


# Computer-Graphik II

## Ray-Tracing



G. Zachmann  
Clausthal University, Germany  
[cg.in.tu-clausthal.de](http://cg.in.tu-clausthal.de)



## Die antike Erklärung des Sehens: Sehstrahlen



G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 2

## Effekte für eine realistische Darstellung

- Das lokale Beleuchtungsmodell versagt bei folgenden Effekten
- (Soft) Shadows (Halbschatten)
- Reflexion (Spiegel und Glanz)
- Transparenz (Wasser, Glas)
- Interreflexion ("*color bleeding*")
- ...


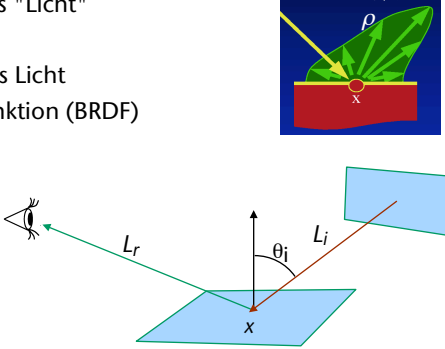
G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 3

## Globale Beleuchtungsrechnung

- Ziel: **Photorealistisches Rendering**
- Die "Lösung": die **Rendering-Gleichung** [Kajiya, Siggraph 1986]


$$L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega} \rho(x, \omega_r, \omega_i) L_i(x, \omega_i) \cos(\theta_i) d\omega_i$$

$L_i$  = aus Richtung  $\omega_i$  inzidentes "Licht"  
 $L_e$  = emittiertes Licht  
 $L_r$  = in Richtung  $\omega_r$  reflektiertes Licht  
 $\rho$  = Reflexionskoeffizientenfunktion (BRDF)  
 $\Omega$  = Halbkugel um Normale

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 4

- Analytische Lösung ist unmöglich!
- Die Rendering Gleichung kann als rekursive Funktion aufgefaßt werden
- Daraus folgen praktische **Approximations-Verfahren**, die auf der Verfolgung des Lichts entlang Strahlen beruhen
  - **Ray tracing** [Whitted, Siggraph 1980, "An Improved Illumination Model for Shaded Display"]
  - **Radiosity** [Goral et. al, Siggraph 1984, "Modeling the Interaction of Light between diffuse Surface"]
  - **Monte Carlo Verfahren**




Turner Whitted,  
Microsoft Research

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 5

## Rekursives Ray-Tracing

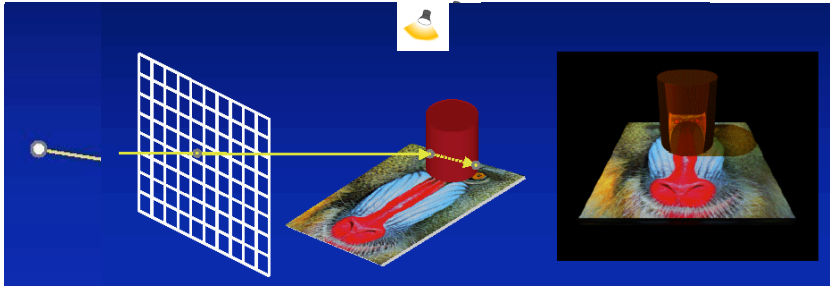
- Algorithmus zur Approximation der Rendering-Gleichung
- Modelliert werden nur:
  - Reflektion
  - Beugung
  - Verdeckungsrechnung
  - Schatten
- Strahlen werden nur in Richtung des **reflektierten** bzw. **gebrochenen** Strahls verfolgt
- Annahmen:
  - Punktlichtquellen
  - Phong-Modell
  - keine Halbschatten



