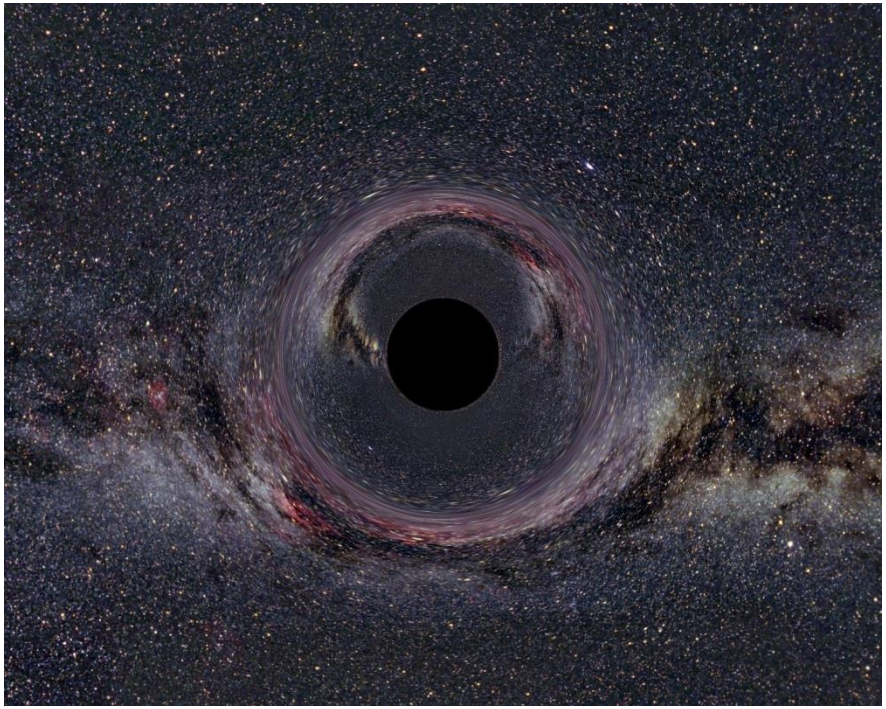
Game Technology

Lecture 7 – 28.11.2015
Physically Based Rendering

TECHNISCHE UNIVERSITÄT DARMSTADT



Dipl-Inform. Robert Konrad
Dr.-Ing. Florian Mehm

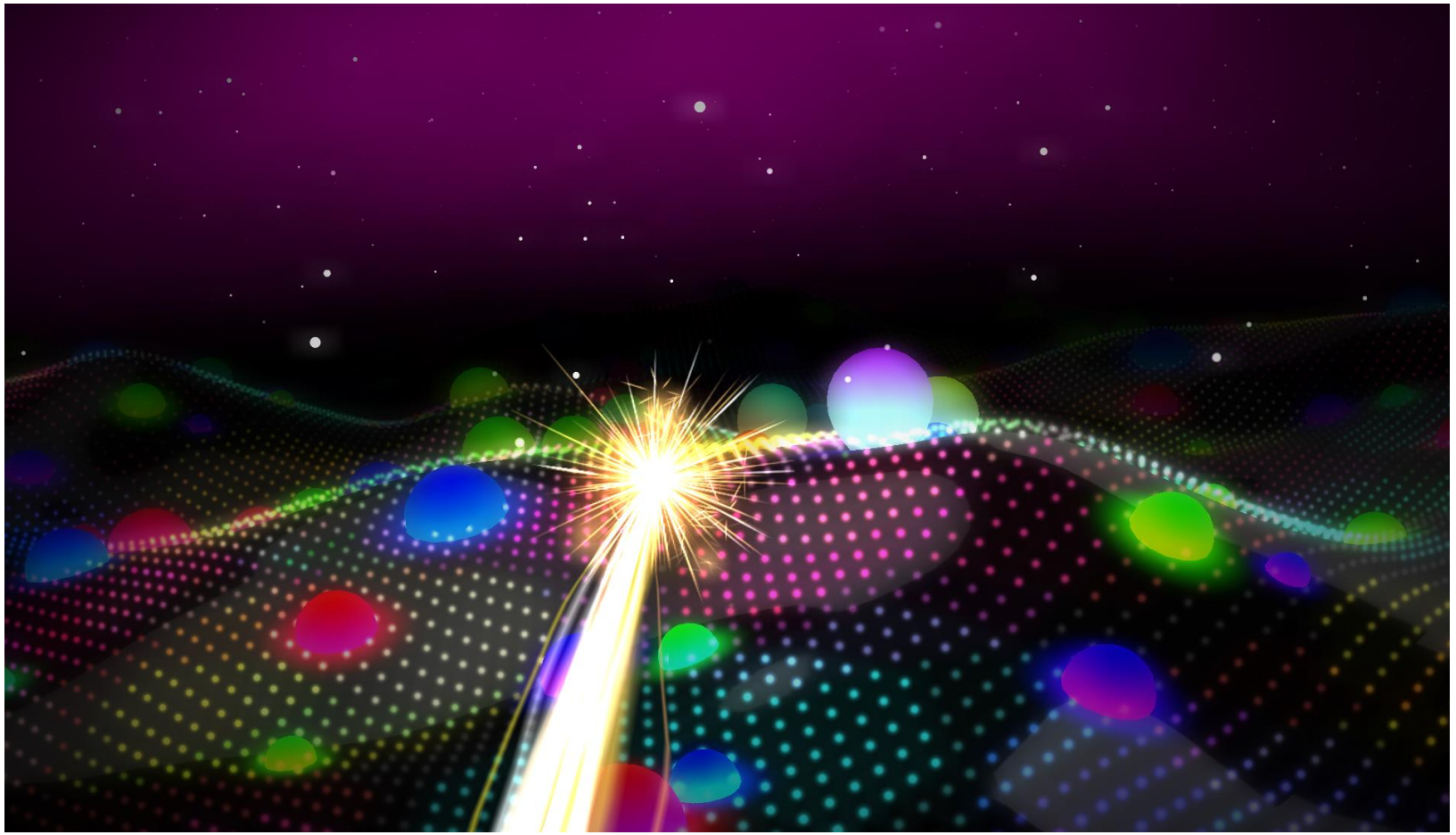
Prof. Dr.-Ing. Ralf Steinmetz
KOM - Multimedia Communications Lab

Light

Light

- Starts from light source
- Bounces around
  - Looses intensity with each collision
- Eventually reaches the camera

