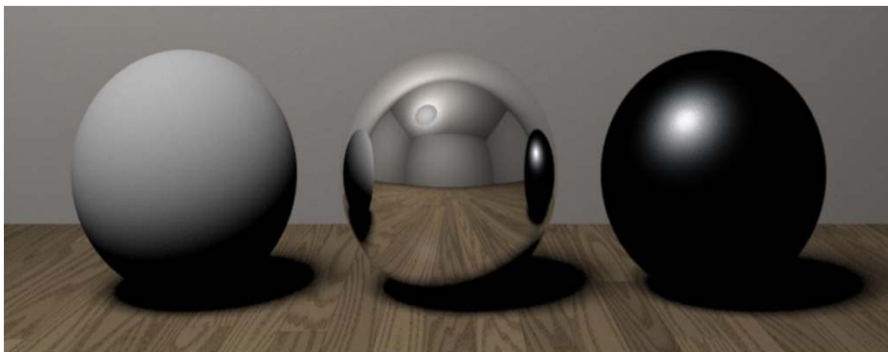


Global Illumination: Raytracing

Die drei Reflexionstypen



ideal diffus
(Lambertian)

ideal
reflektierend

richtungsabhängig
diffus (Phong)



Ray Casting:

Verfolgen eines Strahls durch ein System.

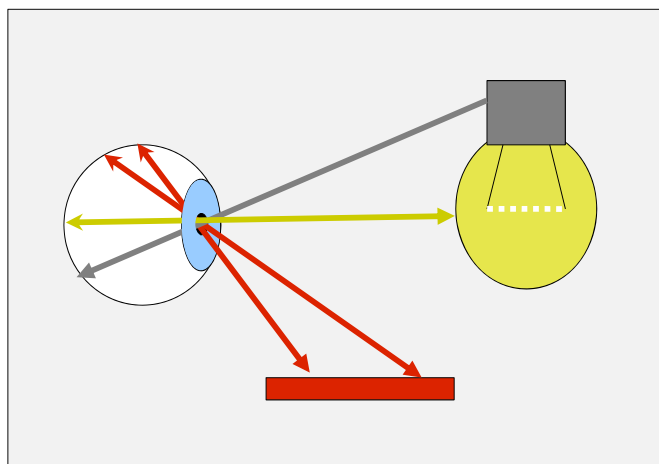
Kann zum Aufbau eines Renderings verwendet werden
(System = 3D Szene).

Vorgeschlagen von **Arthur Appel** in 1968.

Raytracing:

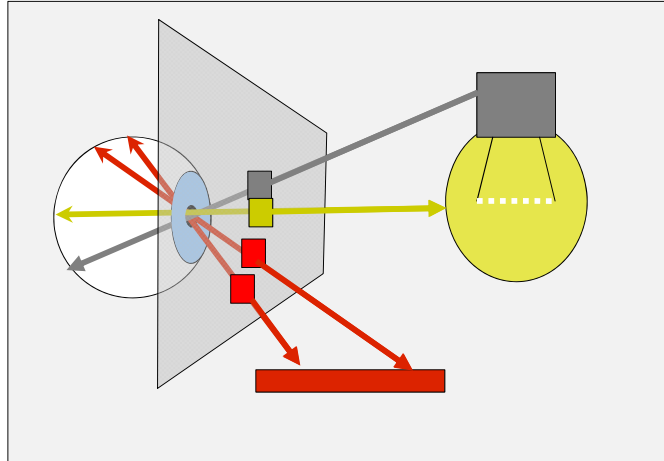
Rendering-Verfahren von Turner & Whitted aus 1980

Ray Casting



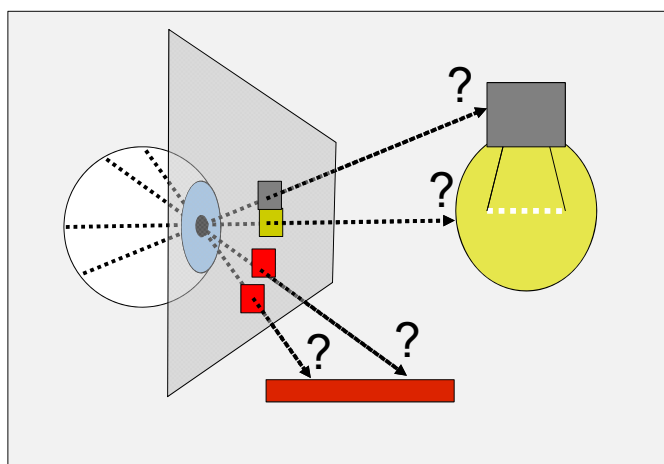
Das Auge misst einfallendes **Licht**.

Ray Casting



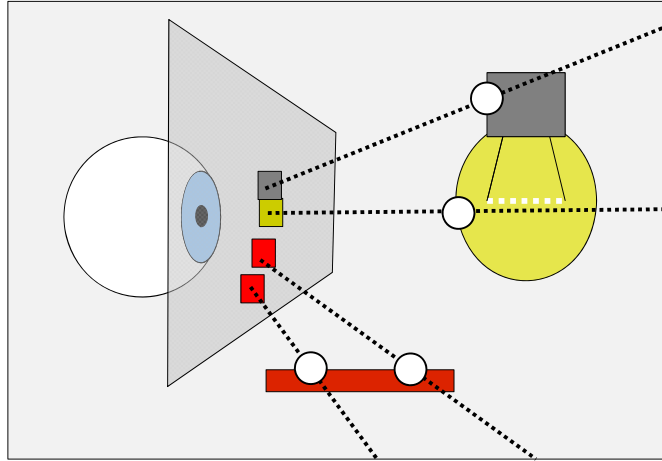
Für jeden Pixel wird ein Strahl verfolgt.

Ray Casting



Für jeden Pixel wird ein Strahl verfolgt.

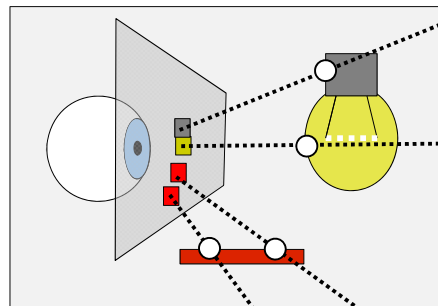
Ray Casting



Die vordersten Schnittpunkte werden ermittelt.

Ray Casting

Liefert Korrespondenz:
Bild → Szene



Offene Frage: Wie wird der Pixel eingefärbt?
Die **gesehene Farbe** hängt vom Licht ab.

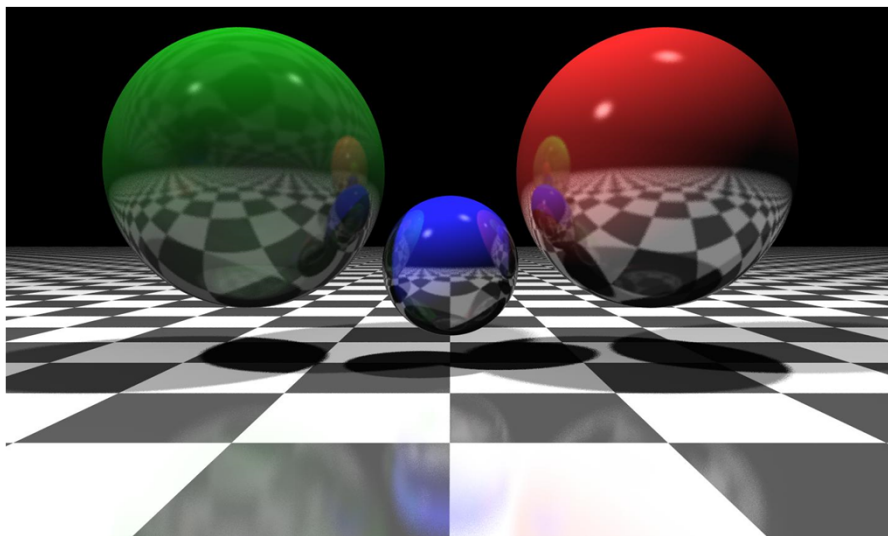
Optionen: Lokales Beleuchtungsmodell
 Raytracing
 Path Tracing, Radiosity, Photon Mapping

...

