

# Teil 8: Ray Tracing

**Beleuchtung,  
inkl. Schatten, Reflexionen**

# Ray Tracing – Einleitung

## Forward Ray Tracing:

- ◆ Lichtwege verfolgen: wohin fällt Licht?
- ◆ vgl. shooting (Radiosity)
- ◆ Strahlen verfehlen Auge: aufwendig!

## Backward Ray Tracing:

- ◆ „Blickstrahl“-Verfolgung: was sehe ich?
- ◆ vgl. gathering (Radiosity)
- ◆ Beleuchtung rekonstruieren



# (Bwd.) Ray Tracing

**Idee:** für jedes Pixel:

- ◆ berechne Blickstrahl (**Kameramodell**)
- ◆ verfolge Blickstrahl in die Szene, ermittle nächsten Schnittpunkt (**Schnittpunktberechnung**)
- ◆ ermittle **Beleuchtung** (Phong, Schatten)
- ◆ ermittle **indirekte Beleuchtung** (Reflexionen, Brechungen)
- ◆ kombinieren Farbeindruck = Pixel-Wert
- ◆ evtl. mehr . . .

# Ray Tracing – Bausteine

## Kameramodell

- ◆ Blickstrahlermittlung

## Sichtbarkeitsberechnung = Schnittberechnungen++

- ◆ welches Objekt sehe ich (pro Pixel)

## Schattierung bzw. Beleuchtung

- ◆ Schattenermittlung (Schattenfühler)
- ◆ lokales Beleuchtungsmodell (Phong)

## Indirekte Einflüsse, Rekursion

- ◆ Reflexionen, Brechungen

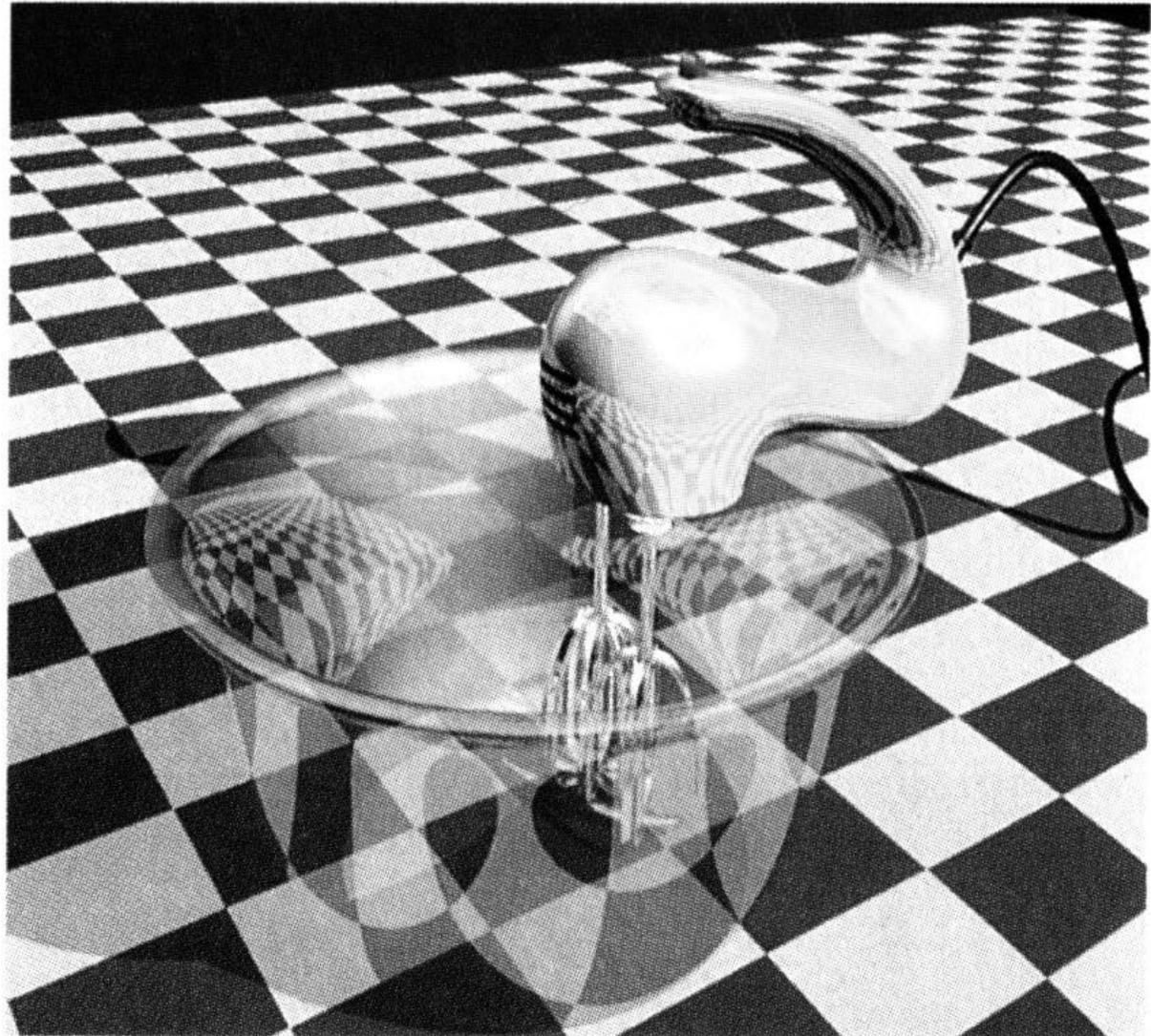
# Ray Tracing

## Charakteristika anhand von Beispielen

# RT: Reflexionen, Brechungen

Normalerw.  
aber nicht:

- ◆ Kaustiken
- ◆ Prismen-  
effekt



# RT: Photo-Realismus

Normalerw.  
aber nicht:

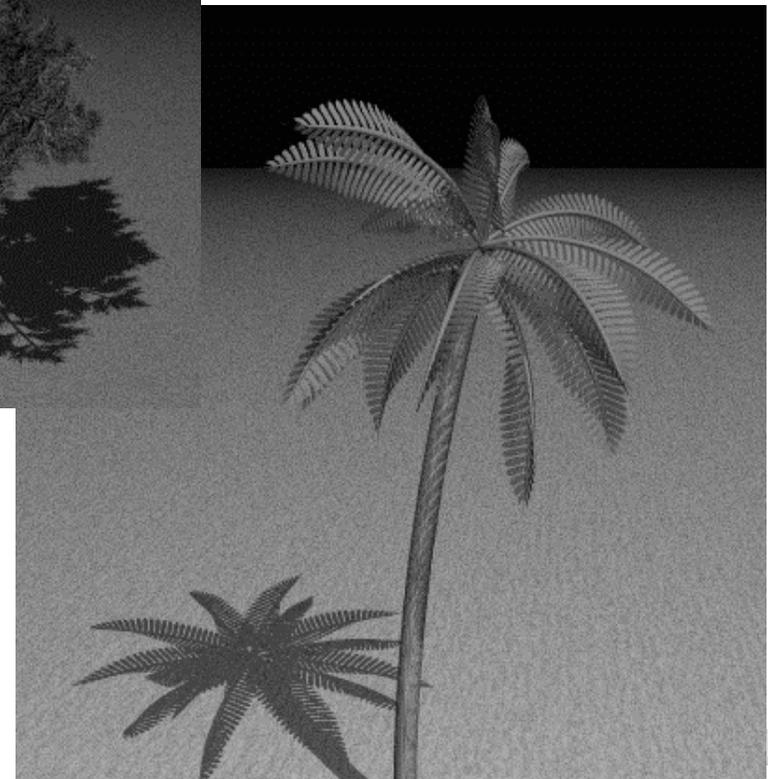
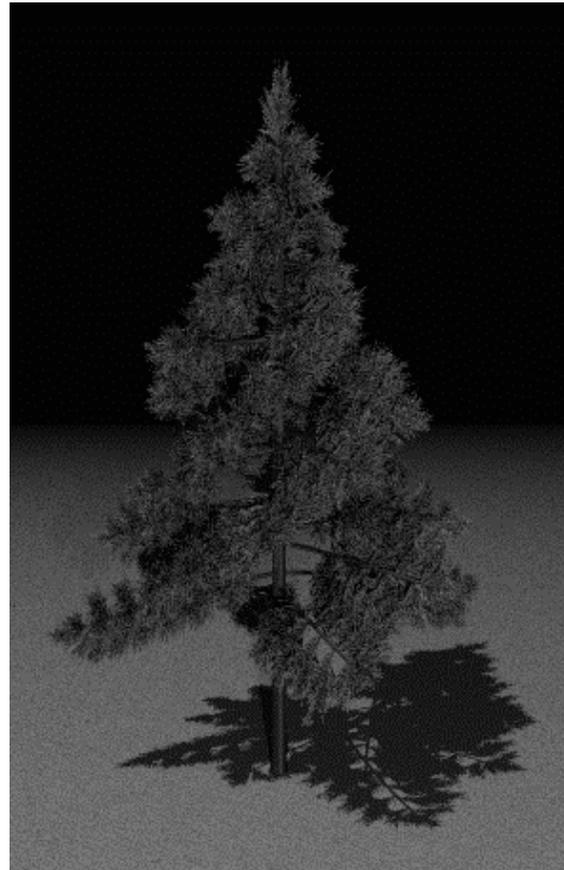
- ◆ Schmutz
- ◆ Out-door
- ◆ Tiefen-  
unschärfe



# RT: Licht und Schatten

Normalerw.  
aber nicht:

- ◆ Halbschatten, i.e., flächige bzw. räuml. Lichtquellen



# RT: Komplexe Objekte

Normalerw.  
aber nicht:

- ◆ komplexe Modelle realer Obj. (Tiere, etc.)



# Ray Tracing – Charakteristika

Photo-realistische Ergebnisbilder

Sichtbarkeitsalgorithmus

Lichtquellen (auch mehrere), Schatten

Reflexionen, Brechungen

Komplexe Objekte (gekrümmt, fraktal)

Sehr aufwendig

# Ray Tracing – Strahlverfolgung

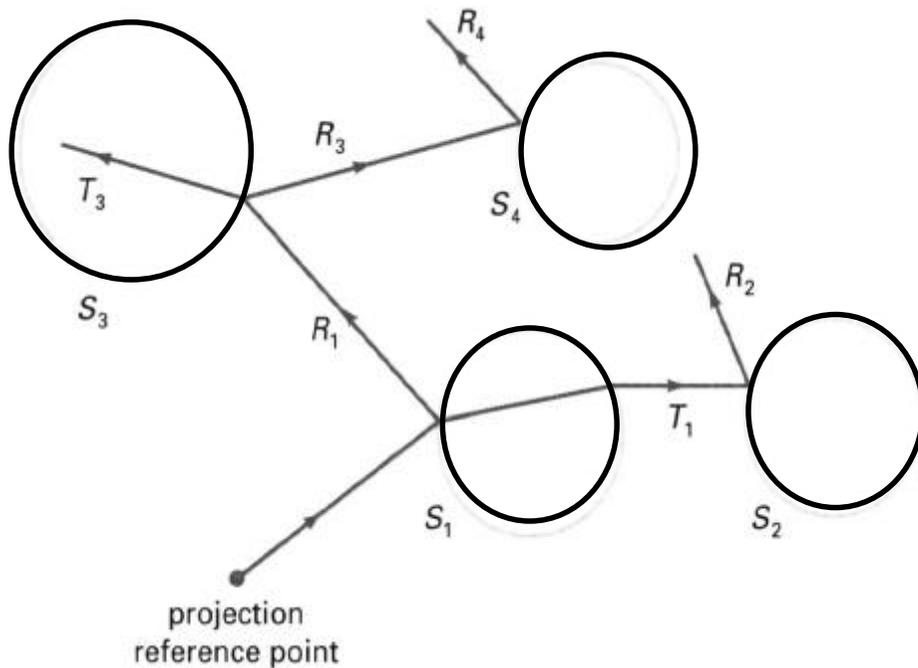
## Primärstrahlen, Sekundärstrahlen

# Ray Tracing – Strahlen

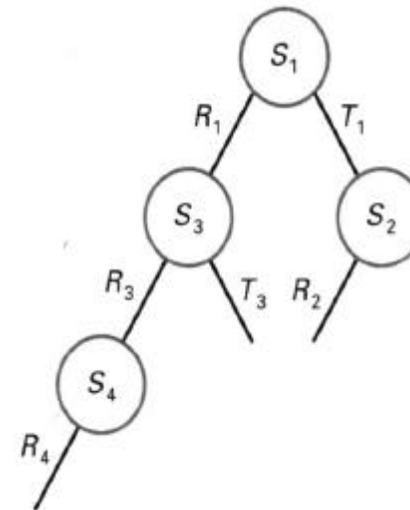
## Primärstrahlen

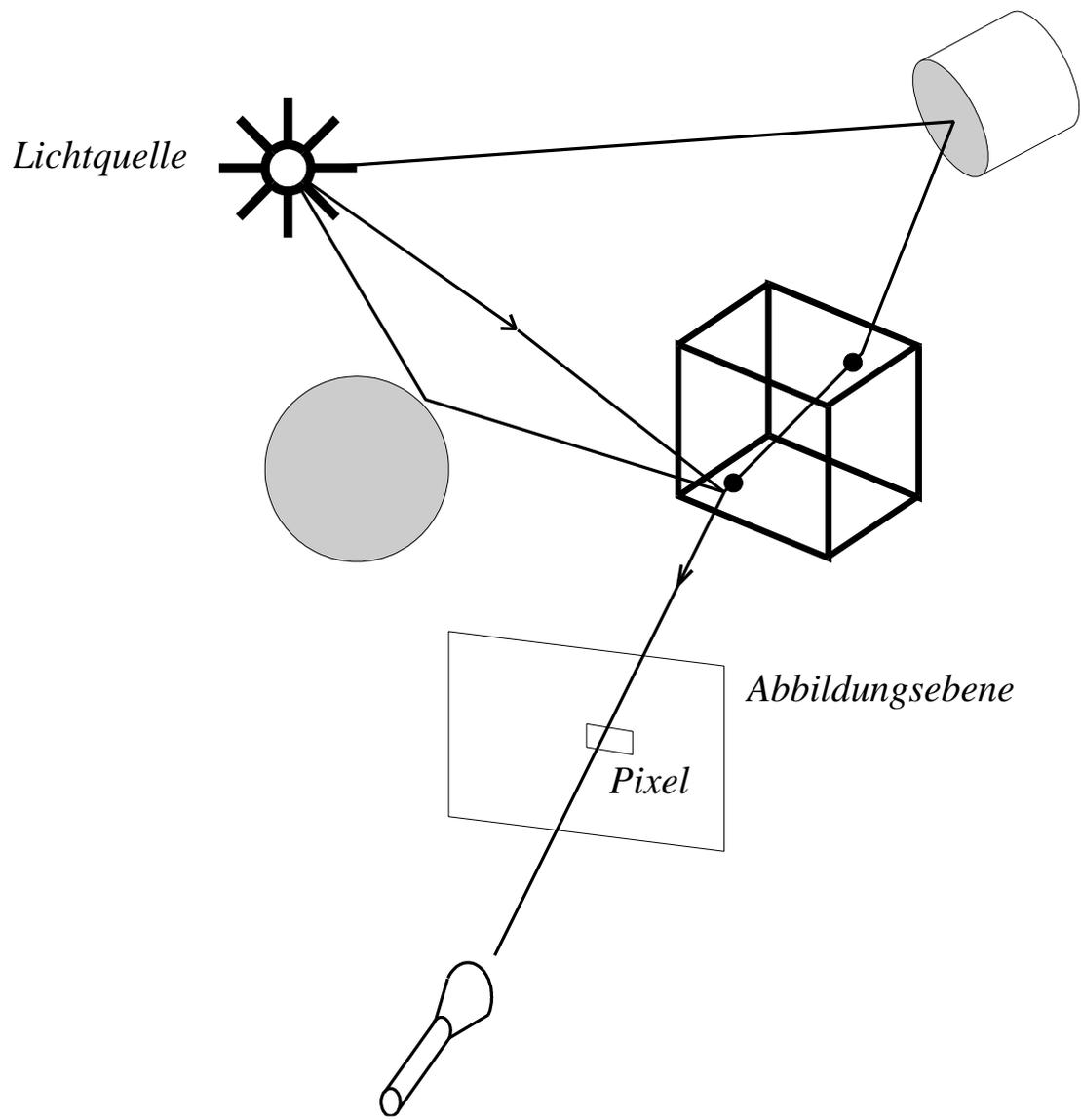
## Sekundärstrahlen:

- ◆ Reflexionen R, Brechungen T

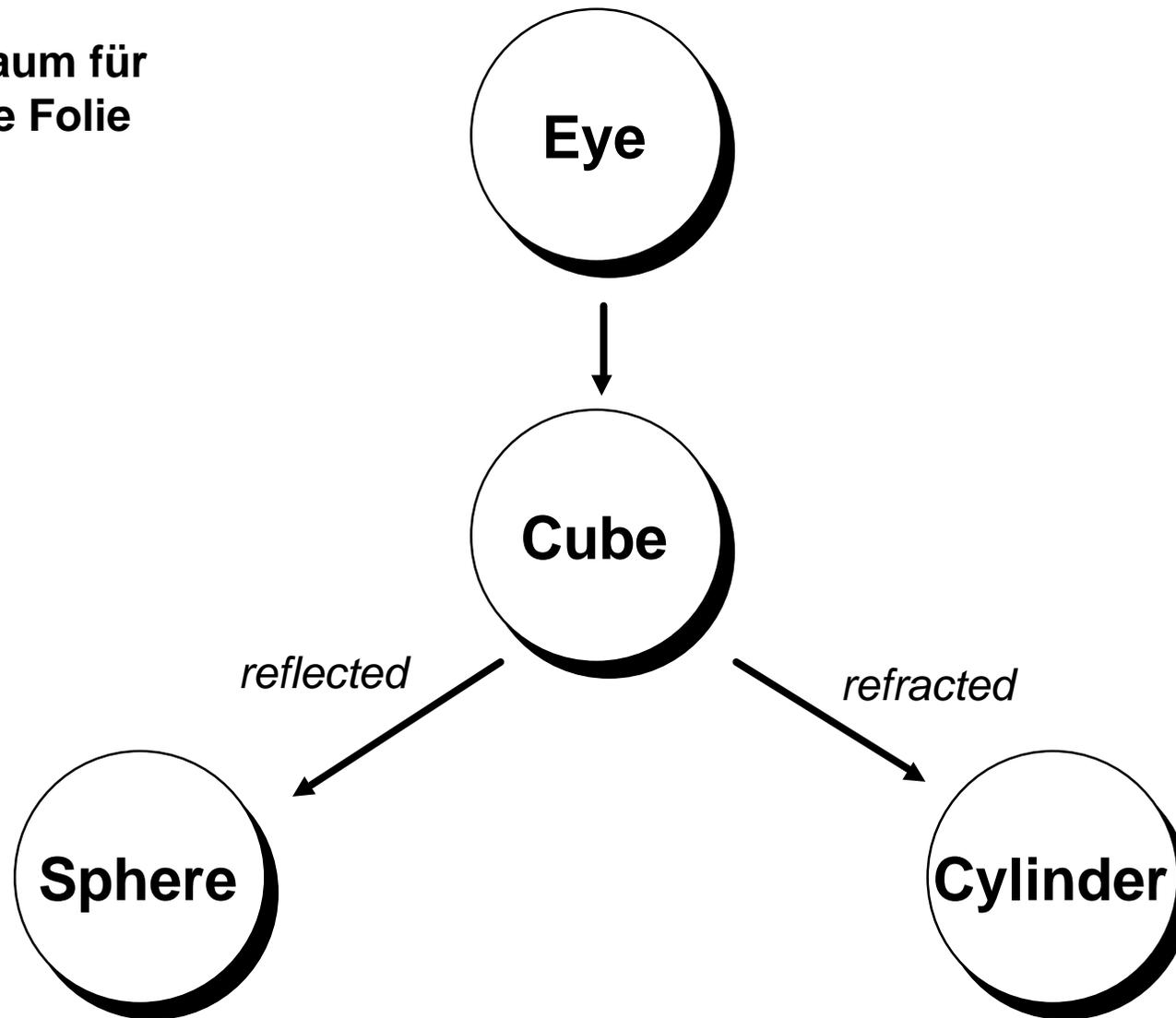


## RT-Graph





**\*) RT Baum für  
gezeigte Folie**



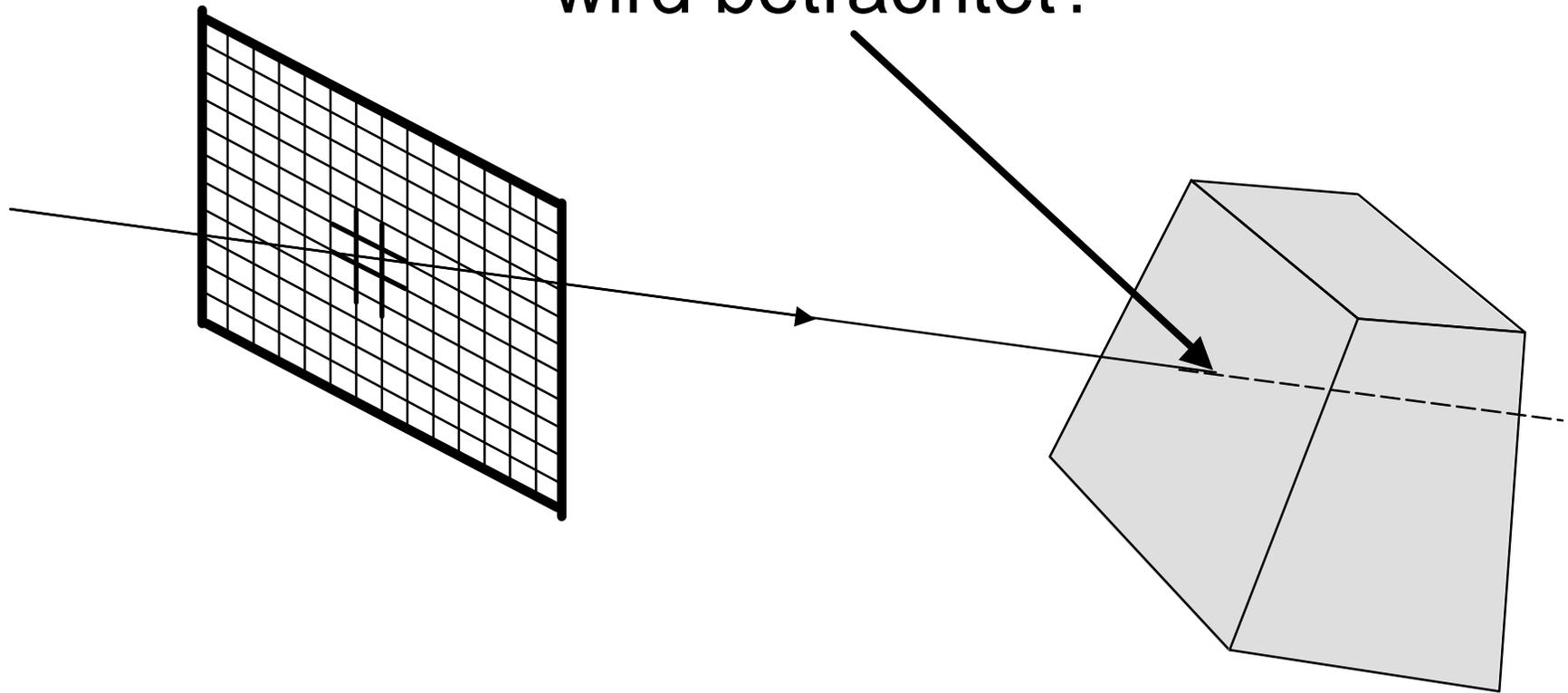
# Ray Tracing – Algorithmus

**Sehstrahl,  
Schnittberechnung,  
Beleuchtung & Schatten,  
Rekursion**

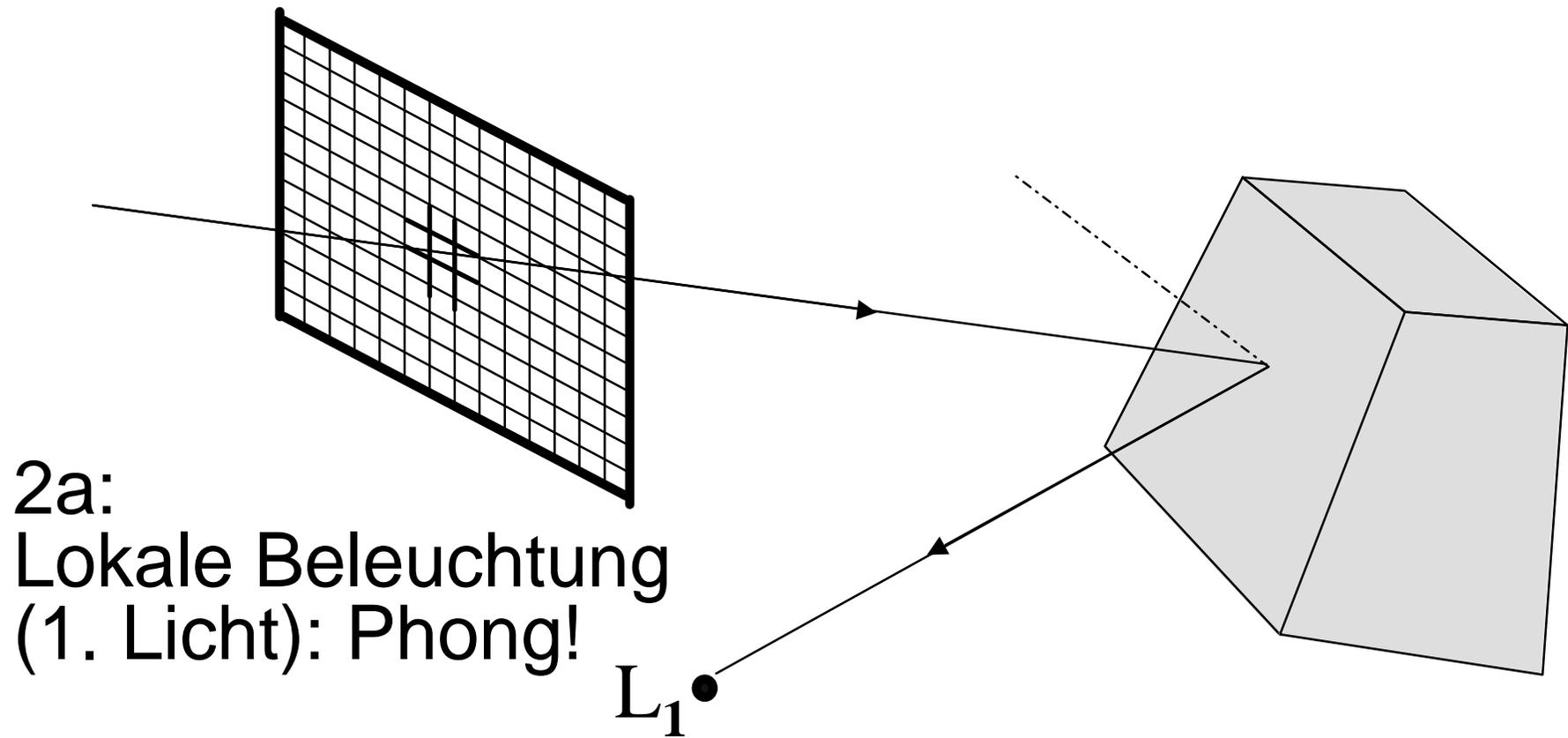
```
FOR every pixel p DO
1. trace primary ray
   find closest intersection s
2. FOR every light source l DO
   trace shadow feeler l -> s
   IF no intersection THEN
     illumination += local influence of l
3. IF surface of s is reflective THEN
   trace secondary ray
   illumination += influence of reflection
   IF surface of s is transparent THEN
   trace secondary ray
   illumination += influence of refraction
```