Light Sources



http://lights.helloenjoy.com

Point Lights

TECHNISCHE UNIVERSITÄT DARMSTADT

Defined by a position and light color/intensity

Directional Light

TECHNISCHE UNIVERSITÄT DARMSTADT

Just a direction and light intensity/color

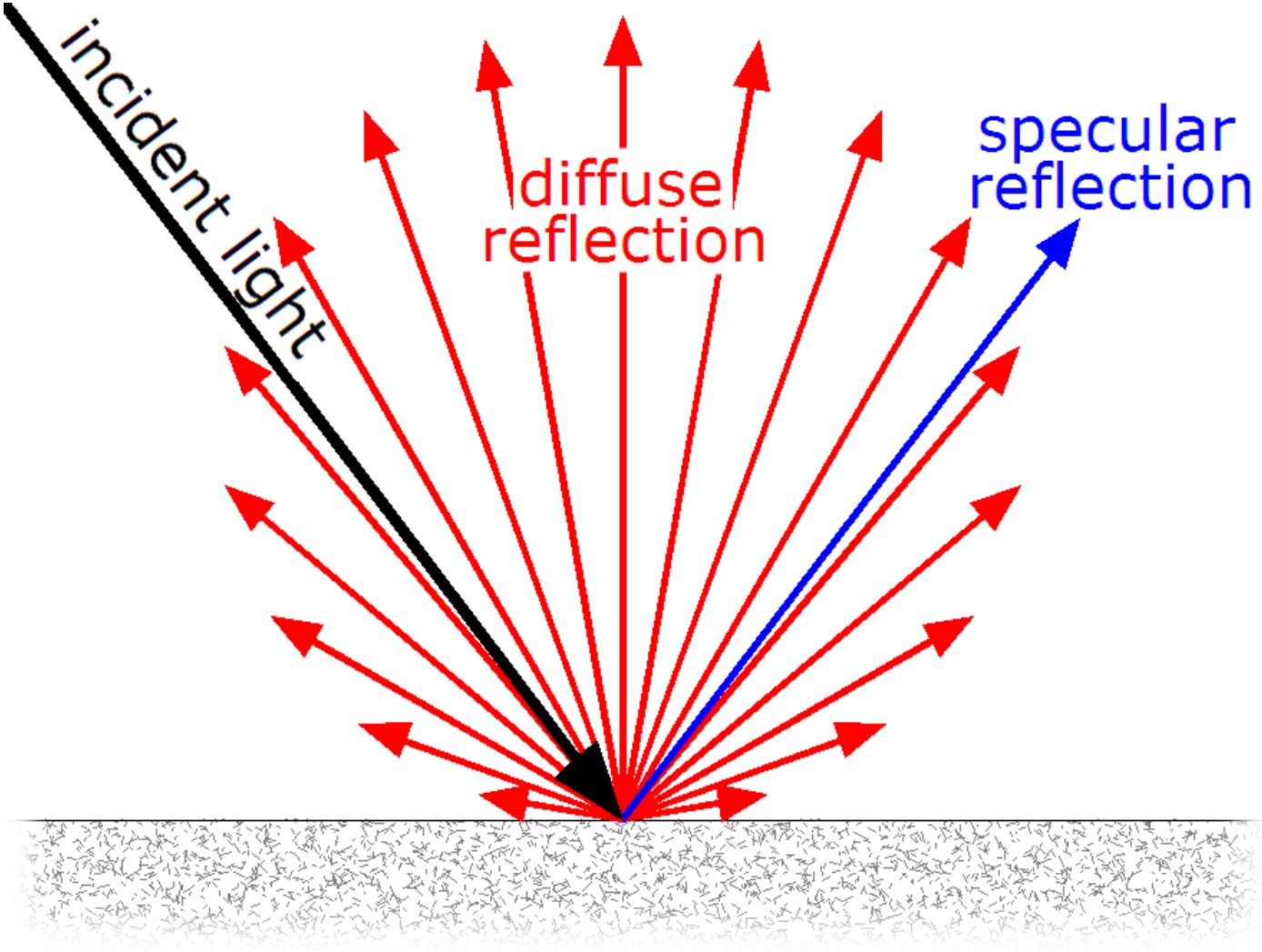
Area Lights

TECHNISCHE UNIVERSITÄT DARMSTADT

More light sources possible

Not supported by most game engines

Bounces



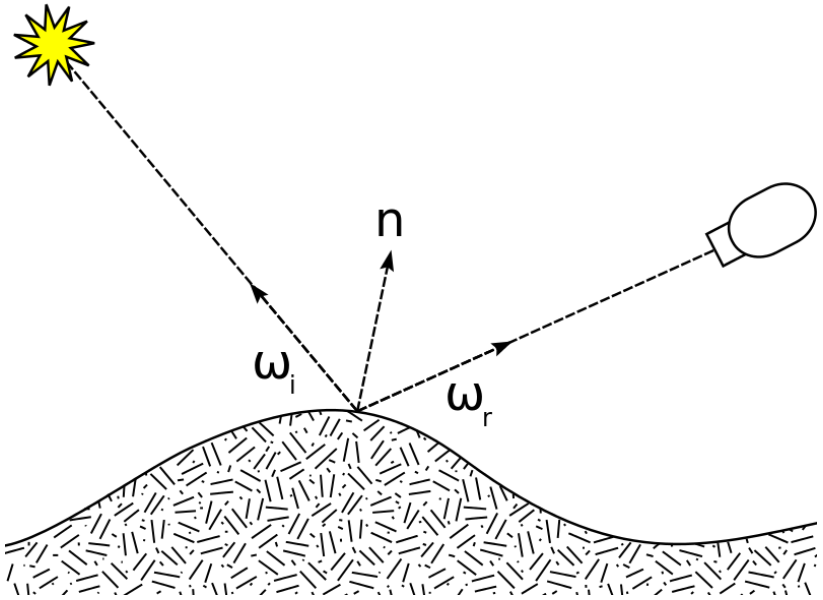
BRDF

TECHNISCHE UNIVERSITÄT DARMSTADT

Bidirectional reflectance distribution function

- incoming light direction
- outgoing direction (for example to the camera)
- returns the ratio of reflected radiance

$$f_r(\omega_i, \omega_r)$$



Path Tracing

TECHNISCHE UNIVERSITÄT DARMSTADT

foreach (pixel)

- bounce around a lot

Use BRDF at each collision

Very slow

- but useful to create reference images
- and for prerendered lighting informations

