



hochschule aschaffenburg
university of applied sciences

Repetitorium – Teil 2

Computergrafik

Rendering Pipeline

Computergrafik

Definitionen

Rendern

Erzeugen eines Bildes aus einem virtuellen räumlichen Modell (Szene).

Anforderungen

- Harte Echtzeit (ca. > 30 fps)
- Weiche Echtzeit (ca. < 30 fps)
- Offline

Modellbeschreibung

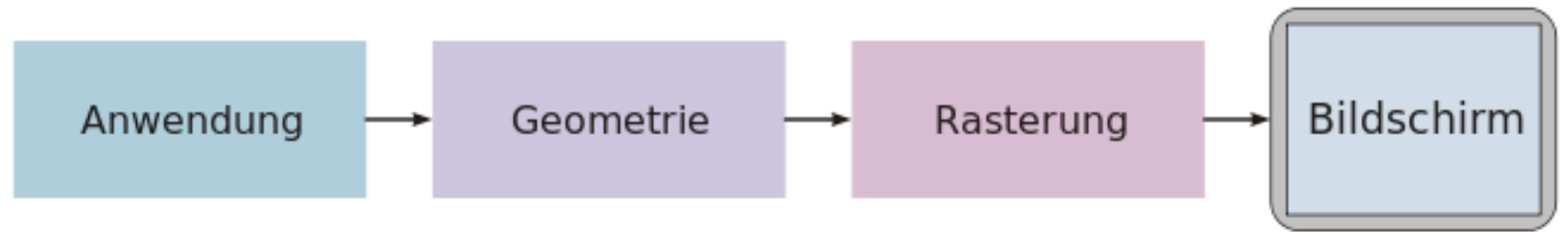
- Geometrie (Raumkoordinaten, Normalen)
- Materialeigenschaften

Szene

- Modelle
 - Hintergrund (Skybox)
 - Lichtquellen
 - Betrachter (Kamera)
-

Rendering Pipeline Schritte

- Anwendung
 - Logik (Kollision, Animation ...)
 - Interaktion
- Geometrie
 - Modellbeschreibung
 - Transformation, Zuschnitt
 - Beleuchtungsmodelle, Shader (3D)
- Rasterung
 - Berechnung Farbwert pro Pixel
 - Verdeckungsrechnung
 - Postprocessing (2D)



(Quelle: Wikipedia)

Geometrieschritt

- Transformation (homogene Koordinaten) längs Sichtbereich (z-Achse)
 - Frustum: near/far clipping plane; Zentralprojektion, Fluchtpunkt
 - Aussehen auf Basis von Materialeigenschaften, Texturen und Beleuchtung
 - Transformation Frustum in Würfel (vgl. Bildschirmkoordinaten x/y)
 - Zuschnitt: Clipping, Culling
-

Geometrieschritt

- Shading
 - Flat Shading: Farbe pro Dreieck aus Eck-Vertices
 - Gouraud Shading: Interpolation zwischen Vertices
 - Phong Shading: Interpolation auf Basis Normalenvektoren
 - Mapping (uv map)
 - Texture Mapping (“bekleben”)
 - Bump Mapping (Normalenvektoren) (+ Blinn-Phong = Struktur)
 - Reflection Mapping (auf Basis Skybox)
-

Beleuchtungsmodelle

Computergrafik

Geometrieschritt

- Beleuchtungsmodelle
 - Lambert (Winkel zu Lichtquelle)
 - Phong (Winkel zu Lichtquelle, Winkel zu Betrachter)
 - Blinn-Phong (Halfway Vektor)
 - Oberflächenbeschaffenheit (rau vs. glatt)
 - Bidirektionale Reflektanzverteilungsfunktion (BRDF)
 - Reflektionskegel
 - Photorealistisch: Sub Surface BRDF
-

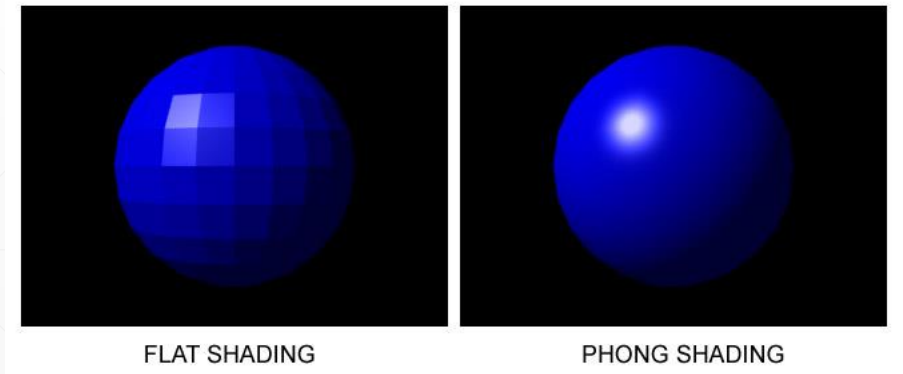
Beleuchtung (hier: Unity)

- direkte Beleuchtung
 - Realtime Lighting
 - Shading: Lambert, Blinn-Phong
 - indirekte Beleuchtung (Global Illumination, bounced lighting)
 - Dynamische Objekte: Lightprobes, Reflection Probes
 - Statische Objekte: baked lighting
 - Schatten
-

Shader

- **Vertex Shader:**
Form, Position, Geometrie
Manipulation der Vertex Koordinaten
Koordinatentransformation (bspw. UnityObjectToClipPos)
- **Fragment Shader:**
(aka Pixel Shader)
Oberfläche, Material, Texturdarstellung
Berechnung der Pixel-Eigenschaften im 2D Bild
Transparenz, Spiegelung, Schattierung, Phong Shading
- **Surface Shader**

