



# 6.7 Renderpipeline

## Genereller Aufbau

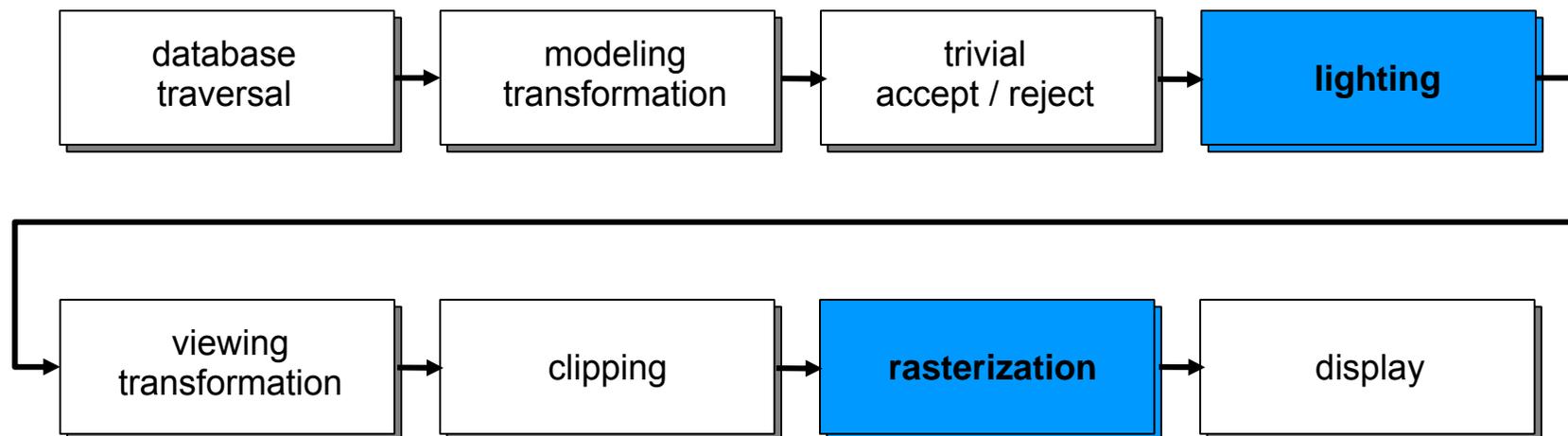


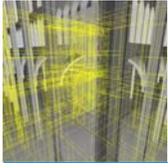


## 6.7 Renderpipeline

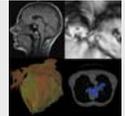
### Beispiel: Lokales Beleuchtungsmodell

- Hier: **Gouraud-Shading mit z-Buffer**



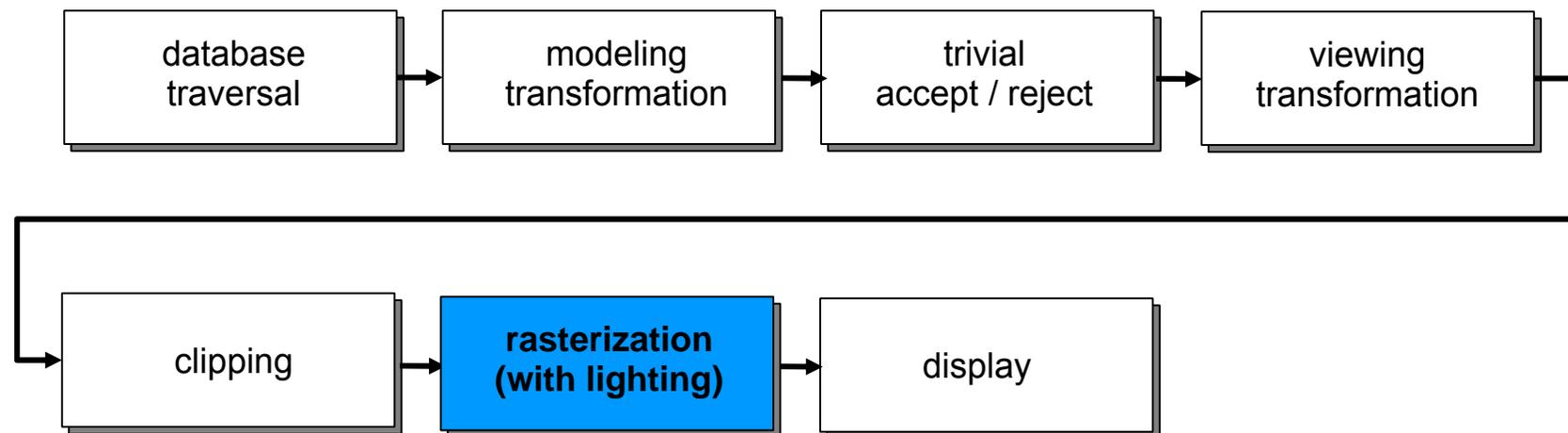


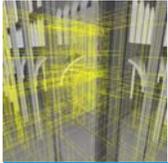
## 6.7 Renderpipeline



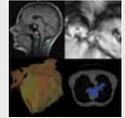
### Beispiel: Lokales Beleuchtungsmodell