# Ray Tracing – Beispiel (1)

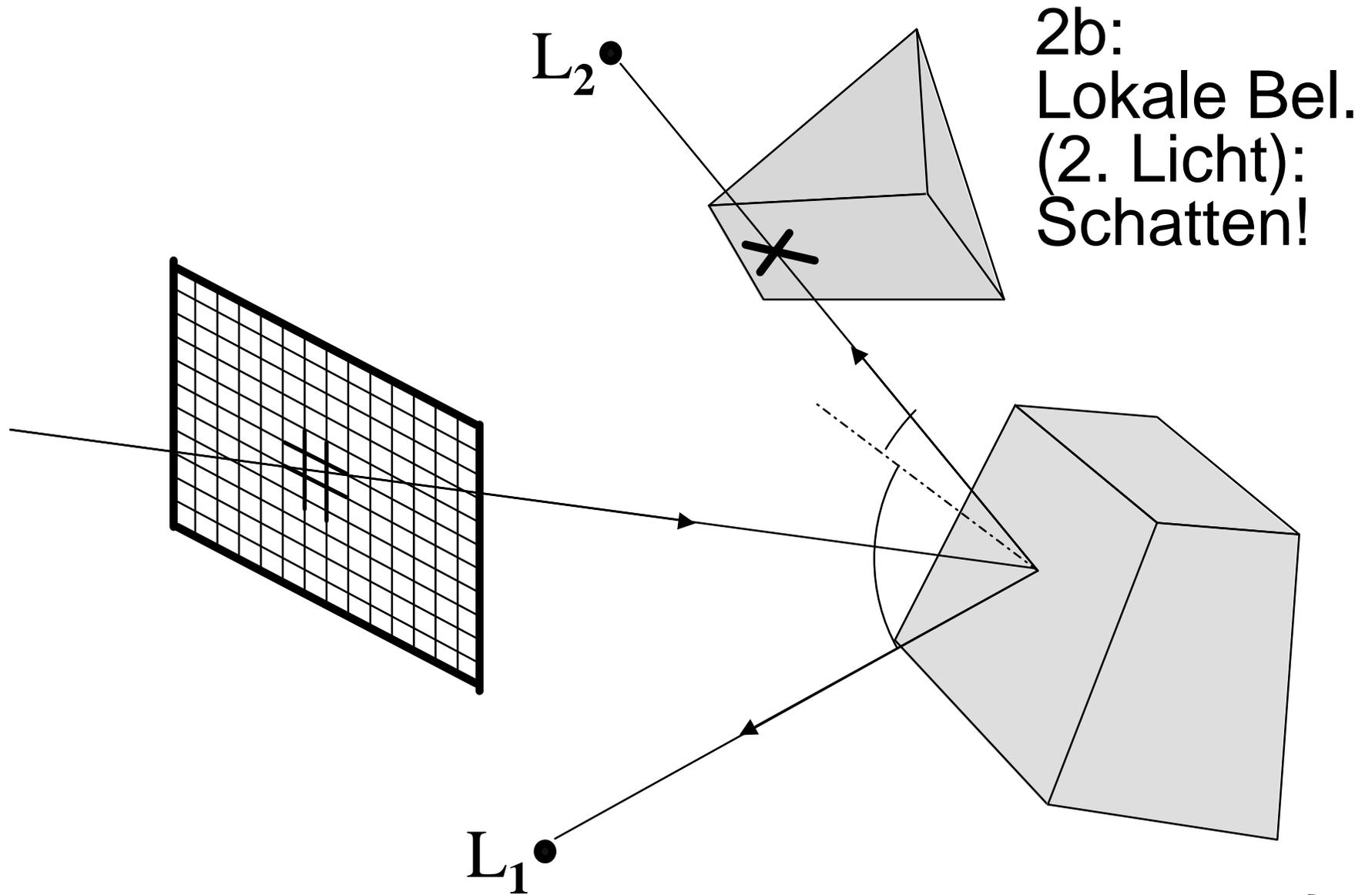
1.  
Sichtbarkeitsberechnung:  
welcher Objektpunkt  
wird betrachtet?



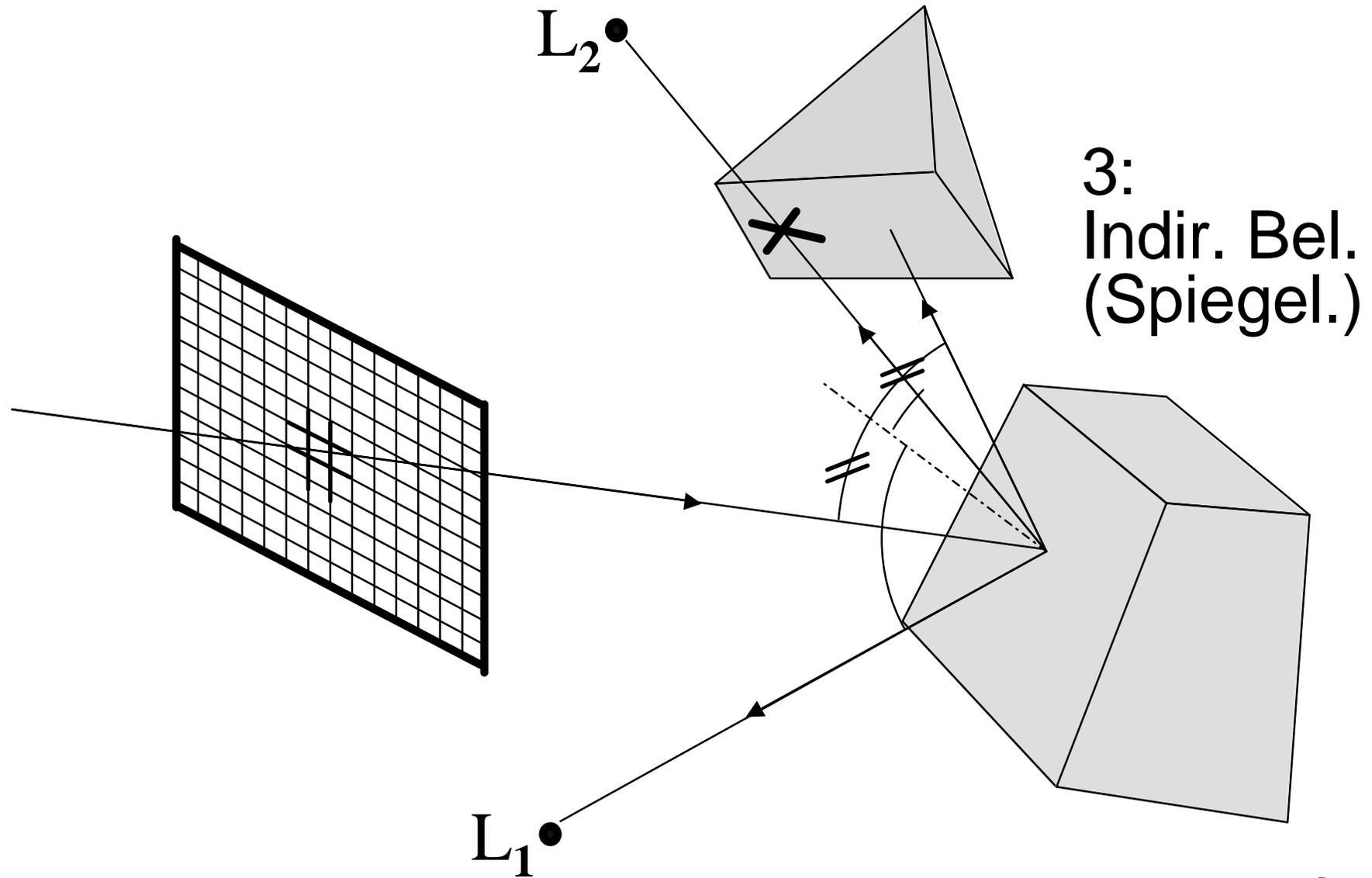
# Ray Tracing – Beispiel (2)



# Ray Tracing – Beispiel (3)

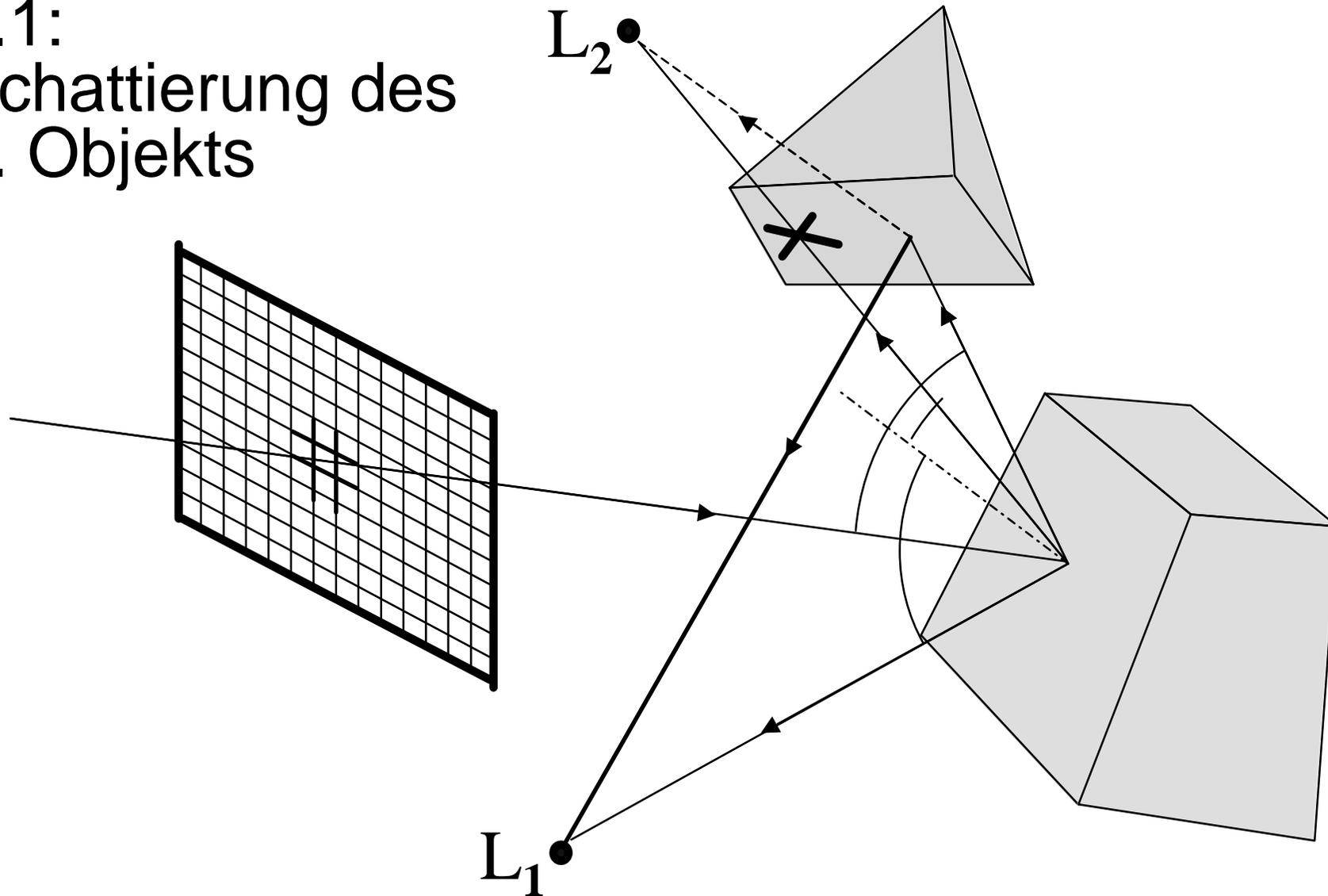


# Ray Tracing – Beispiel (4)



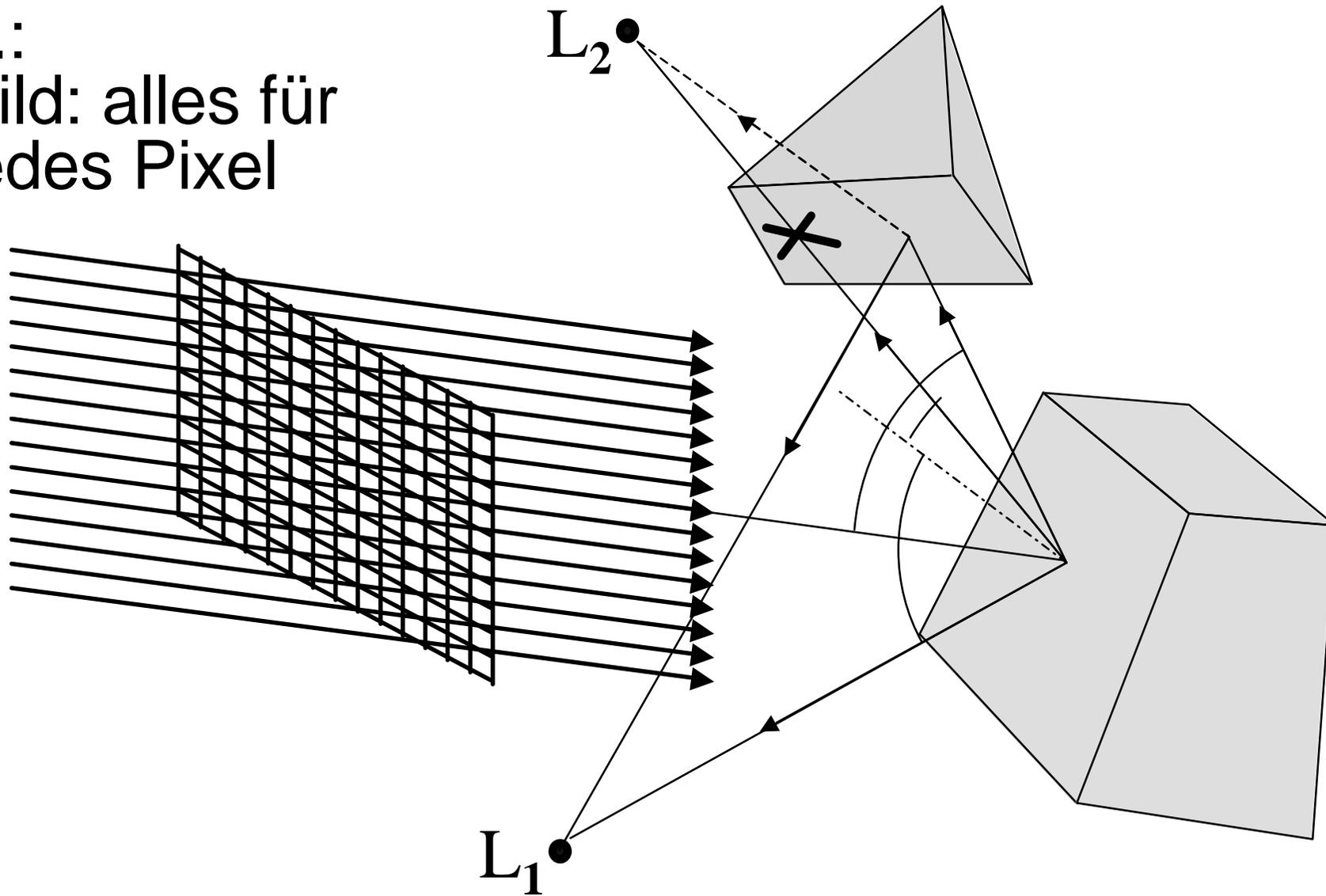
# Ray Tracing – Beispiel (5)