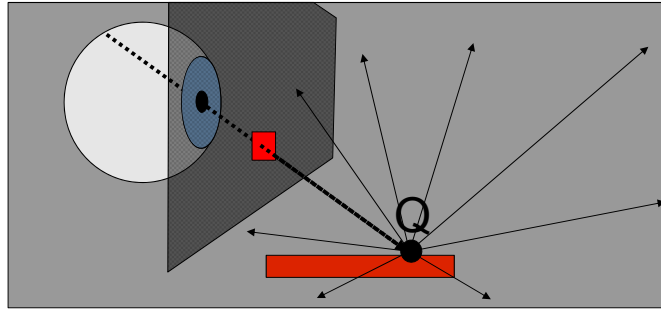
Raytracing (Turner & Whitted)



Raytracing (Turner & Whitted)



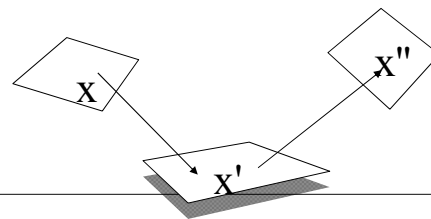
Raytracing



Prinzip: weitere Strahlen von Punkt Q aus verfolgen.
Frage: welche (und wieviele) Richtungen?

Raytracing: Welche Richtungen?

Rendering Equation:

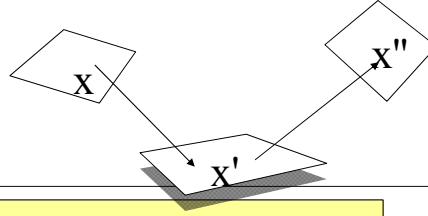


$$L(x' \rightarrow x'') = E(x' \rightarrow x'') + \int_x f_r(x, x', x'') L(x \rightarrow x') V(x \rightarrow x') G(x \rightarrow x') dx$$

$L(A \rightarrow B)$:	Licht von A nach B
$E(A \rightarrow B)$:	Emittiertes Licht von A nach B
$V(A \rightarrow B)$:	Verdeckungsfaktor
$G(A \rightarrow B)$:	Geometriefaktor
$f_r(A \rightarrow B \rightarrow C)$:	Reflektierter Anteil von $L(A \rightarrow B)$ nach C

Raytracing: Welche Richtungen?

Rendering Equation:

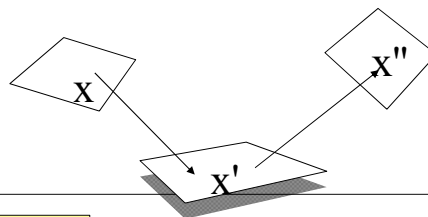


$$L(x' \rightarrow x'') = \overset{\text{aktiv:}}{E(x' \rightarrow x'')} + \overset{\text{passiv:}}{\int_x f_r(x, x', x'') L(x \rightarrow x') V(x \rightarrow x') G(x \rightarrow x') dx}$$

- $L(A \rightarrow B)$: Licht von A nach B
- $E(A \rightarrow B)$: Emittiertes Licht von A nach B
- $V(A \rightarrow B)$: Verdeckungsfaktor
- $G(A \rightarrow B)$: Geometriefaktor
- $f_r(A \rightarrow B \rightarrow C)$: Reflektierter Anteil von $L(A \rightarrow B)$ nach C

Raytracing: Welche Richtungen?

Rendering Equation:

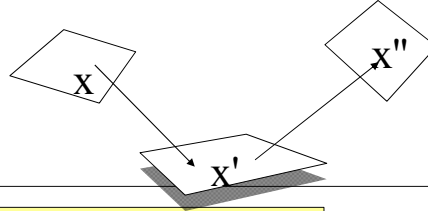


$$L(x' \rightarrow x'') = \overset{\text{aktiv:}}{E(x' \rightarrow x'')} + \int_x \overset{\text{abgestrahlt } x \rightarrow x'}{f_r(x, x', x'') L(x \rightarrow x')} V(x \rightarrow x') G(x \rightarrow x') dx$$

- $L(A \rightarrow B)$: Licht von A nach B
- $E(A \rightarrow B)$: Emittiertes Licht von A nach B
- $V(A \rightarrow B)$: Verdeckungsfaktor
- $G(A \rightarrow B)$: Geometriefaktor
- $f_r(A \rightarrow B \rightarrow C)$: Reflektierter Anteil von $L(A \rightarrow B)$ nach C

Raytracing: Welche Richtungen?

Rendering Equation:



aktiv:

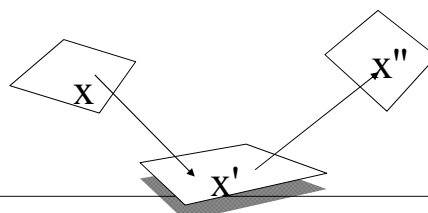
$$L(x' \rightarrow x'') = E(x' \rightarrow x'') + \int_x f_r(x, x', x'') L(x \rightarrow x') V(x \rightarrow x') G(x \rightarrow x') dx$$

angekommen bei x'

- $L(A \rightarrow B)$: Licht von A nach B
- $E(A \rightarrow B)$: Emittiertes Licht von A nach B
- $V(A \rightarrow B)$: Verdeckungsfaktor
- $G(A \rightarrow B)$: Geometriefaktor
- $f_r(A \rightarrow B \rightarrow C)$: Reflektierter Anteil von $L(A \rightarrow B)$ nach C

Raytracing: Welche Richtungen?

Rendering Equation:



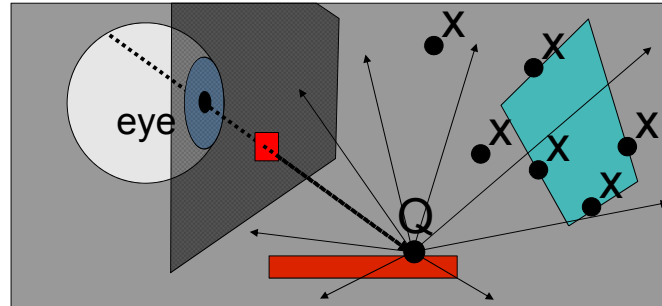
aktiv:

$$L(x' \rightarrow x'') = E(x' \rightarrow x'') + \int_x f_r(x, x', x'') L(x \rightarrow x') V(x \rightarrow x') G(x \rightarrow x') dx$$

weitergeleitet nach x''

- $L(A \rightarrow B)$: Licht von A nach B
- $E(A \rightarrow B)$: Emittiertes Licht von A nach B
- $V(A \rightarrow B)$: Verdeckungsfaktor
- $G(A \rightarrow B)$: Geometriefaktor
- $f_r(A \rightarrow B \rightarrow C)$: Reflektierter Anteil von $L(A \rightarrow B)$ nach C

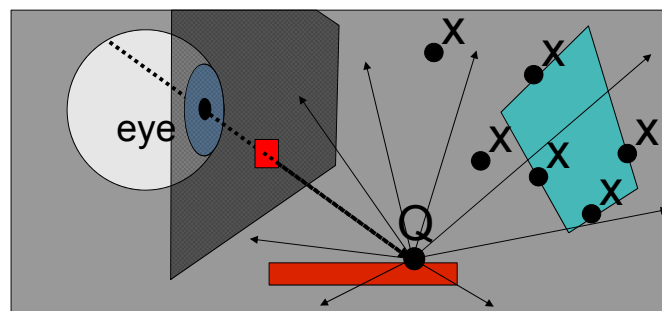
Raytracing: Welche Richtungen?



In unserem Fall:

$$L(Q \rightarrow \text{eye}) = E(Q \rightarrow \text{eye}) + \int_x f_r(x \rightarrow Q \rightarrow \text{eye}) L(x \rightarrow Q) V(x \rightarrow Q) G(x \rightarrow Q) dx$$

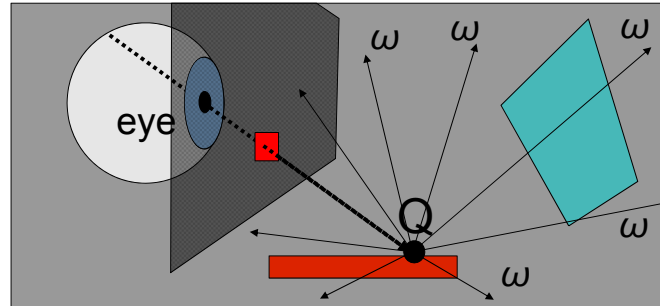
Raytracing: Welche Richtungen?



Anstatt Punkte **X** zu betrachten, ...

$$L(Q \rightarrow \text{eye}) = E(Q \rightarrow \text{eye}) + \int_x f_r(x \rightarrow Q \rightarrow \text{eye}) L(x \rightarrow Q) V(x \rightarrow Q) G(x \rightarrow Q) dx$$

Raytracing: Welche Richtungen?