G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 6

## Funktionsweise

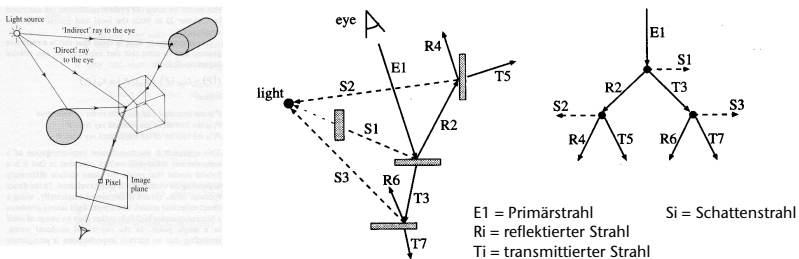
1. Synthetische Kamera = Augpunkt + Bildebene in Weltkoordinaten
2. Schieße Strahlen vom Augpunkt aus durch die Pixel in die Szene
3. Falls der Strahl mehr als ein Objekt schneidet, betrachte nur den ersten Schnittpunkt
4. Schieße weitere Strahlen vom dort zu allen Lichtquellen (Schattenstrahlen; "shadow feelers")
5. Treffen diese Schattenstrahlen auf ein Objekt, so liegt der betrachtete Flächenpunkt im Schatten. Ansonsten wird das Phong-Beleuchtungsmodell ausgewertet
6. Ist das sichtbare Objekt spiegelnd, dann schieße weiteren reflektierten Strahl in die Szene
7. Ist das Objekt transparent, so wird zusätzlich ein gebrochener Strahl weiterverfolgt



G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 7

## Der Strahlbaum

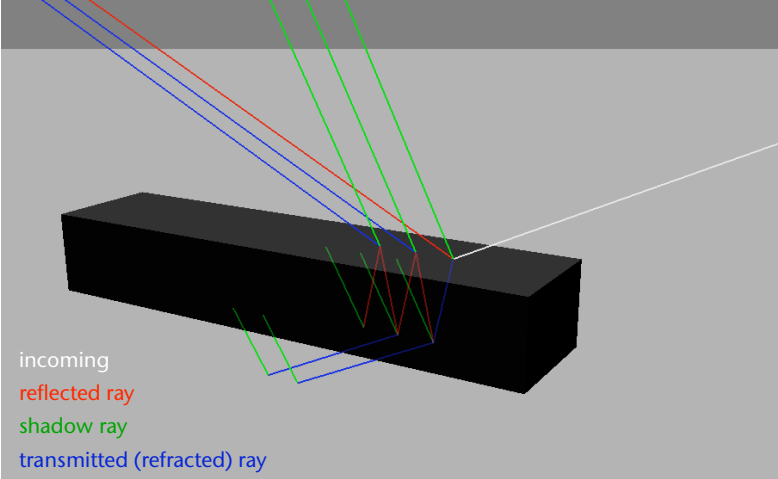
- Grundidee des Raytracing: Strahlengänge von den Lichtquellen bis zum Auge konstruieren, aber dabei beim Auge starten und diese Strahlengänge rückwärts "suchen"
- Ergibt (konzeptionell!) einen Strahlenbaum:



E1 = Primärstrahl  
 Ri = reflektierter Strahl  
 Ti = transmittierter Strahl  
 Si = Schattenstrahl

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 8

Visualisierung eines Strahlbaumes (eignet sich hervorragend zum Debugging)



incoming  
reflected ray  
shadow ray  
transmitted (refracted) ray

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 9

## Das Beleuchtungsmodell

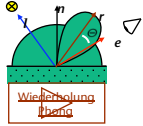
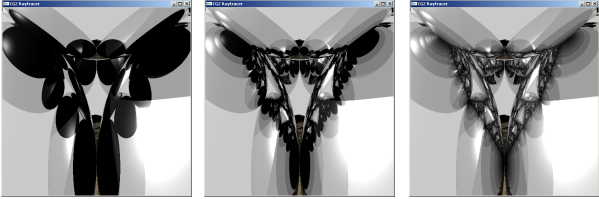

Beleuchtung auf der Fläche

$$L_{\text{ges}} = L_{\text{Phong}} + r_s L_s + r_t L_t$$

$r_s$  = Reflexionskoeffizient für das reflektierte Licht  $L_s$   
 $r_t$  = Transmissionskoeffizient für das transmittierte Licht  $L_t$

Abbruch der Rekursion:

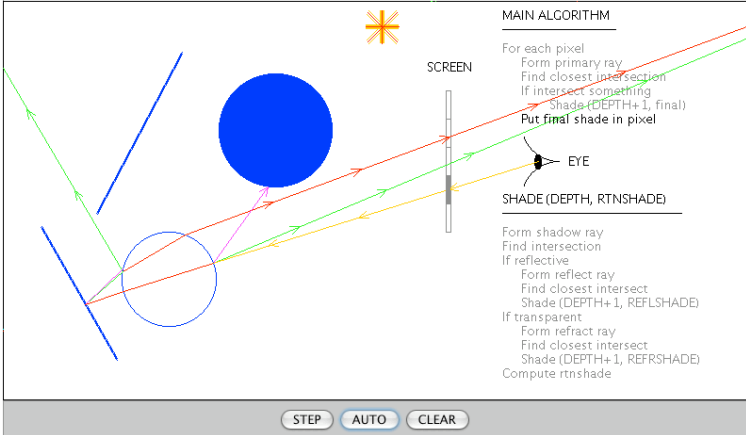
- Falls maximale Rekursionstiefe erreicht; oder/und,
- Falls Beitrag zur Beleuchtung zu klein (schrumpft wie  $r_t^n$ )

Rek. Tiefe: 3      Rek. Tiefe: 5      Rek. Tiefe: 100

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 10

## Demo



**MAIN ALGORITHM**

```

For each pixel
  Form primary ray
  Find closest intersection
  If intersect something
    Shade (DEPTH+1, final)
  Put final shade in pixel

```

**SHADE (DEPTH, RTNSHADE)**

```

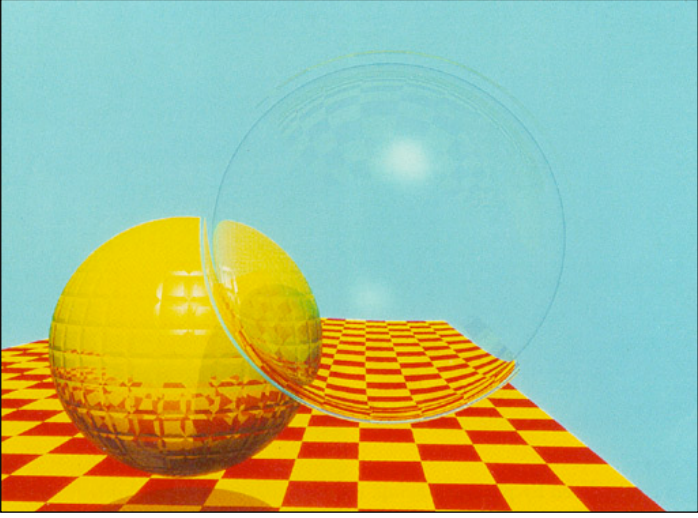
Form shadow ray
Find intersection
If reflective
  Form reflect ray
  Find closest intersect
  Shade (DEPTH+1, REFLSHADE)
If transparent
  Form refract ray
  Find closest intersect
  Shade (DEPTH+1, REFRSHADE)
Compute rtshade

```

[http://www.siggraph.org/education/materials/HyperGraph/raytrace/rt\\_java/raytrace.html](http://www.siggraph.org/education/materials/HyperGraph/raytrace/rt_java/raytrace.html)

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 11

## Eines der ersten Ray-Tracing-Bilder



Turner Whitted 1980

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 12

## Albrecht Dürers "Ray-Casting-Maschinen" [16. Jhrdt.]



The image shows two parts: an engraving of a ray-casting machine and a detailed diagram of its pulley system. The engraving depicts a man in a workshop using a device to project a shadow of a lute onto a screen. The diagram shows a wooden frame with a hinged shutter that can be moved to reveal a grid of points. A string is stretched across the frame, and a pulley system is used to keep it taut. A pointer is used to trace the points on the string.