3.1:  
Schattierung des  
2. Objekts

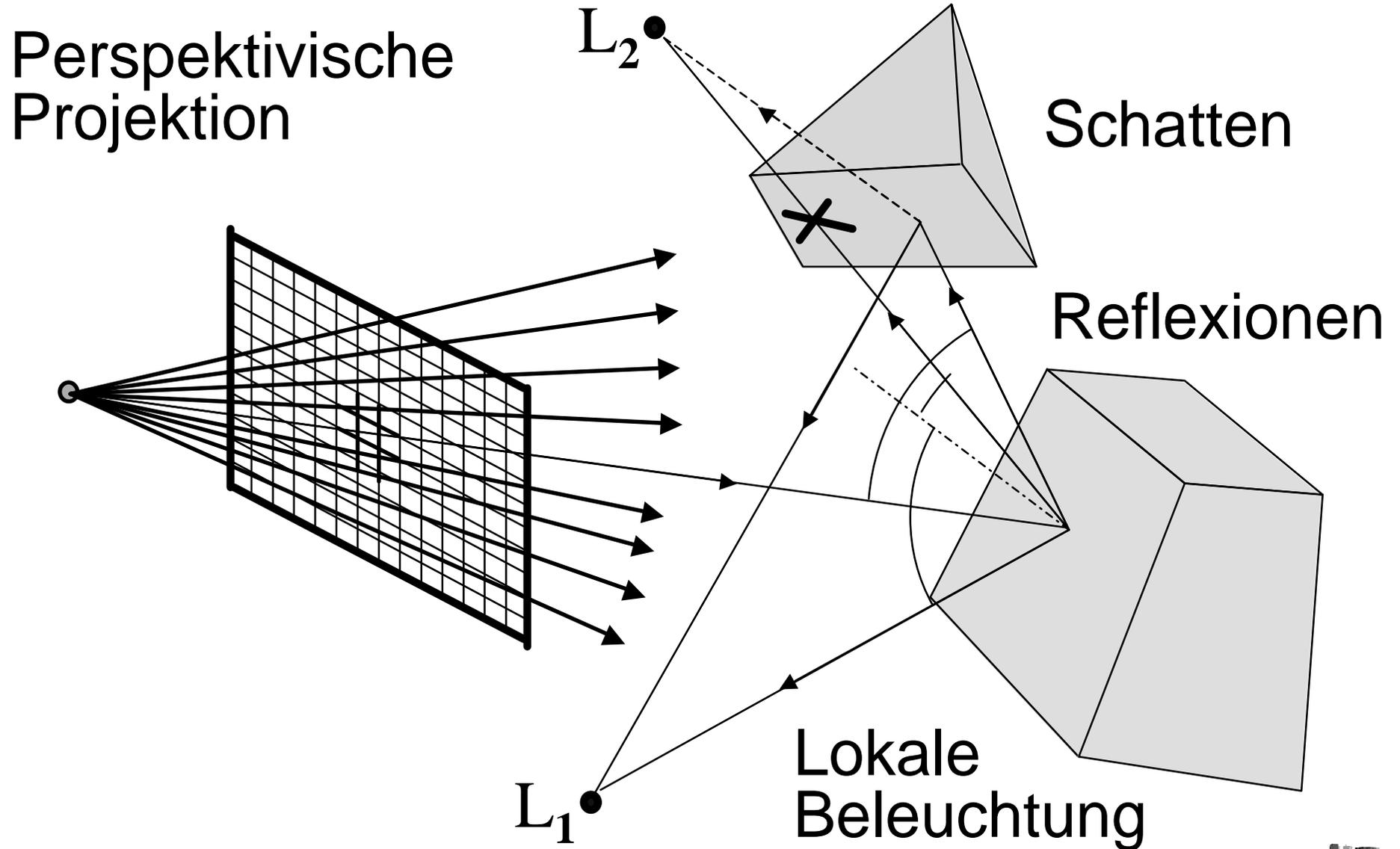


# Ray Tracing – Beispiel (6)

0.:  
Bild: alles für  
jedes Pixel



# Ray Tracing – Beispiel (7)



# Ray Tracing – Schnittberechnungen

Kugel,  
Polygon,  
Box

# Anforderungen an Objekte

## Schnittberechnung mit Strahl

- ◆ Lösung in geschlossener Form
- ◆ Iterativer, numerischer Ansatz

## Normalenberechnung im Schnittpunkt

### Beispiele:

- ◆ Implizit definierte Objekte –  $f(x)=0$  – wie Kugel, Ebene, etc.: oft relativ einfach
- ◆ Sonst: oft Ausnutzen von Hilfsstrukturen: bounding boxes, regular grid, BSP, etc.

# Strahl-Kugel Schnitt (1)

Strahl (parametrisch):

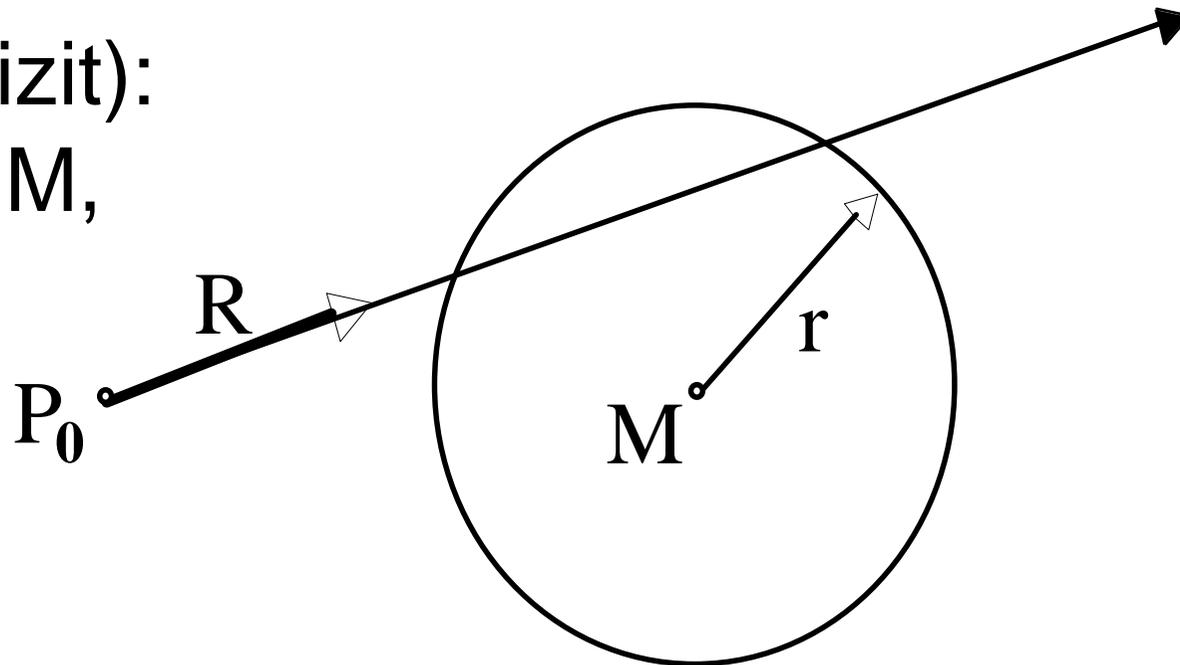
Augpunkt  $P_0$ , Richtungsvektor  $R$ , i.e.

$$P(t) = P_0 + t \cdot R \quad (|R| = 1)$$

Kugel (implizit):

Mittelpunkt  $M$ ,

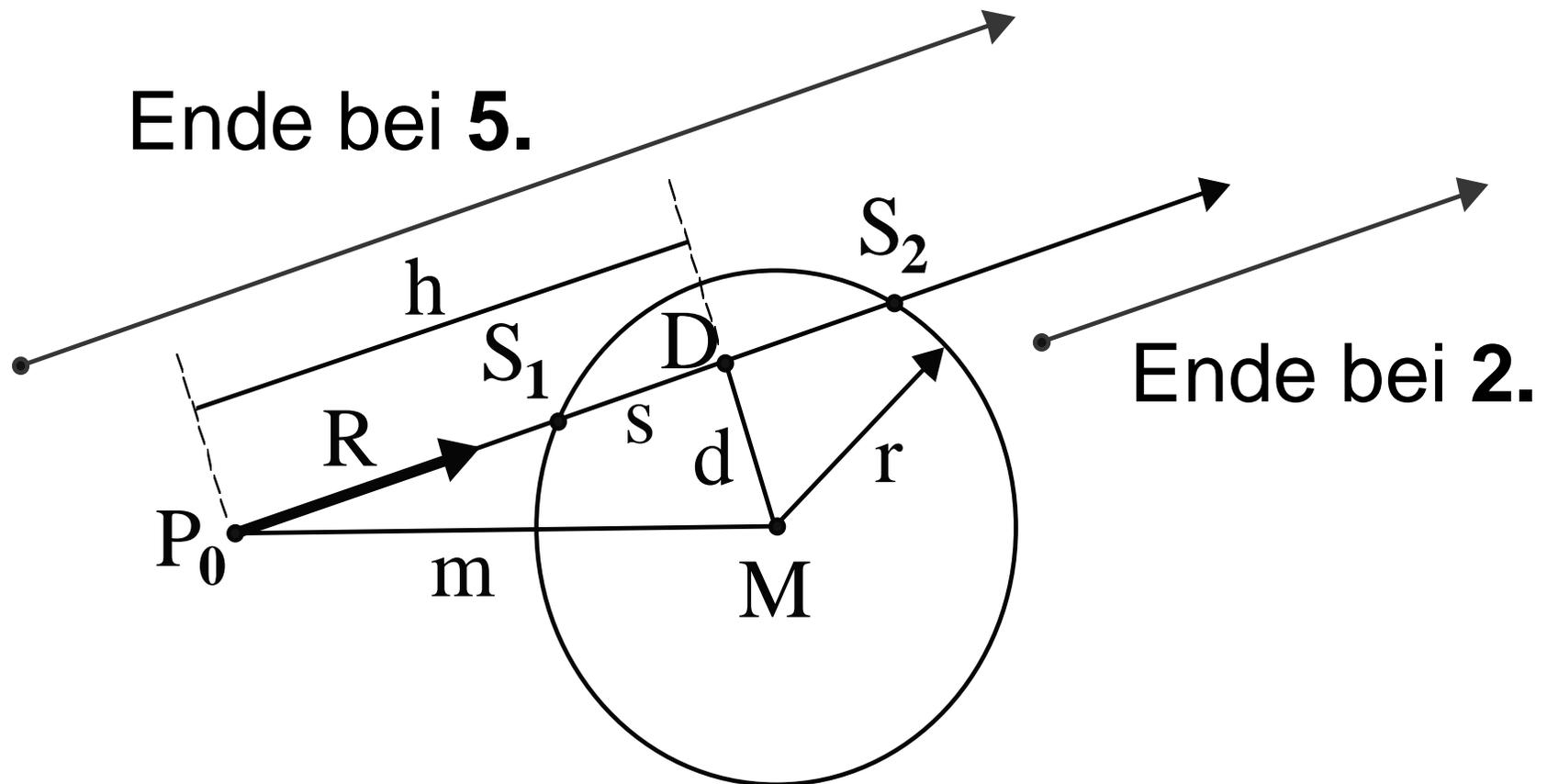
Radius  $r$



# Strahl-Kugel Schnitt (2)

1. Abstand<sup>2</sup>  $(M-P_0)^2 \equiv m^2 \rightarrow P_0$  außerhalb?
2.  $P_0$  außerhalb +  $R$  zeigt weg von Kugel  
 $R \cdot (M-P_0) \equiv h < 0 \Rightarrow$  fertig (kein Schnitt)
3. Def.  $D$  ( $\equiv$  Strahlpunkt am nächsten zu  $M$ )
4. Abstand<sup>2</sup>  $(M-D)^2 \equiv d^2 \rightarrow$  Strahl schneidet?
5.  $d^2 > r^2 \Rightarrow$  fertig (Strahl geht vorbei)
6. Ber. t-Werte von Schnittpunkten ( $\equiv t_1, [ t_2 ]$ )
7. Ber. Schnittpunkte  $S_1, [ S_2 ]$
8. Ber. Kugelnormale  $N_1, [ N_2 ]$ , fertig (Schnitt)

# Strahl-Kugel Schnitt (3)



# Strahl-Kugel Schnitt, Beispiel (1)

Bsp.:  $P_0 = (1 \ -2 \ -1)^T$ ,  $R \parallel (1 \ 2 \ 4)^T$ ,  $M = (3 \ 0 \ 5)^T$ ,  $r = 3$

---

1.  $m^2 = (M-P_0)^2 = (2 \ 2 \ 6)^T \cdot (2 \ 2 \ 6)^T = 44$   
 $44 > r^2 = 9 \Rightarrow P_0$  außerhalb
2.  $R \cdot (M-P_0) = (1 \ 2 \ 4)^T / \sqrt{21} \cdot (2 \ 2 \ 6)^T = 6.546$   
 $6.546 > 0 \Rightarrow R$  zeigt in Richtung Kugel  
!!  $R \cdot (M-P_0) = h = m \cdot \cos(\angle MP_0D) = |D-P|$  !!
4.  $d^2 = m^2 - h^2 = 44 - 42.850 = 1.150$
5.  $1.150 < r^2 = 9 \Rightarrow$  Strahl schneidet wirklich

# Strahl-Kugel Schnitt, Beispiel (2)

Bsp.:  $P_0 = (1 \ -2 \ -1)^T$ ,  $R \parallel (1 \ 2 \ 4)^T$ ,  $M = (3 \ 0 \ 5)^T$ ,  $r = 3$

---

6.  $s^2 = r^2 - d^2 = 9 - 1.150 = 7.850 \rightarrow s = 2.802$

$$t_1 = h - s = 6.546 - 2.802 = 3.744,$$

$$t_2 = \dots$$

7.  $S_1 = P_0 + t_1 \cdot R = (1.816 \ -0.368 \ 2.269)^T,$

$$S_2 = \dots$$

8.  $N_1 = (S_1 - M) / r = (-0.395 \ -0.123 \ -0.910)^T,$

$$N_2 = \dots$$

# Strahl-Polygon Schnitt (1)

$$\text{Strahl: } P = P_0 + t \cdot R \quad |R| = 1$$

Polygon-Ebene:

$$A \cdot x + B \cdot y + C \cdot z + D = 0 \quad (A^2 + B^2 + C^2) = 1$$

Normalvektor auf Polygon:  $N = (A \ B \ C)$

Vektorschreibweise:  $N \cdot P = -D$

$$N \cdot (P_0 + t \cdot R) = -D \Rightarrow \boxed{t = - (D + N \cdot P_0) / N \cdot R}$$