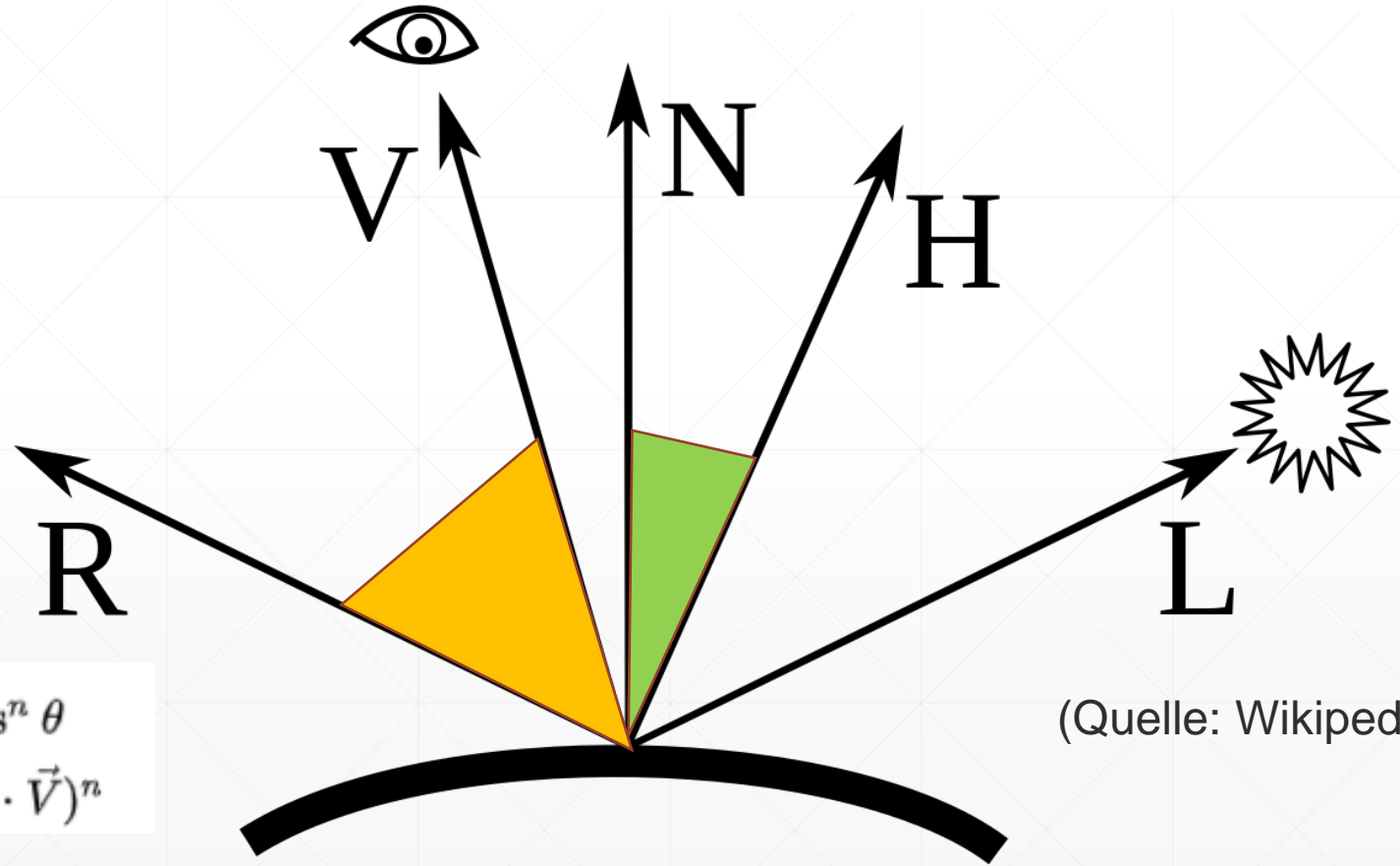
(Quelle:
Wikipedia)



Beleuchtung

- Lambert: Einfallswinkel
- Phong: Reflexionen
- Blinn: Vereinfachung

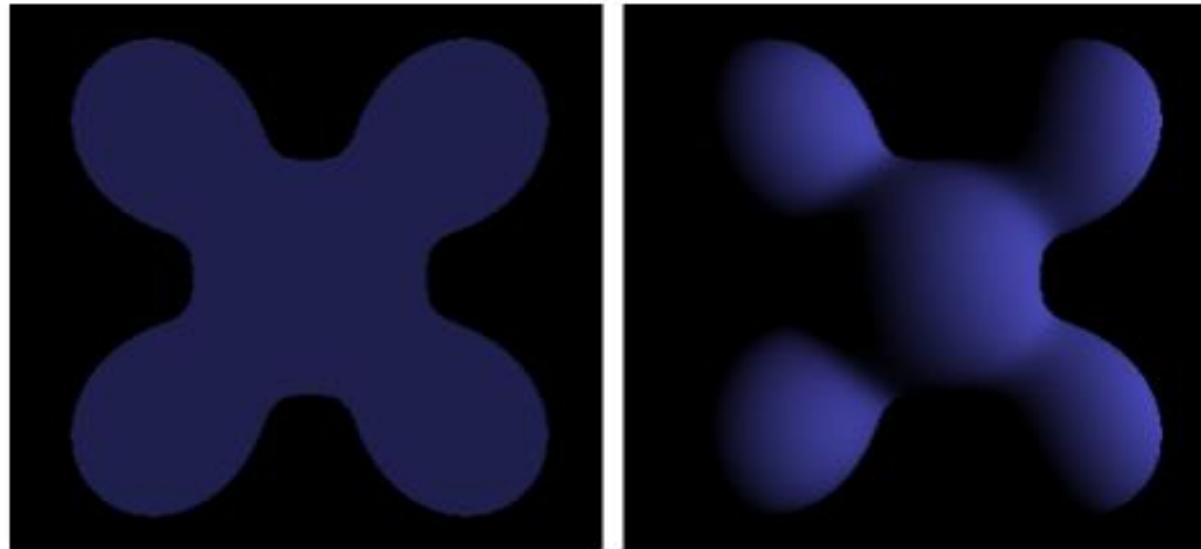
$$\begin{aligned} I_{\text{specular}} &= I_{\text{in}} \cdot k_{\text{specular}} \cdot \cos^n \theta' \\ &= I_{\text{in}} \cdot k_{\text{specular}} \cdot \left(\frac{(V+L) \cdot N}{\|(V+L)\| \cdot \|N\|} \right)^n \end{aligned}$$



$$\begin{aligned} I_{\text{specular}} &= I_{\text{in}} k_{\text{specular}} \cos^n \theta \\ &= I_{\text{in}} k_{\text{specular}} (\vec{R} \cdot \vec{V})^n \end{aligned}$$

(Quelle: Wikipedia)

Lambert Beleuchtung



Ambient

+

Diffuse

(Quelle: Wikipedia)

