Interaktive Computergrafik

Prof. Dr. Frank Steinicke
Human-Computer Interaction
Fachbereich Informatik
Universität Hamburg

Interaktive Computergrafik

Kapitel Kameras

Prof. Dr. Frank Steinicke
Human-Computer Interaction, Universität Hamburg

Inhalt

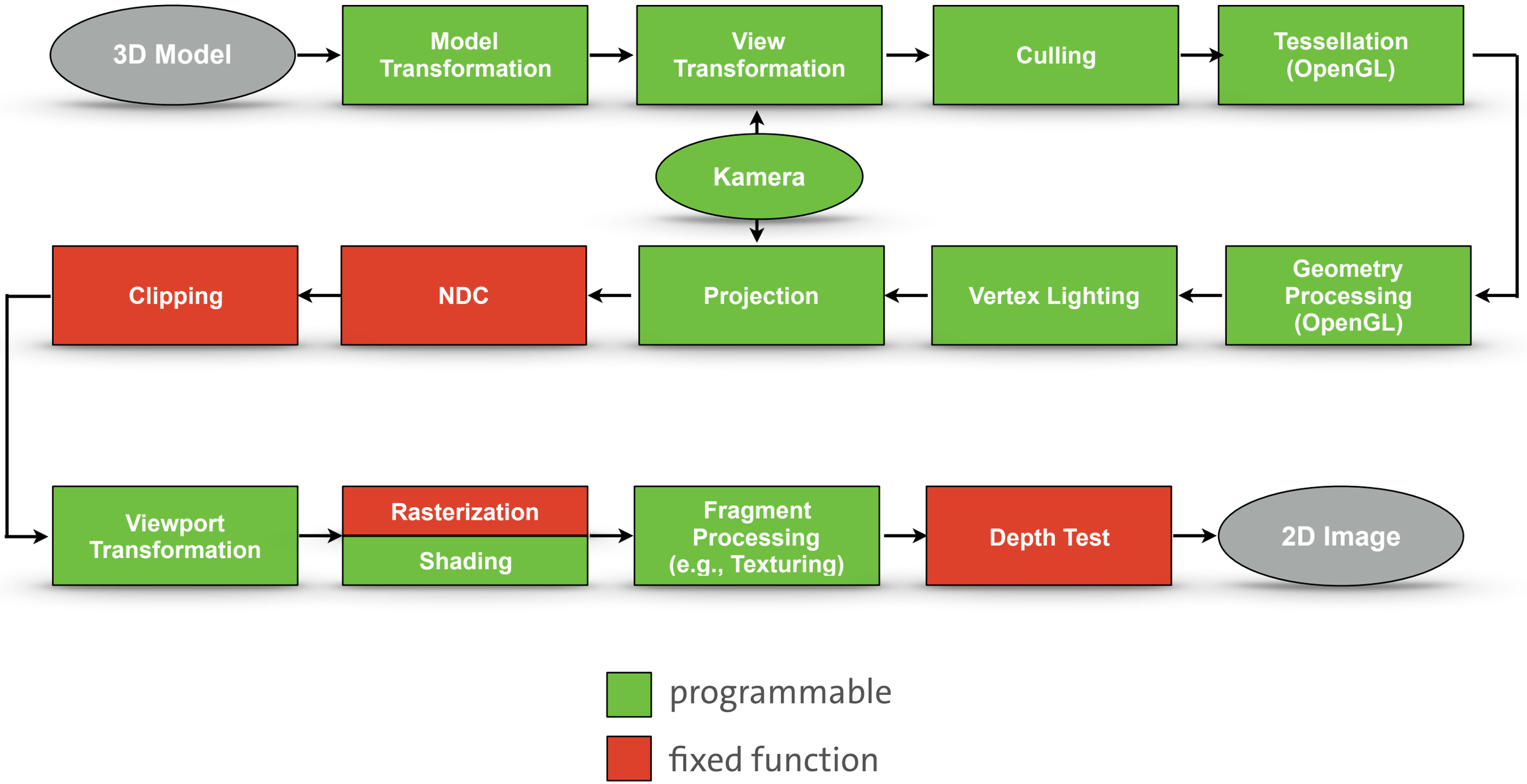
- Kamerakoordinaten
- Euler-Winkel vs. Quaternionen
- View-Matrix
- Sichtbarkeitsermittlung