Realtime Lighting

TECHNISCHE UNIVERSITÄT DARMSTADT

Consider only light rays from direct light sources

- First bounce

Use shadow maps

- Second bounce

Ignore further light bouncing

- No reflections
- No ambient light

Image-Based Lighting

TECHNISCHE UNIVERSITÄT DARMSTADT

Put surroundings in cube map

- Use for example path tracing to generate the cube map

Ignore lights, instead sample cube map

A cube map is only correct for one position

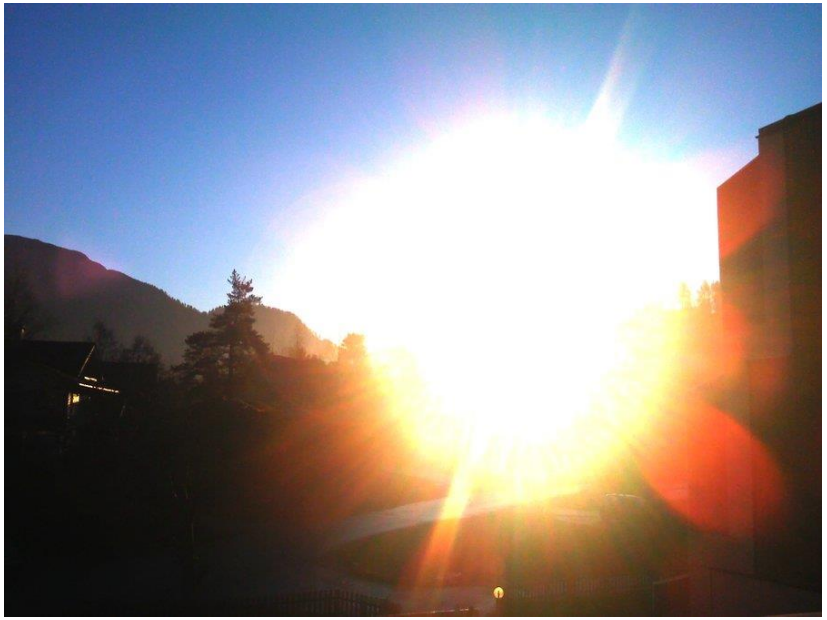
Ignores dynamic objects

HDR

TECHNISCHE UNIVERSITÄT DARMSTADT

„High dynamic range“

Use more than 32 bits of data for one pixel



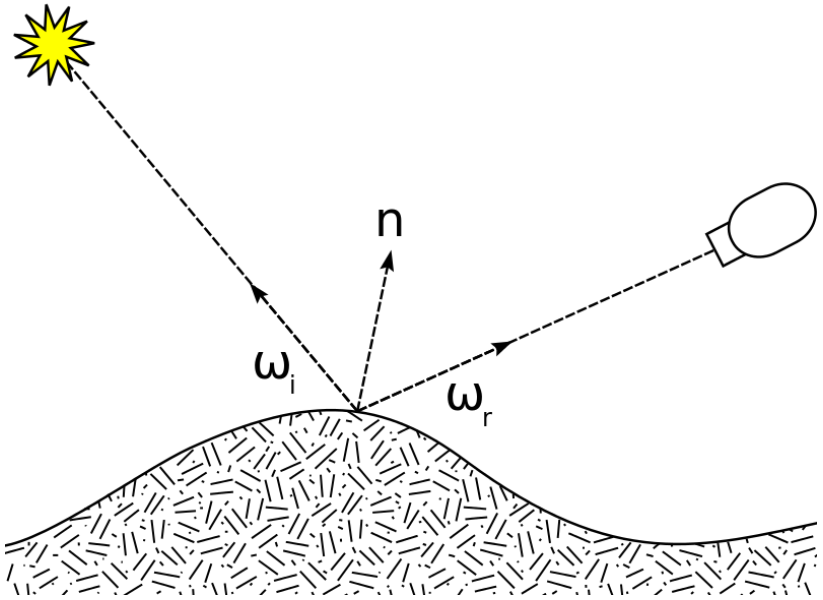
BRDF

TECHNISCHE UNIVERSITÄT DARMSTADT

Bidirectional reflectance distribution function

- incoming light direction
- outgoing direction (for example to the camera)
- returns the ratio of reflected radiance

$$f_r(\omega_i, \omega_r)$$

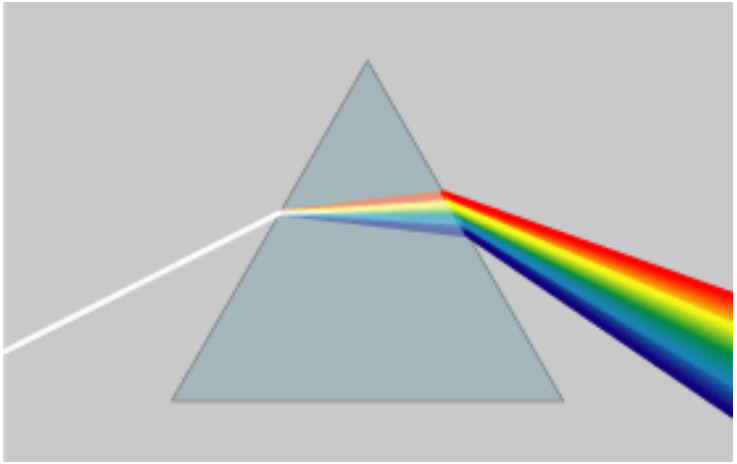


BRDF Shortcomings

TECHNISCHE UNIVERSITÄT DARMSTADT

Subsurface-Scattering

Wavelength dependence



BRDF

TECHNISCHE UNIVERSITÄT DARMSTADT

Only positive light

$$f_r(\omega_i, \omega_r) \geq 0$$

BRDF

TECHNISCHE UNIVERSITÄT DARMSTADT

Inverted

$$f_r(\omega_i, \omega_r) = f_r(\omega_r, \omega_i)$$

BRDF

TECHNISCHE UNIVERSITÄT DARMSTADT

Energy conserving

$$\forall \boldsymbol{\omega_i}, \int_{\Omega} \boldsymbol{f_r(\omega_i, \omega_r) cos(\theta_r) d\omega_r} \leq \mathbf{1}$$