Ambiente
Beleuchtung:

$$I_{\text{ambient}} = I_a k_{\text{ambient}}$$

mit

- I_a ... Intensität des Umgebungslichts
- k_{ambient} ... Materialkonstante

Diffuse Beleuchtung:

$$\begin{aligned} I_{\text{diffus}} &= I_{\text{in}} k_{\text{diffus}} \cos \varphi \\ &= I_{\text{in}} k_{\text{diffus}} (\vec{L} \cdot \vec{N}) \end{aligned}$$

mit

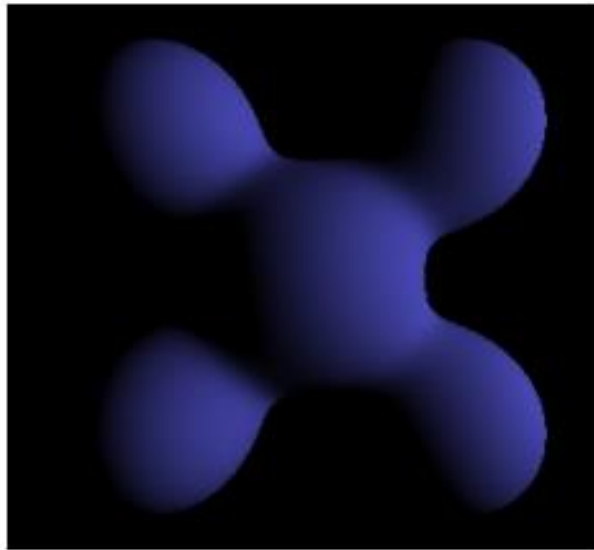
- I_{in} ... Lichtstärke des einfallenden Lichtstrahls
- k_{diffus} ... empirisch bestimmter Reflexionsfaktor
- φ ... Winkel zwischen Normalenvektor

Phong Beleuchtung



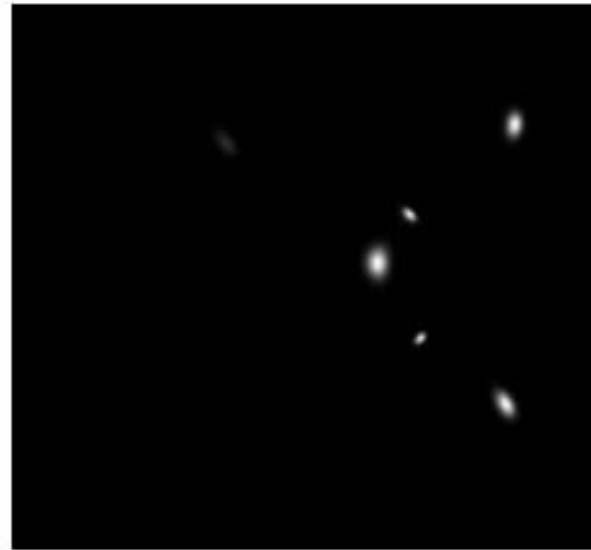
Ambient

+



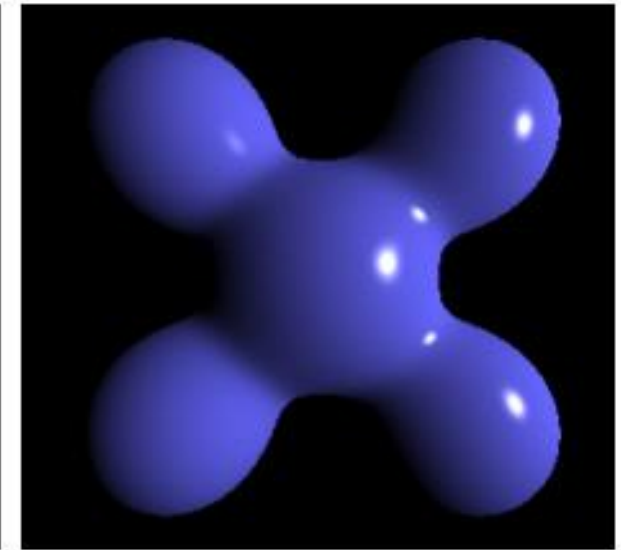
Diffuse

+



Specular

=



Phong Reflection

(Quelle: Wikipedia)

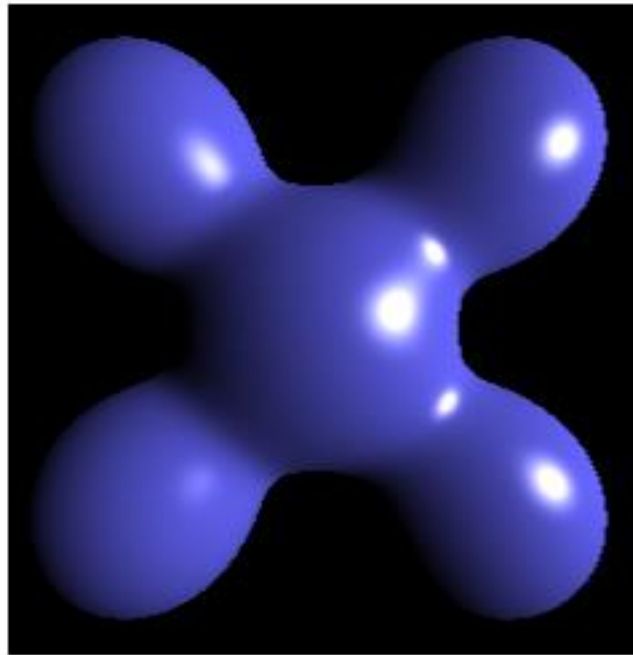
Specular:

$$\begin{aligned} I_{\text{specular}} &= I_{\text{in}} k_{\text{specular}} \cos^n \theta \\ &= I_{\text{in}} k_{\text{specular}} (\vec{R} \cdot \vec{V})^n \end{aligned}$$