Interaktive Computergrafik

Kapitel Kameras

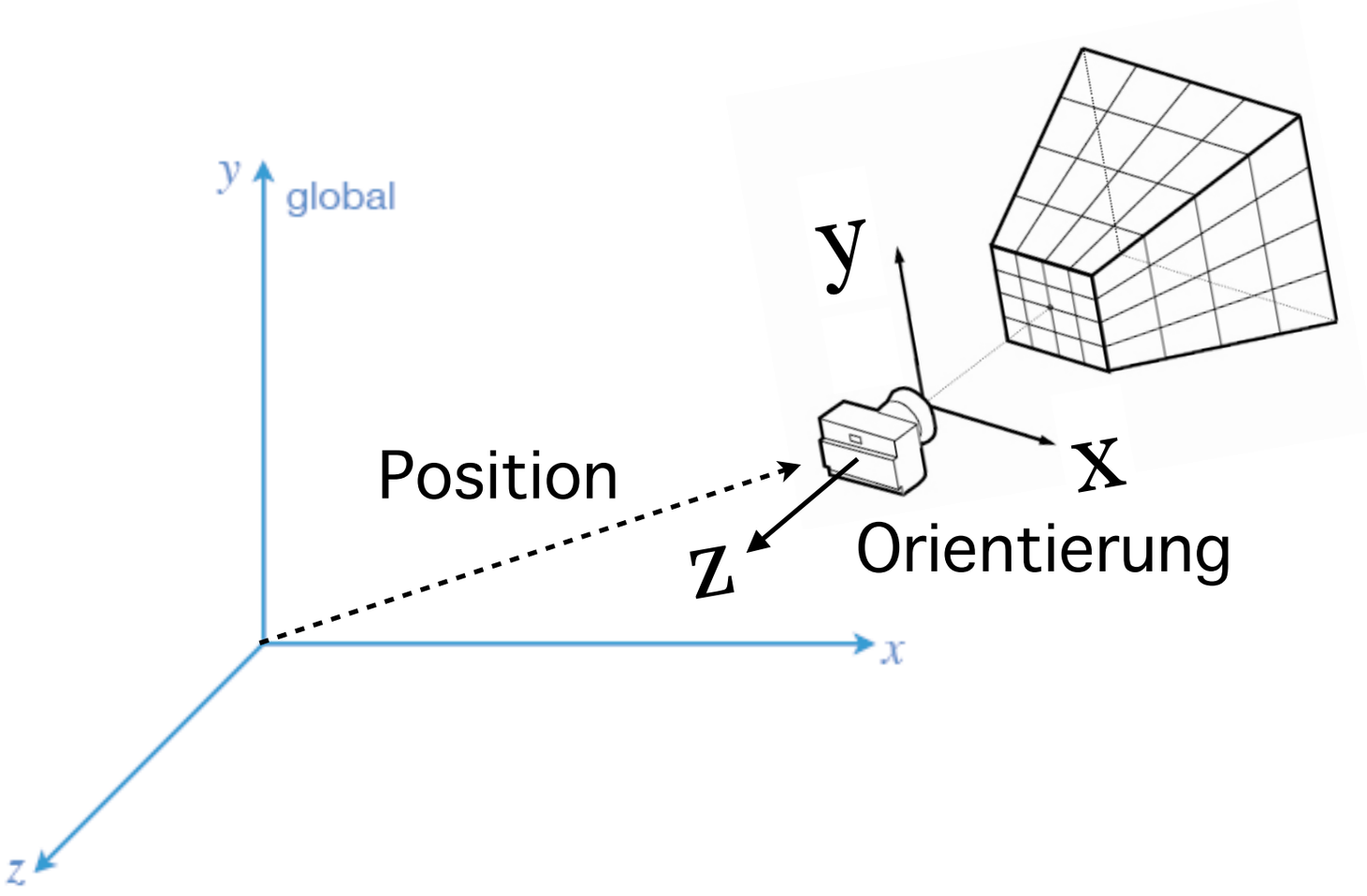
Kamerakoordinaten

3D Rendering Pipeline