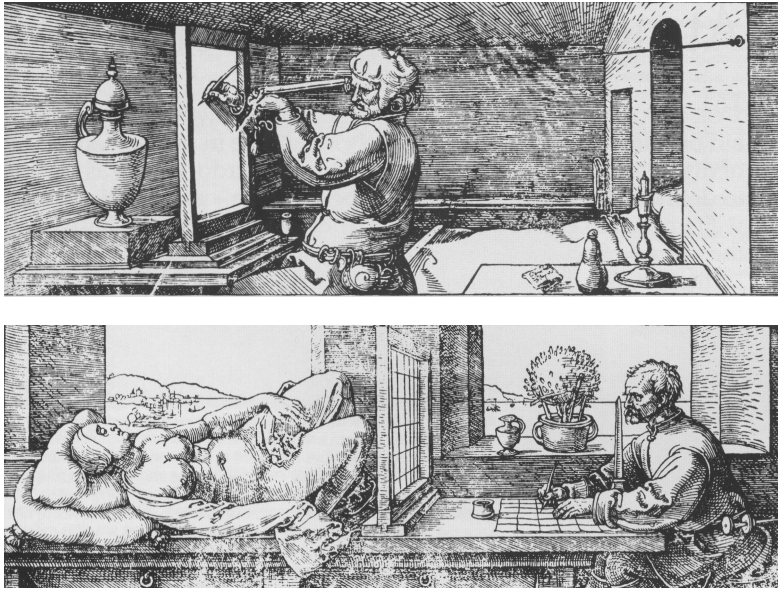
*The movable threads (probably made of silk) were stretched across the frame at right angles to each other*

**PULLEY SYSTEM**  
At the wall, the string was attached to a weight, which acted as a pulley (see engraving), keeping the string taut as it passed through the needle eye and frame to the pointer at its other end.

*Pulley weight*      *Pointer*

*The foreshortened lute, plotted point by point*      *Hinged shutter*      *Wooden frame*

G. Zachmann    Computer-Graphik 2 - SS 08 Ray-Tracing 13



The top engraving shows a man in a workshop using a ray-casting device to project the shadow of a vase onto a screen. The bottom engraving shows a man in a workshop using a ray-casting device to project a landscape onto a screen. The man is sitting at a table, and the screen is positioned to capture the projection of the landscape.

G. Zachmann    Computer-Graphik 2 - SS 08 Ray-Tracing 14

## Beispiele

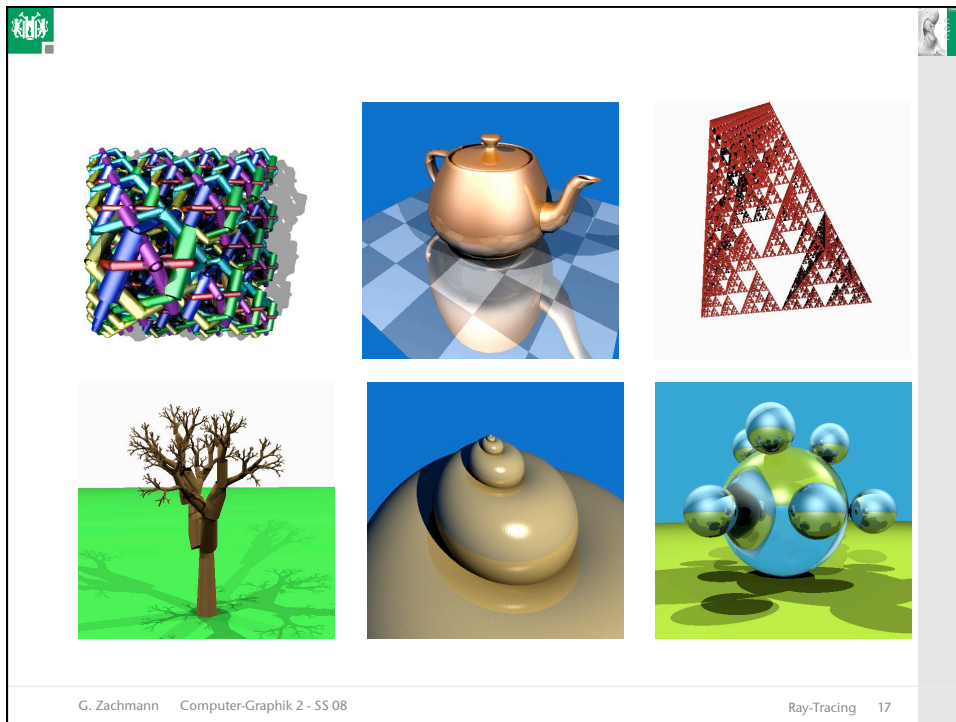
Jensen, Lightscape

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 15

Objekt "sphere flake" aus der "Standard Procedural Databases" (SPD) von Eric Haines  
[<http://www.acm.org/tog/resources/SPD/>].