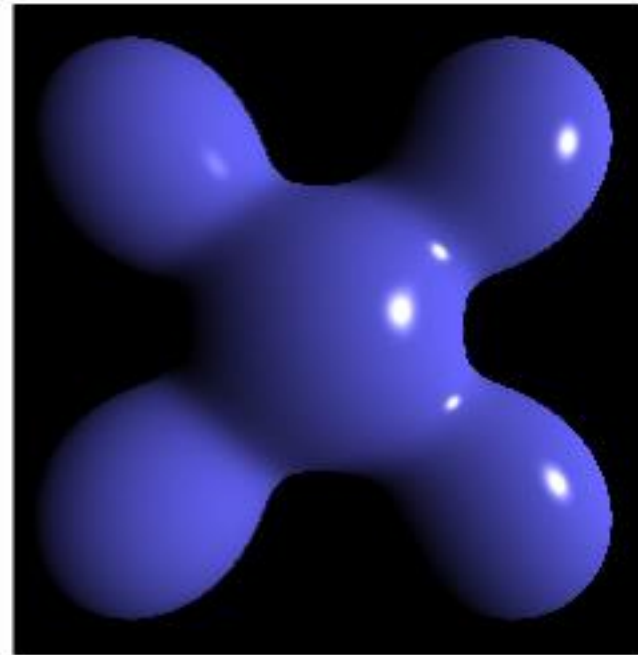
mit

- I_{in} ... Lichtstärke des einfallenden Lichtstrahls der Punktlichtquelle
- k_{specular} ... empirisch bestimmter Reflexionsfaktor für spiegelnde Komponente der Reflexion
- θ ... Winkel zwischen idealer Reflexionsrichtung des ausfallenden Lichtstrahls \vec{R} und Blickrichtung des Betrachters \vec{V}
- n ... konstanter Exponent zur Beschreibung der Oberflächenbeschaffenheit

Blinn-Phong Beleuchtung



Blinn-Phong



Phong

(Quelle: Wikipedia)

Shader

Computergrafik

Shader

- **Properties**
 - aus Unity Inspector
 - Deklaration im Programm
- **CG (C for Graphics) Programm**
 - aka HLSL (High Level Shader Language)
 - Input aus dem Modell
 - “surf”: Berechnungsblock
- **FallBack**
 - Default Shader
- **Lambert**
 - Beleuchtungsmodell
 - Standard: Blinn-Phong

```
Shader "HSA/SimpleSurfaceShader" {
```

```
    Properties {
```

```
        _myColour ("Example Colour", Color) = (1,1,1,1)
```

```
        _myEmission ("Example Emission", Color) = (1,1,1,1)
```

```
    }
```

```
    SubShader {
```

```
        CGPROGRAM
```

```
        #pragma surface surf Lambert
```

= “surface” shader,
Typ: Lambert

```
        struct Input {
```

```
            float2 uvMainTex;
```

```
        };
```

```
        fixed4 _myColour;
```

```
        fixed4 _myEmission;
```

```
        void surf (Input IN, inout SurfaceOutput o){
```

```
            o.Albedo = _myColour.rgb;
```

```
            o.Emission = _myEmission.rgb;
```

```
        }
```

```
        ENDCG
```

```
    FallBack "Diffuse"
```

```
}
```

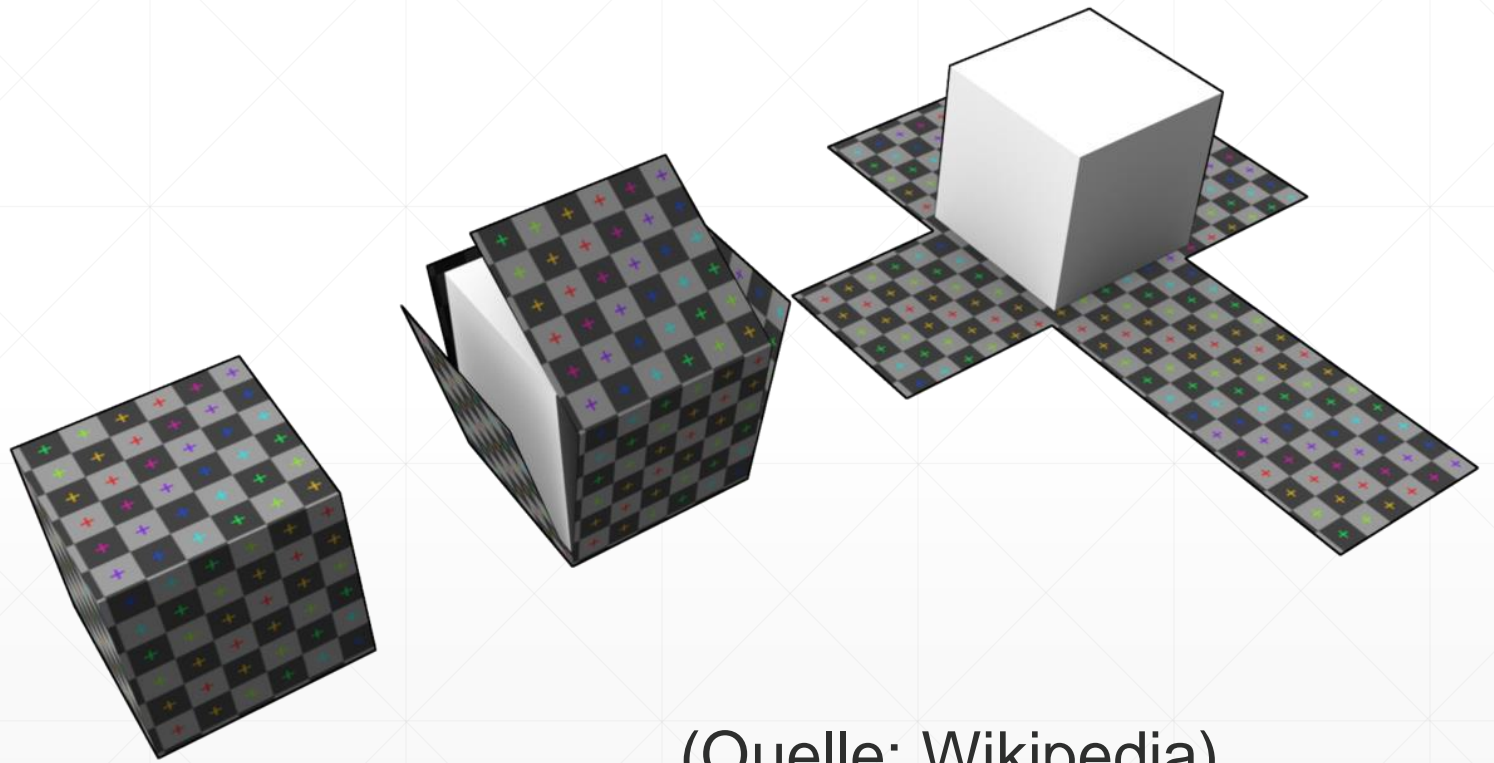

Input Structure (per Pixel)

