

Interaktive Computergrafik



Prof. Dr. Frank Steinicke
Human-Computer Interaction
Department of Computer Science
University of Hamburg



Interaktive Computergrafik

Lektion 12

Prof. Dr. Frank Steinicke

Human-Computer Interaction, Universität Hamburg



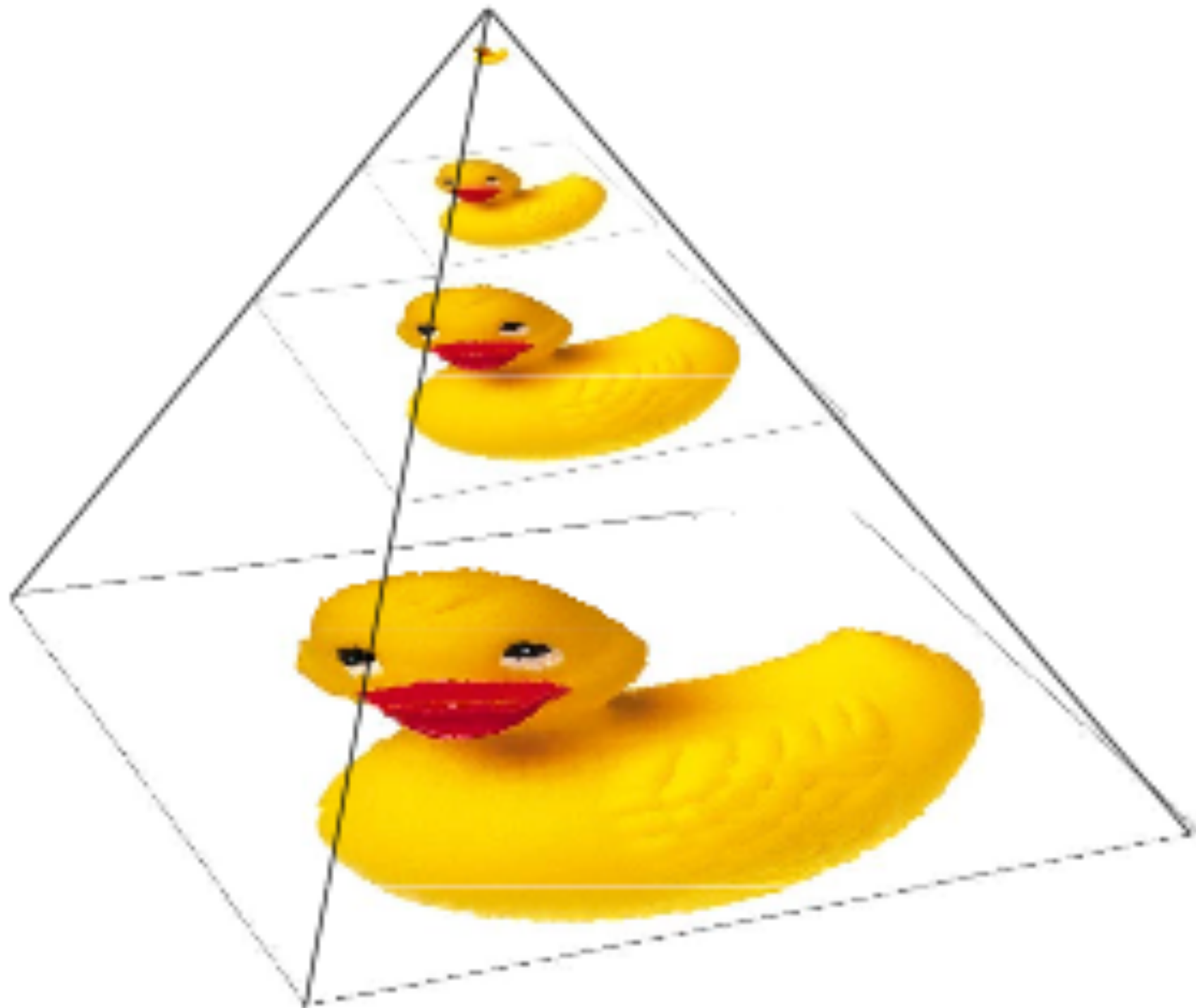
Interaktive Computergrafik

Lektion 12

Texturen 2

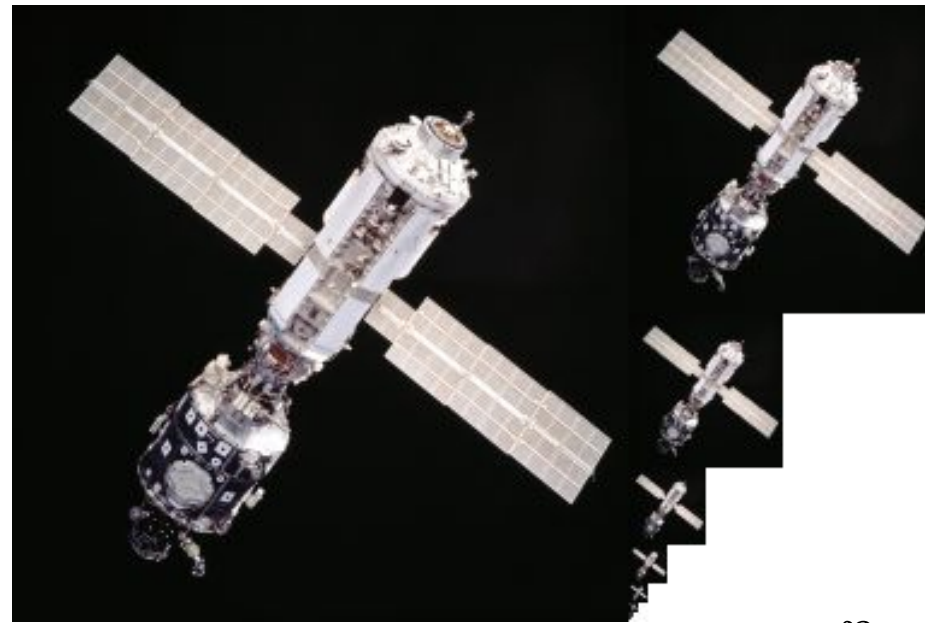
MIP-Mapping

Beispiel: Texturpyramide



MIP-Mapping

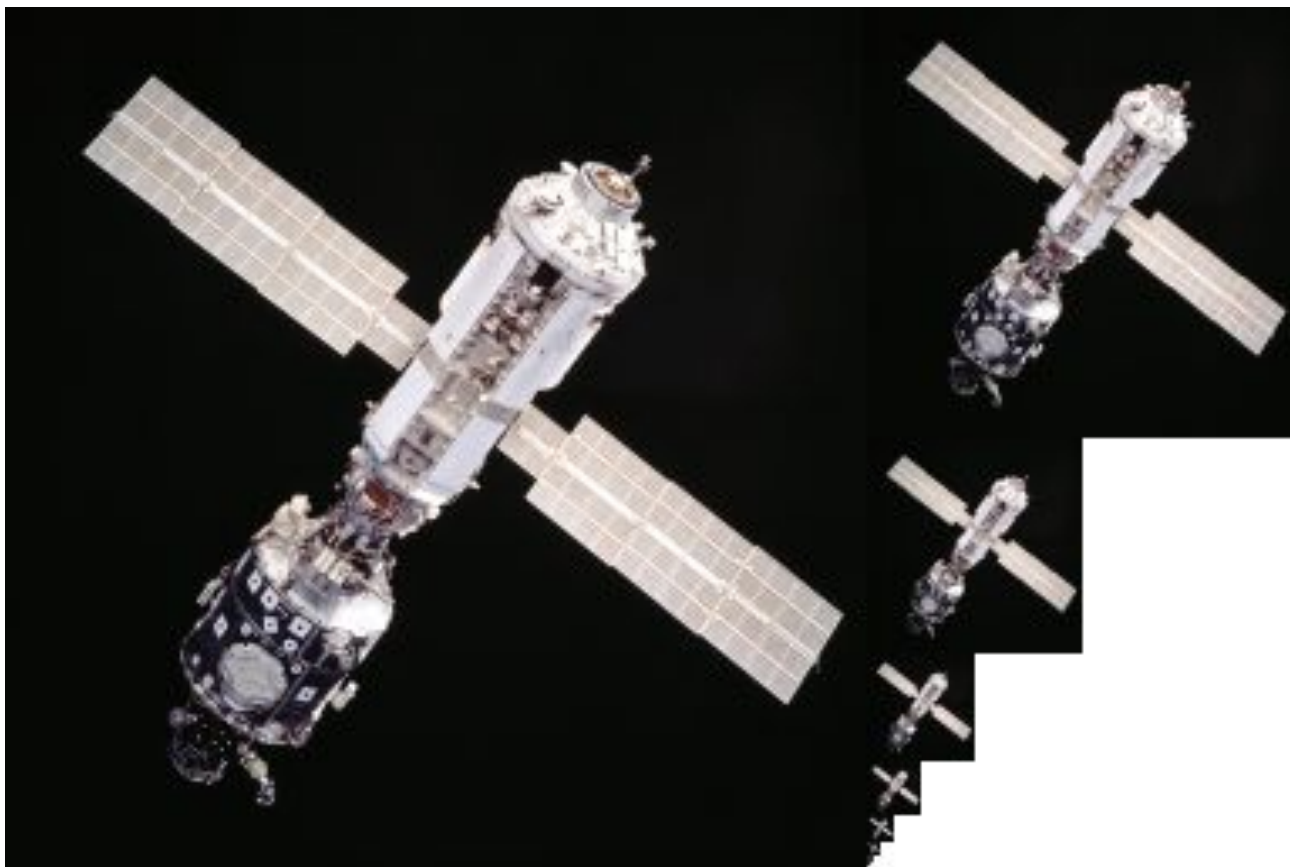
Speicherung Texturpyramide



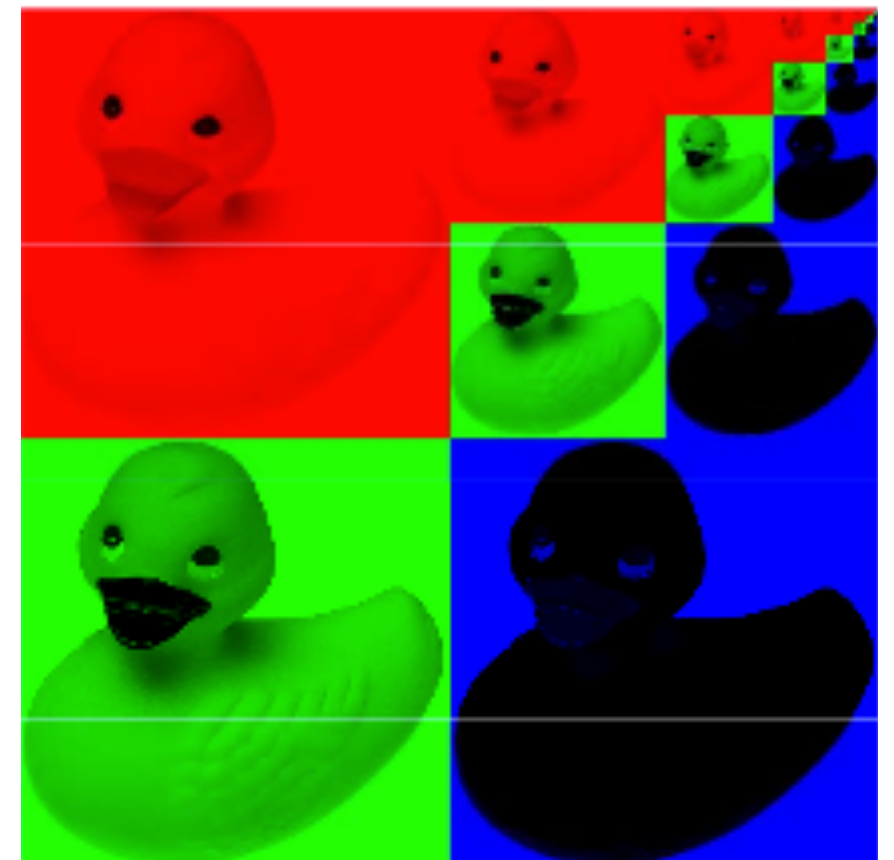
- Nach geometrischer Reihe $\sum_{i=0}^n q^i = \frac{q^{n+1} - 1}{q - 1}$ gilt:
$$\sum_{i=0}^{\infty} \left(\frac{1}{4}\right)^i = \frac{1}{1 - \frac{1}{4}} = \frac{4}{3} = 1 + \frac{1}{3}$$
- ➔ MIP-Maps haben höchstens um 1/3 höheren Speicherbedarf als das größte Bild

MIP-Mapping

Speicherung Texturpyramide



VS



- Effiziente Speicherung durch volle Ausnutzung der Textur mit 1 Byte (Graustufen) statt 3 Byte pro Pixel