Phong Lighting

TECHNISCHE UNIVERSITÄT DARMSTADT

color = ambient + diffuse + specular

Phong Lighting

TECHNISCHE UNIVERSITÄT DARMSTADT

color = ~~ambient~~ + diffuse + specular

Gamma

TECHNISCHE UNIVERSITÄT DARMSTADT

sRGB

TECHNISCHE UNIVERSITÄT DARMSTADT

See Exercise 1

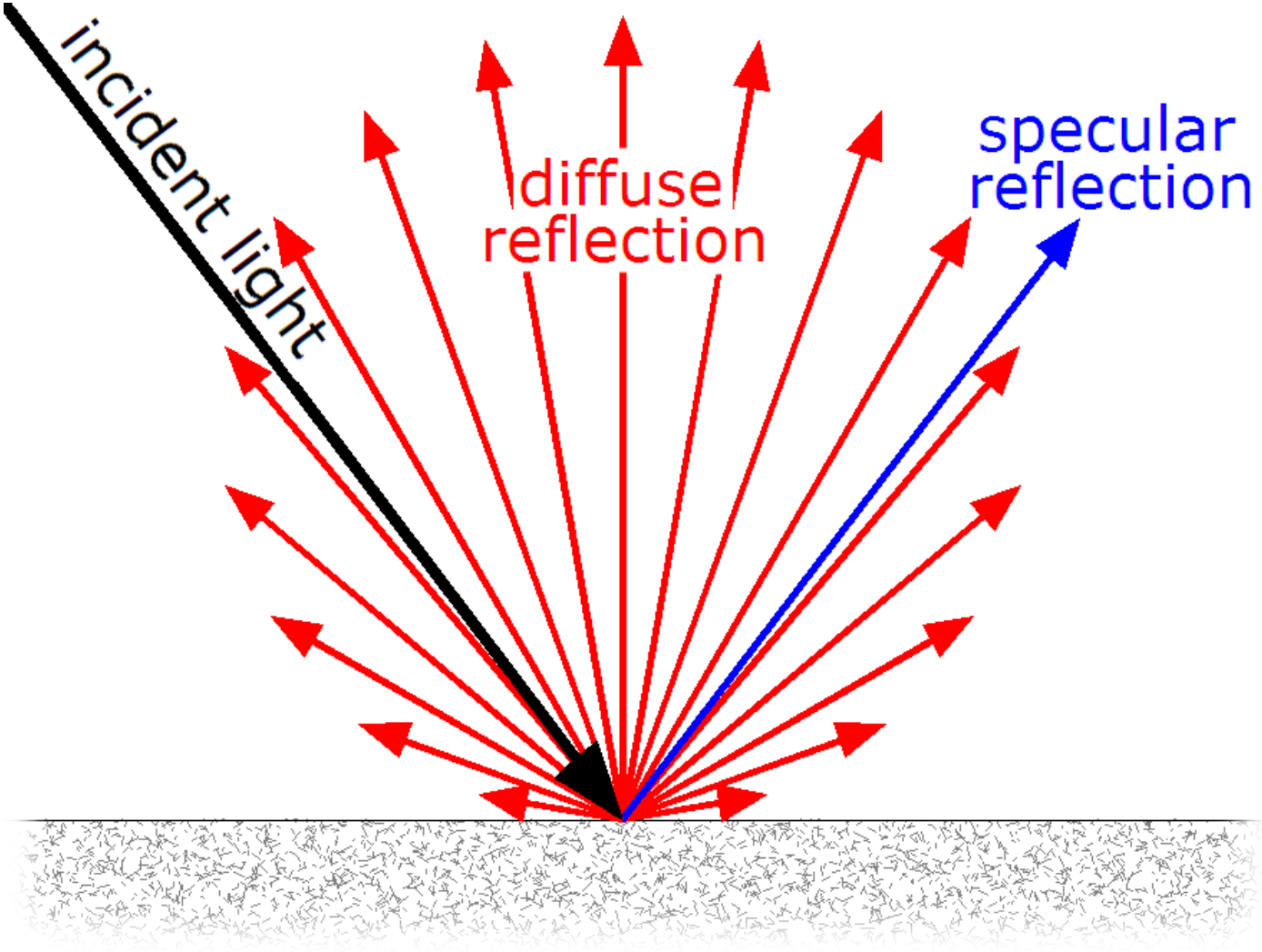
Transform textures to linear (pow 2.2)

- Or use sRGB texture reading (also allows proper filtering)

Lighting calculations in linear space (gamma 1)

Then transform for sRGB (pow 1 / 2.2)

Diffuse & Specular



Diffuse

TECHNISCHE UNIVERSITÄT DARMSTADT

Lambertian reflectance / Phong diffuse

I = L*N

Good enough for modern engines

- Used for example in Unreal Engine 4

Specular

Brick

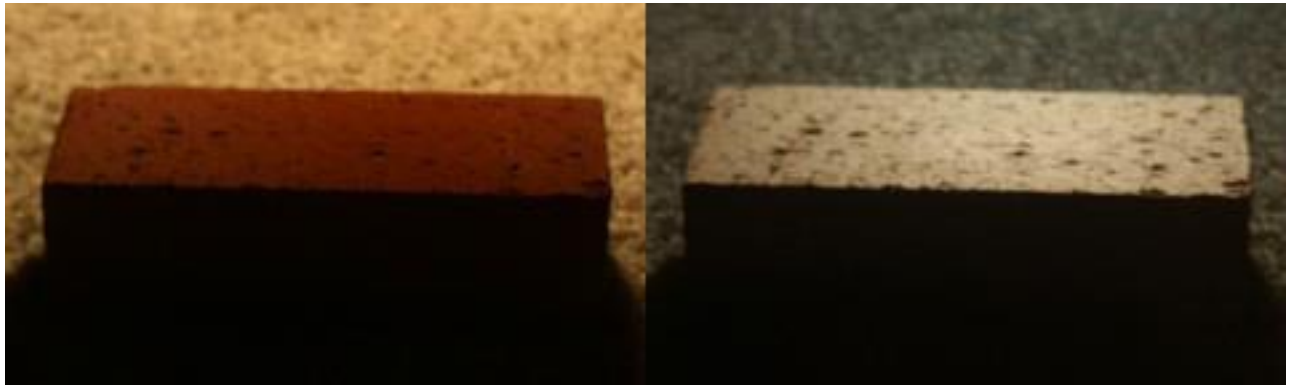
TECHNISCHE UNIVERSITÄT DARMSTADT

angle(normal, light) = angle(normal, camera)