- float2 uv_MainTex;
- float2 uv2_MainTex;
- float3 viewDir; // Betrachtungswinkel
- float3 worldPos; // Modellkoordinaten (3D)
- float3 worldRefl; // Reflektionen

```
struct Input {  
    float2 uv_mainTex;  
    float2 uv2_mainTex;  
    float3 viewDir;  
    float3 worldPos;  
    float3 worldRefl;  
};
```

Beispiel: UV Map

- Zwischen 0 und 1
- Vertex-spezifisch
- `Float2 uv_MainTex;`



(Quelle: Wikipedia)

Surface Output

- fixed3 Albedo; // diffuse color
 - fixed3 Normal; // Normalenvektor
 - fixed3 Emission; // Leuchtkraft
 - half Specular; // Blinn-Phong (Specular Highlight)
 - fixed Gloss; // Strength of Specular Reflection
 - fixed Alpha; // Transparency
-

Postprocessing

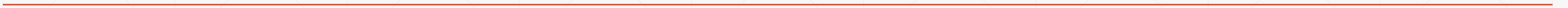
Computergrafik

Post Processing in Unity

- Fog
 - Anti Aliasing
 - Ambient Occlusion
 - Screen Space Reflection
 - Depth of Field
 - Motion Blur
 - Eye Adaption
 - Bloom
 - Color Grading
 - User Lut
 - Chromatic Aberration
 - Grain
 - Vignette
 - Dithering
-

Aufgaben

- Nennen und beschreiben Sie zwei mögliche Postprocessing Funktionen von Unity



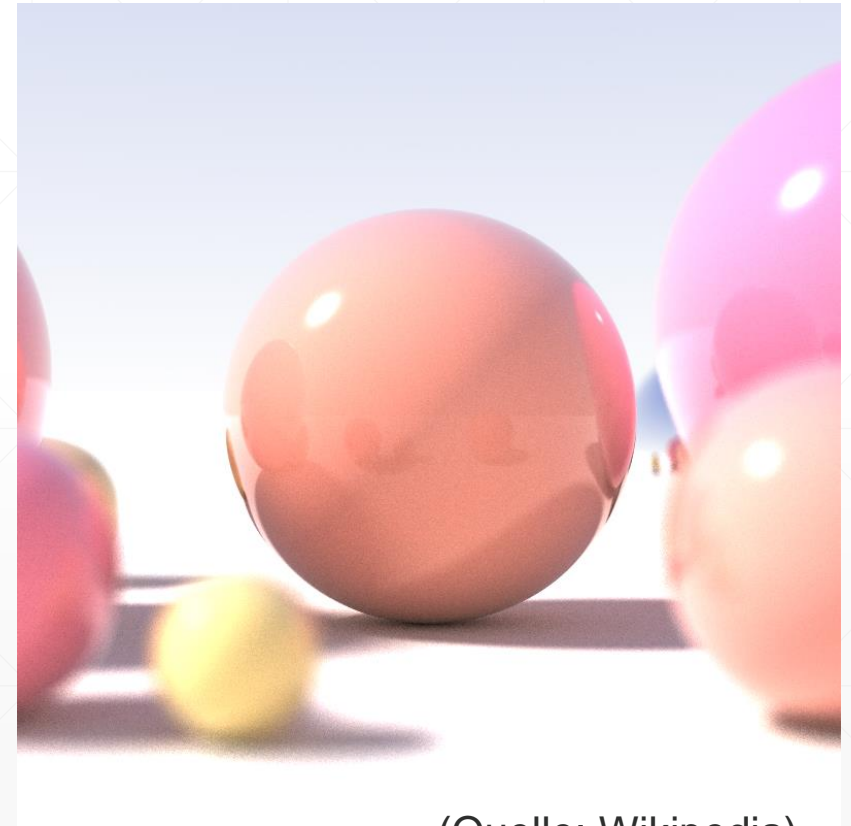
Raytracing

Computergrafik

Rendering

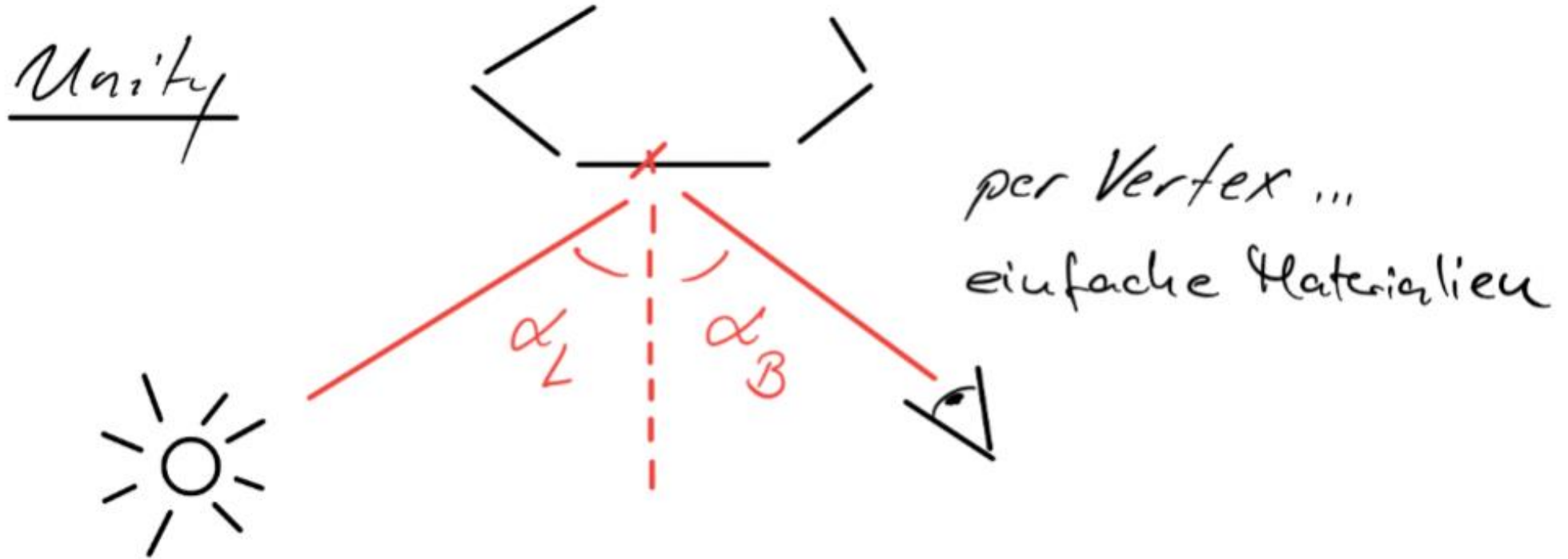
Raytracing

- Strahlverfolgung
 - Verdeckungsberechnung
 - Streuung und Reflektion an Oberflächen (physikalisch)
- Berechnung der Lichtverteilung
- Bildsynthese:
 - Path Tracing
 - Photon Mapping
 - Particle Tracing
- Alternative (historisch): Radiosity
 - Finite Elemente Verfahren
 - Indirekte, diffuse Beleuchtung