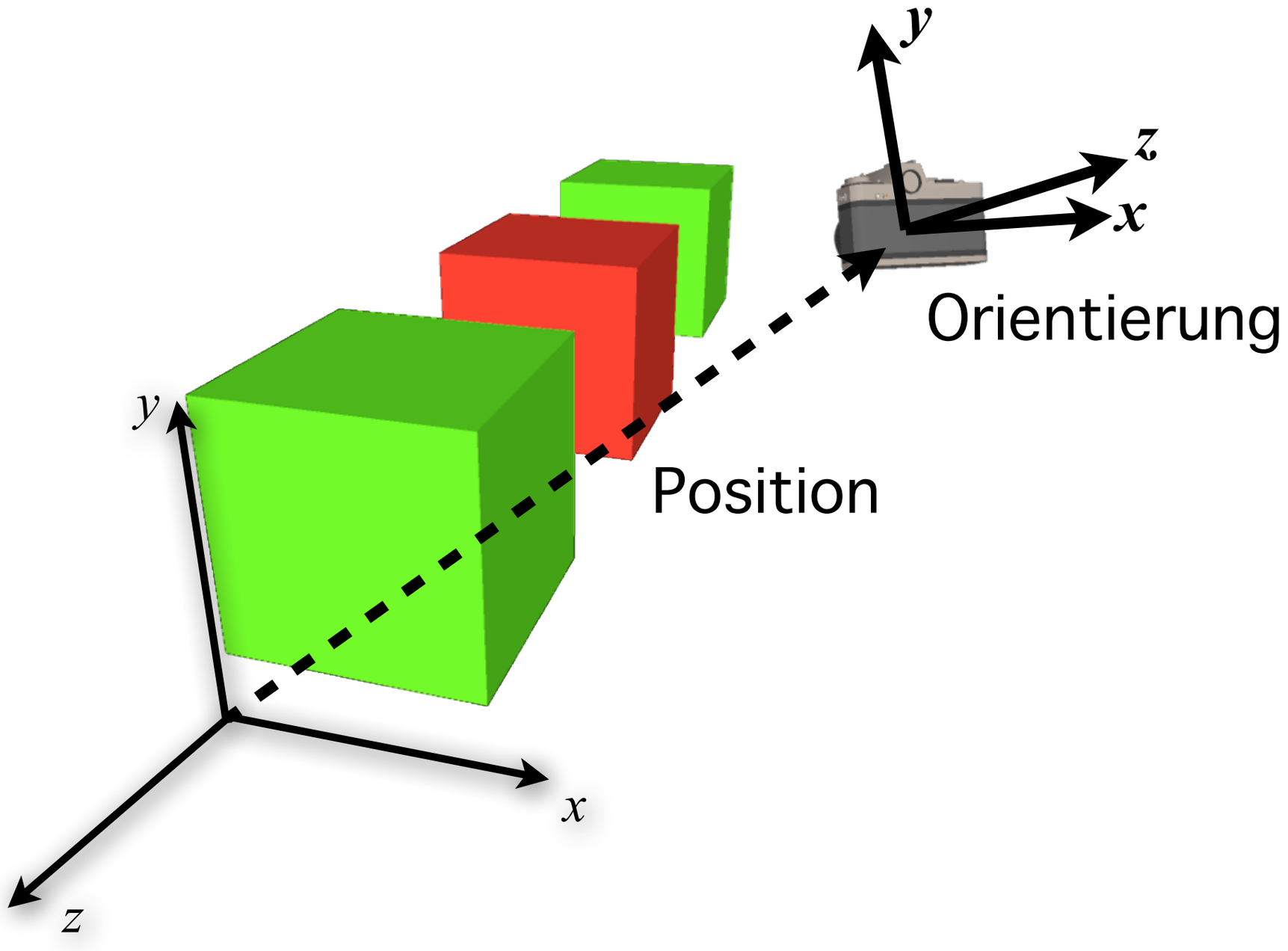
Kamera

- **Position** und **Orientierung** der Kamera in Weltkoordinaten spezifizieren sichtbaren Bereich