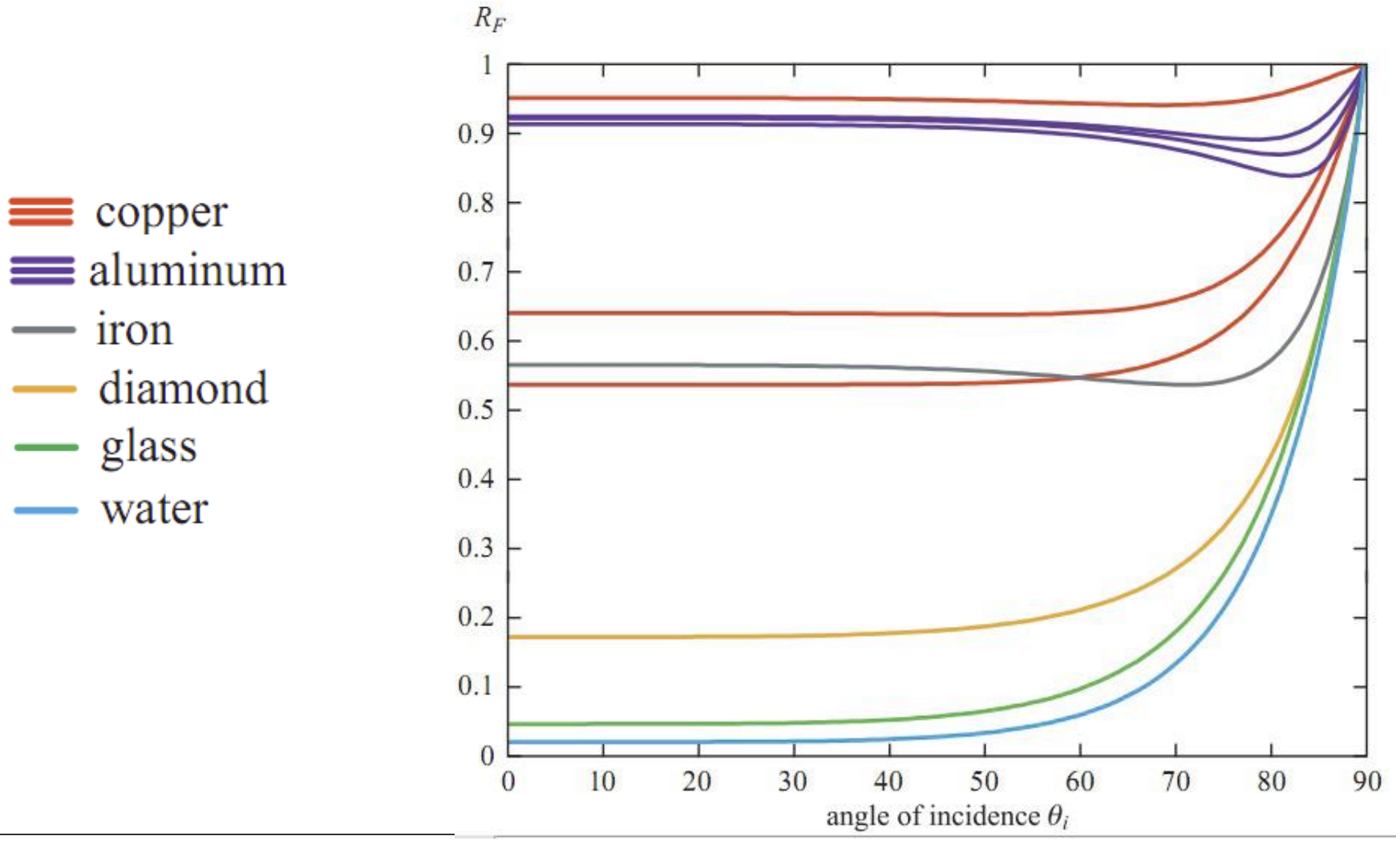
Brick

TECHNISCHE UNIVERSITÄT DARMSTADT

angle(normal, light) = angle(normal, camera)



Fresnel


copper
aluminum
iron
diamond
glass
water
R_F
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
angle of incidence θ_i

Fresnel

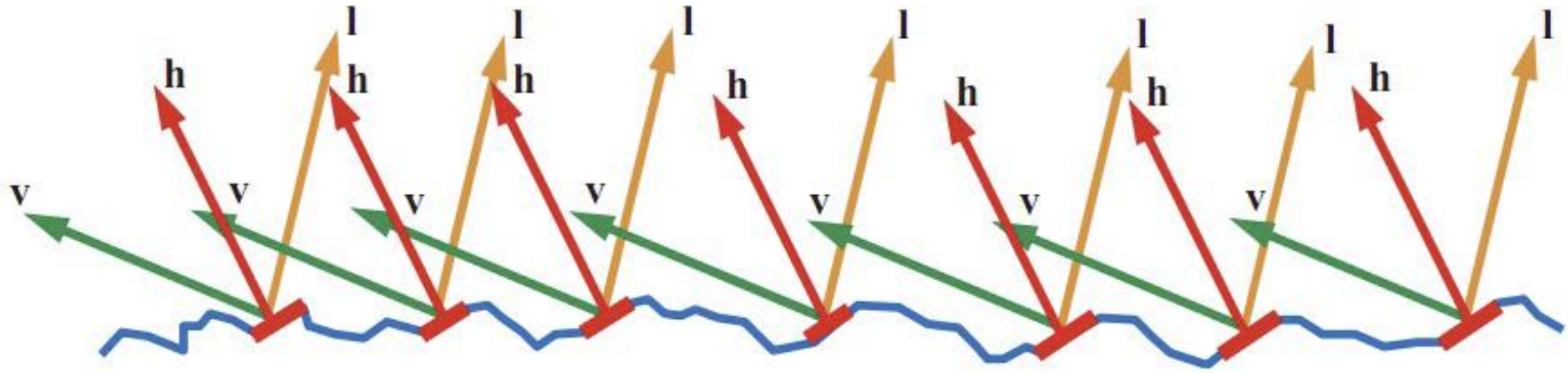
Schlick Approximation

TECHNISCHE UNIVERSITÄT DARMSTADT

Schlick(spec, light, normal) = spec + (1 - spec) (1 - (light*normal))^5

Microfacet Model

TECHNISCHE UNIVERSITÄT DARMSTADT



Microfacet BRDF

$$f(l,v) = \frac{F(l,h)G(l,v,h)D(h)}{4(n \cdot l)(n \cdot v)}$$

Normal Distribution

TECHNISCHE UNIVERSITÄT DARMSTADT

D(h)

Portion of microfacets pointing to h

$$D_{tr}(m) = \frac{\alpha_{tr}^2}{\pi\left((n \cdot m)^2(\alpha_{tr}^2 - 1) + 1\right)^2}$$

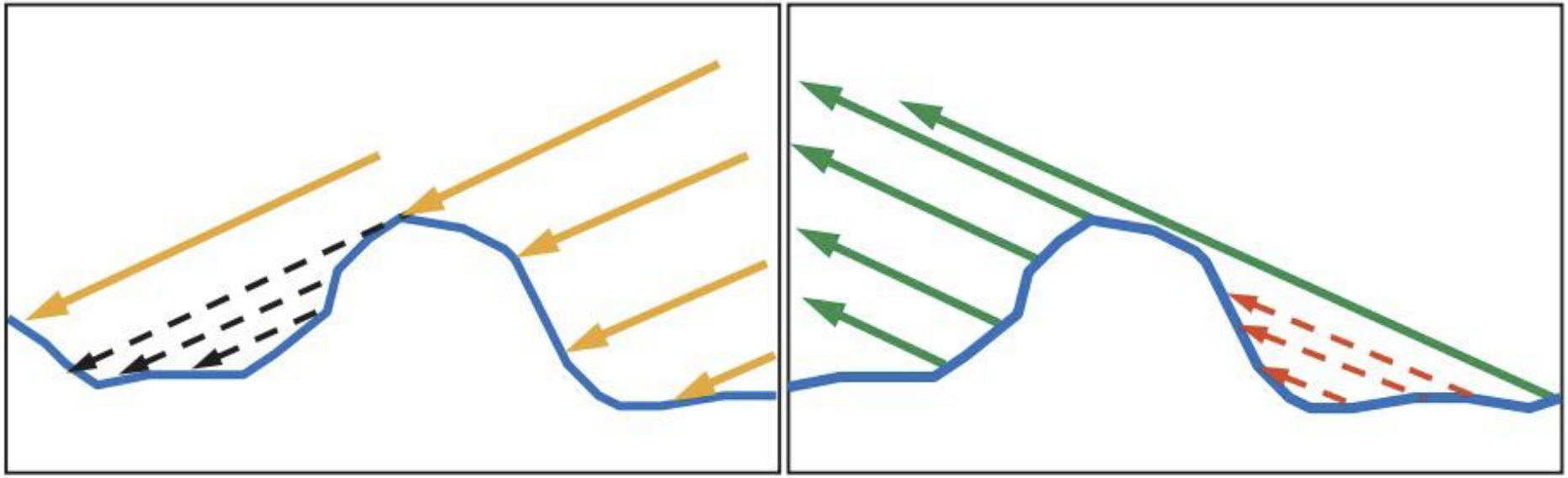
Trowbridge-Reitz (GGX)