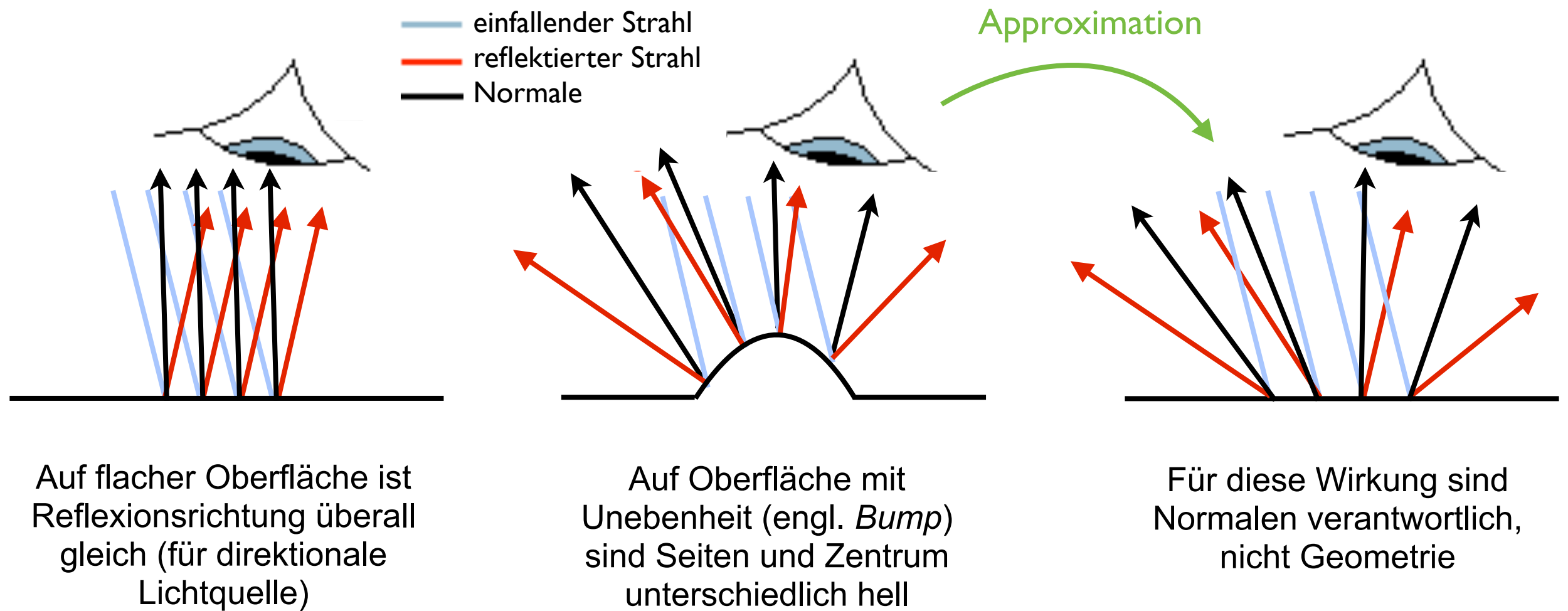
Interaktive Computergrafik

Lektion 12

Bump Mapping

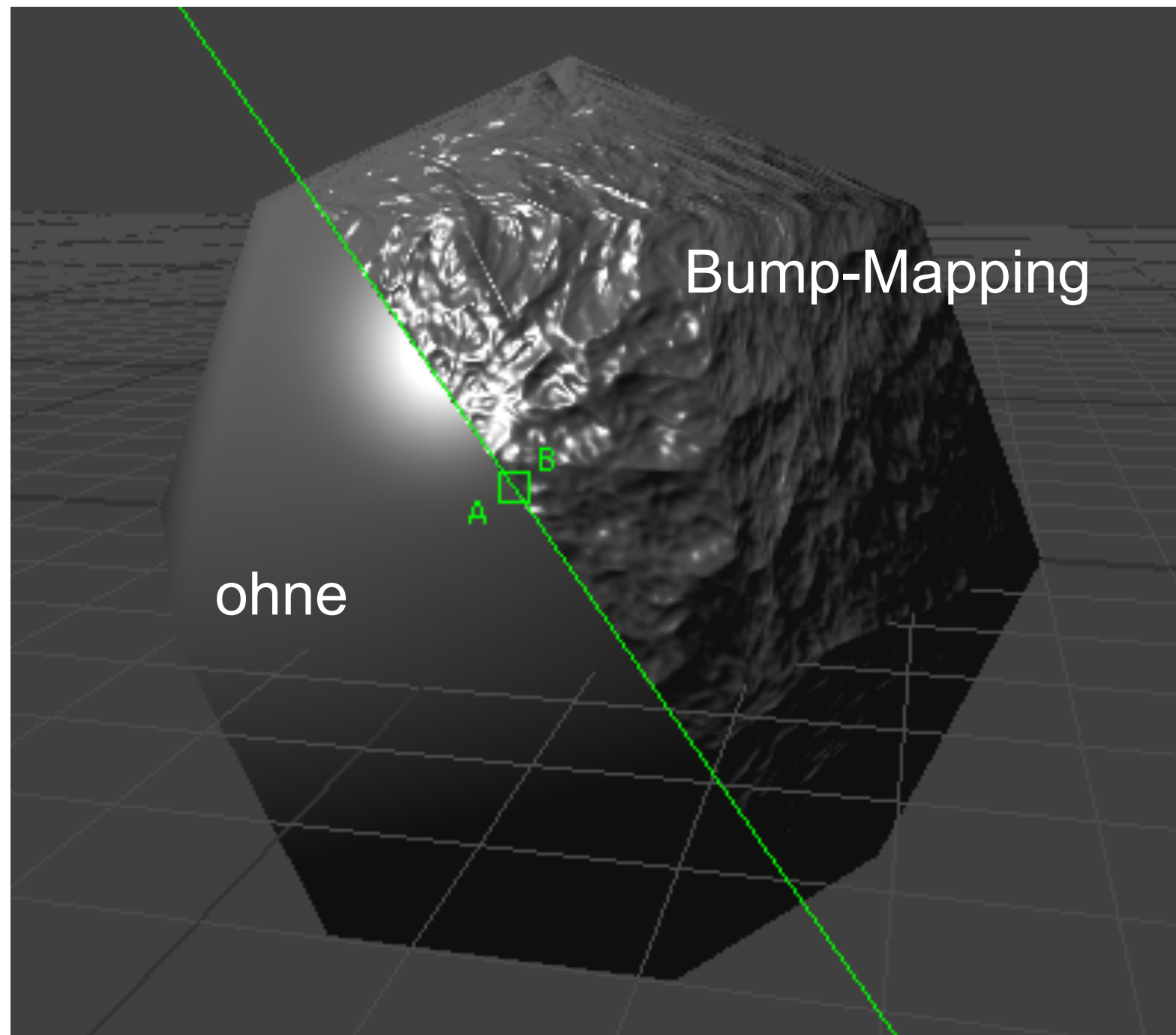
Bump-Mapping

Phong-Beleuchtung



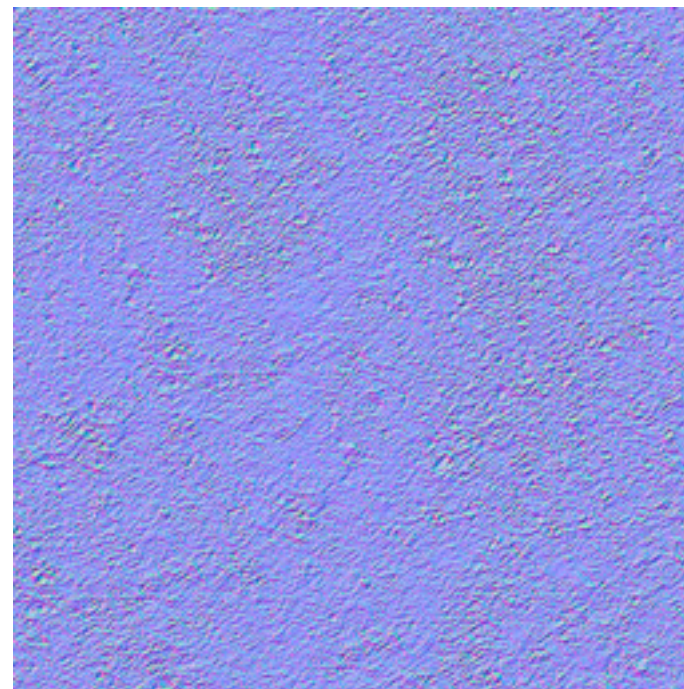
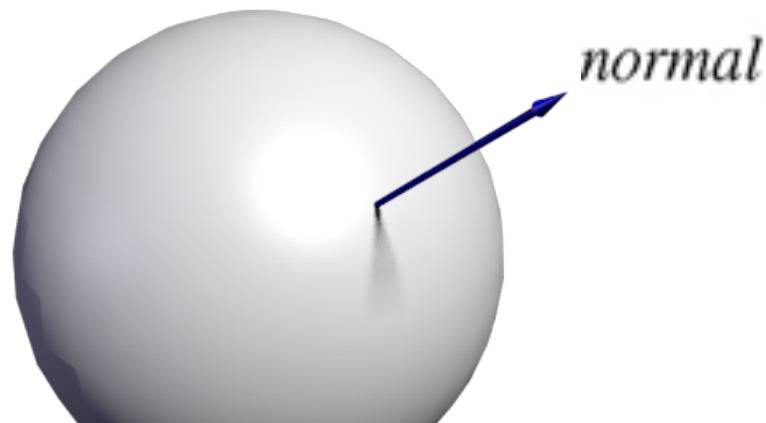
Bump-Mapping

Beispiele



Normal-Map

- Texel speichern (x,y,z)-Werte der Normale in RGB-Kanälen einer 2D-Textur



Normal-Map

Normal-Map

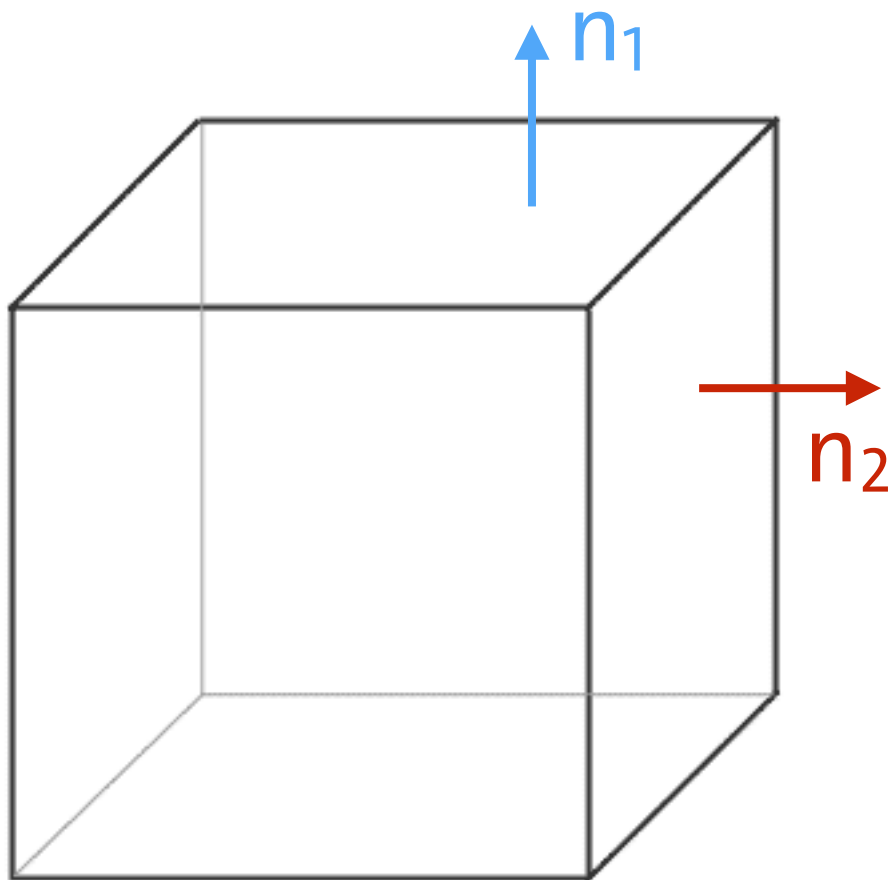
- Texel speichern (x,y,z)-Werte der Normale in RGB-Kanälen einer 2D-Textur
- Vorsicht: $x,y,z \in [-1..1]$ und $r,g,b \in [0..1]$
→ Konvertierung notwendig

Color.rgb = Normal.xyz / 2.0 + 0.5 bzw.