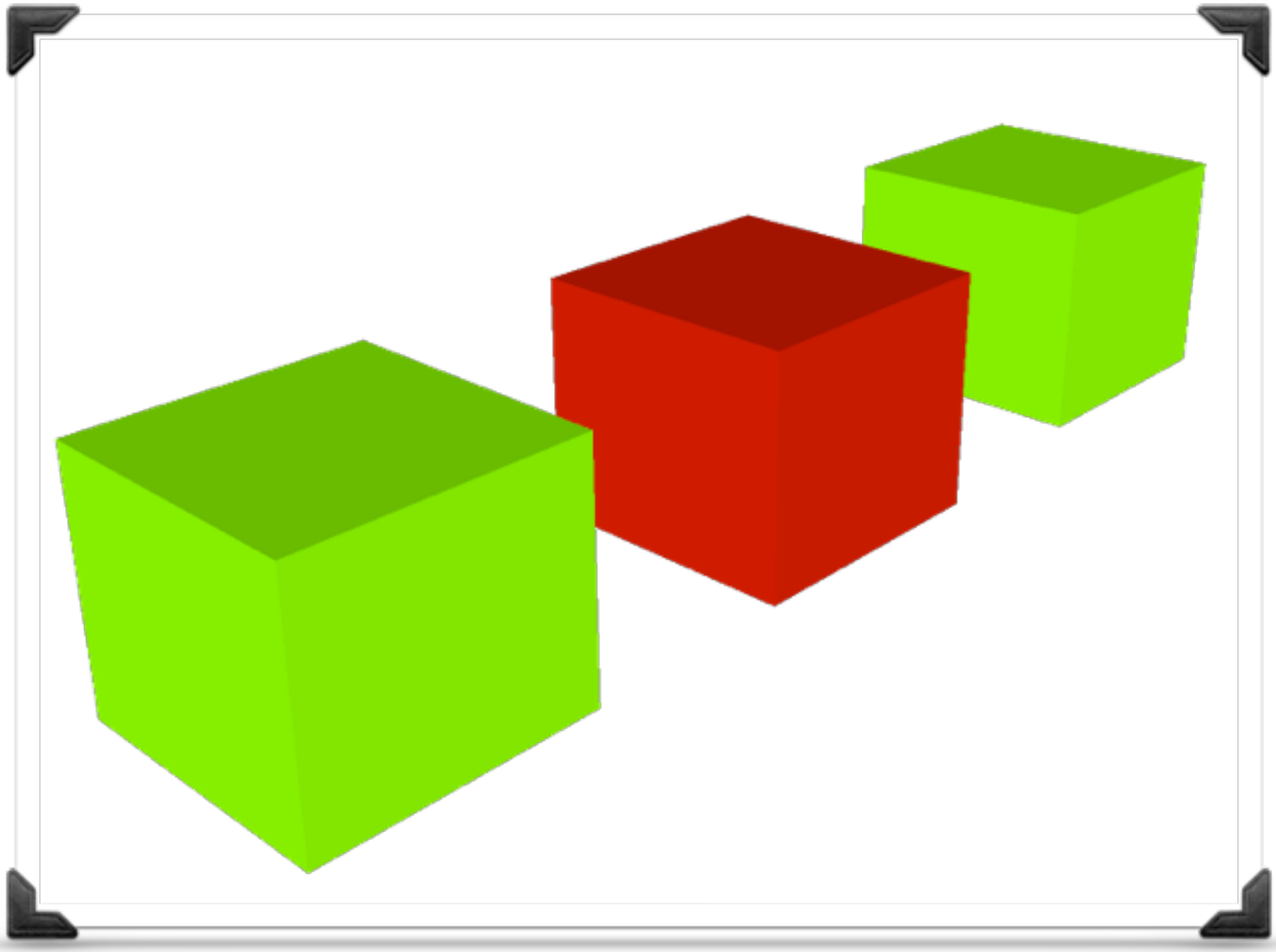
Kamera

Beispiel



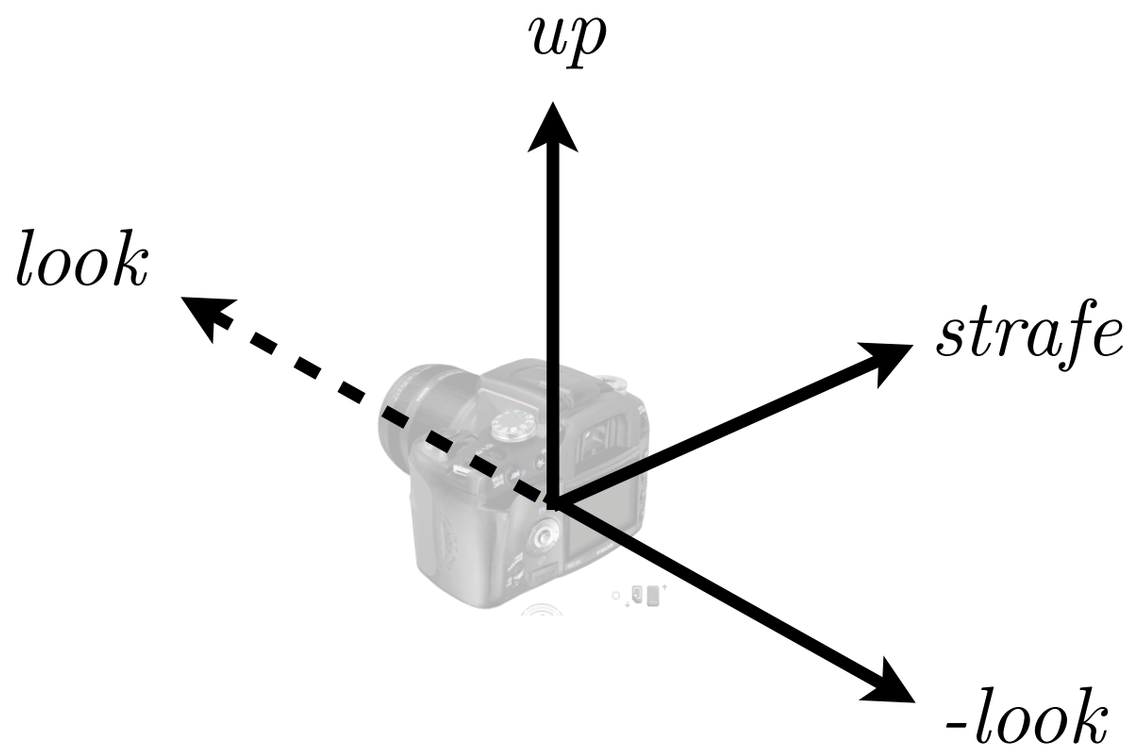
Kamera

Beispiel



Sichtkoordinaten

- Kamera schaut entlang negativer z-Achse ($look$), x-Achse zeigt nach rechts ($strafe$) und y-Achse zeigt nach oben (up)

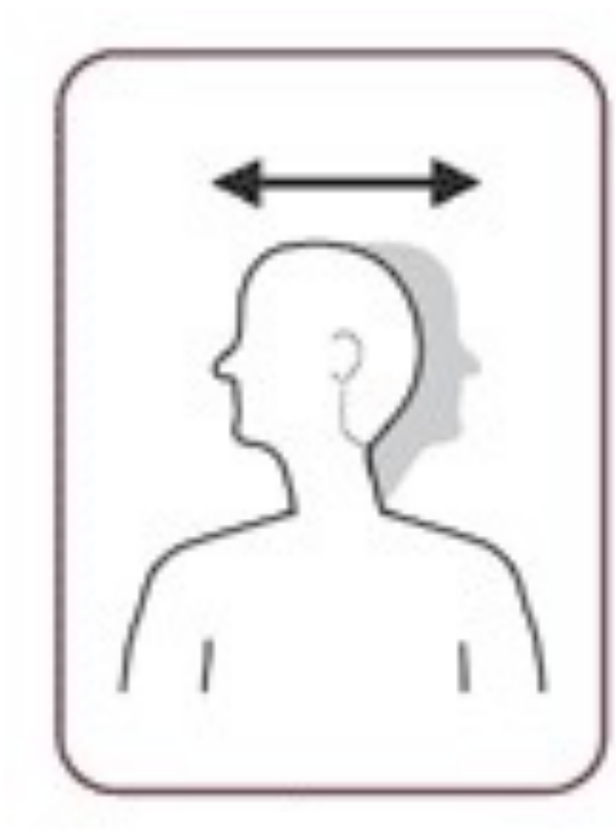