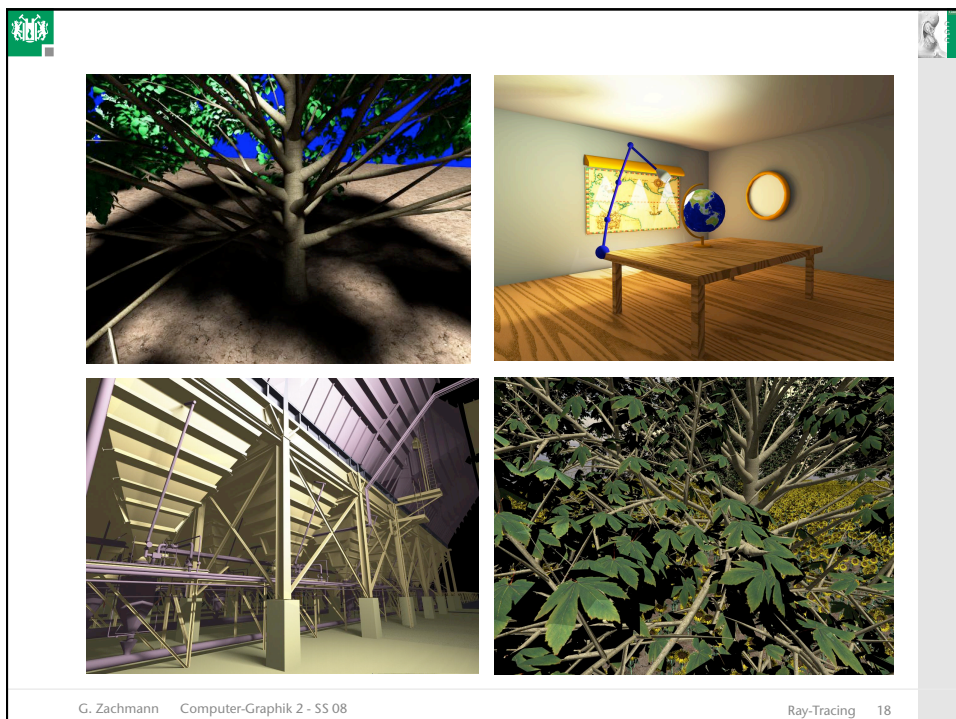
G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 16





G. Zachmann Computer-Graphik 2 - SS 08

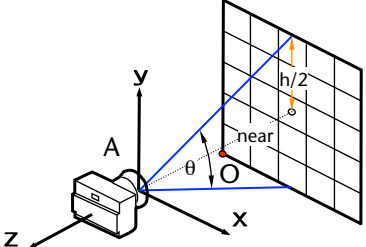
Ray-Tracing 17



G. Zachmann Computer-Graphik 2 - SS 08

Ray-Tracing 18

## Die Kamera (ideale Lochkamera)



$$h = 2 \cdot \text{near} \cdot \tan \frac{\theta}{2}$$

$$O = A - \text{near} \cdot \mathbf{z} - \frac{b}{2} \mathbf{x} - \frac{h}{2} \mathbf{y}$$

Die Main-Loop eines Ray-Tracers

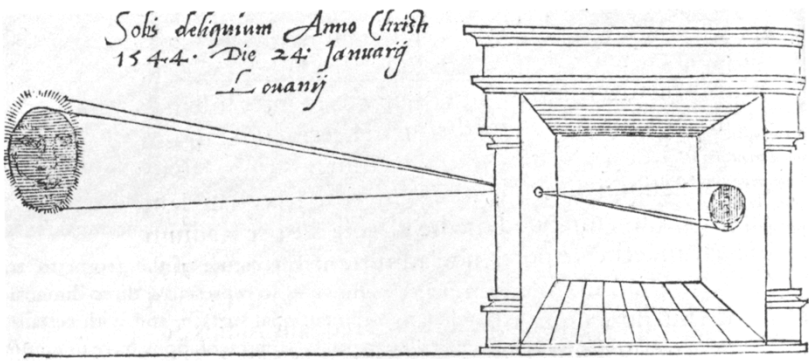
```

for ( t = 0; t < h; t ++ )
  for ( s = 0; s < b; s ++ )
    ray.from = A
    ray.at = O + s·x + t·y
    trace( 0, ray, &color );
    putPixel( x, y, color );