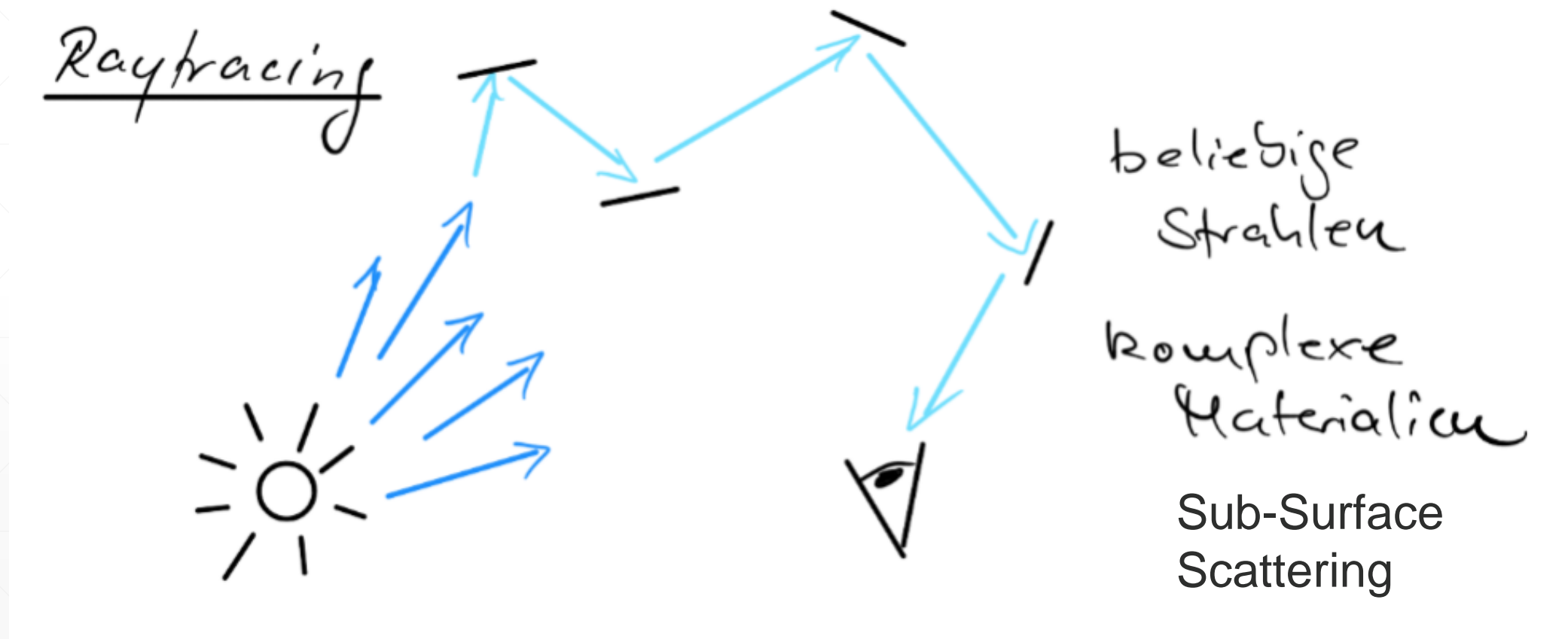
(Quelle: Wikipedia)

Rendering



Berechnung der Winkel
Berechnung des Farborts des Pixels
(Näherung)

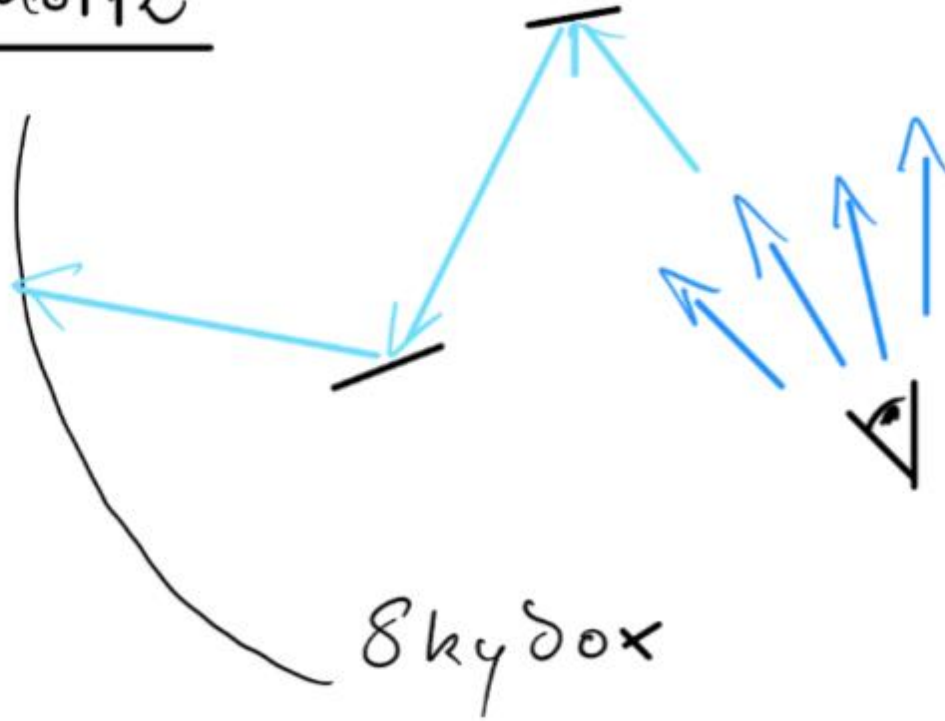
Light Tracing / Particle Tracing



Realistische
Lichtverteilung in der

Path Tracing

Helmholtz



es gibt einen
Betrachter
aber viele
Lichtquellen

Globale Beleuchtung

- direkte Beleuchtung + indirekte Beleuchtung
- Rendergleichung

$$L(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_{\Omega} f_r(x, \vec{\omega}', \vec{\omega}) L(x, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}'$$