...betrachten wir Richtungen ω .

$$L(Q \rightarrow \text{eye}) = E(Q \rightarrow \text{eye}) + \int_{\Omega} f_r(\omega \rightarrow Q \rightarrow \text{eye}) L(\omega \rightarrow Q) G(\omega \rightarrow Q) d\omega$$

Raytracing: Welche Richtungen?

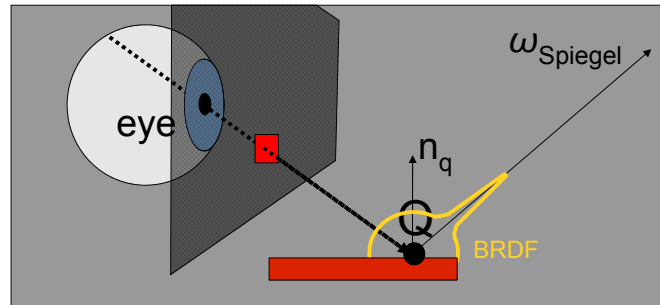
Uns interessieren Richtungen ω_i ,
wo der **Integrand h** besonders gross wird.

$$L(Q \rightarrow \text{eye}) = E(Q \rightarrow \text{eye}) + \int_{\Omega} \overset{h:}{f_r(\omega \rightarrow Q \rightarrow \text{eye}) L(\omega \rightarrow Q) G(\omega \rightarrow Q)} d\omega$$

Dann gilt: $\sum_i h(\omega_i) \approx \int h(\omega) d\omega$.

i

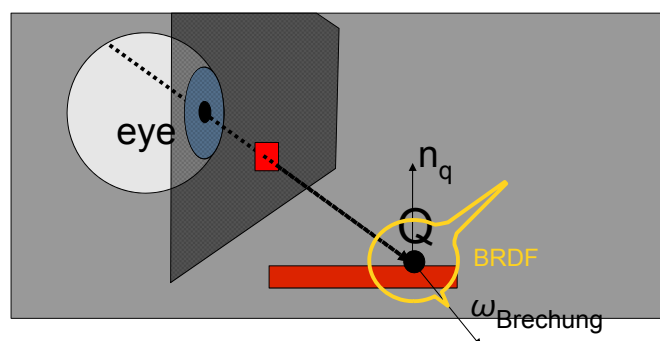
Raytracing: Welche Richtungen?



BRDF $f_r(\omega \rightarrow Q \rightarrow \text{eye})$: peak in Spiegelrichtung bei glattem Q

$$L(Q \rightarrow \text{eye}) = E(Q \rightarrow \text{eye}) + \int_{\Omega} f_r(\omega \rightarrow Q \rightarrow \text{eye}) L(\omega \rightarrow Q) G(\omega \rightarrow Q) d\omega$$

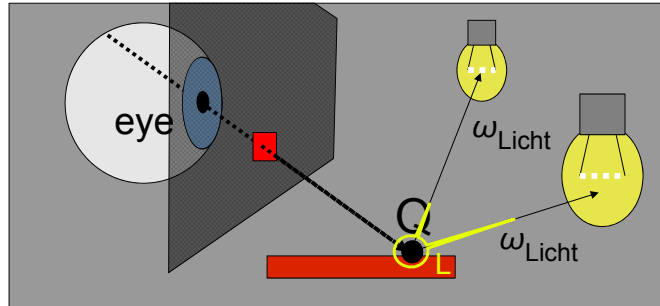
Raytracing: Welche Richtungen?



BRDF $f_r(\omega \rightarrow Q \rightarrow \text{eye})$: weiterer peak in Brechungsrichtung.

$$L(Q \rightarrow \text{eye}) = E(Q \rightarrow \text{eye}) + \int_{\Omega} f_r(\omega \rightarrow Q \rightarrow \text{eye}) L(\omega \rightarrow Q) G(\omega \rightarrow Q) d\omega$$

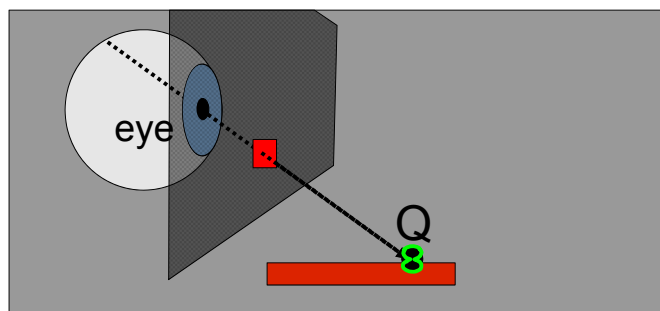
Raytracing: Welche Richtungen?



$L(\omega \rightarrow Q)$: peaks in die Richtungen der Lichtquellen.

$$L(Q \rightarrow \text{eye}) = E(Q \rightarrow \text{eye}) + \int_{\Omega} f_r(\omega \rightarrow Q \rightarrow \text{eye}) L(\omega \rightarrow Q) G(\omega \rightarrow Q) d\omega$$

Raytracing: Welche Richtungen?



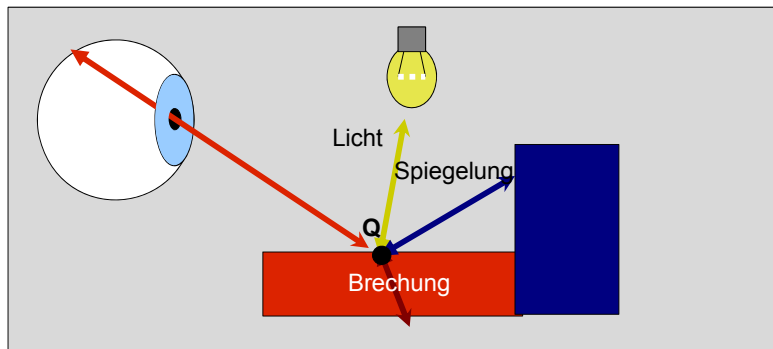
$G(\omega \rightarrow Q)$ ist glatt. Keine "peaks".

$$L(Q \rightarrow \text{eye}) = E(Q \rightarrow \text{eye}) + \int_{\Omega} f_r(\omega \rightarrow Q \rightarrow \text{eye}) L(\omega \rightarrow Q) G(\omega \rightarrow Q) d\omega$$

Basic Raytracing: 3 Richtungen

ω_S : Spiegelrichtung
 ω_B : Brechungsrichtung
 ω_{L_i} : Lichtquellenrichtungen i

Für jede Richtung ω_i wird ein Strahl von Q aus verfolgt.



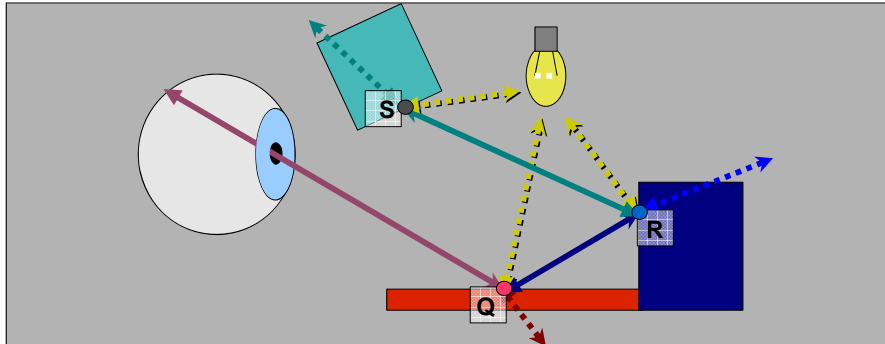
Raytracing: Richtungen

ω_S : Spiegelrichtung
 ω_B : Brechungsrichtung
 ω_{L_i} : Lichtquellenrichtungen i

Für jede Richtung ω_i wird ein Strahl von Q aus verfolgt.