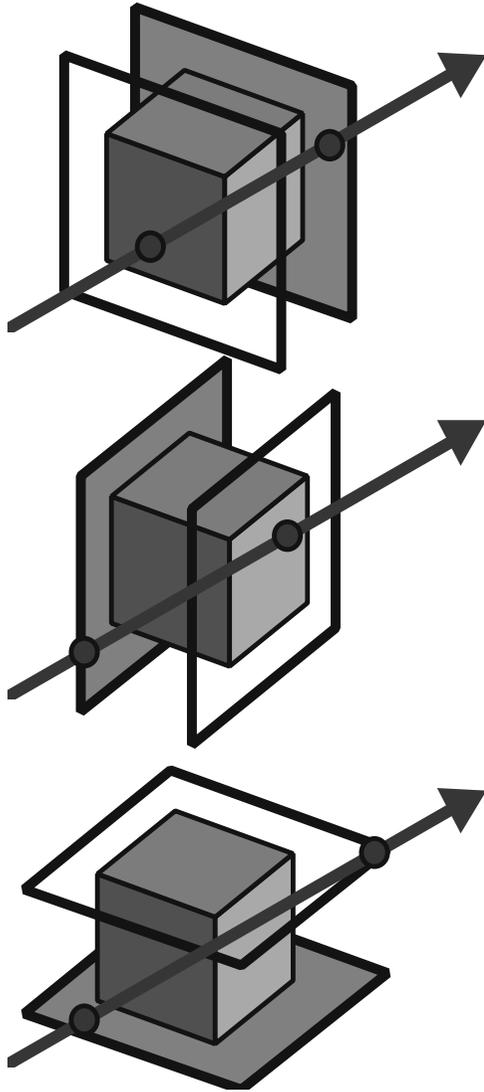
# Strahl-Polygon Schnitt (2)

$N \cdot R = 0 \Rightarrow$  Strahl ist parallel zu Polygon

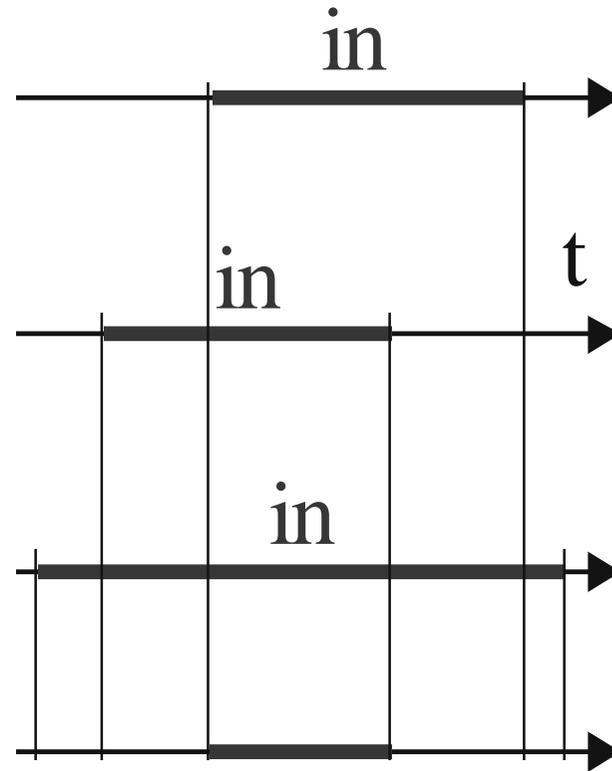
$t < 0 \Rightarrow$  Schnittpunkt hinter Augpunkt

Schnittpunkt:  $t$  in Strahlgleichung einsetzen  
inside/outside-Test für Polygon (z.B.) mit  
even/odd-Regel

# Strahl-Box Schnitt



t-Werte:



Für alle **konvexen** Vielflächer gilt:  
Schnitte nur mit Ebenen, kein Polygon  
(inside/outside)  
Test notwendig!

Durchschnitt: box

# Ray Tracing – Beleuchtung

**Beleuchtung = lokale Beleuchtung  
+ Reflexionsanteil  
+ Brechungsanteil**

# Beleuchtungsmodell

**Am Schnittpunkt:**

$$I = k_{\text{local}} \cdot I_{\text{local}} + k_{\text{refl}} \cdot I_{\text{refl}} + k_{\text{trans}} \cdot I_{\text{trans}}$$

$$k_{\text{local}} + k_{\text{refl}} + k_{\text{trans}} = 1$$

**Lokal (Phong):**

$$I_{\text{local}} = k_a I_a + k_d I_L \cos \theta + k_s I_L \cos^{n_s} \phi$$

**Rekursive Definition:**

$$I(P) = k_{\text{local}} I_{\text{local}} + k_{\text{refl}} I(P_r) + k_{\text{trans}} I(P_t)$$

# Beleuchtung: lokale Beleuchtung

Schattenfühler in Richtung L

Umgebungslicht

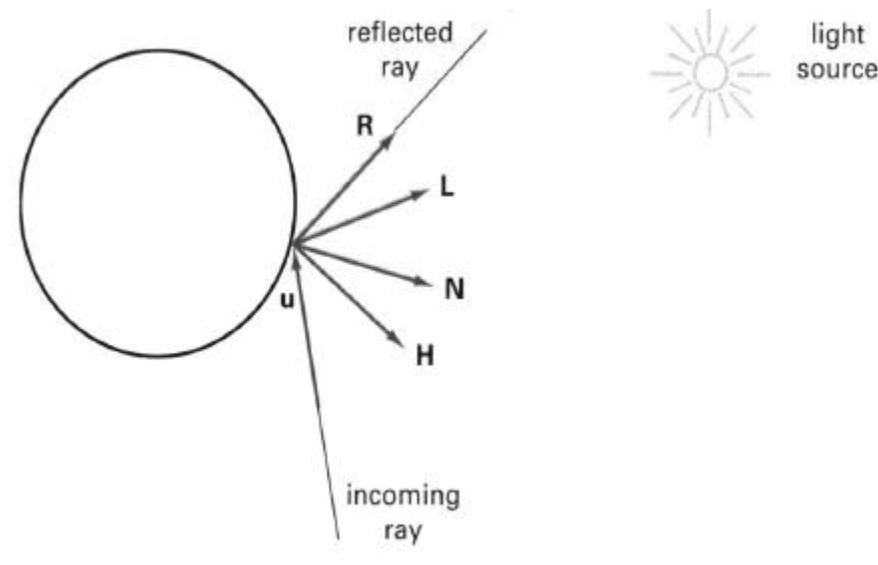
$$k_a I_a$$

Diffuse Reflexion

$$k_d (N \cdot L)$$

Spiegelnde Reflexion, z.B.

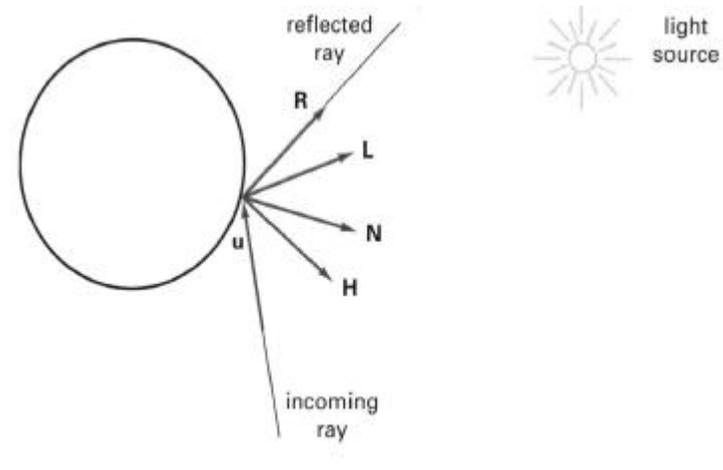
$$k_s (H \cdot N)^{n_s}$$



# Ray Tracing: Reflexion, Brechung

## Reflexionsstrahl

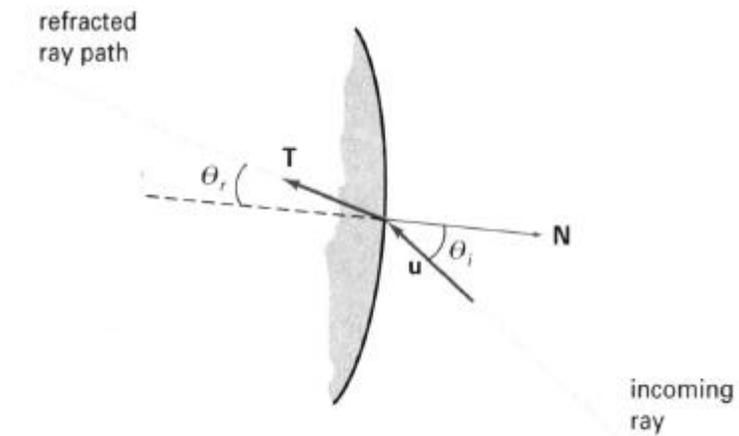
$$R = u - (2u \cdot N)N$$



## Brechungsstrahl

### ◆ Brechungsgesetz

$$\sin \theta_r = \frac{\eta_i}{\eta_r} \sin \theta_i$$



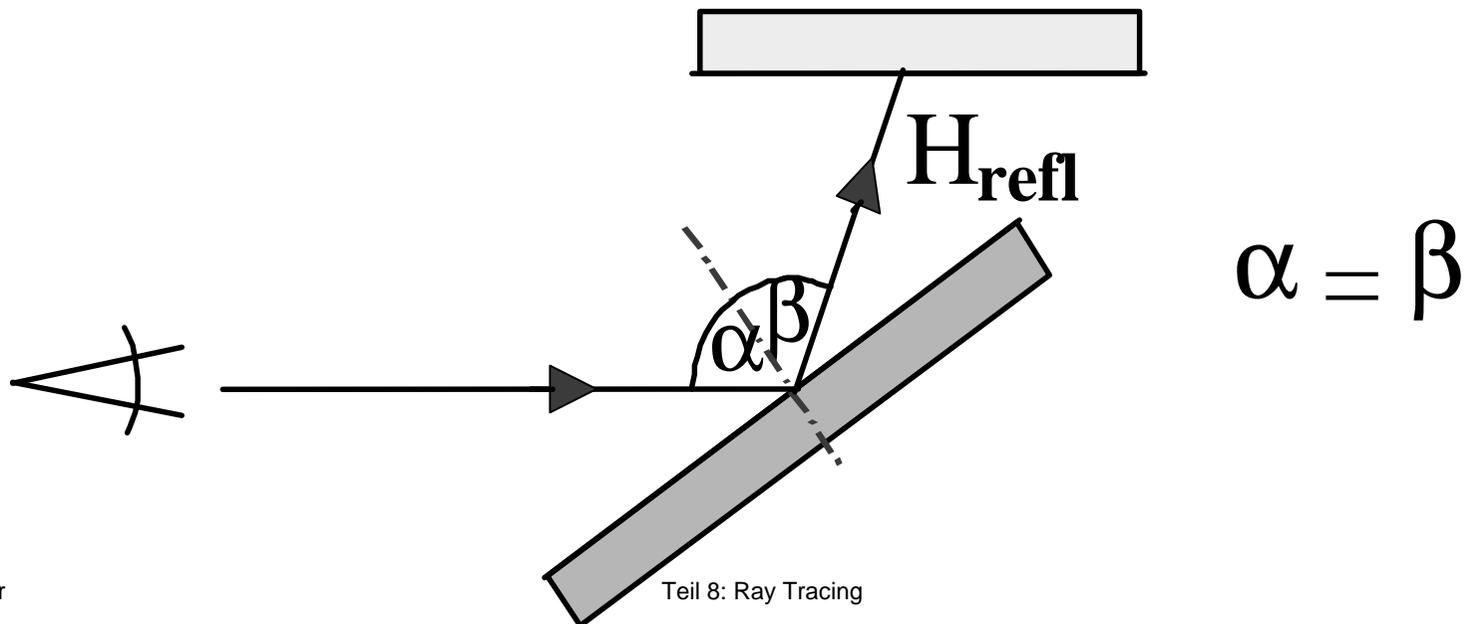
# Beleuchtung: Reflexion

$$I_{\text{refl}} = k_{\text{refl}} \cdot H_{\text{refl}}$$

$I_{\text{refl}}$  ... Reflexionsanteil

$k_{\text{refl}}$  ... Reflexionskoeffizient des Materials

$H_{\text{refl}}$  ... Einfallender Teil in Reflexionsrichtung



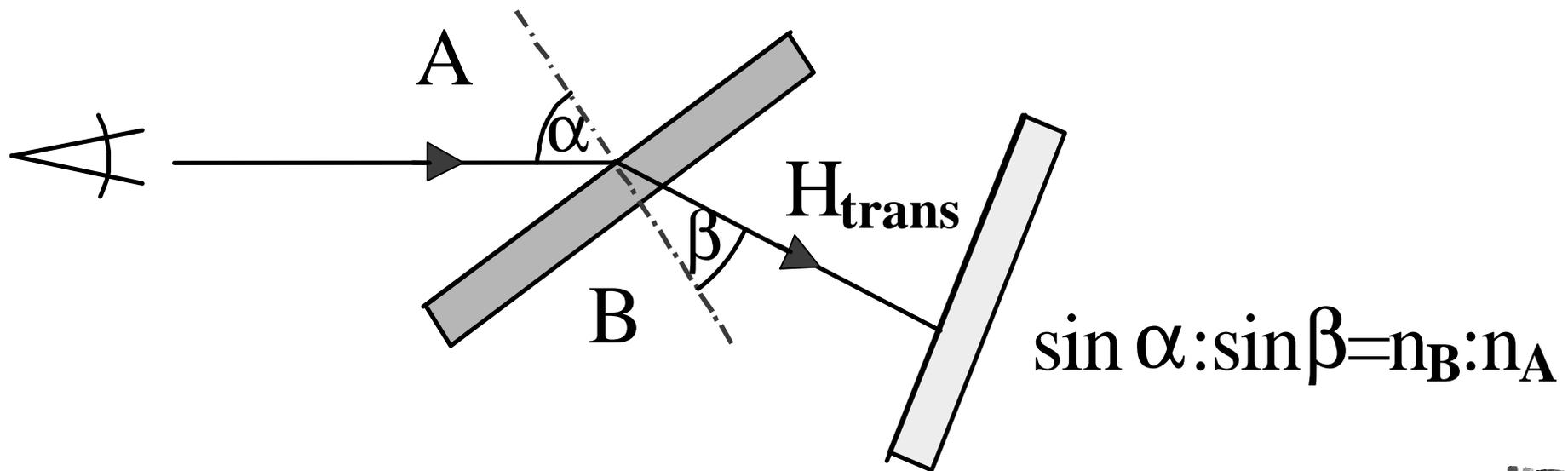
# Beleuchtung: Transparenz

$$I_{\text{trans}} = k_{\text{trans}} \cdot H_{\text{trans}}$$

$I_{\text{trans}}$  ... Transparenzanteil

$k_{\text{trans}}$  ... Transparenzkoeffizient des Materials

$H_{\text{trans}}$  ... Einfallender Teil in Brechungsrichtung



# Ray Tracing – Beispiele

## 3 Beispiele

# Ray Tracing Beispiel 1

Szene:

37 Kugeln

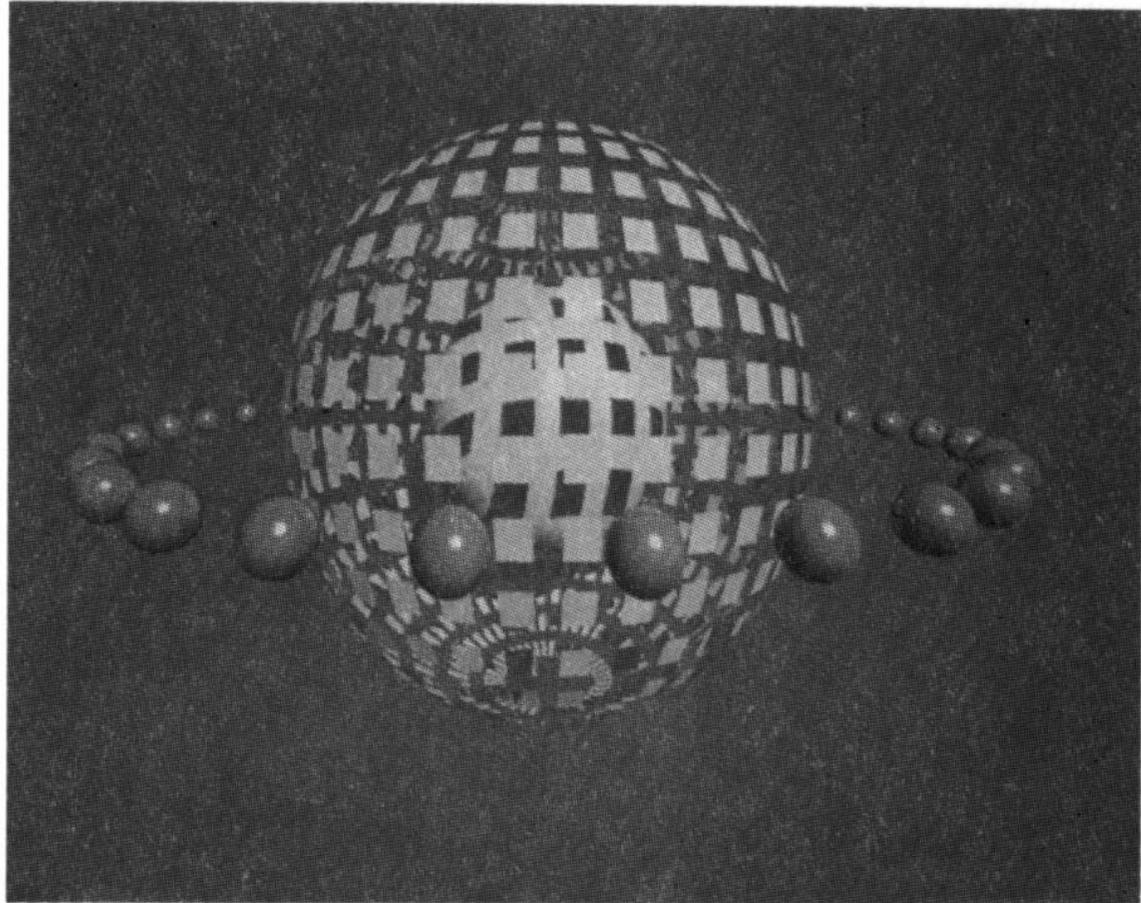
720 Polyg.

Algorithmus:

9 Strahlen

per Pixel

Tiefe 5



# Ray Tracing Beispiel 2

Szene:

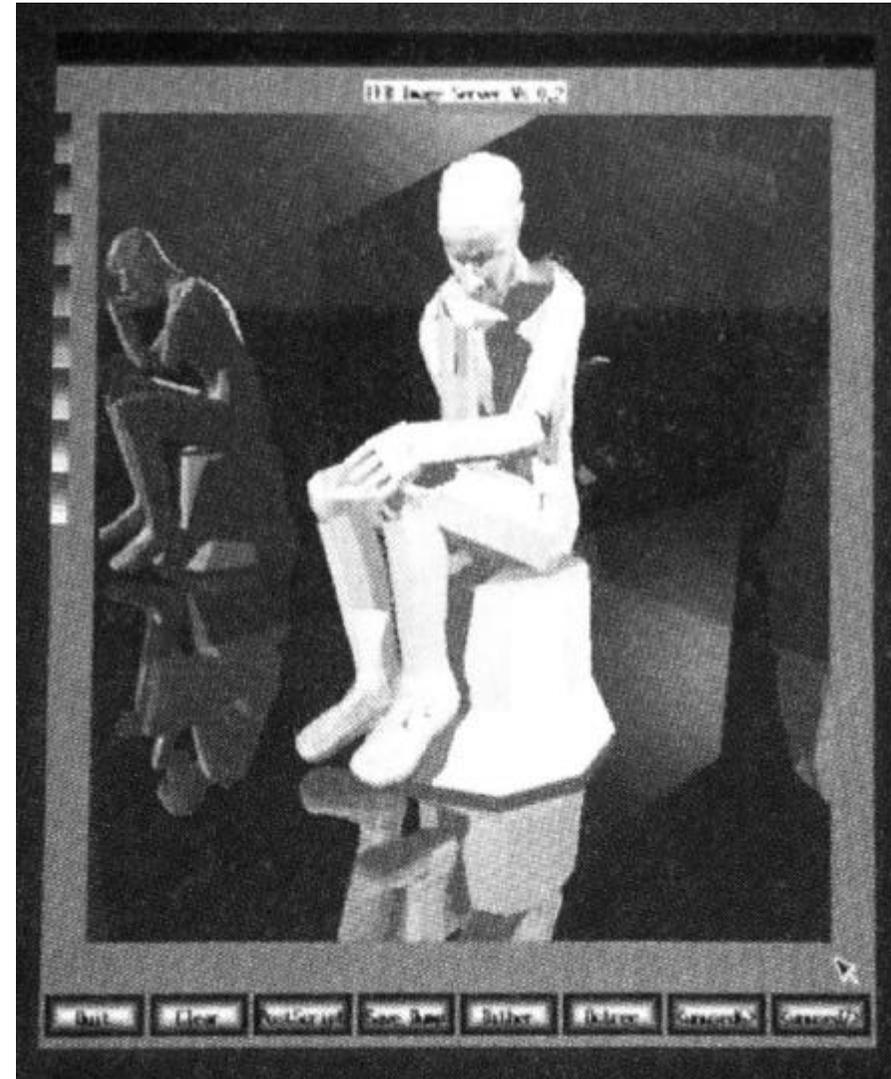
3036 Primitive

2 Lichtquellen

Algorithmus:

1 Strahl per Pixel

1.6 Mio. Strahlen



# Ray Tracing Beispiel 3

Szene:

1298 Polyg.

4 Kugeln

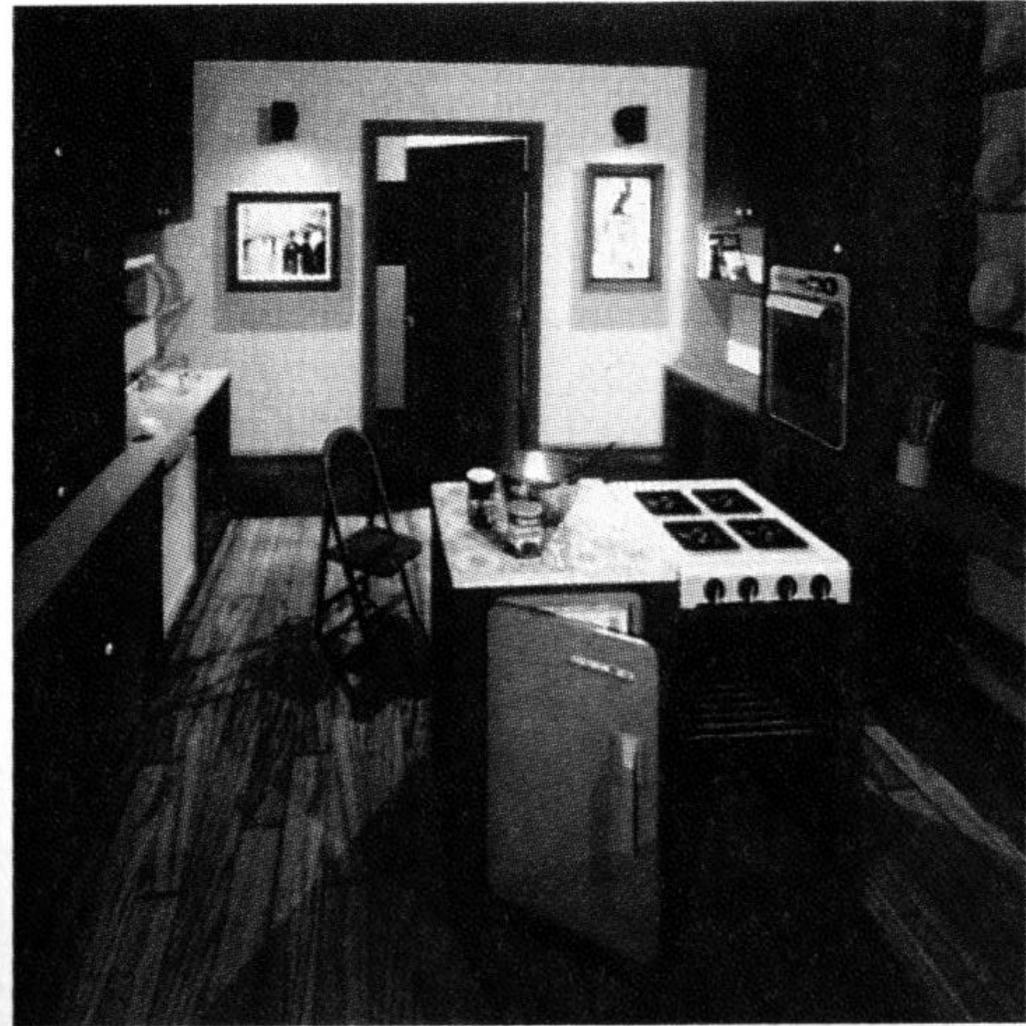
76 Zylinder

35 Quadriken

5 Lichtquellen

Algorithmus:

Light-Buffer



# Ray Tracing Beispiel 3 – Detail



# Mängel

Komplexität durch **Menge an Strahlen**  
(und dabei entstehende Schnitte)

Restriktion der „Globalität“ auf  
**Spiegelreflexion und Brechnung**  
(keine globale diffuse Beleuchtung)

Visuelle Mängel

- ◆ Anti-Aliasing
- ◆ Kein Halbschatten
- ◆ Tiefenschärfeneffekte

# Distributed Ray Tracing

**Anti-Aliasing, Halbschatten, diffuse  
Reflexion, etc.**

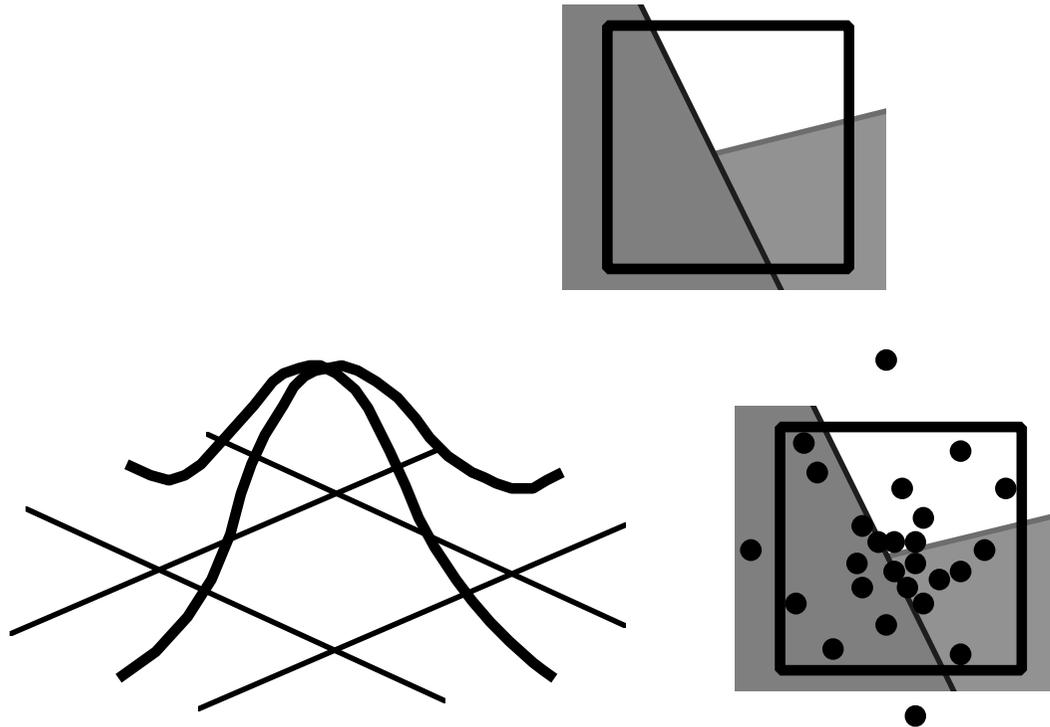
# Distributed Ray Tracing

## Berücksichtigung folgender Effekte:

- ◆ Aliasing
- ◆ Unschärfe Spiegelung
- ◆ Milchglaseffekte
- ◆ Halbschatten
- ◆ Tiefenunschärfe
- ◆ Bewegungsunschärfe