*Normal.xyz = Color.rgb * 2.0 - 1.0*

Normal-Map

- Normalen gespeichert in **Tangentenraum**
- Beispiel:



$$n_{1\text{welt}} = (0, 1, 0)$$

\neq

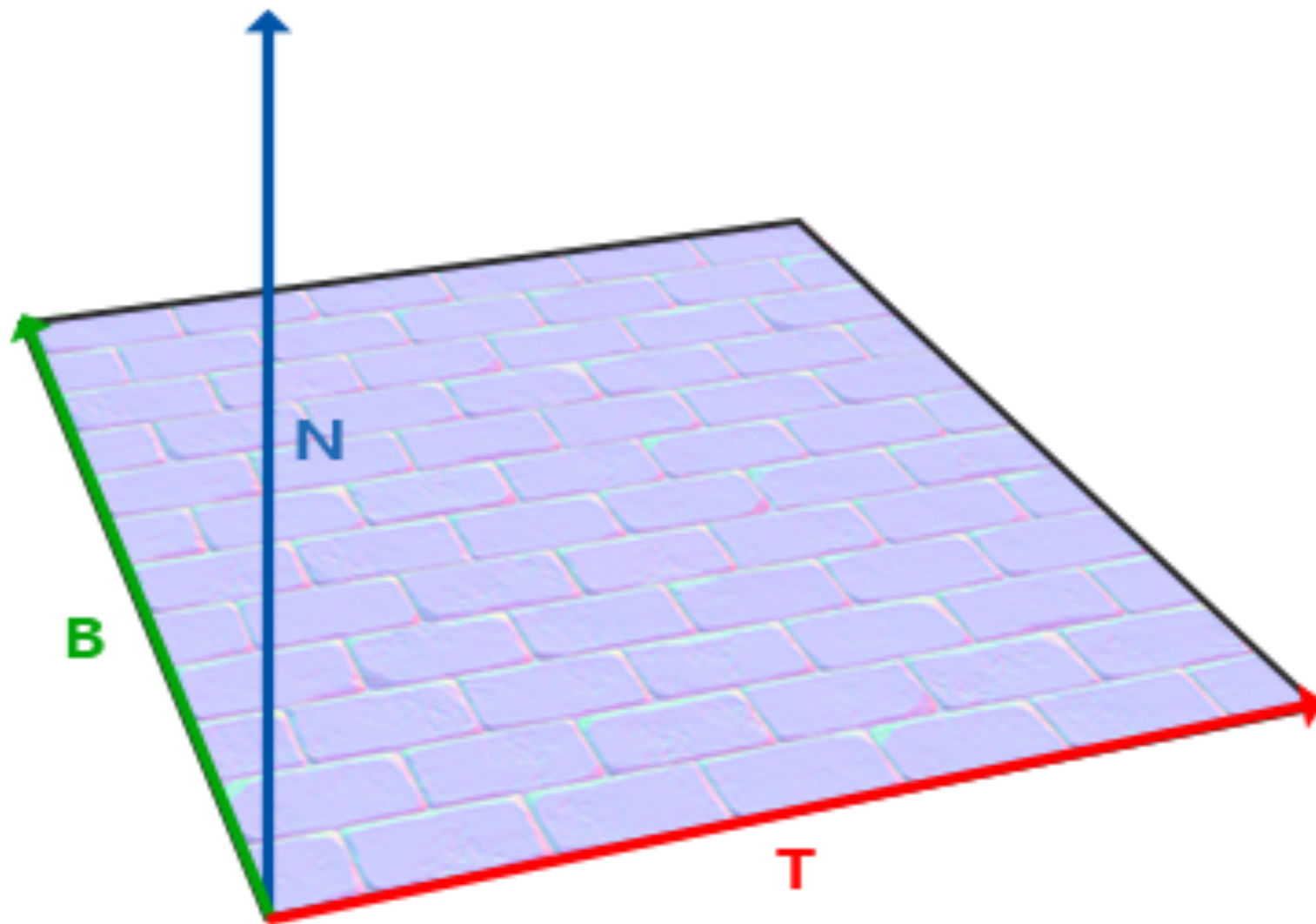
$$n_{2\text{welt}} = (1, 0, 0)$$

$$n_{1\text{tangente}} = (0, 0, 1)$$

$=$

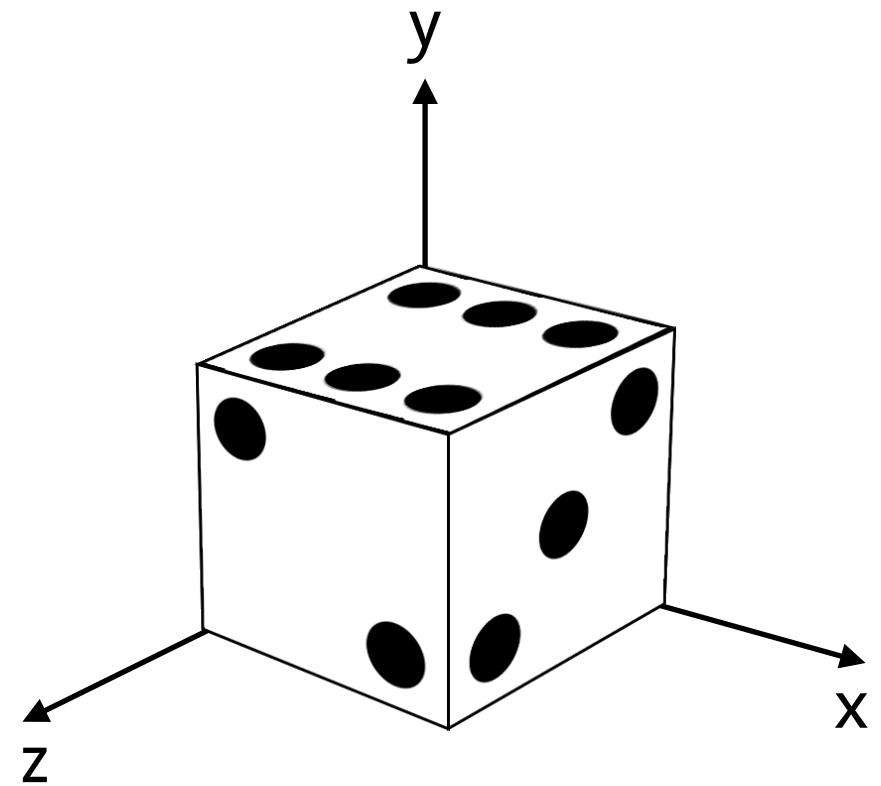
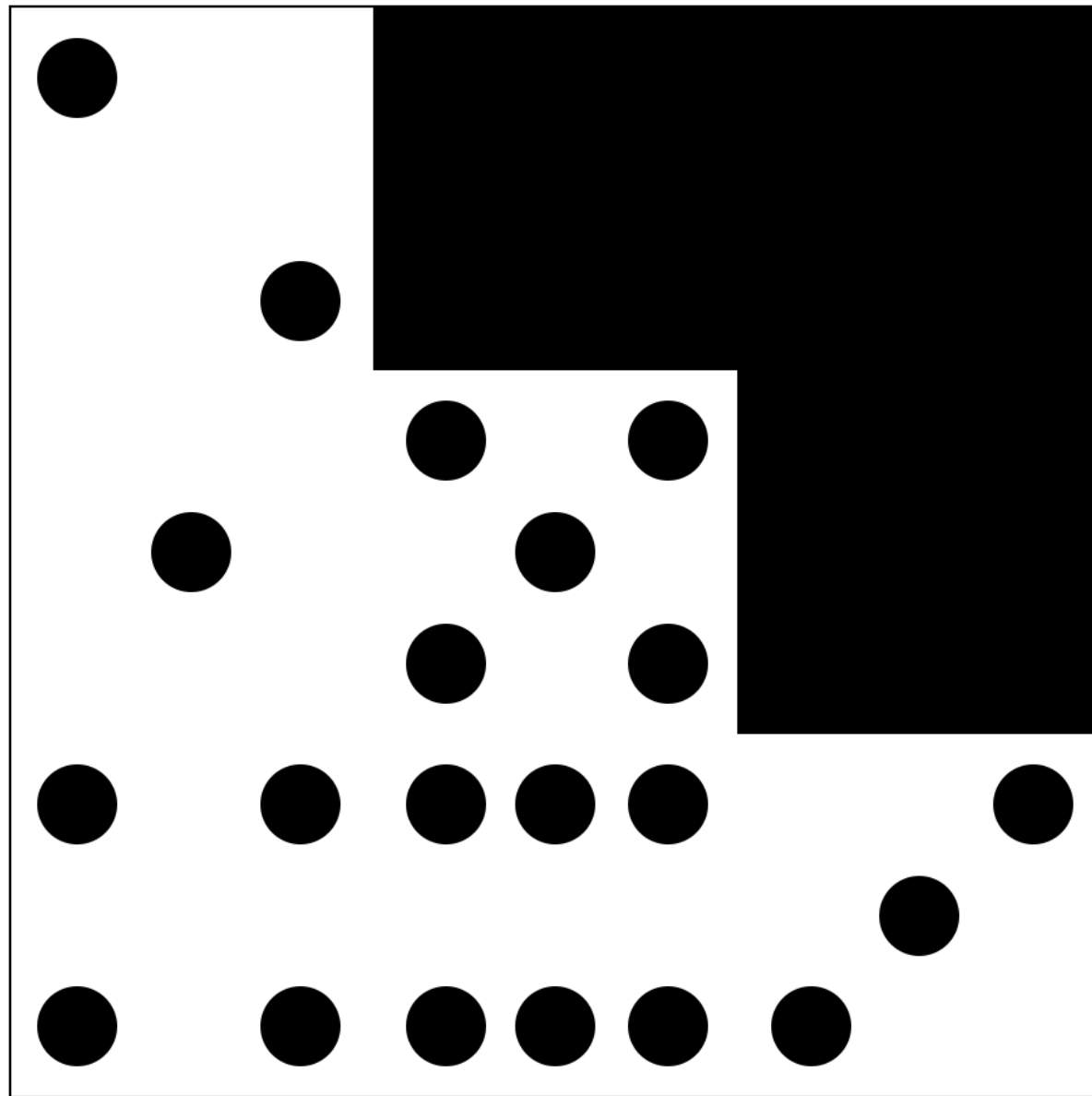
$$n_{2\text{tangente}} = (0, 0, 1)$$

Tangentenraum



Beispiel

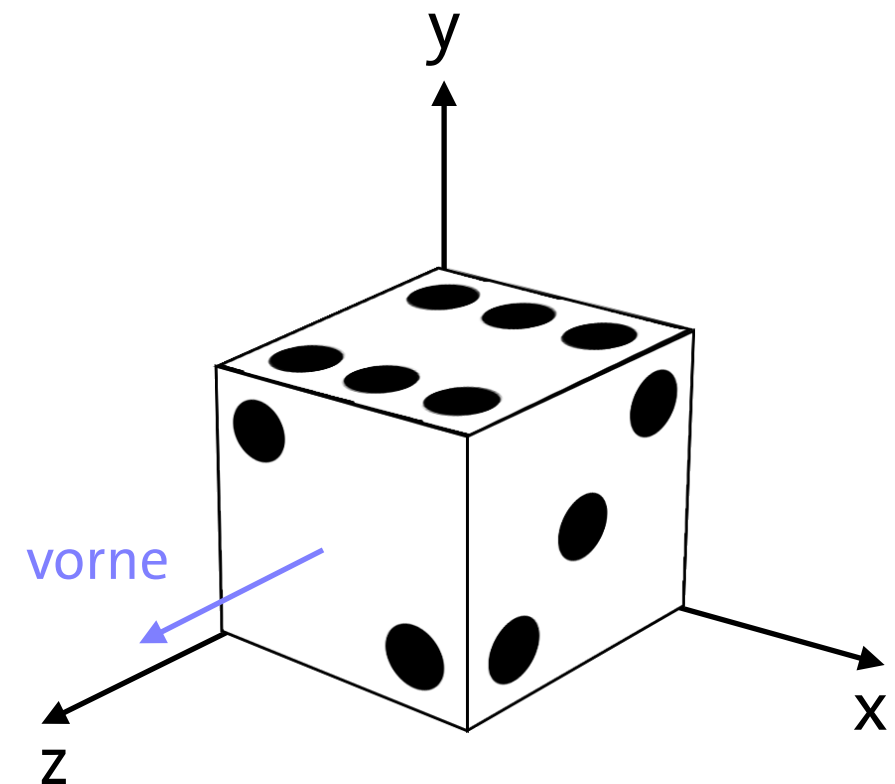
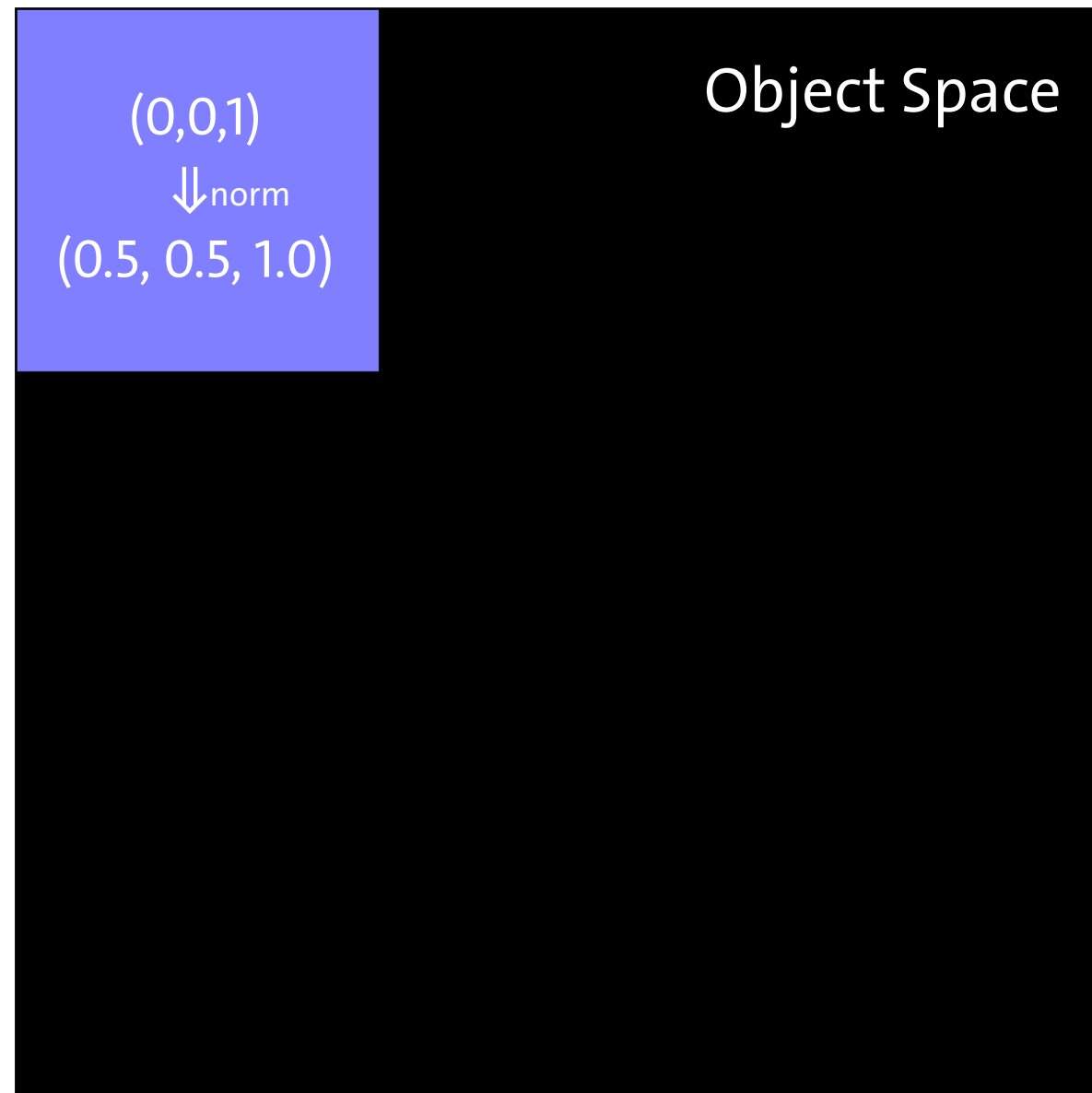
Diffuse Map