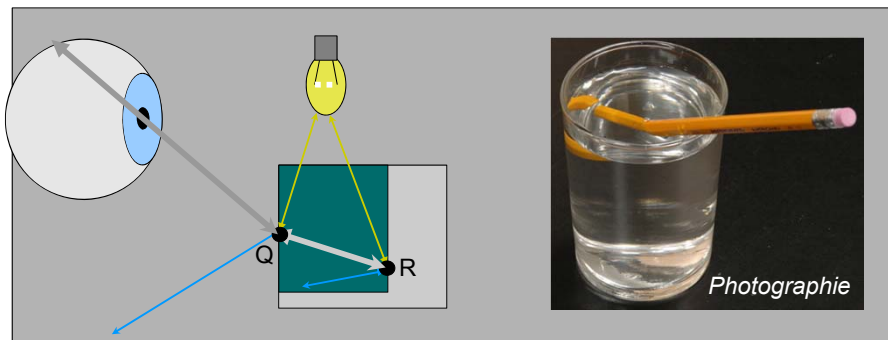
$$\begin{aligned} L(Q \rightarrow \text{eye}) \approx & E(Q \rightarrow \text{eye}) + f_r(\omega_S \rightarrow Q \rightarrow \text{eye}) L(\omega_S \rightarrow Q) G(\omega_S \rightarrow Q) \\ & + f_r(\omega_B \rightarrow Q \rightarrow \text{eye}) L(\omega_B \rightarrow Q) G(\omega_B \rightarrow Q) \\ & + \sum_i f_r(\omega_{L_i} \rightarrow Q \rightarrow \text{eye}) L(\omega_{L_i} \rightarrow Q) G(\omega_{L_i} \rightarrow Q) \end{aligned}$$

Anwendung: Reflexionsstrahl



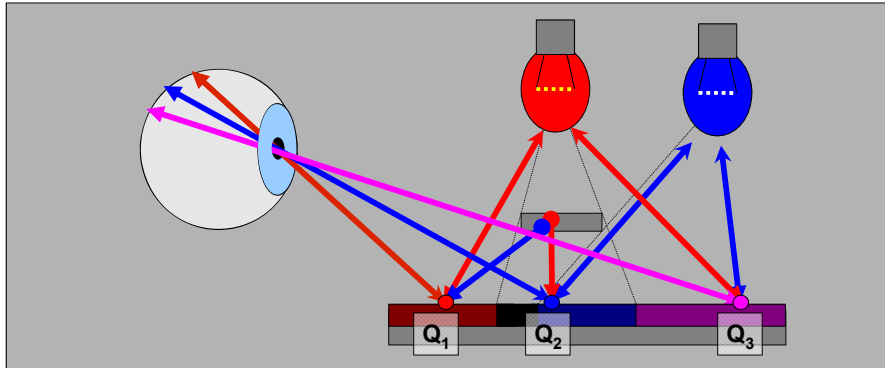
- Ein Strahl in Spiegelrichtung (ω_S), um Punkt **R** zu finden (ray cast).
- Liefert den Spiegelterm: $f_r(R \rightarrow Q \rightarrow \text{eye})$ $L(R \rightarrow Q)$ $G(R \rightarrow Q)$
- Spiegelterm enthält selbst wieder ein $L(R \rightarrow Q)$: Rekursion!

Anwendung: Brechungsstrahl



- Ein Strahl in Brechungsrichtung (ω_B), um R zu finden.
- Wie bei der Reflexion, $L(R \rightarrow Q)$ rekursiv auswerten.

Anwendung: Schattenstrahl (ω_{L_i})



- Zu jeder Lichtquelle (hier 2) wird ein Strahl geschossen.
- Falls er ankommt, wird ein lokales Beleuchtungsmodell evaluiert.
- Keine Rekursion, weil $L(\text{Licht} \rightarrow Q) \approx E(\text{Licht} \rightarrow Q)$.

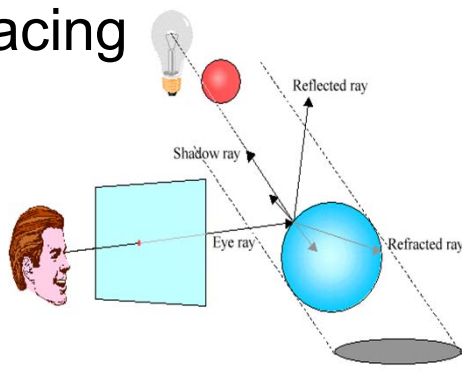
Raytracing: Real

$$I = I_E + k_A I_A + \sum_i I_{L_i} (k_D \langle N, L_i \rangle + k_S \langle V, R_i \rangle^n) S_i + k_S I_S + k_B I_B$$

phong
0 falls im Schatten
1 sonst
Spiegelung
Brechung

Ray Tracing

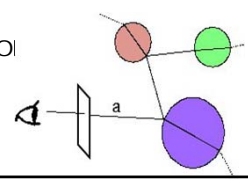
Es ist günstig vom Betrachter die Strahlen in die Szene zu verfolgen.



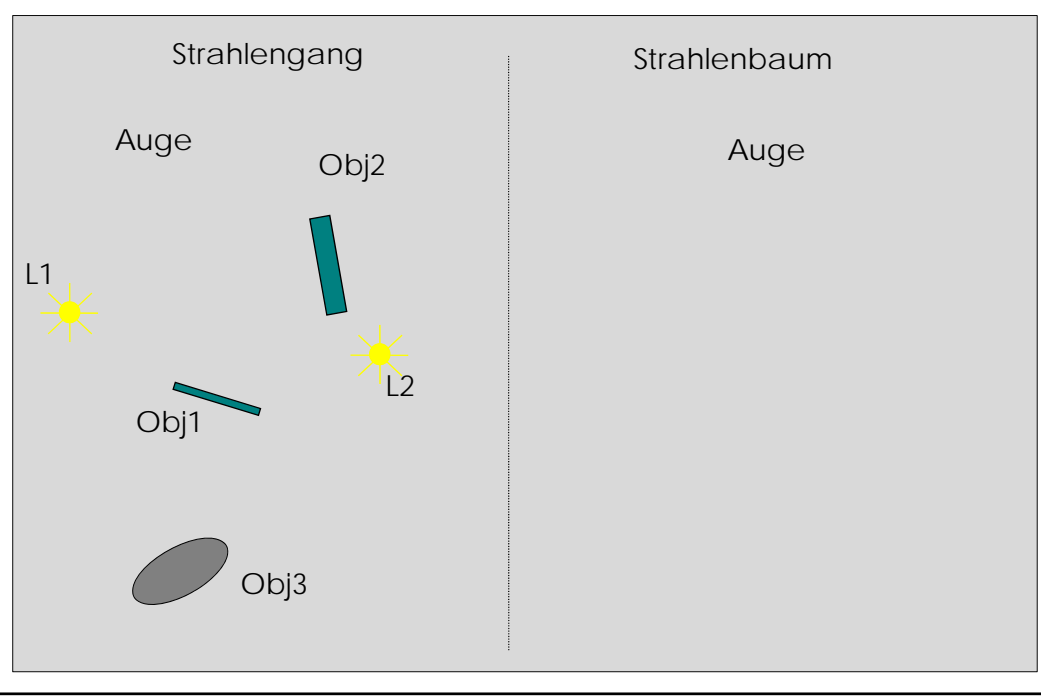
$$I = I_E + k_A I_A + \underbrace{\sum_{L_i} (k_D \langle NL_i \rangle + k_S \langle VR_i \rangle^n)}_{\text{Phong}} I_{L_i} \cdot S_i + \underbrace{k_R I_R + k_B I_B}_{\substack{\text{Reflektion} \\ \text{+ Brechung}}}$$

0 falls im Schatten
1 sonst

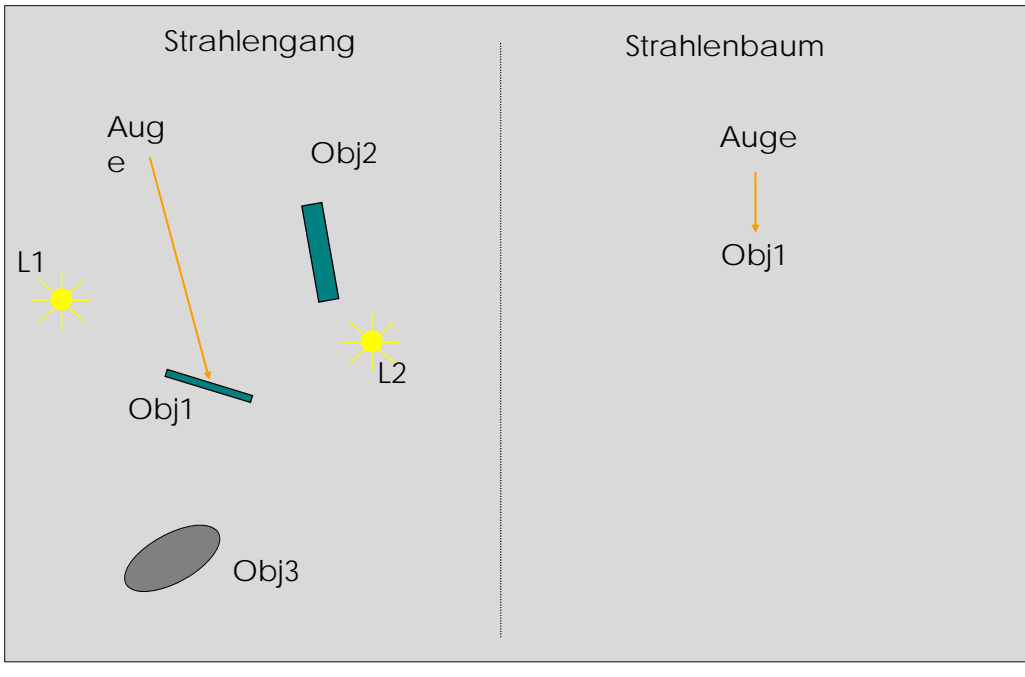
Die Genauigkeit hängt dann von der Anzahl der berechneten Reflexionen ab.