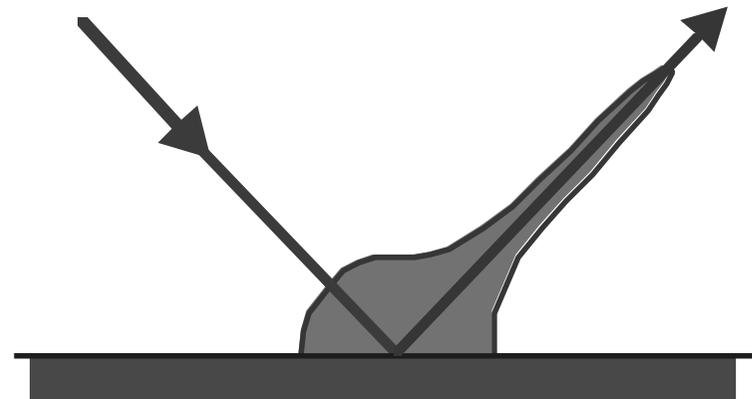
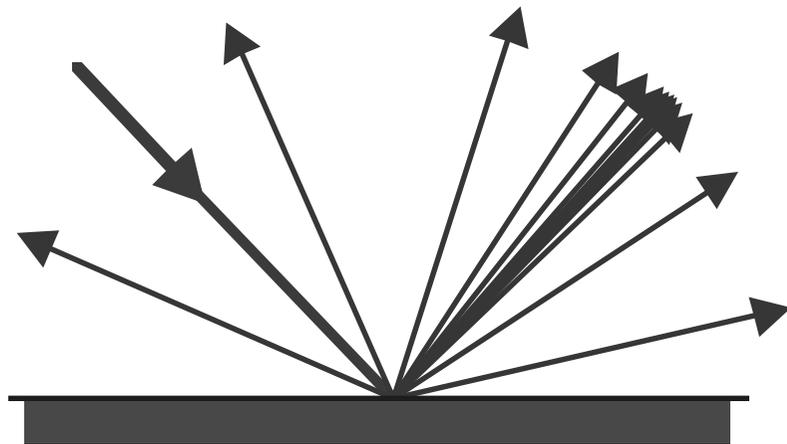
# Anti-Aliasing

## Stochastisches Sampling mit Gauss-Gewichtung

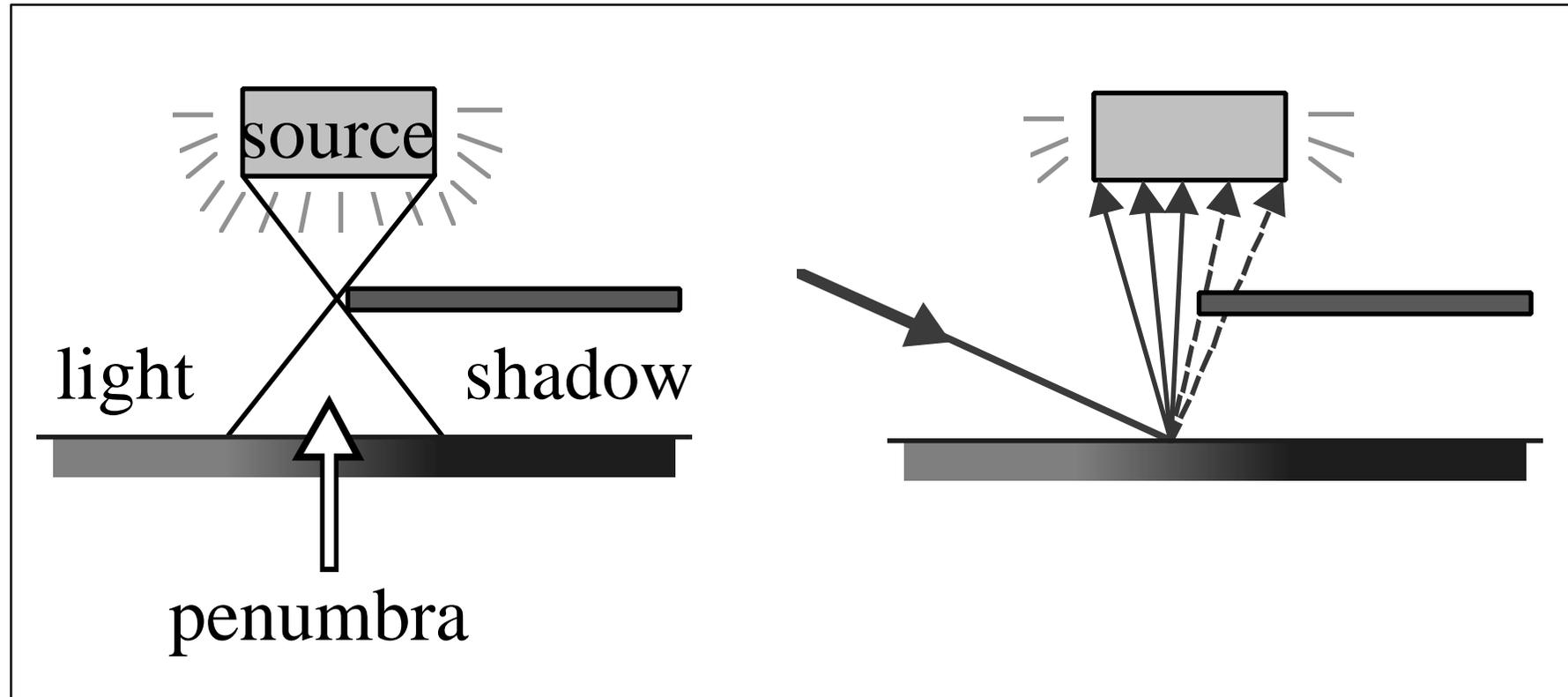


# Unscharfe Spiegelung

Simulation von gerichtet-diffusen Reflexionen durch Variation des Reflexionswinkels

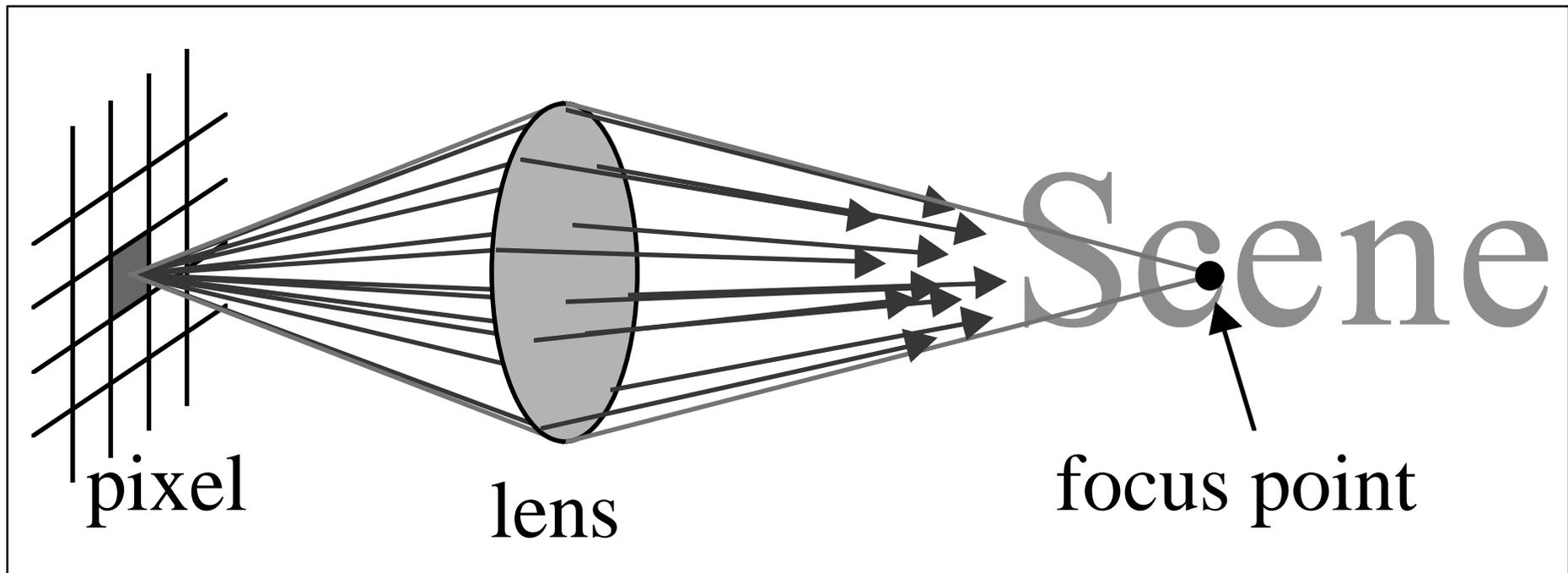


# Halbschatten



# Tiefenunschärfe

Abhängigkeit von der Blende  
berücksichtigen (Durchmesser)



# Algorithmus

(Distributed Ray Tracing)

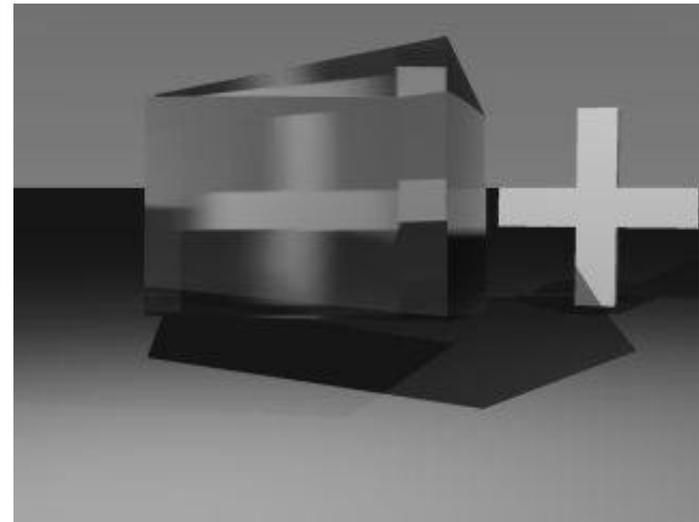
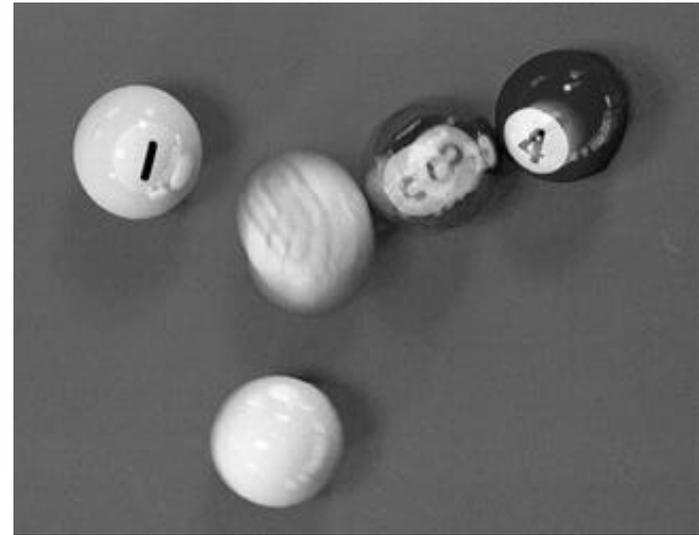
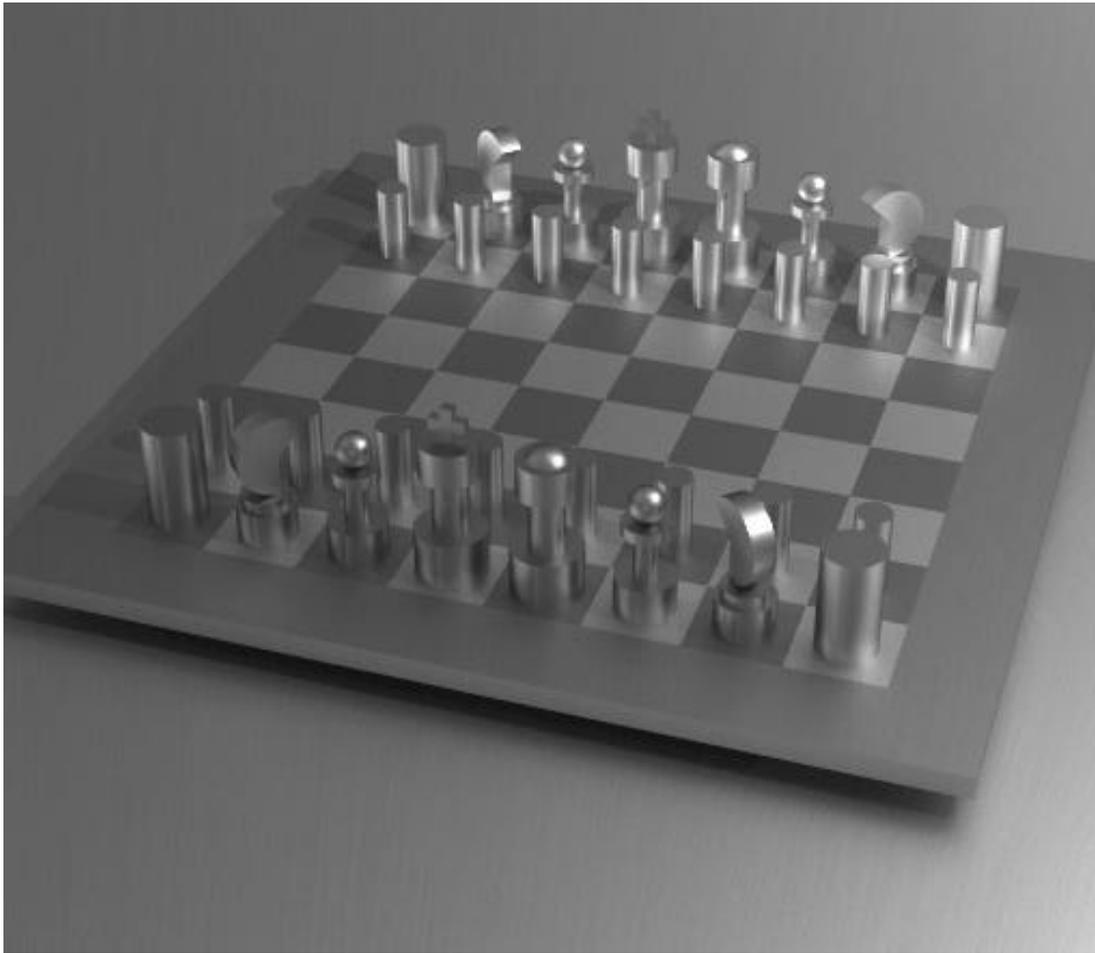
**Große Anzahl an Strahlen - aber:  
Monte-Carlo Integration möglich**