```

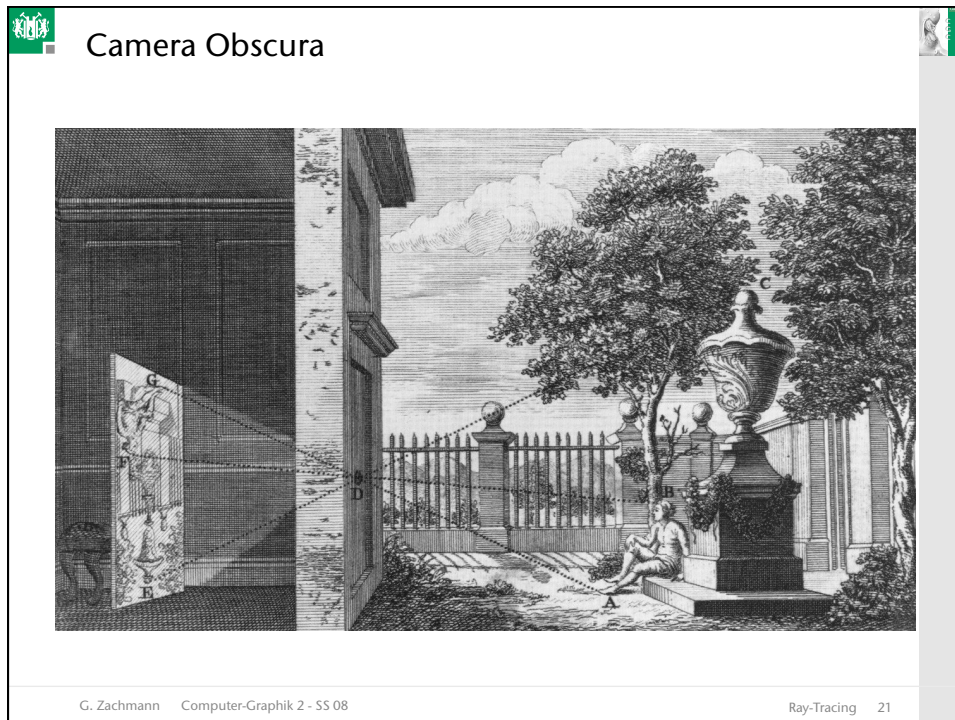
G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 19

## Älteste Abbildung einer Lochkamera



Von R. Gemma Frisius, 1545

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 20



### Andere sonderbare Kameras

- Mit Ray-Tracing sind andere Projektionen sehr einfach
- Z.B. Fischaugen, Omnimax, Panorama




G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 22



## Sekundärstrahlen

- Annahme: Hit zwischen Primärstrahl und Szene gefunden
- Reflektierter Strahl:

$$((-d \cdot n) \cdot n - (-d)) \cdot 2 + (-d) = d - 2(d \cdot n) \cdot n$$

$$\|n\| = 1$$

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 24

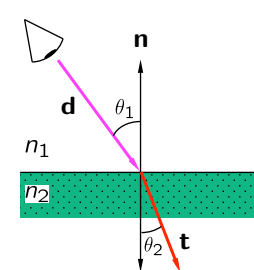
## Gebrochener Strahl

- Brechungsgesetz [Snell ~1600] :
 
$$n_1 \sin \theta_1 = n_2 \sin \theta_2$$
- Der transmittierte Strahl:
 
$$\mathbf{t} = \frac{n_1}{n_2} (\mathbf{d} + \mathbf{n} \cos \theta_1) - \mathbf{n} \cos \theta_2$$

$$\cos \theta_1 = -\mathbf{d} \cdot \mathbf{n}$$

$$\cos^2 \theta_2 = 1 - \frac{n_1^2}{n_2^2} (1 - (\mathbf{d} \cdot \mathbf{n})^2)$$
- Brechungsindizes:
 