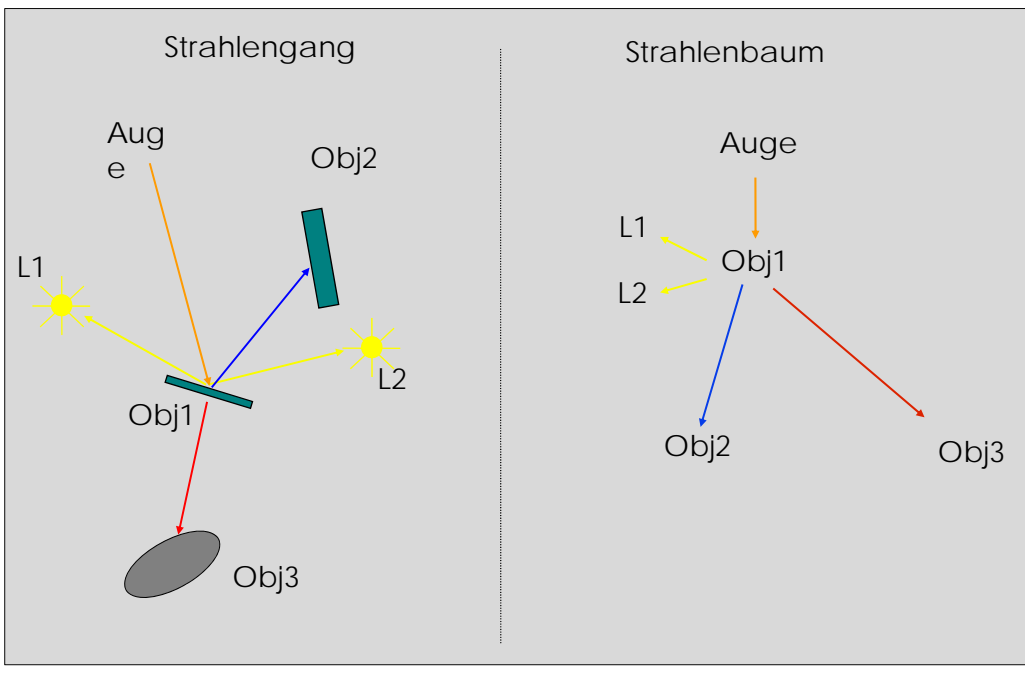
Ray Tracing: ein rekursiver Prozess



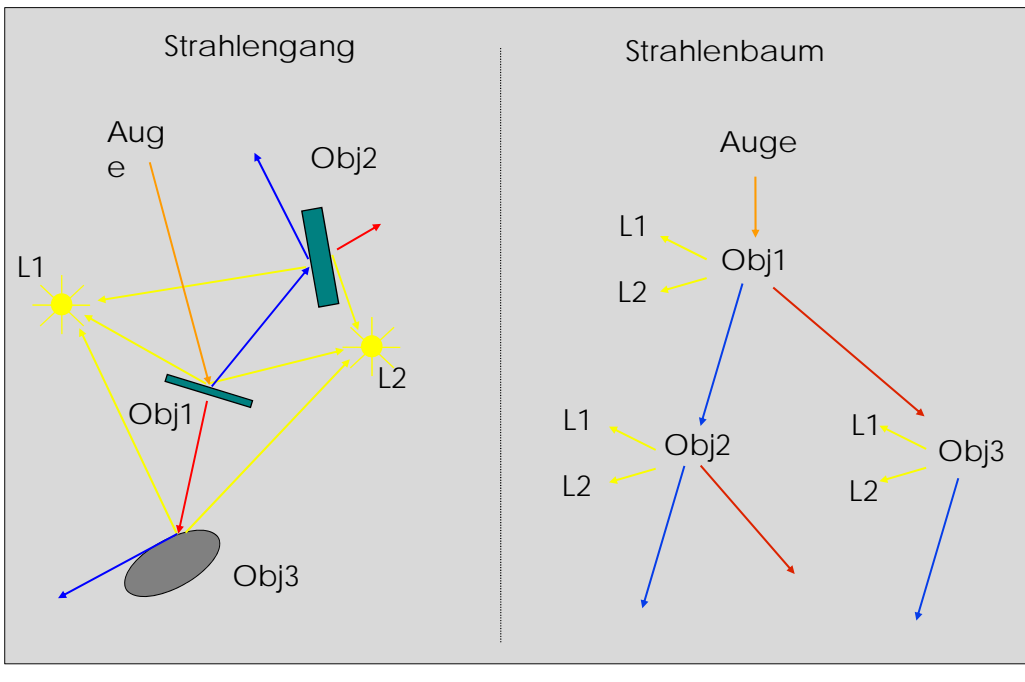
Ray Tracing: ein rekursiver Prozess



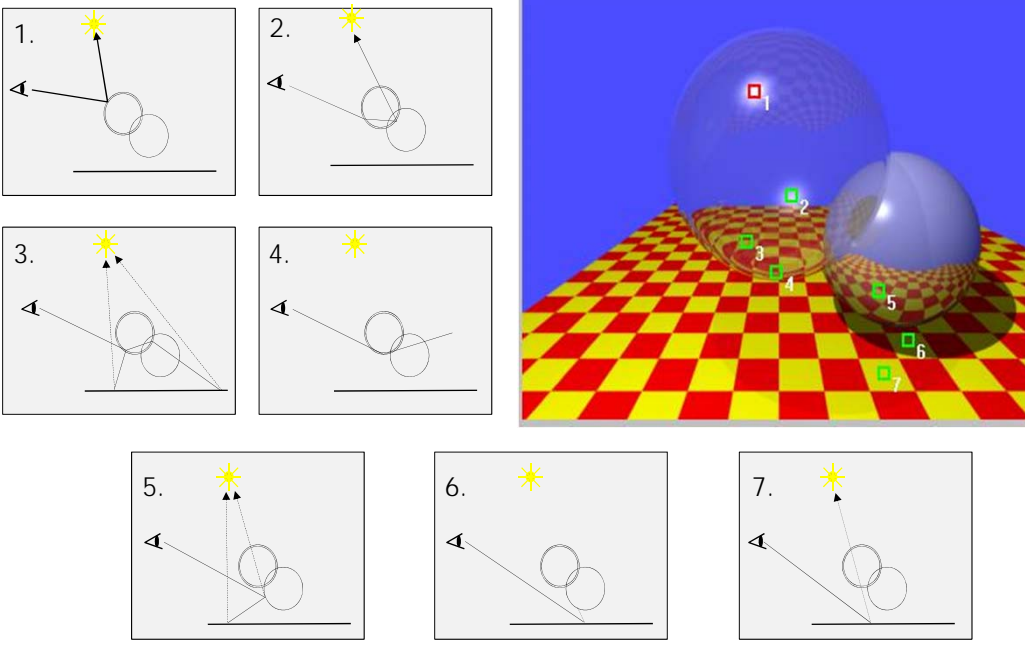
Ray Tracing: ein rekursiver Prozess



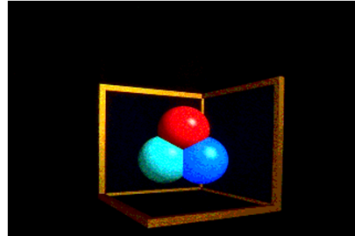
Ray Tracing: ein rekursiver Prozess



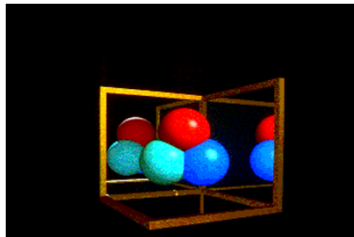
Fallbeispiele



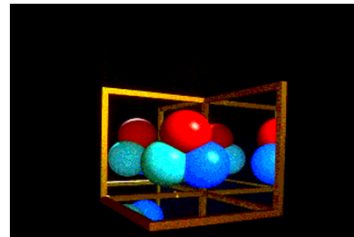
Mehrfach Reflektion



1 Reflektion



2 Reflektionen



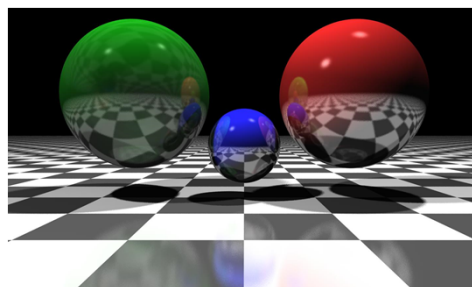
3 Reflektionen

Ray Tracing Basics: Fazit

Grosszügige Approximation eines allgemeinen Beleuchtungsmodells?