α: Roughness

Geometry Factor

- **G(l, v, h)**
- **Cook-Torrance:**

$$G_{ct}(l,v,h) = \min\left(1, \frac{2(n\cdot h)(n\cdot v)}{(v\cdot h)}, \frac{2(n\cdot h)(n\cdot l)}{(v\cdot h)}\right)$$



Physically Based Rendering

TECHNISCHE UNIVERSITÄT DARMSTADT

In practice:

- Gamma correction
- Microfacet BRDF
- Lots of Cube Maps

Polarization of Reflected Light

TECHNISCHE UNIVERSITÄT DARMSTADT

Specular Reflection

- Polarization does not change

Diffuse Reflection

- Polarization is randomized

Cardboard

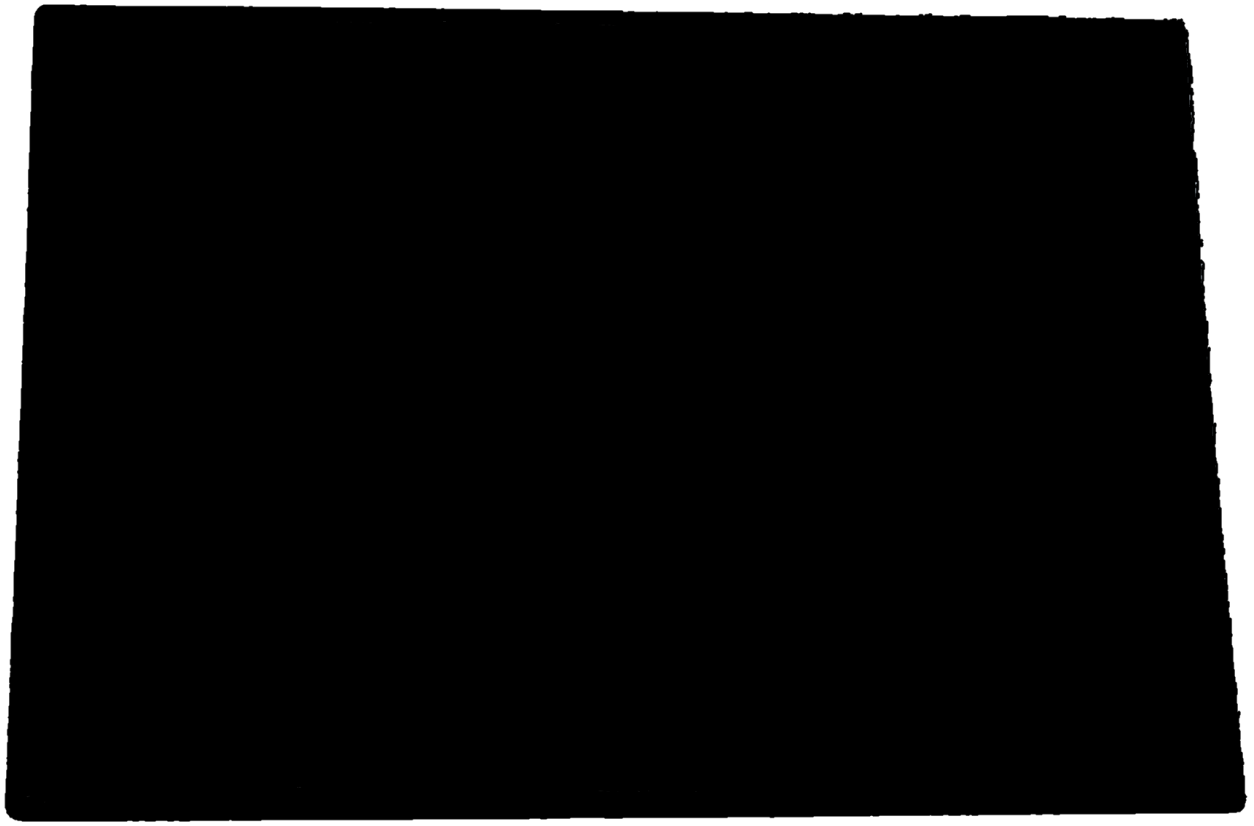
Cardboard Diffuse

Cardboard Specular

Metal

Metal Specular

Metal Diffuse



Metals and Dielectrics

TECHNISCHE UNIVERSITÄT DARMSTADT

Metals:

- No diffuse
- High Specular

Dielectrics

- Diffuse
- Low Specular

Note: Specular value is specified at low angles

Textures

TECHNISCHE UNIVERSITÄT DARMSTADT

Typical Setup

- Diffuse texture
- Specular texture
- Roughness texture
- Normal Map

Incorporating Image Based Lighting

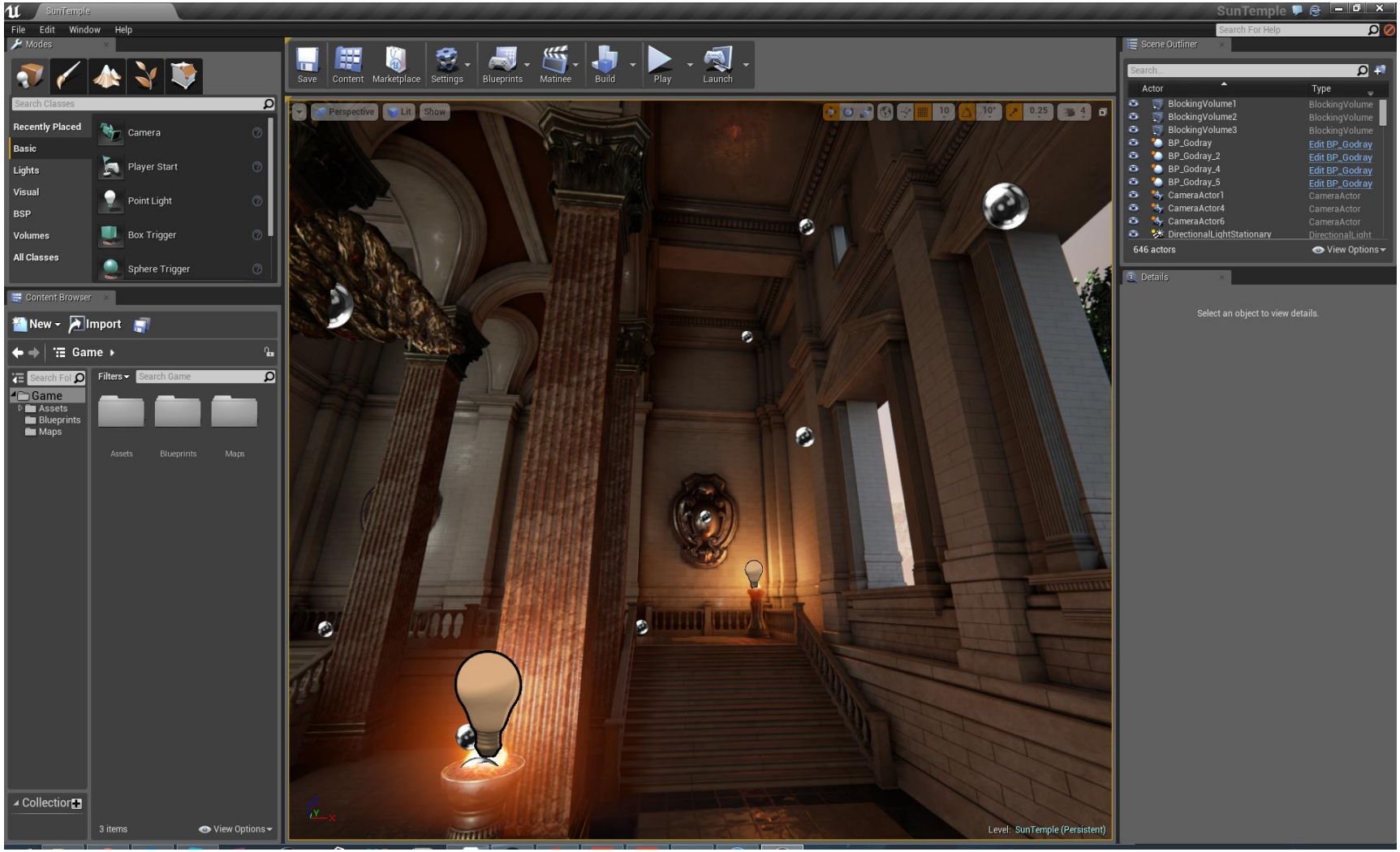
TECHNISCHE UNIVERSITÄT DARMSTADT

Precalculate Cube Maps

- Lots of Cube Maps
- Manually placed in level editor

Cube Maps

TECHNISCHE UNIVERSITÄT DARMSTADT



Cube Maps

TECHNISCHE UNIVERSITÄT DARMSTADT

Can be interpolated

- Which is a rough approximation

Can not capture dynamic objects

Reflection Rendering

TECHNISCHE UNIVERSITÄT DARMSTADT

As done in Unreal Engine 4, Killzone Shadow Fall,…

Deferred Rendering Pass

Raytrace depth buffer

If no hit

- Interpolate local cube maps

If no hit

- Use global cube map

Ambient Occlusion

TECHNISCHE UNIVERSITÄT DARMSTADT

Small notches are normally shadowed

- Unless lit directly

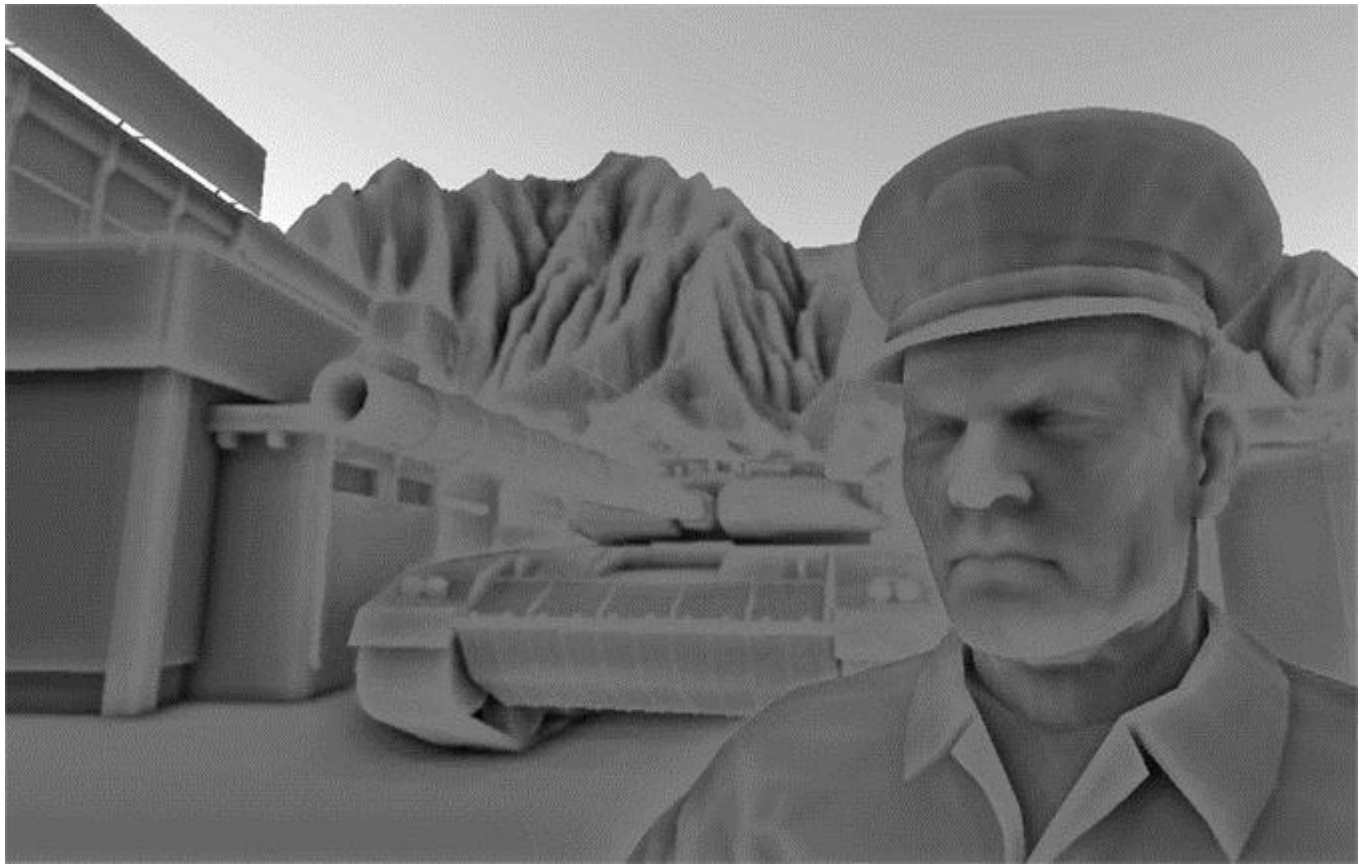
Calculations need very exact light bounces