Object Space

Beispiel

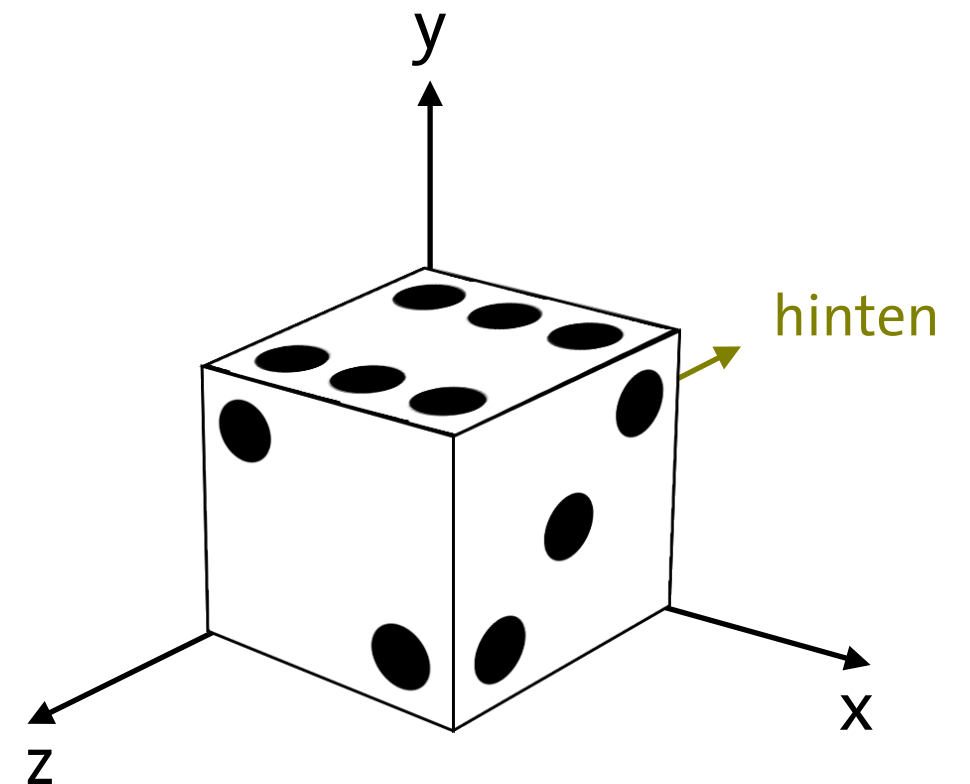
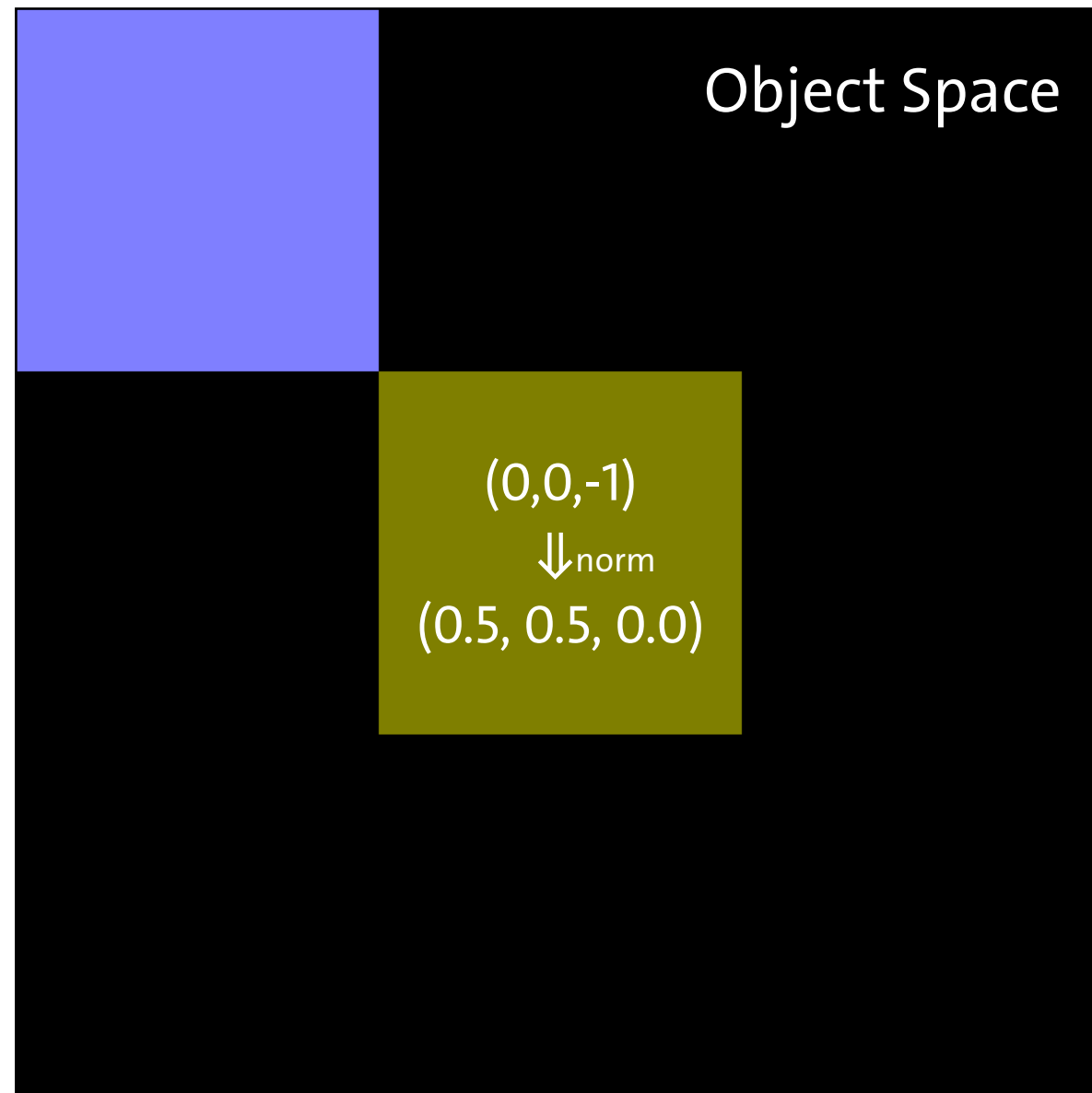
Normal Map - Object Space



Object Space

Beispiel

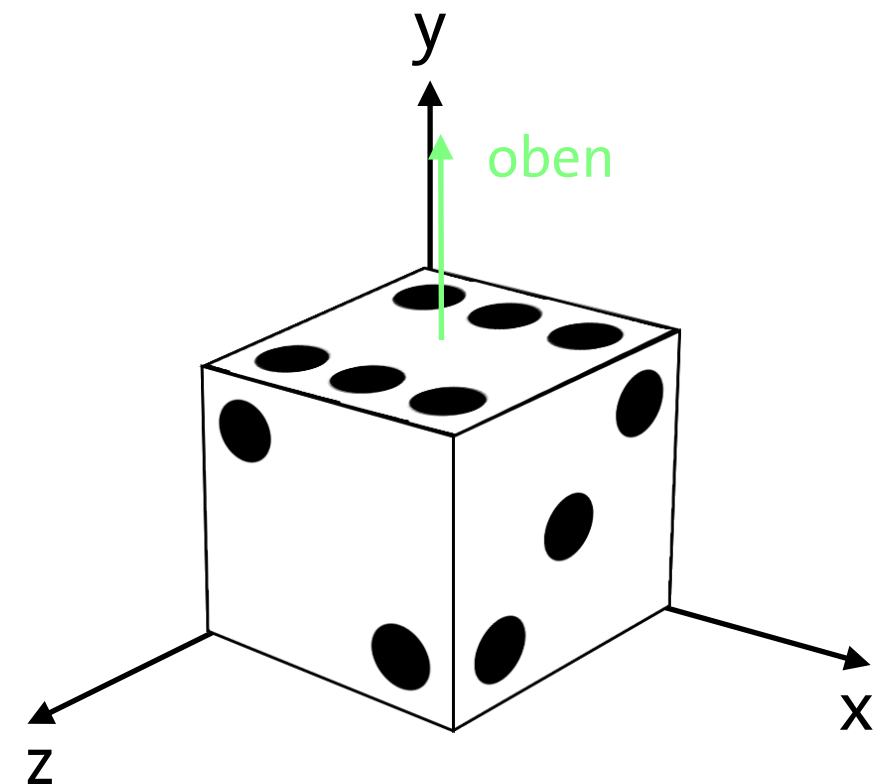
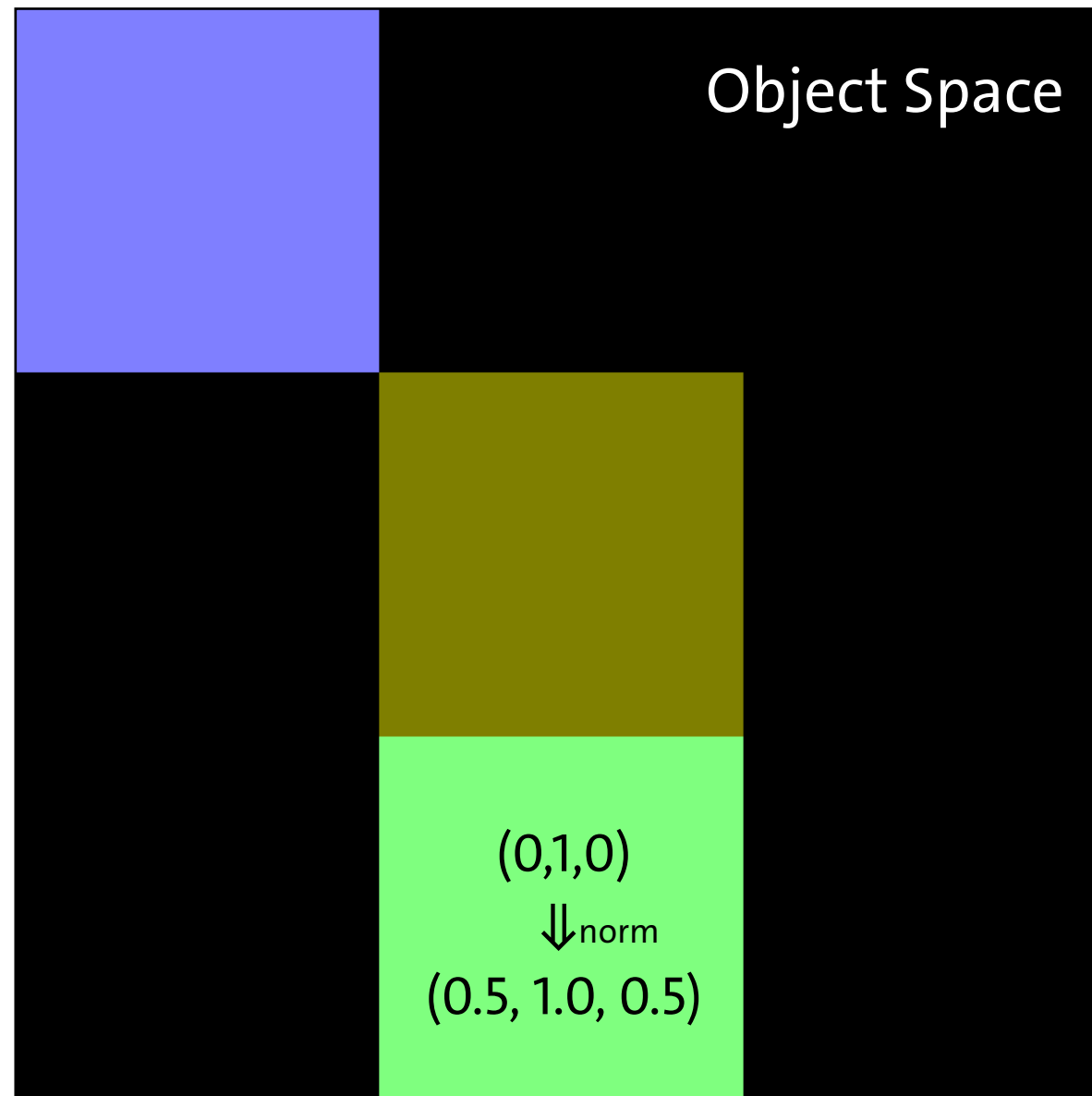
Normal Map - Object Space