Es wird berücksichtigt:

1. Schatten
2. Spiegelung
3. Brechung



Es wird **nicht** berücksichtigt:

1. Diffuse Reflexion zwischen Objekten
2. Ausgedehnte Lichtquellen (weiche Schatten)
3. Spiegelung und Brechung an rauen Oberflächen
4. Bewegungs- und Tiefenunschärfe
5. Brechung oder Spiegelung des Schattenstrahls (Kauistik)

Global Illumination: Advanced Raytracing

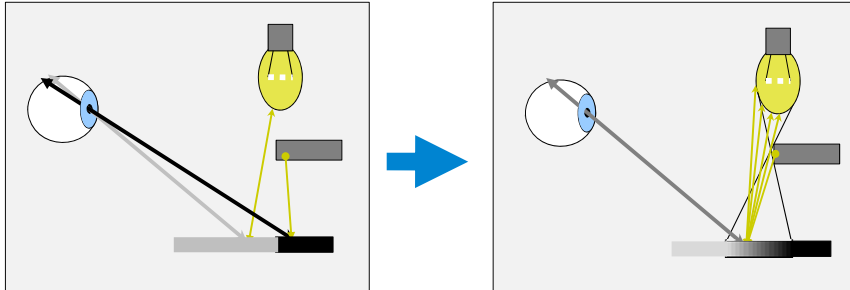
Distribution Raytracing

Bisher wurde **nicht** berücksichtigt:

1. Diffuse Reflexion zwischen Objekten
2. **Ausgedehnte Lichtquellen (weiche Schatten)**
3. **Spiegelung und Brechung an rauhen Oberflächen**
4. **Bewegungs- und Tiefenunschärfe**
5. Brechung oder Spiegelung des Schattenstrahls (Kaustik)

Die drei hervorgehobenen Effekte lassen sich ebenfalls simulieren, indem **mehr als nur die vier Strahlen** (Sicht, Spiegelung, Brechung und Schatten) verfolgt werden.

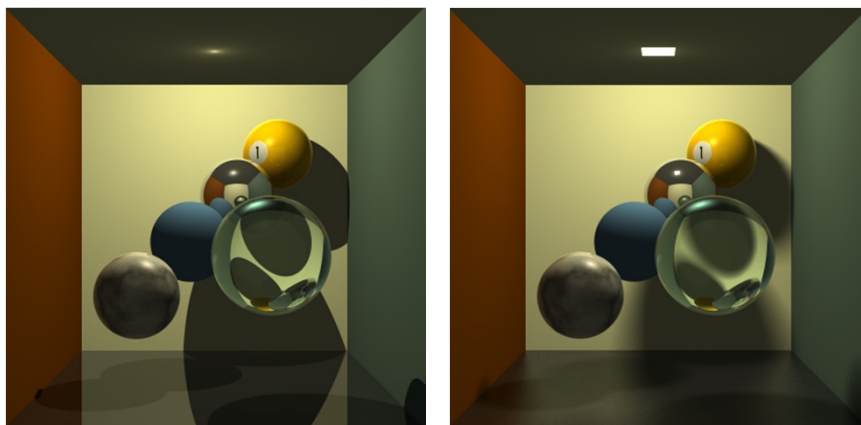
Ausgedehnte Lichtquellen (weiche Schatten)



Statt einem Schattenstrahl zum **Zentrum** der Lichtquelle, werden mehrere solche Strahlen zu **verschiedenen Punkten auf der Oberfläche** der Lichtquelle geschossen.

Wenn nur ein Teil der Strahlen ankommt, liegt der Punkt im Halbschatten (Penumbra).

Ausgedehnte Lichtquellen (Beispiel)

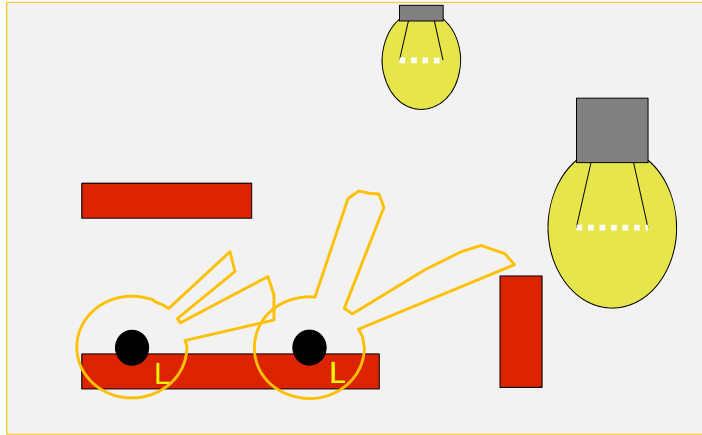


Punktlichtquelle

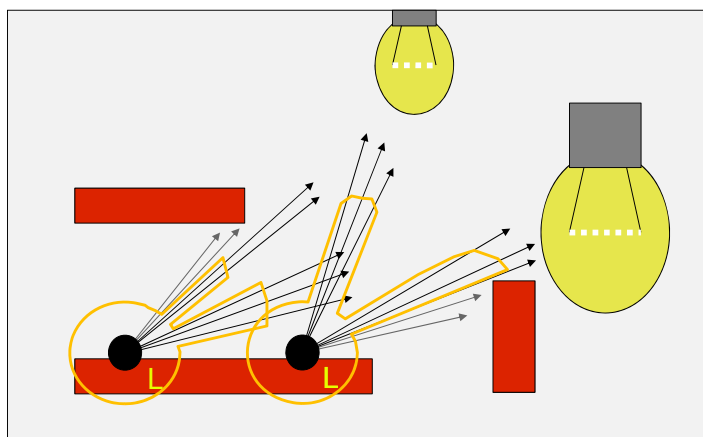
Ausgedehnte Lichtquelle

Bilder von Jason Waltman

Ausgedehnte Lichtquellen: Rechtfertigung

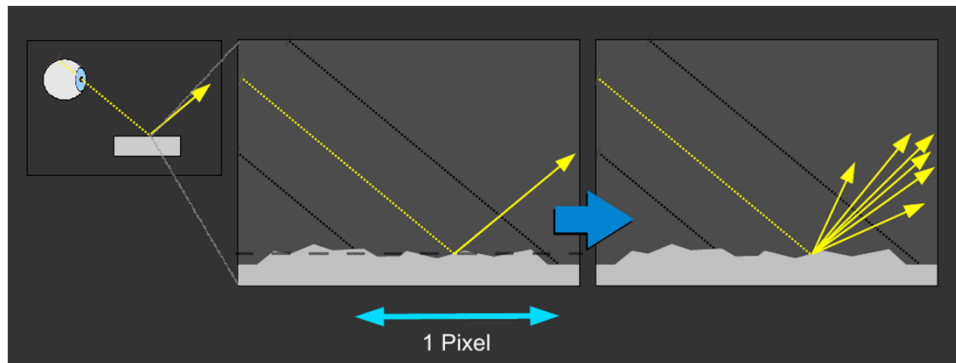


Ausgedehnte Lichtquellen: Rechtfertigung



Genauere Integration durch mehr Samples.
Samples zufällig verteilt → **Monte Carlo Integration**

Spiegelung an rauen Oberflächen
Ein Pixel deckt einen Bereich der Oberfläche ab, nicht nur einen Punkt.
Die Oberfläche weist in diesem Bereich unterschiedliche Ausrichtungen auf.
Das führt zu leicht verwaschenen Reflexionen.
Dieser Effekt lässt sich simulieren indem statt einem Reflexionsstrahl
mehrere leicht versetzte Strahlen pro Pixel geschossen werden.