- hier: Phong-Shading mit z-Buffer



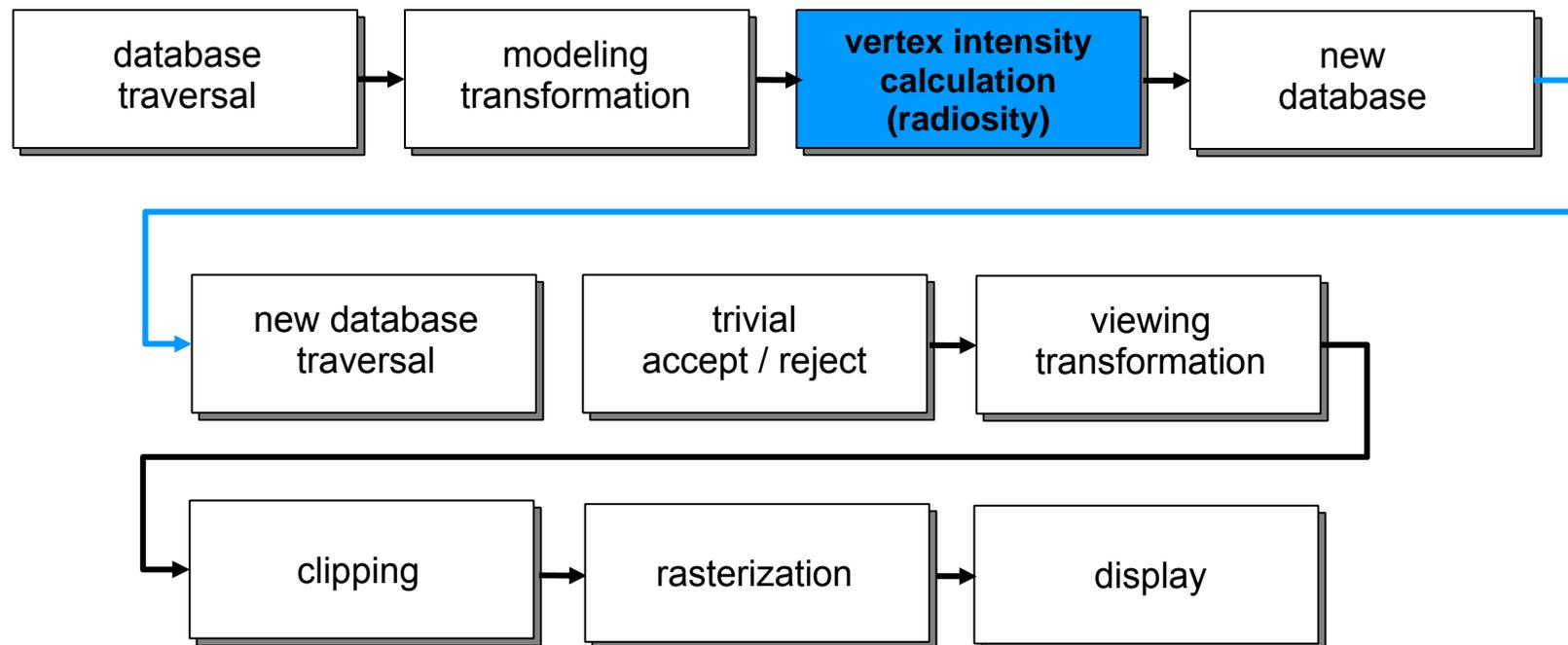


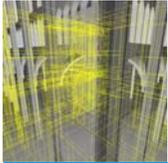
## 6.7 Renderpipeline



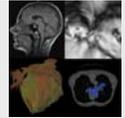
### Beispiel: Globales Beleuchtungsmodell

- hier: Radiosity mit Gouraud-Shading und z-Buffer



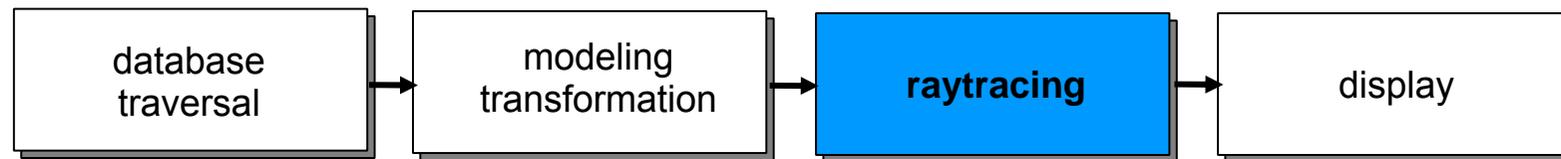


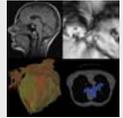
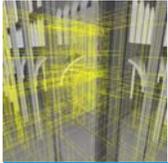
## 6.7 Renderpipeline



### Beispiel: Globales Beleuchtungsmodell

- hier: [Raytracing](#)





- Computergraphik, Universität Leipzig (Prof. G. Scheuermann)
- Graphische Datenverarbeitung I, Universität Tübingen (Prof. W. Straßer)
- Graphische Datenverarbeitung I, TU Darmstadt (Prof. M. Alexa)
- Computergraphik III, Uni Siegen (Rezk-Salama)