Object Space

Beispiel

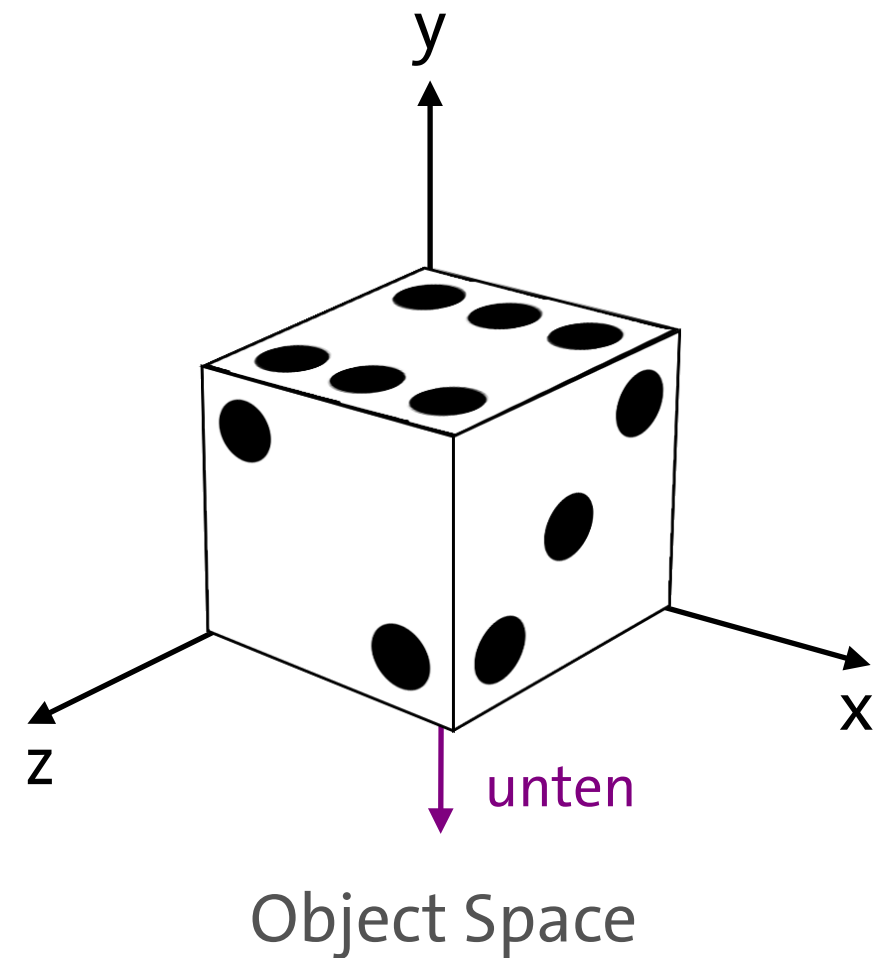
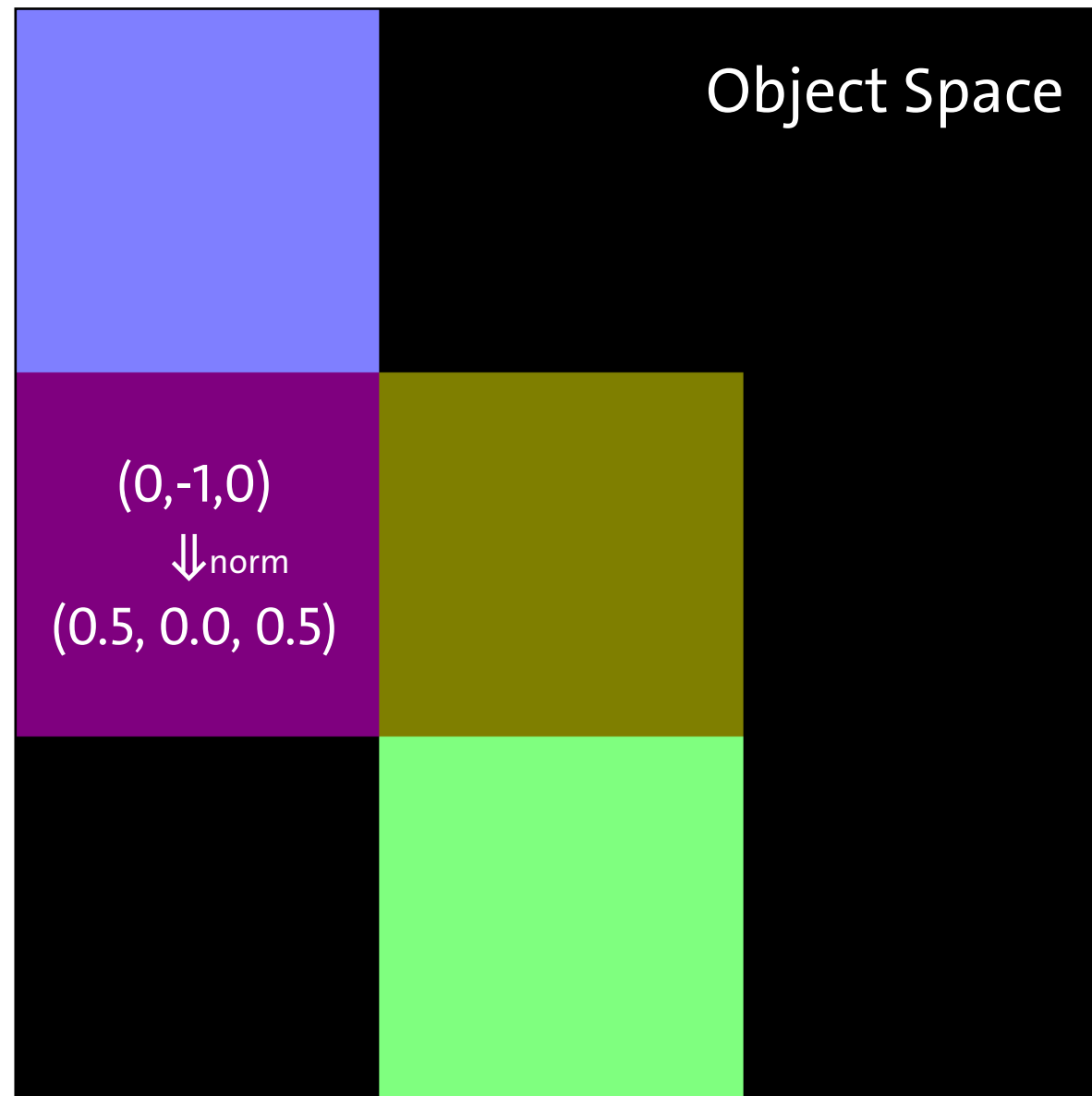
Normal Map - Object Space



Object Space

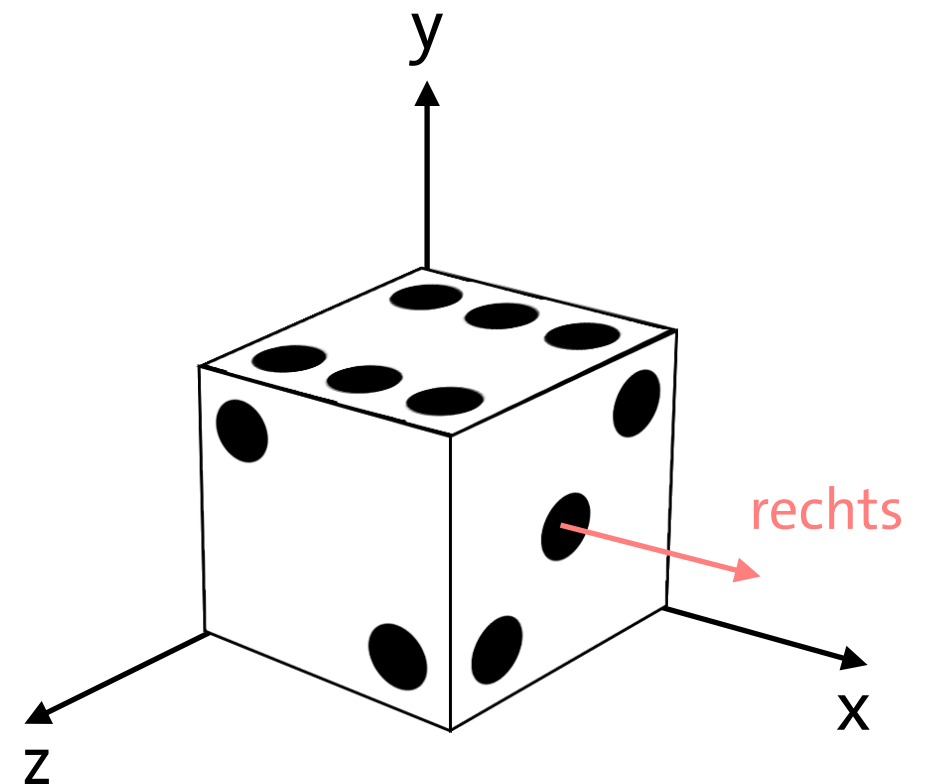
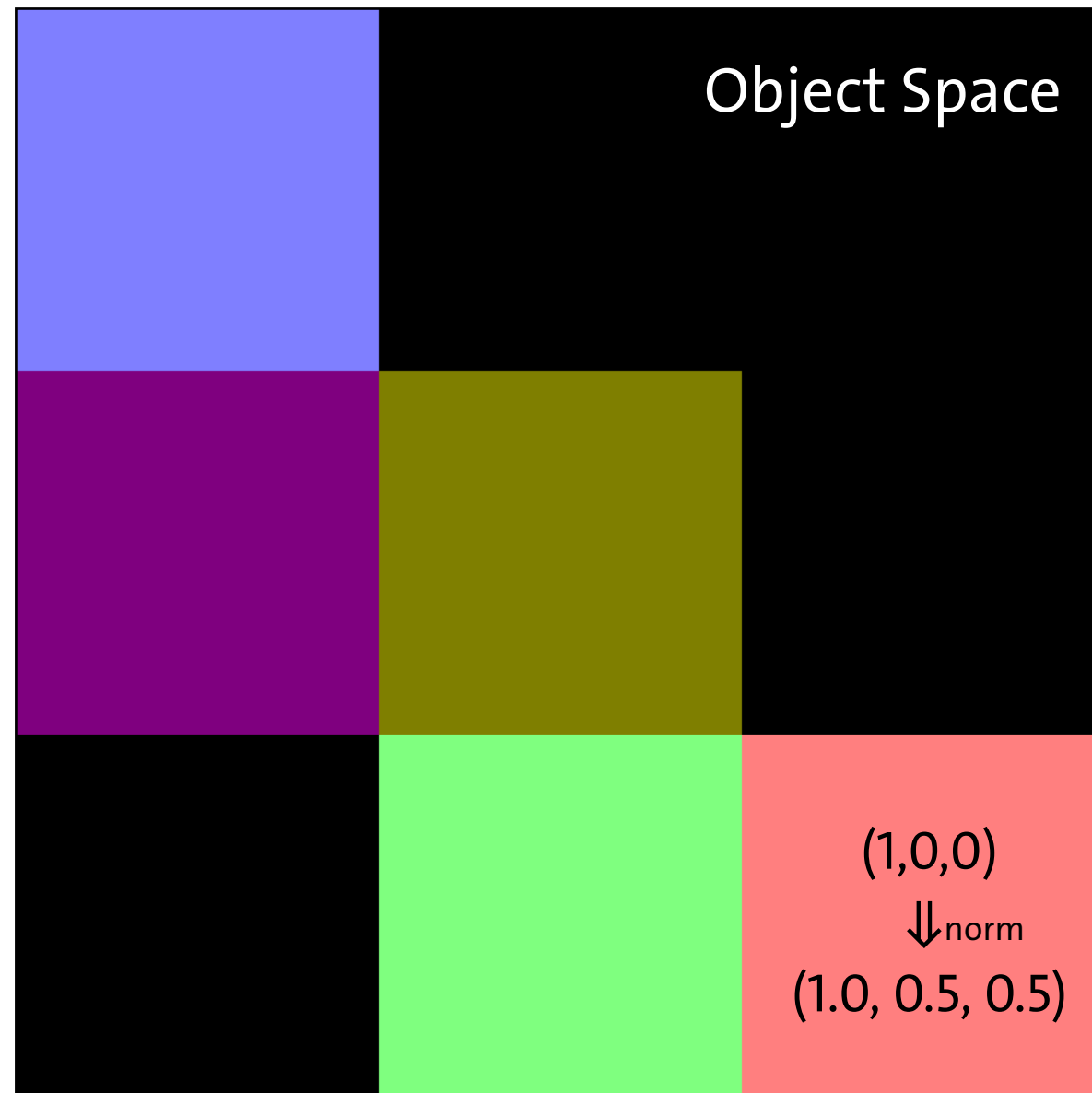
Beispiel