Analog: breitere BRDF

Rauhere Oberfläche durch mehrere **Spiegelstrahlen**.



Perfekte Spiegelung



Verwaschene Spiegelung

Bilder von Henrik Wann Jensen

Spiegelung an rauhen Oberflächen (Beispiel)



"A McIntosh in the Kitchen" by Marc Jacquier (2004)

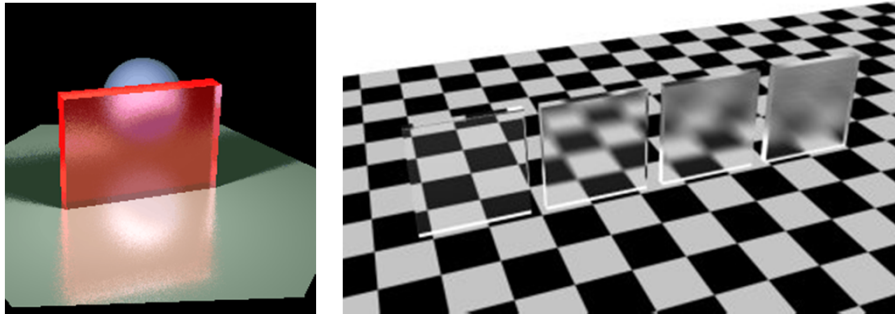
Spiegelung an rauhen Oberflächen



"Rocket" by Jochen Diehl (2004)

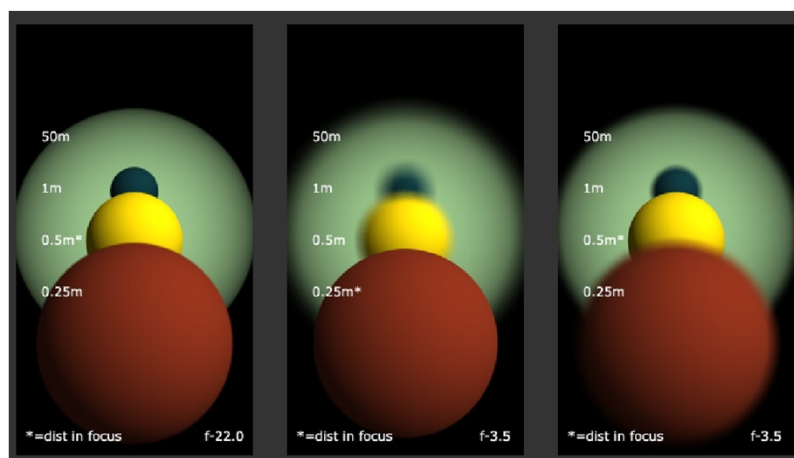
Brechung an rauhen Oberflächen

Verwaschene Brechung durch mehrere **Brechungsstrahlen**.



Tiefenschärfe

Eine reale Linse (auch jene im Auge) kann nur Punkte in einem bestimmten Abstand scharf abbilden. Punkte, die zu nahe oder zu weit entfernt sind, werden unscharf.



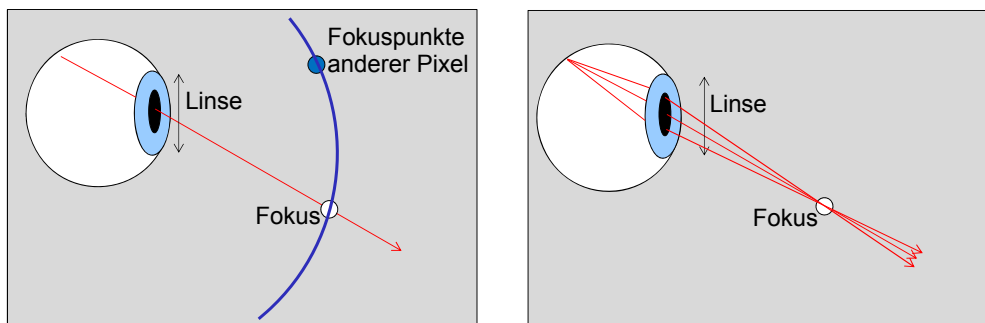
Bilder von Jason Waltman

Tiefenschärfe

Es werden für jeden Pixel mehrere Sehstrahlen von verschiedenen Punkten auf der Linsenoberfläche zu einem gegebenen Fokuspunkt ausgesandt.

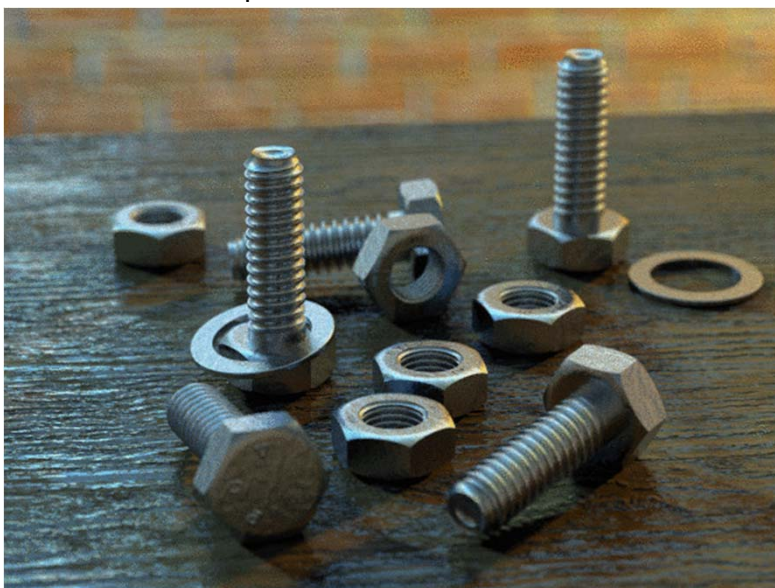
Dieser Fokuspunkt variiert über den Pixeln.

Er fährt eine Projektion des Bildes im Fokusabstand ab.



Tiefenschärfe

Mehrere **Sichtstrahlen** pro Pixel:



"Still with Bolts" by Jaime Vives Piqueres (2002)

Tiefenschärfe



"Columns" by Jaime Vives Piqueres

Bewegungsunschärfe

Eine Kamera sammelt Licht über einen endlichen Zeitraum hinweg (Belichtungszeit).

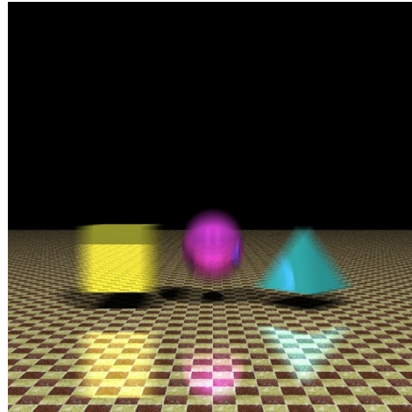
Dies führt zu Bewegungsunschärfe.

Lösung:

- Für jeden Pixel werden mehrere Strahlen vom Auge ausgesandt.
- Als Zeitpunkt für jeden Strahl wird ein Punkt in der Nähe des Moments der Bildaufnahme gewählt.
- Jeder Strahl wird wie ein gewöhnlicher Sehstrahl durch die Szene geschickt. Diese befindet sich in einem Zustand, der dem Zeitpunkt des Strahls entspricht.
- Die Ergebnisse der Strahlen werden gemittelt.

Bewegungsunschärfe (2)

Eine Kamera sammelt Licht über einen endlichen Zeitraum hinweg (Belichtungszeit).
Dies führt zu Bewegungsunschärfe.



Bewegungsunschärfe (3)

Eine Kamera sammelt Licht über einen endlichen Zeitraum hinweg (Belichtungszeit).
Dies führt zu Bewegungsunschärfe.