Screen Space Ambient Occlusion

TECHNISCHE UNIVERSITÄT DARMSTADT

Filter after rendering

Darken at sharp normal changes



Global Illumination

TECHNISCHE UNIVERSITÄT DARMSTADT

„Ambient light“

Spherical Harmonic Lighting

Voxel Cone Tracing

…

Summary

GPU Internals

TECHNISCHE UNIVERSITÄT DARMSTADT

NVidia GeForce GT 6600, 2004

Memory

TECHNISCHE UNIVERSITÄT DARMSTADT

Memory bandwidth is extremely important

- Textures
- Framebuffer
- Depth Buffer

Memory access times not very important

- Most data is streamed
- Access times can be hidden by switching tasks

Memory

TECHNISCHE UNIVERSITÄT DARMSTADT

Gigantic discrepancy from low-end to high-end

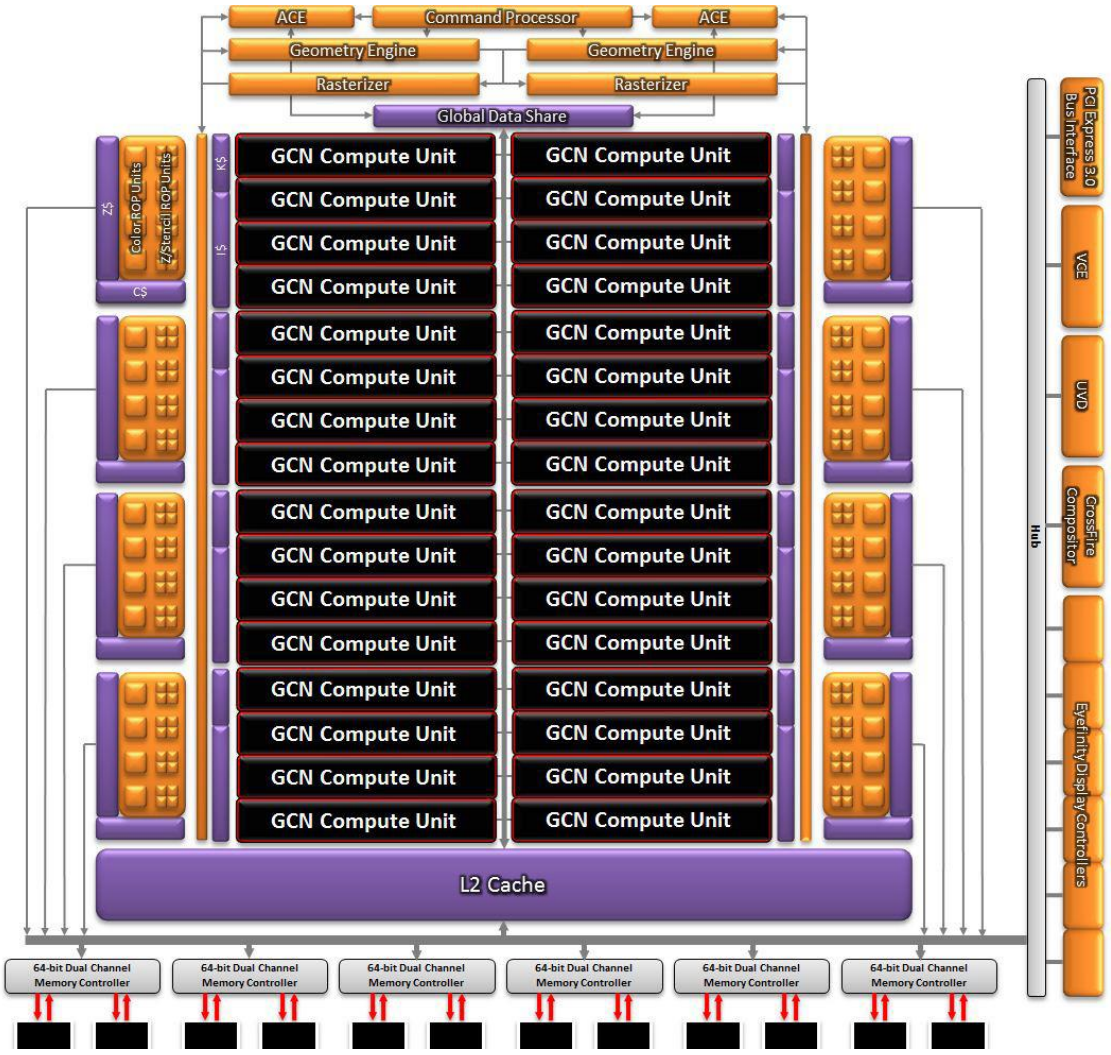
- GeForce 720 starts at 14.4 GB/s
- PS4: 176 GB/s
- GeForce 780 Ti: 336 GB/s
- GeForce GTX 980: 224 GB/s

Vertex and Fragments Shaders

TECHNISCHE UNIVERSITÄT DARMSTADT

Run on the same hardware

Dynamically scheduled


ACE
Command Processor
ACE
Geometry Engine
Geometry Engine
Rasterizer
Rasterizer
Global Data Share
GCN Compute Unit
L2 Cache
Z$
Color ROP Units
Z/Stencil ROP Units
C$
K$
I$
Hub
PCI Express 3.0 Bus Interface
VCE
UVD
CrossFire Compositor
Eyefinity Display Controllers
64-bit Dual Channel Memory Controller

SMT

TECHNISCHE UNIVERSITÄT DARMSTADT

Also used in CPUs (Hyperthreading)

Switch to different thread when stalled (for example waiting for memory)

SIMD

TECHNISCHE UNIVERSITÄT DARMSTADT

Compiler can put calculations on multiple vertices/pixels in one instruction

Problem: Flow control

- Wrong paths pseudo-executed
  - Can be efficient when all vertices/pixel in one pack take the same paths

Shader variants

- „Typically when you create a simple surface shader, it internally expands into 50 or so internal shader variants” (Shader Compilation in Unity 4.5)

Parallelization

TECHNISCHE UNIVERSITÄT DARMSTADT

Small work packages can prevent parallelization

- Performance dip for tiny triangles

CPU <-> GPU

TECHNISCHE UNIVERSITÄT DARMSTADT

Biggest performance trap

Minimize state changes

Minimize draw calls

Send little data to the GPU

If possible never read data from the GPU

Summary