$L(x, w)$: Energiefluss von Punkt x der Oberfläche in Richtung w

L_e : Eigenemission

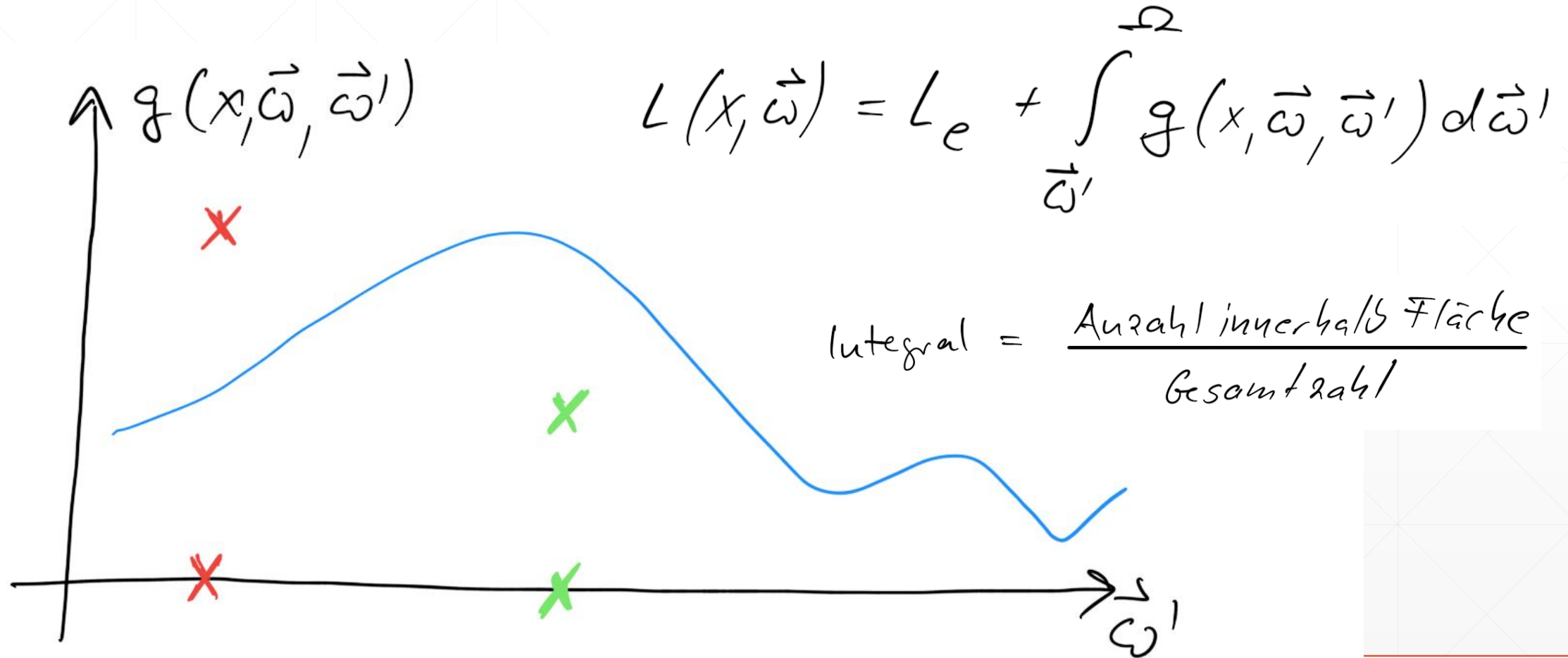
f_r : Streuungsterm (BRDF), winkelabhängig

$L(x, w')$: Energiefluss aus Richtung w' auf $x \rightarrow$ direkt + indirekt (Photon Mapping)

$w \cdot n$: Skalarprodukt von Einfallswinkel und Normalenvektor (vgl. Lambert)

Ω : Halbraumwinkel über Fläche

Monte Carlo Integration



Aufgabe

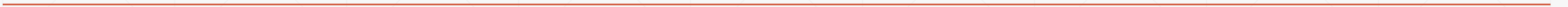
- Beschreiben Sie Raytracing? Nennen Sie zwei Vorgehensweisen.
 - Was ist Monte Carlo Integration?
-