Normal Map - Object Space



Beispiel

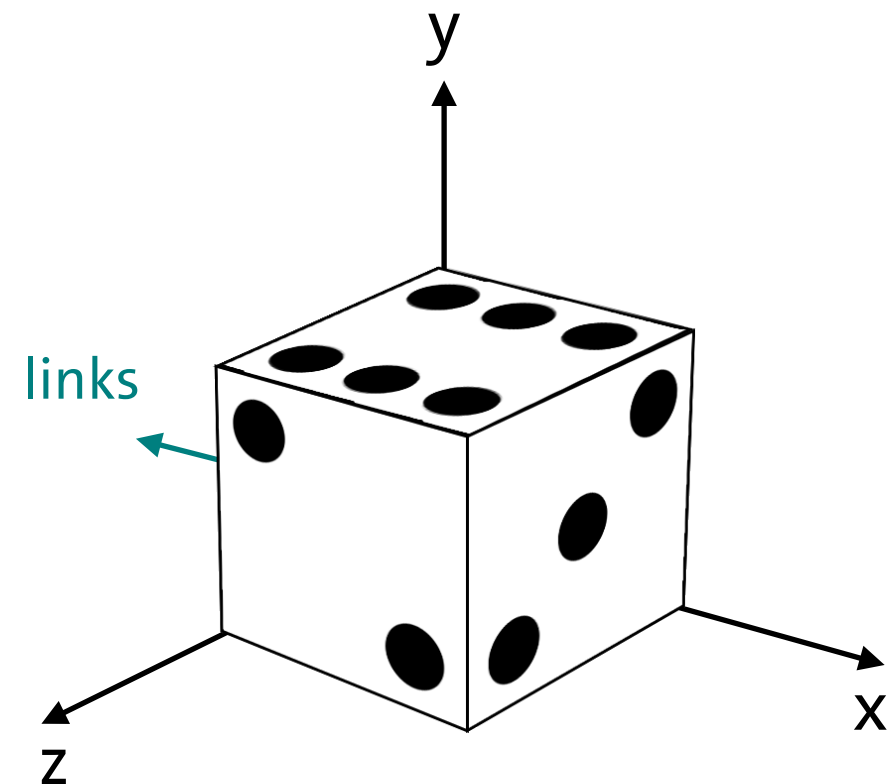
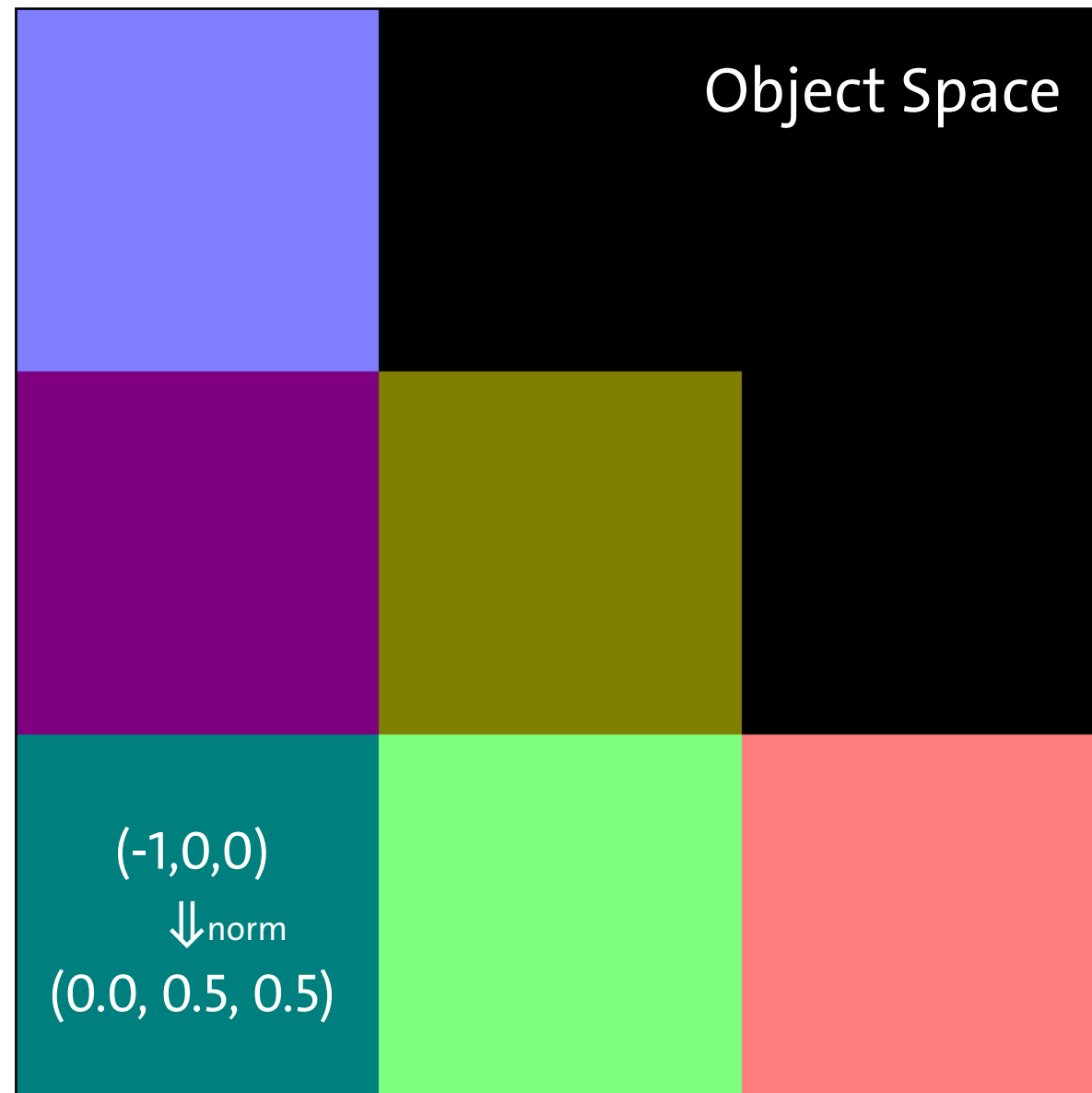
Normal Map - Object Space



Object Space

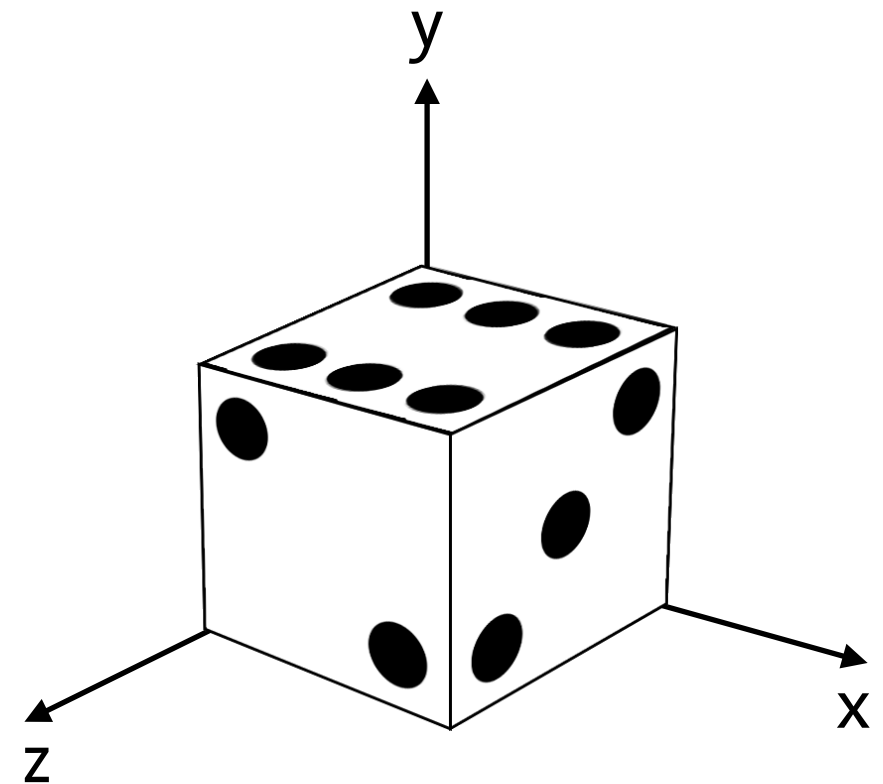
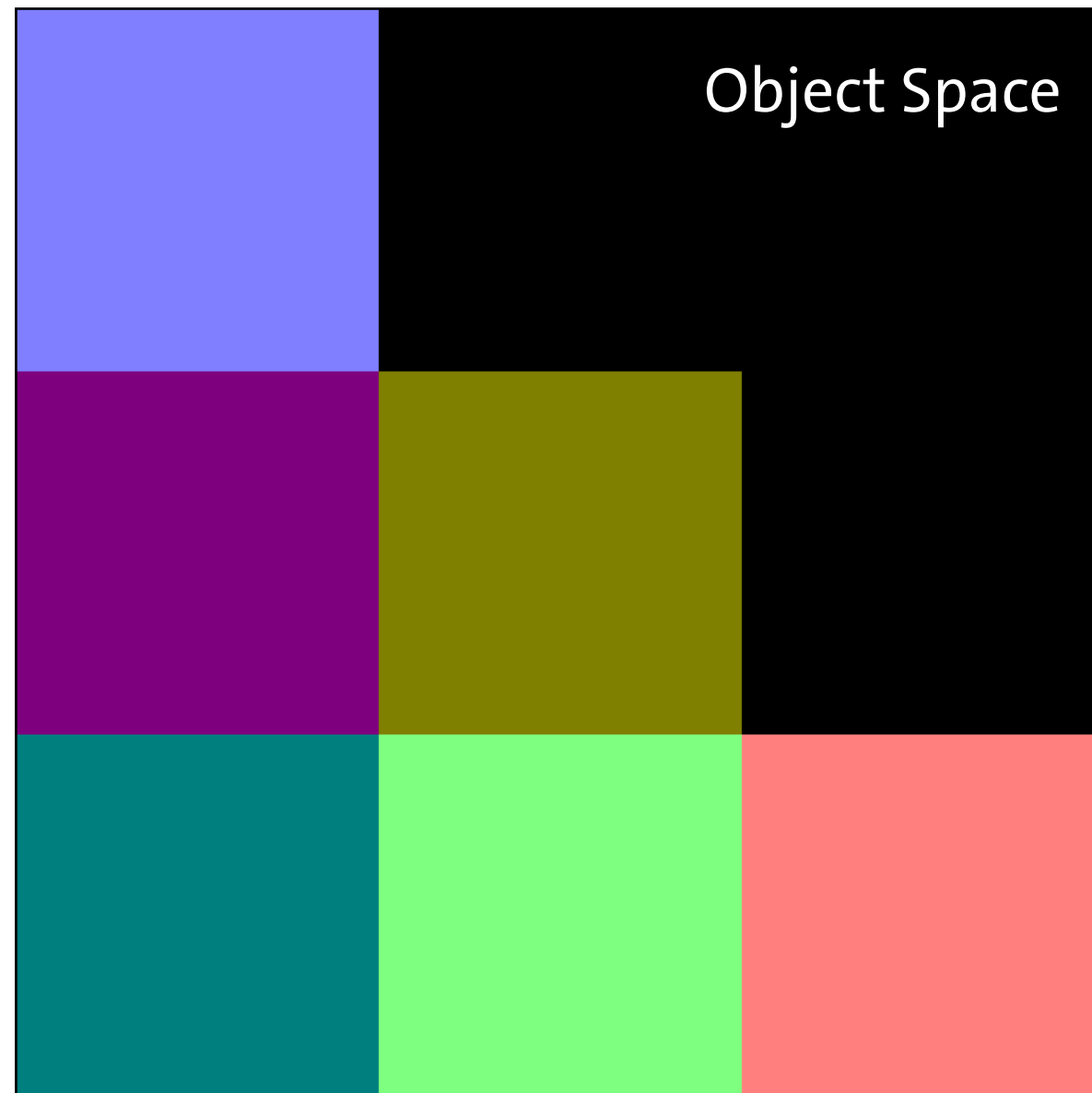
Beispiel