- ◆ Nur einen Strahl weiterverfolgen
- ◆ Integration eines Strahlenbündels

**Benötigte Strahlen (pro Pixel):**

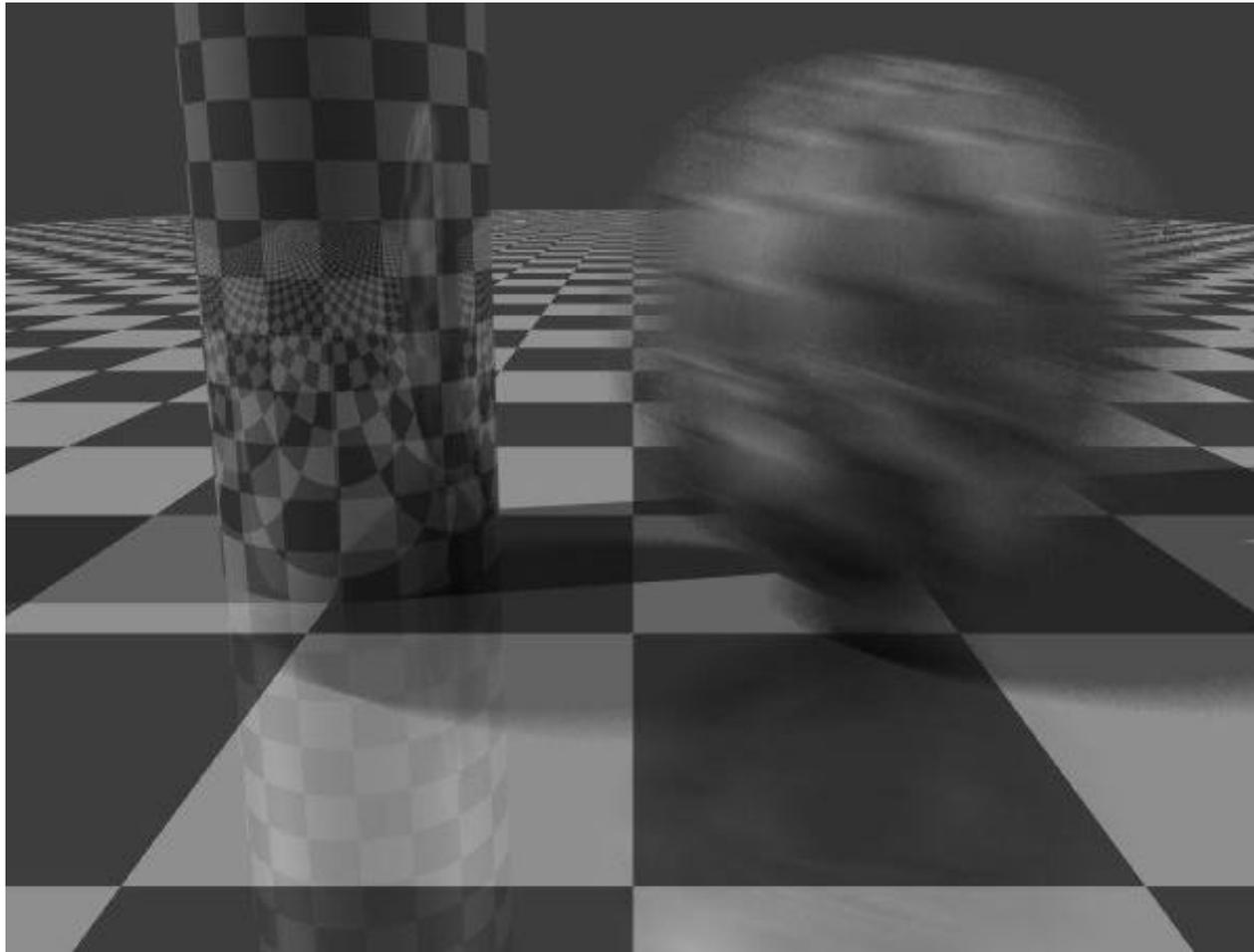
- ◆ Minimum: 5-10
- ◆ Gute Qualität: 10-20 (für einen Effekt)
- ◆ Gute Gesamtqualität: 20-60

# Distributed Ray Tracing – Bsp.



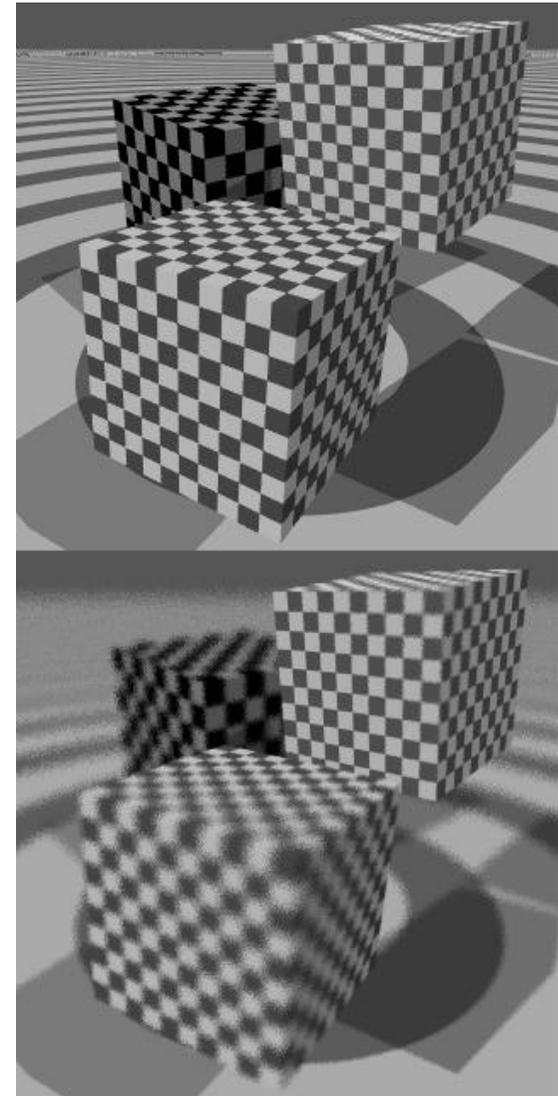
Halbschatten, Blur, Spektren

# Distributed Ray Tracing – Bsp.

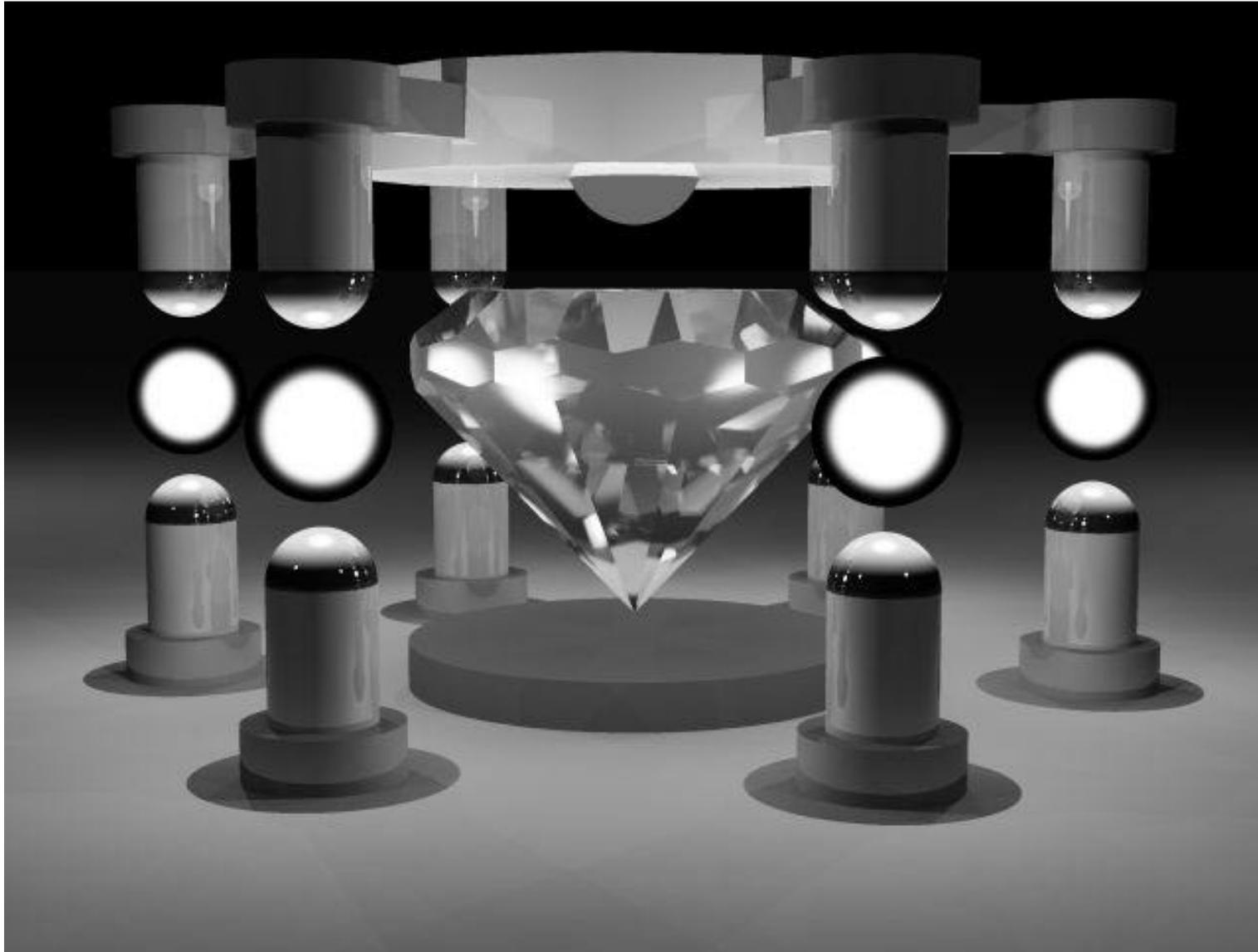


Blur

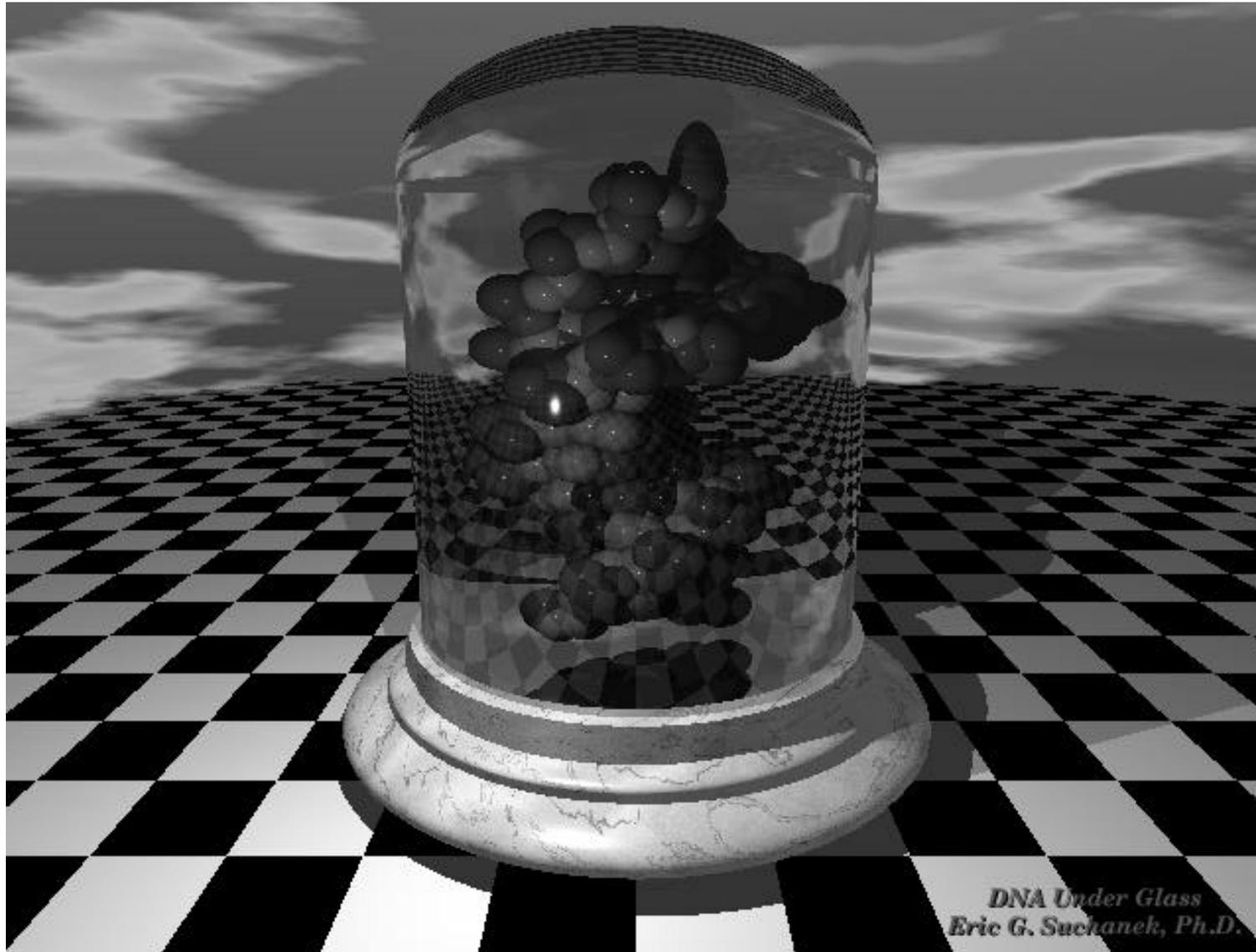
& Tiefenunschärfe



# Distributed Ray Tracing – Bsp.



# www.povray.org – Beispiele



# www.povray.org – Beispiele



# www.povray.org – Beispiele



# www.povray.org – Beispiele



# Ray Tracing – Weitere Beispiele



*Crystals.jpg (c)1995 Pascal.Massimino@ens.fr*