Kameraorientierung

- **Orientierung** der Kamera spezifizierbar durch verschiedene Verfahren
  - 4x4-Rotationsmatrix
  - Euler-Winkel
  - Quaternionen

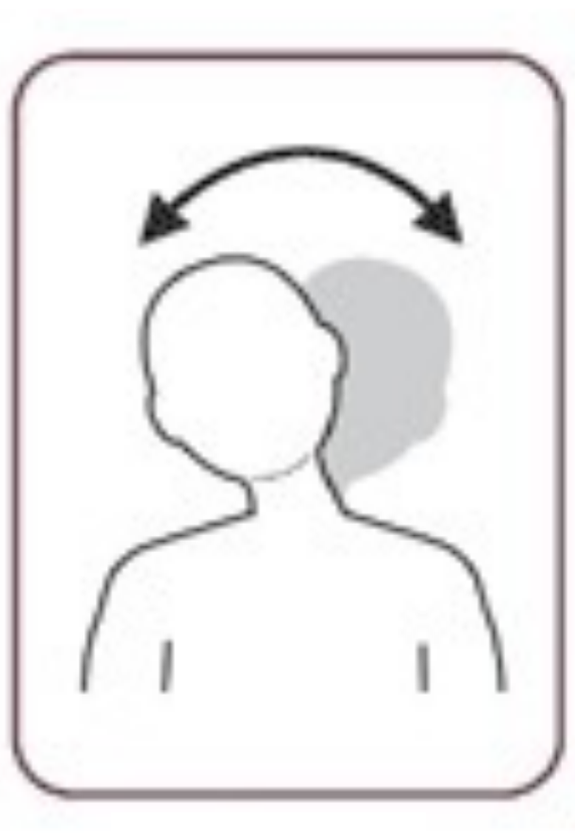
Euler-Winkel

- Euler-Winkel sind inspiriert durch **drei Rotationsachsen** des menschlichen Kopfes