Normal Map - Object Space



Beispiel

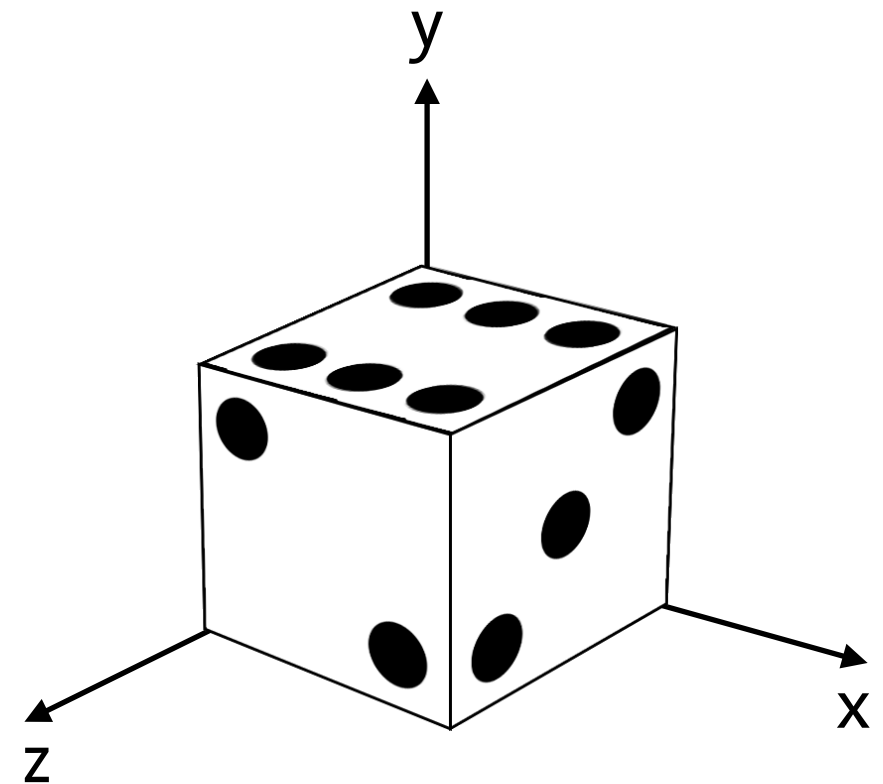
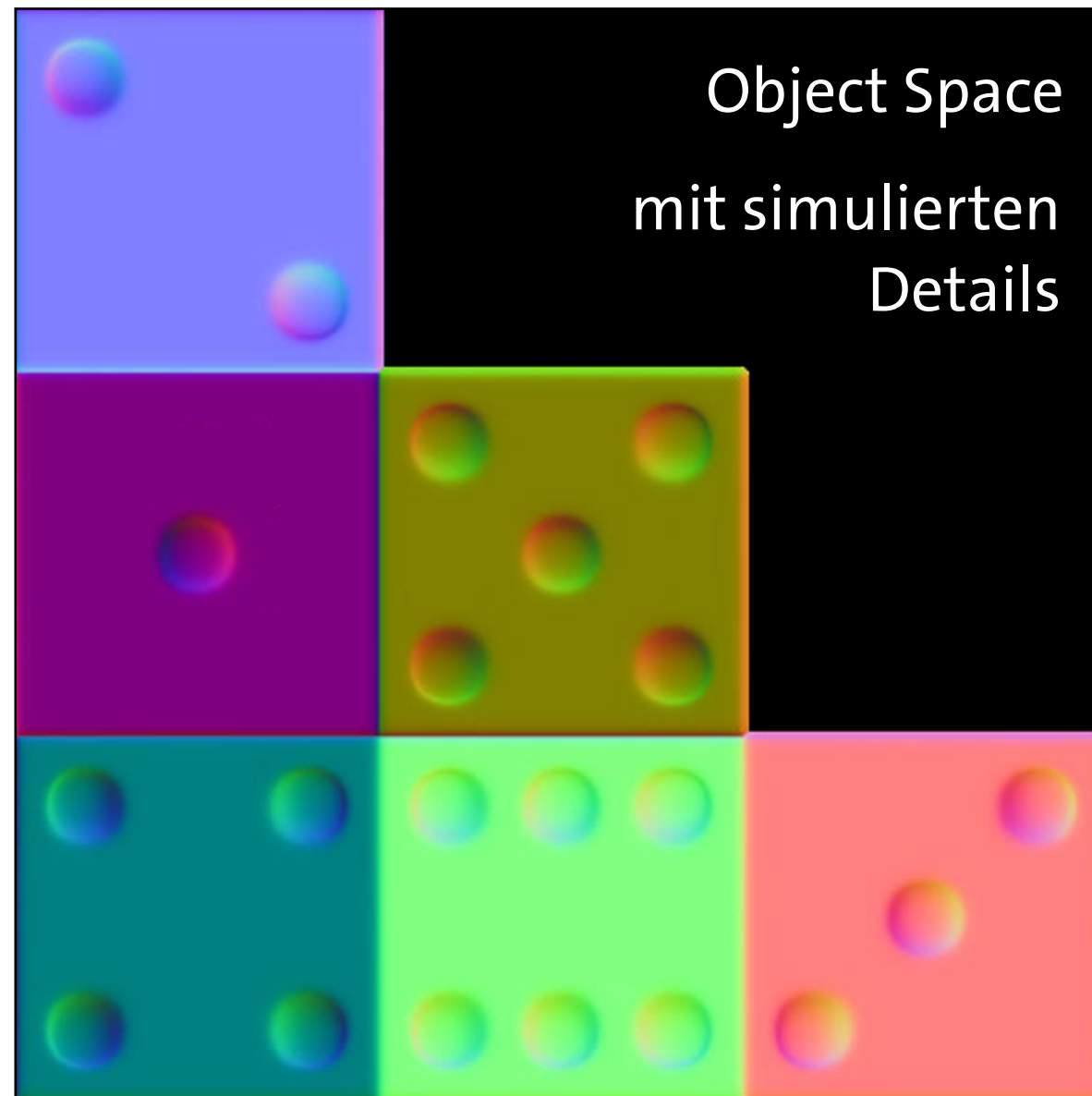
Normal Map - Object Space



Object Space

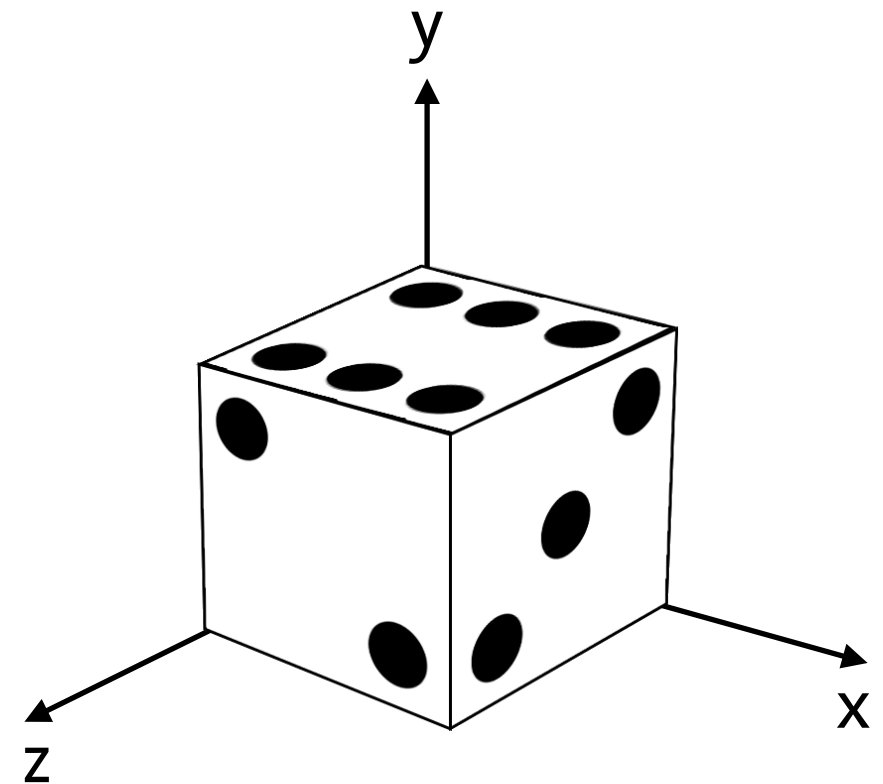
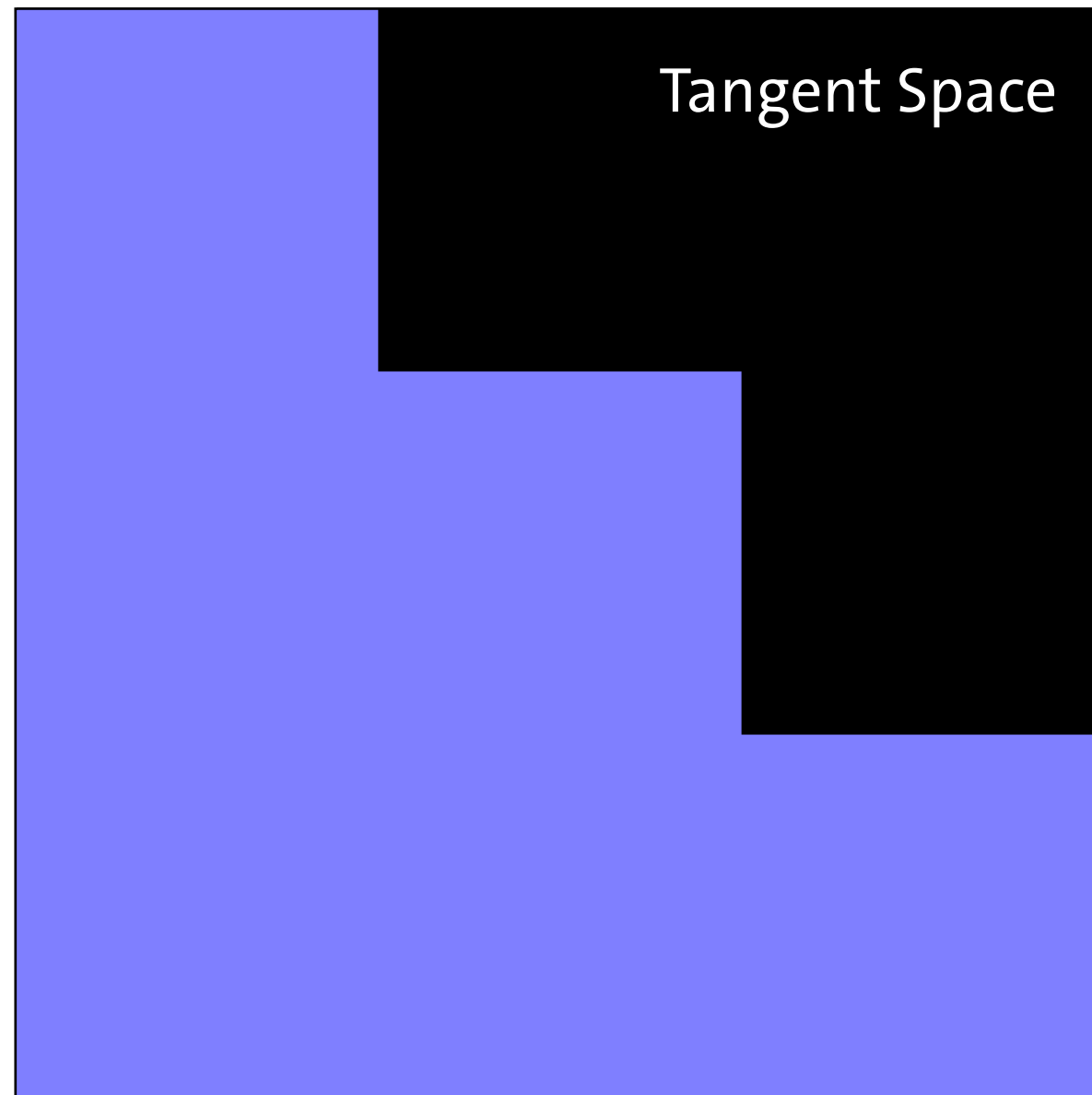
Beispiel