Bildmanipulation

Computergrafik

Aufgabe

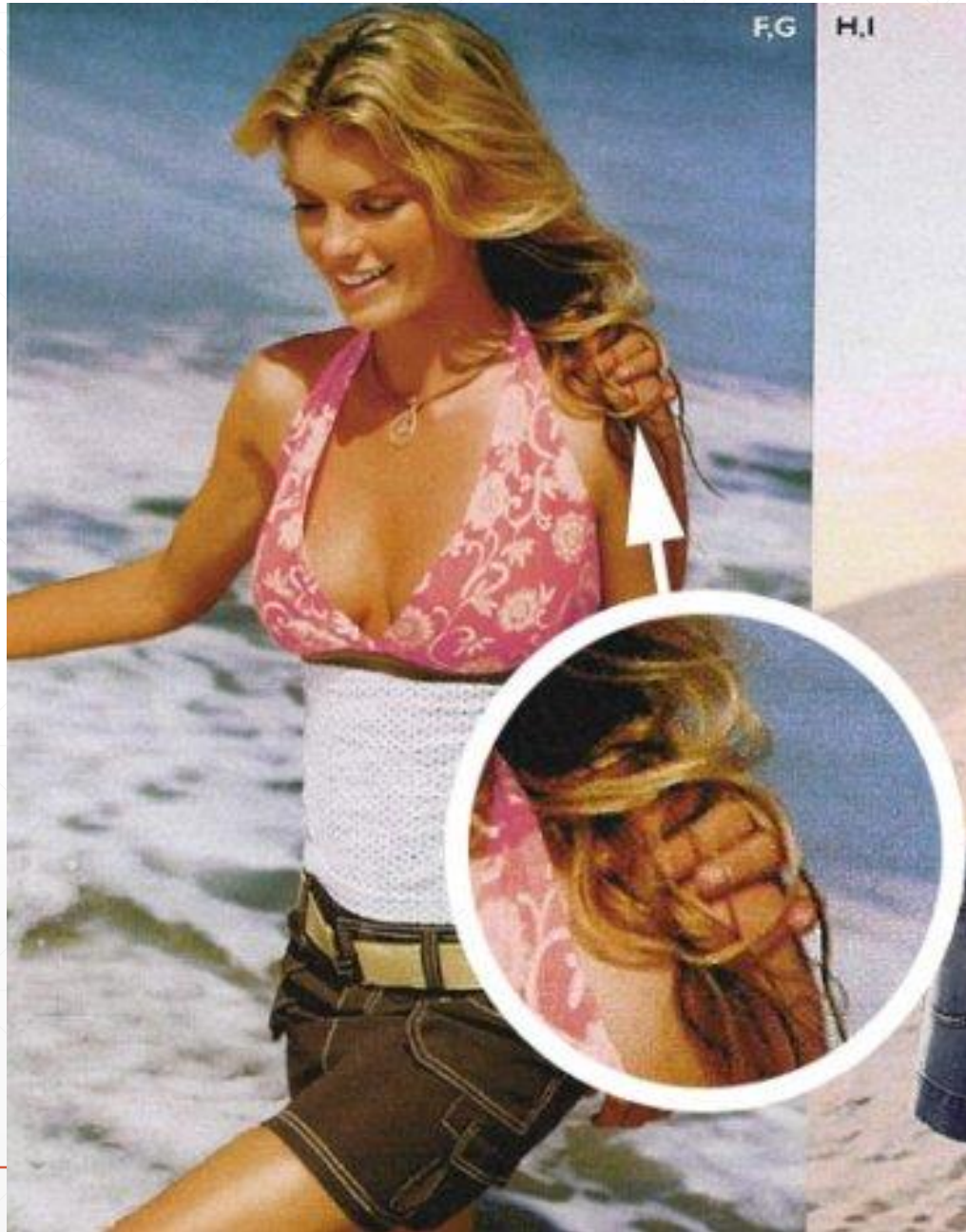
- Woran erkennt man Bildmanipulationen?



Fehler

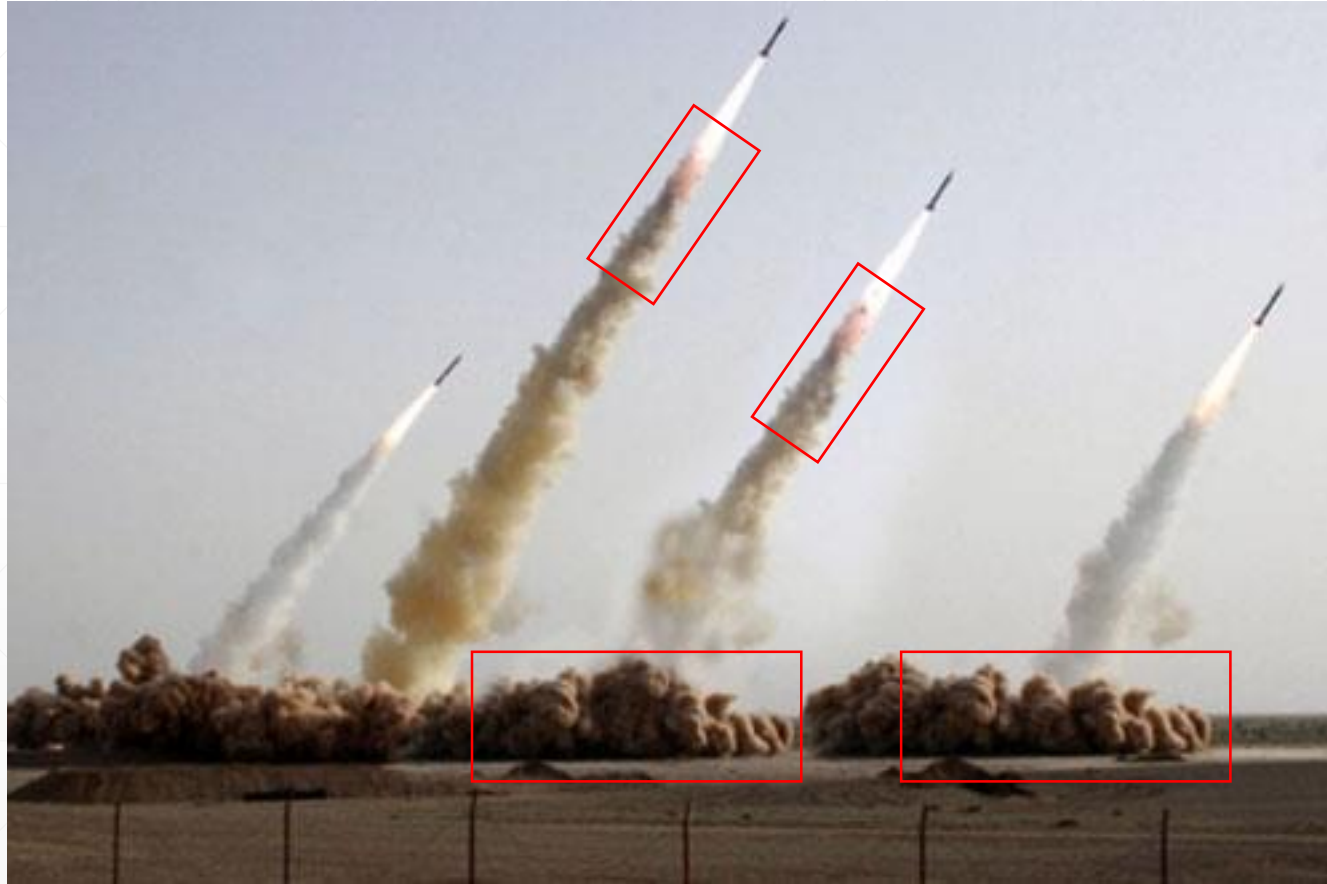
- Übergänge
 - Licht und Schatten
 - Über-Kontrastierung
 - Harte Kanten
 - Artefakte
 - ...
-

Fehler



Quelle: PopSugar

Fotomontage



Quelle: Spiegel vom 10. Juli 2008