“gieren”

“nicken”

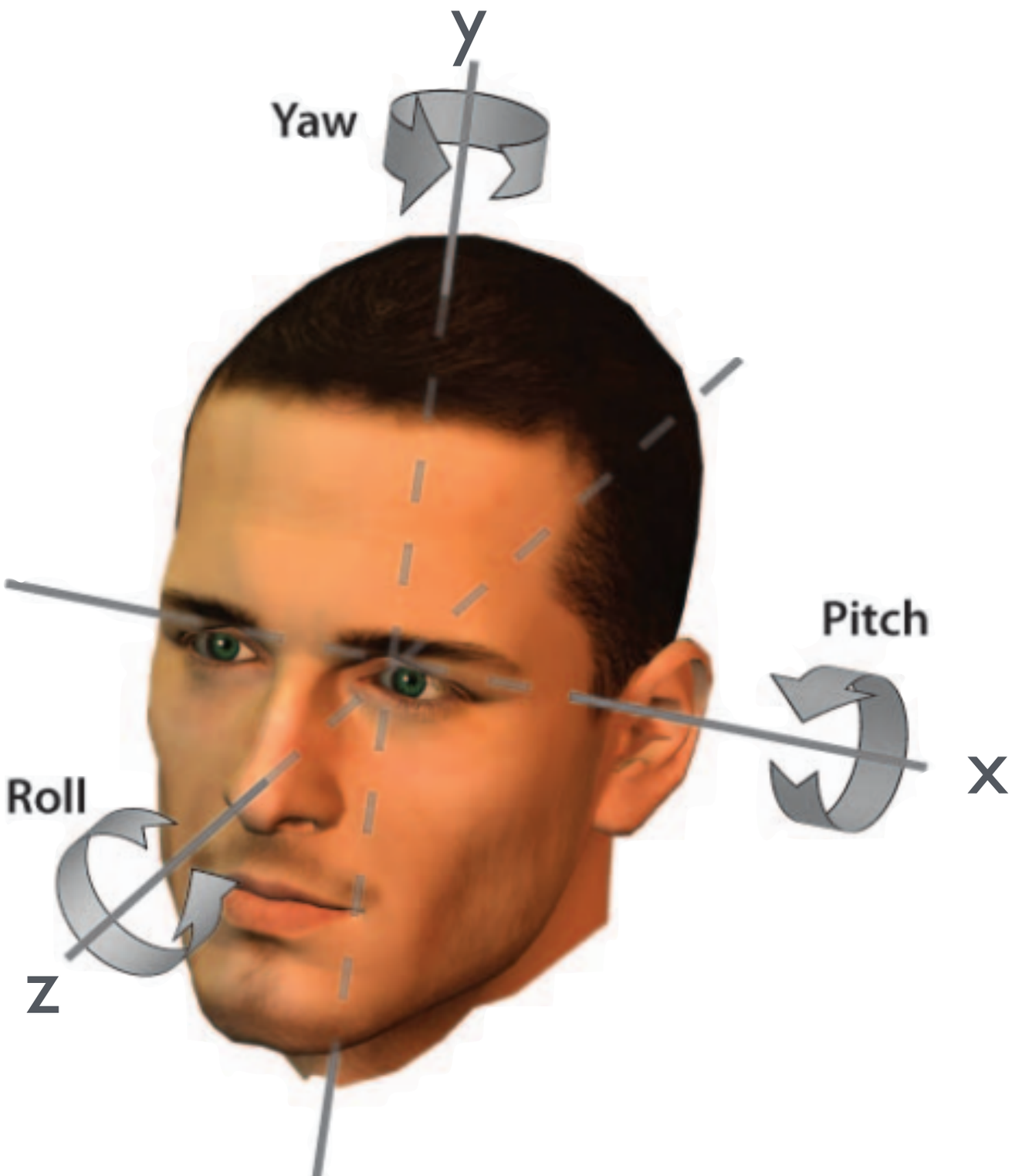


“rollen”