Distribution Raytracing

Kann mit distribution raytracing angegangen werden:

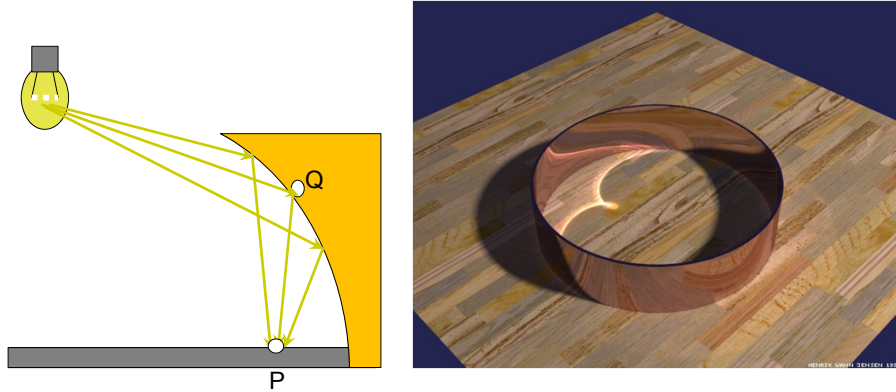
1. Diffuse Reflexion zwischen Objekten
2. Ausgedehnte Lichtquellen (weiche Schatten)
3. Spiegelung und Brechung an rauen Oberflächen
4. Bewegungs- und Tiefenunschärfe
5. Brechung oder Spiegelung des Schattenstrahls (Kaustik)

Distribution Raytracing

Benötigt aufwendigere Methoden, weil das Licht aus **beliebiger** Richtung einfallen kann:

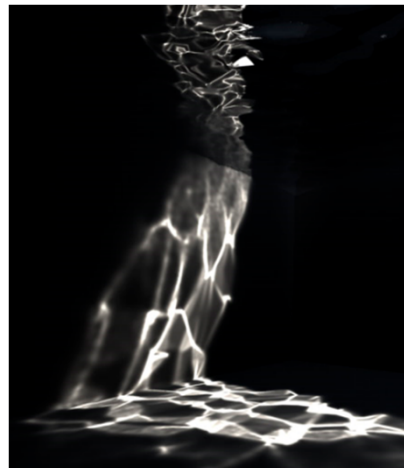
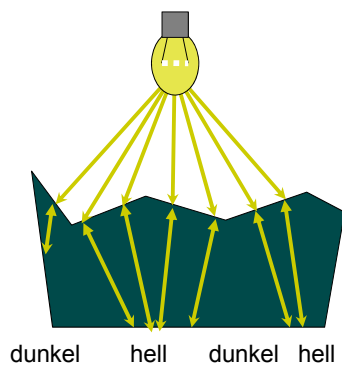
1. Diffuse Reflexion zwischen Objekten
2. Ausgedehnte Lichtquellen (weiche Schatten)
3. Spiegelung und Brechung an rauen Oberflächen
4. Bewegungs- und Tiefenunschärfe
5. Brechung oder Spiegelung des Schattenstrahls (Kaustik)

Reflexionskaustik



- Der Lichtstrahl wird (spekulär) reflektiert, und erreicht Punkt P aus unbekannter Richtung (ua. Punkt Q).
- Ausgehend von Punkt P allein lässt sich Punkt Q nicht ermitteln.
- Jeder glänzende Punkt in der Szene kommt in Frage.

Brechungskastik



- Der Lichtstrahl wird diesmal gebrochen, und erreicht Punkt P wieder aus unbekannter Richtung.
- Jeder Punkt auf der Wasseroberfläche könnte Licht zu jedem Punkt auf dem Grund brechen.

Wie wird Kaustik simuliert?

Erst seit 1995 von Henrik Wann Jensen mittels Photon Mapping effizient simuliert.

Idee:

- Photonen von Lichtquelle aus verfolgen
- Bei Kollision Brechungs- und Reflektionsrichtung berechnen und weiter verfolgen
- Bei weiteren Kollision Position in einer Liste speichern
- Szene rendern und neben Phong Shading auch Photonendichte mit berechnen

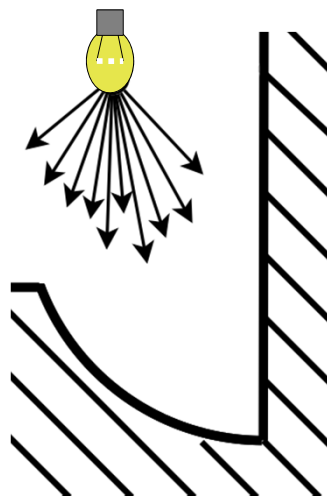


Henrik Wann Jensen: <http://graphics.ucsd.edu/~henrik/>
„Global Illumination using Photon Maps”.
In "Rendering Techniques '96".
Eds. X. Pueyo and P. Schröder.
Springer-Verlag, pp. 21-30, 1996

Photon Mapping: Pseudocode (1)

PhotonMap()

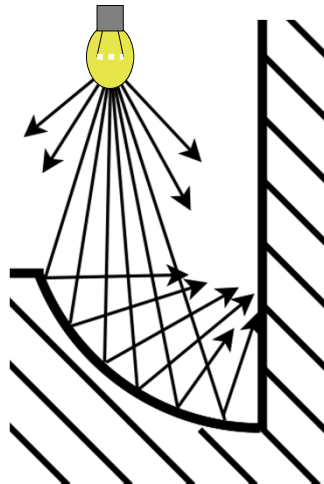
- 1: $N \leftarrow 10000$ { max Photons }
- 2: **for all lights do**
- 3: **for** $j < N$ **do**
- 4: $dir \leftarrow RandomDir()$
- 5: $TracePhoton(light.pos, dir, 1)$
- 6: **end for**
- 7: **end for**



Pseudocode (2)

TracePhoton(pos,dir ,recursion)

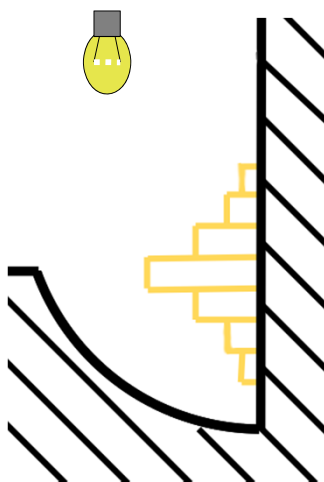
- 1: **for all objects do**
- 2: **if** *closestIntersection* and *recursion* \neq 0
- then**
- 3: TracePhoton(*intersecPos*; *reflDir*; 0)
- 4: TracePhoton(*intersecPos*; *refrDir*; 0)
- 5: **else**
- 6: *photons.push(intersecPos)*
- 7: **end if**
- 8: **end for**



Pseudocode (3)

CalcLightIntensity(pos,surfNormal)

- 1: *intensity* \leftarrow 0
- 2: **for all photons do**
- 3: **if** (*dis* \leftarrow *photon.distanceTo(pos)*) $<$ *maxPhotonDis*
- then**
- 4: *intensity* \leftarrow *intensity* + *photon.dir* · *surfNormal* * $\sqrt{(\text{maxPhotonDis} - \text{dis}) / \text{photonCorrection}}$
- 5: **end if**
- 6: **end for**
- 7: **return** *intensity*



Photon Mapping

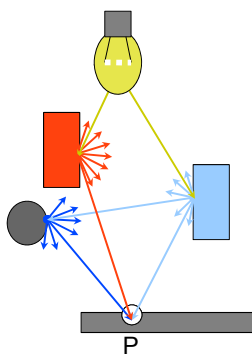
Pro

- Gute Resultate bei wenig Rechenaufwand
- Einfache Implementation
- Simuliert zusätzlich diffuse Reflektion und Subsurface Scattering

Kontra

- Ineffizient bei Aussenszenen (Paralleles Licht)
- Parameter abhängig (Aufwändiges Kalibrieren)
- Physikalisch inkorrekt (Nicht Energie erhaltend)

Diffuse Reflexion zwischen Objekten



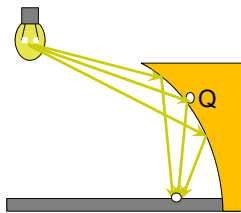
- Der Lichtstrahl wird von einer anderen Fläche diffus reflektiert.
- Jeder von P aus sichtbare Punkt in der Szene kommt in Frage.

Diffuse Reflexion zwischen Objekten

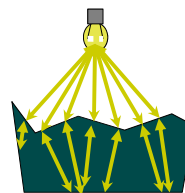
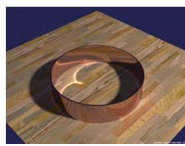


By Spencer & Jones 09

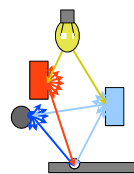
Zusammenfassung



Reflexionskaustik



Brechungskaustik



Diffuse Interreflexion

- Alle drei Effekte werden durch Licht hervorgerufen, das weder aus der Richtung der Lichtquelle, noch aus der Reflexions- oder Brechungsrichtung eintrifft.
- Um das Licht einzufangen, müssten mehr als nur diese drei Richtungen getestet werden.