Luft	Wasser	Glas	Diamant
1.0	1.33	1.5 - 1.7	2.4



G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 25

$$|\mathbf{n}| = |\mathbf{b}| = 1$$

$$\mathbf{t} = \cos \theta_2 \cdot (-\mathbf{n}) + \sin \theta_2 \cdot \mathbf{b}$$

$$\mathbf{d} = \cos \theta_1 \cdot (-\mathbf{n}) + \sin \theta_1 \cdot \mathbf{b}$$

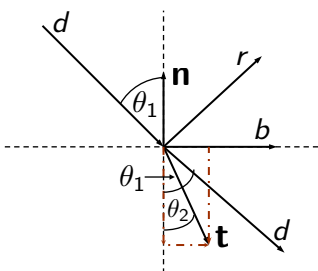
$$\mathbf{b} = \frac{\mathbf{d} + \mathbf{n} \cdot \cos \theta_1}{\sin \theta_1}$$

$$\mathbf{t} = -\mathbf{n} \cdot \cos \theta_2 + \frac{\sin \theta_2}{\sin \theta_1} (\mathbf{d} + \mathbf{n} \cdot \cos \theta_1)$$

cos  $\theta_2$  ausrechnen:

$$\sin \theta_2 = \frac{n_1}{n_2} \sin \theta_1$$

$$\sin^2 + \cos^2 = 1$$

$$\cos^2 \theta_2 = 1 - \left(\frac{n_1}{n_2} \sin \theta_1\right)^2$$


$$\frac{\sin \theta_2}{\sin \theta_1} = \frac{n_1}{n_2}$$

$$\cos \theta_1 = \mathbf{n} \cdot (-\mathbf{d})$$

G. Zachmann Computer-Graphik 2 - SS 08 Ray-Tracing 26

■ Totalreflexion:  
 wenn Radikand  $< 0 \Leftrightarrow \sin \theta_2 \leq \frac{n_1}{n_2}$





G. Zachmann Computer-Graphik 2 - SS 08
Ray-Tracing 27


■ Refraction and the Lifeguard Problem

■ Running is faster than swimming

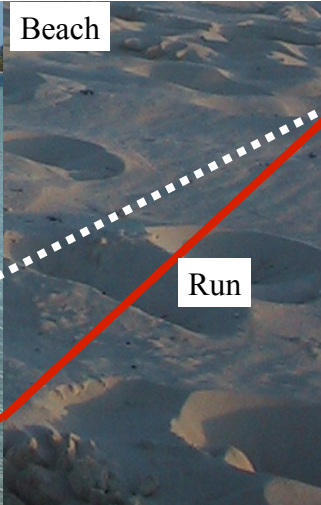
Water




Person in trouble

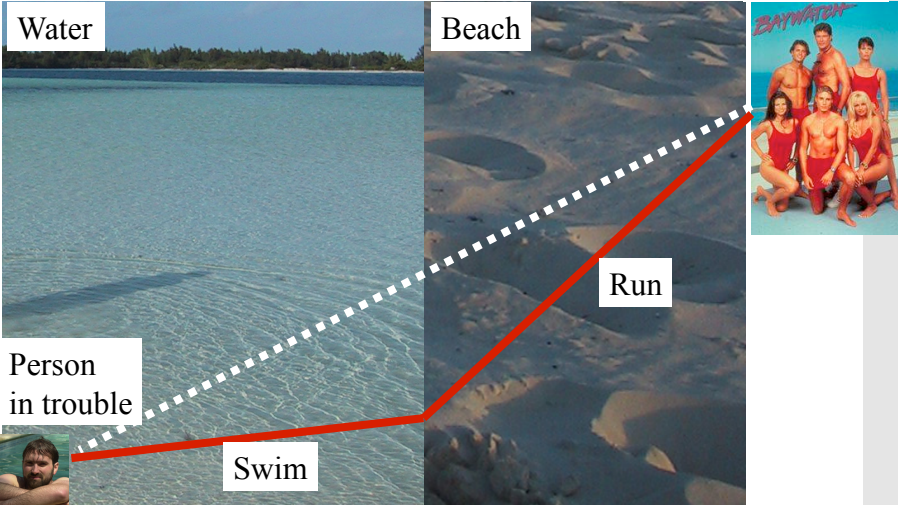


Beach



Lifeguard





G. Zachmann Computer-Graphik 2 - SS 08
Ray-Tracing 28