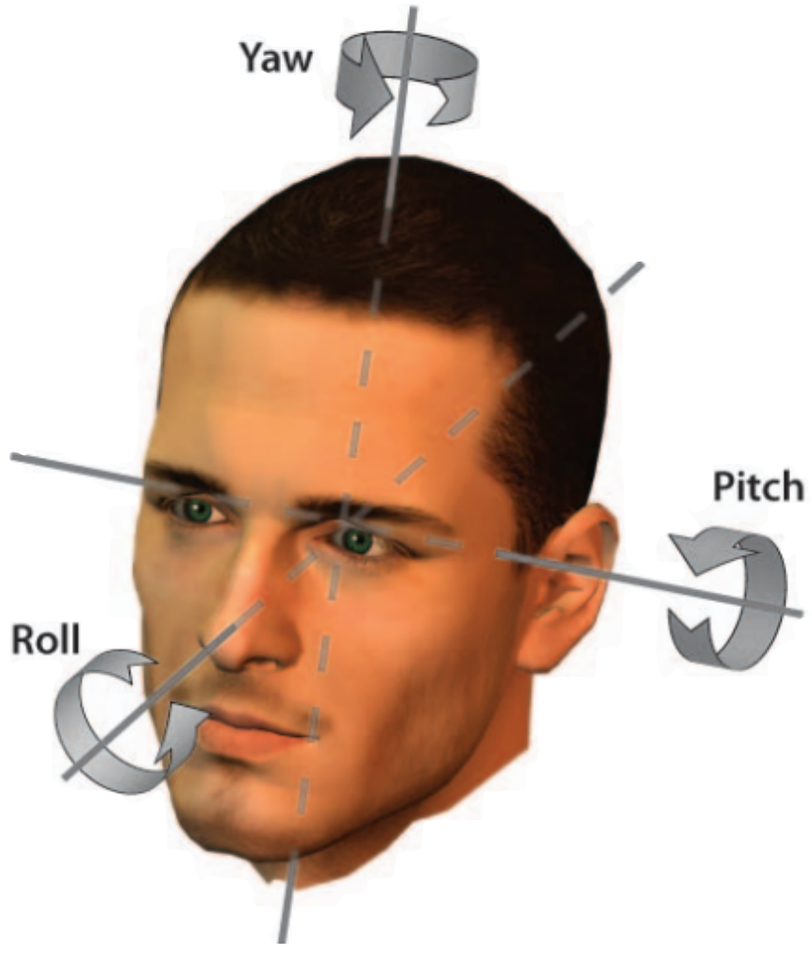
Euler-Winkel

- Orientierung definiert durch **drei Winkel** (engl. *yaw, pitch, roll*)
- Berechnet als **Komposition** von drei aufeinanderfolgenden Rotationen um Achsen $x,\ y$ und z

Euler-Winkel



Diskussion



Nehmen Sie die Kopfpose für Euler-Winkel yaw=45°, pitch=45°, roll=45° ein!