Normal Map - Object Space



Beispiel

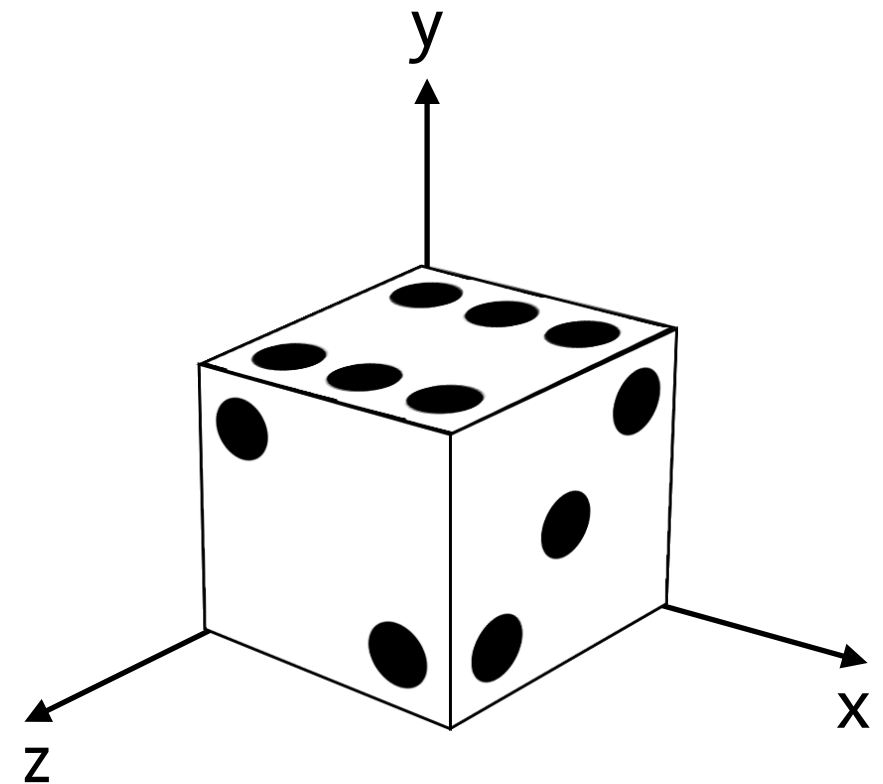
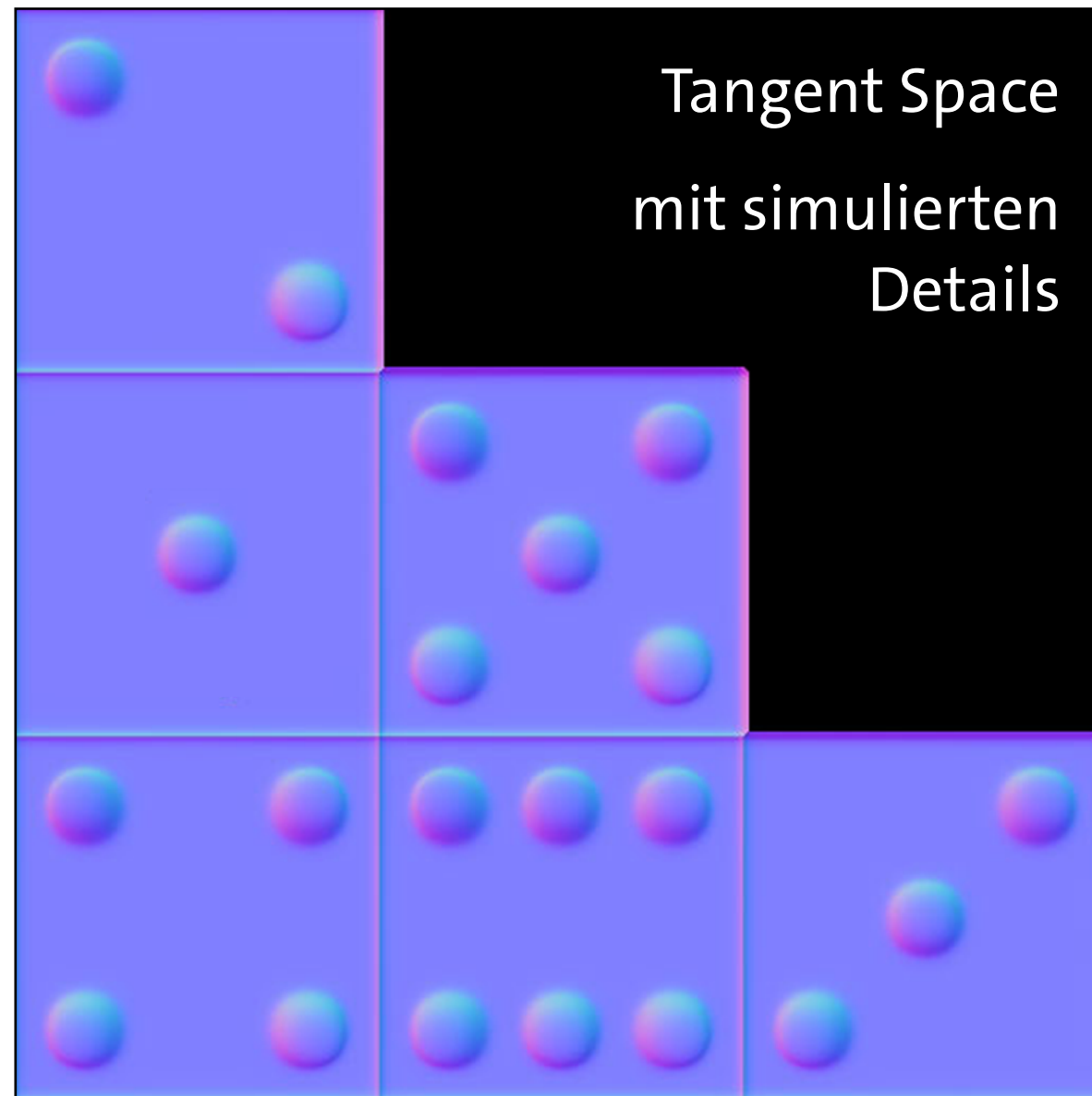
Normal Map - Tangent Space



Object Space

Beispiel

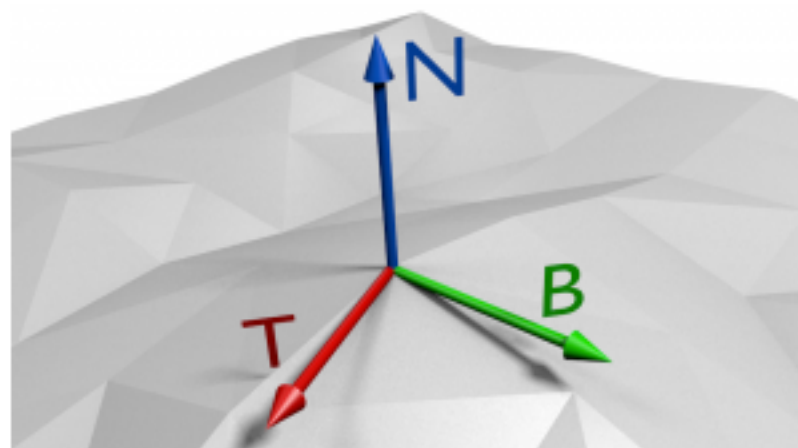
Normal Map - Tangent Space



Object Space

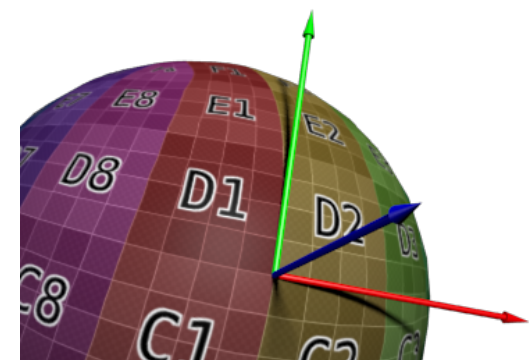
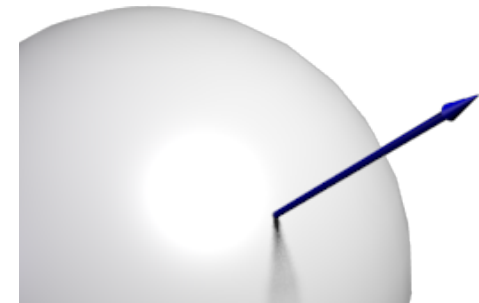
Normal-Map

- Normalen gespeichert in **Tangentenraum**
 - Normale $(0,0,1)$ entspricht Polygonnormale
 - Abbildung von Tangentenraum nach Welt-/Kamerakoordinaten durch **TBN-Matrix** (Tangente, Binormale, Normale)



Normal-Map

- Normalen N sind die vom Anwendungsprogramm übergebenen Normalen
- Tangenten T liegen in der Ebene senkrecht zur Normalen
- Binormalen B werden berechnet durch Kreuzprodukt $N \times T$



Normal Map

Berechnung der TBN-Matrix

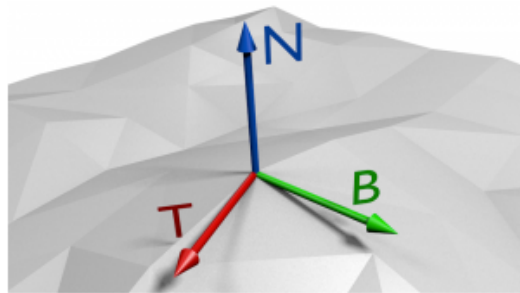
- Berechnung im Fragment Shader:

```
in vec4 normalCam;  
...  
vec3 n = normalize(normalCam.xyz);  
vec3 t = normalize(cross(n, vec3(0.0, 0.0, 1.0)));  
vec3 b = cross(n, t);  
...
```

⇒ praktische Anwendung in Übung

Normal Map

Anwendung der TBN-Matrix



$$TBN = \begin{pmatrix} T_x & B_x & N_x \\ T_y & B_y & N_y \\ T_z & B_z & N_z \end{pmatrix}$$

2 Möglichkeiten:

- Überführung der Normalen von Tangentenraum in Welt- bzw. Kamerakoordinaten: $N_{cam} = TBN \cdot N_{tangent}$
- Überführung aller anderer Vektoren (L und V) in Tangentenraum durch Multiplikation mit TBN^{-1}

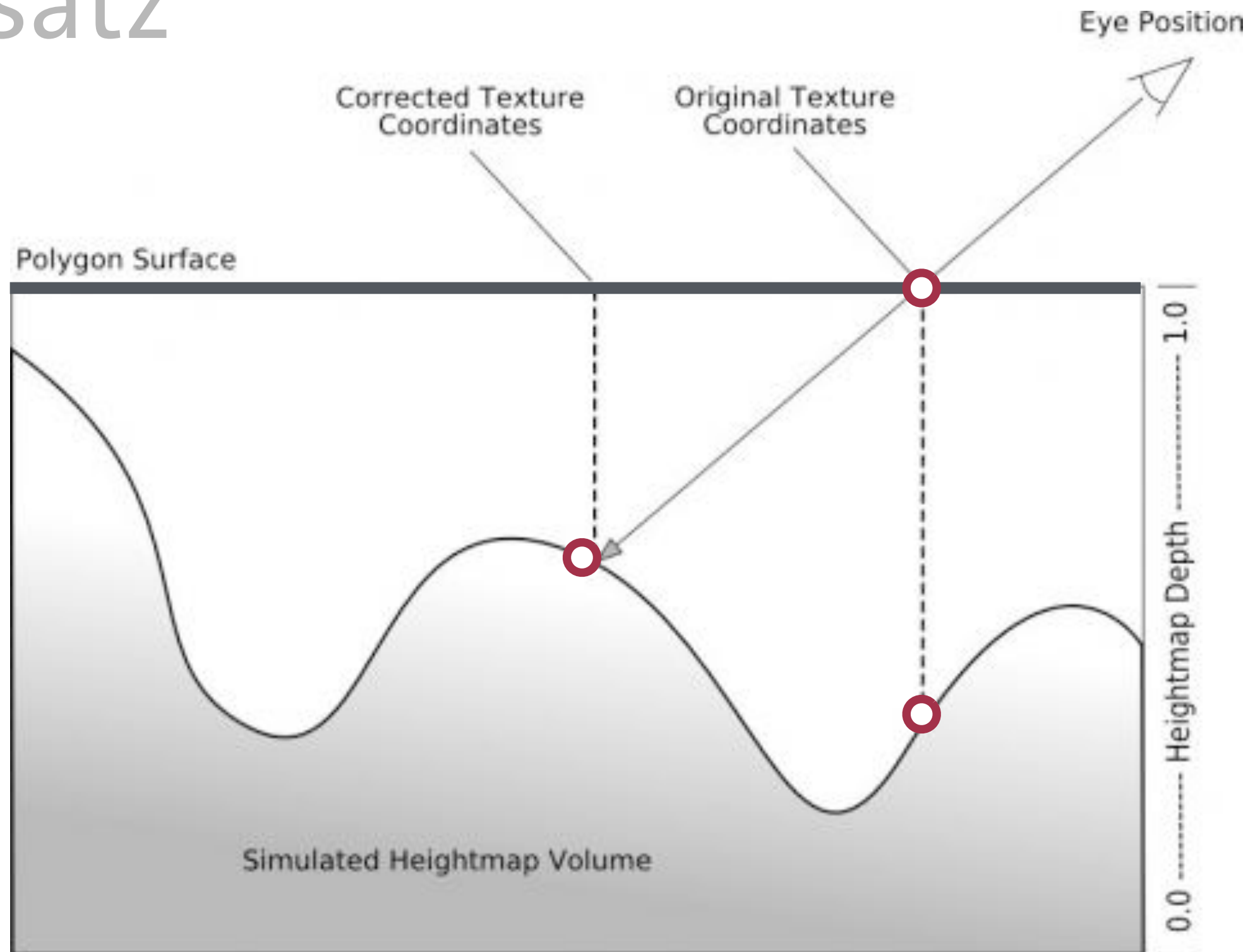


Interaktive Computergrafik

Lektion 12

Parallax Occlusion Mapping

Parallax-Occlusion-Mapping Ansatz



Diskussion



Vergleichen Sie Bump Mapping und Parallax Occlusion Mapping.

Vergleich

Bump vs. Parallax Occlusion

	Bump Mapping	Parallax Occlusion Mapping
Erfordert zusätzliche Textur (neben Diffuse Map)	✓	✓
Simuliert Oberflächenstruktur	✓	✓
Unterstützt Selbstokklusion	✗	✓
Verändert die Objekt-Silhouette	✗	✗
Unterstützt dynamische Objekte / Lichtquellen	✓	✓