Euler-Winkel

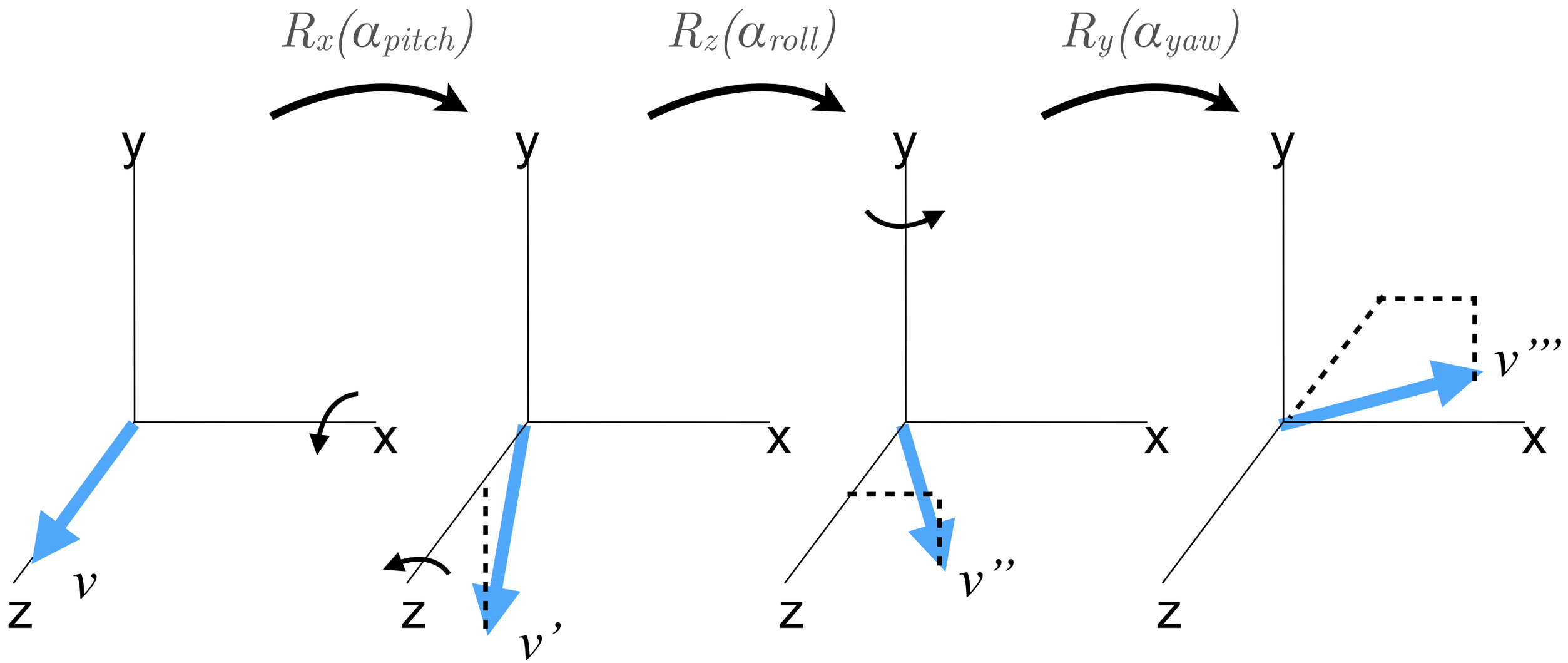
- **Reihenfolge der Rotationen** für Winkel $(\alpha_{yaw}, \alpha_{pitch}, \alpha_{roll})$ wichtig, z.B.:

$$R_{Euler} := R_y(\alpha_{yaw}) \cdot R_z(\alpha_{roll}) \cdot R_x(\alpha_{pitch})$$

- *Beachten: Unterschiedliche Rotationsreihenfolge führt zu unterschiedlichen Ergebnissen!*

Euler Winkel

Beispiel

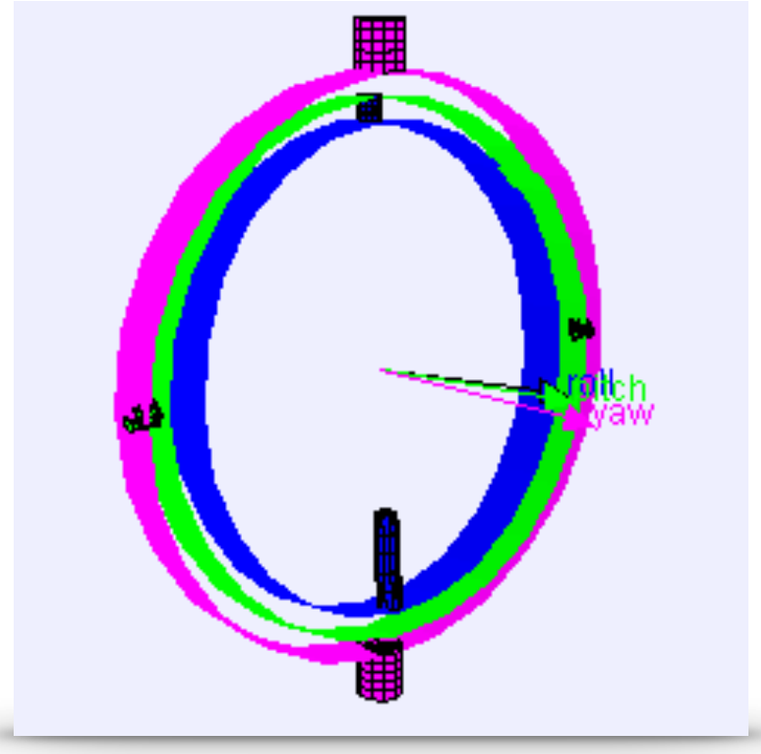


Euler Winkel

Kardanische Blockade

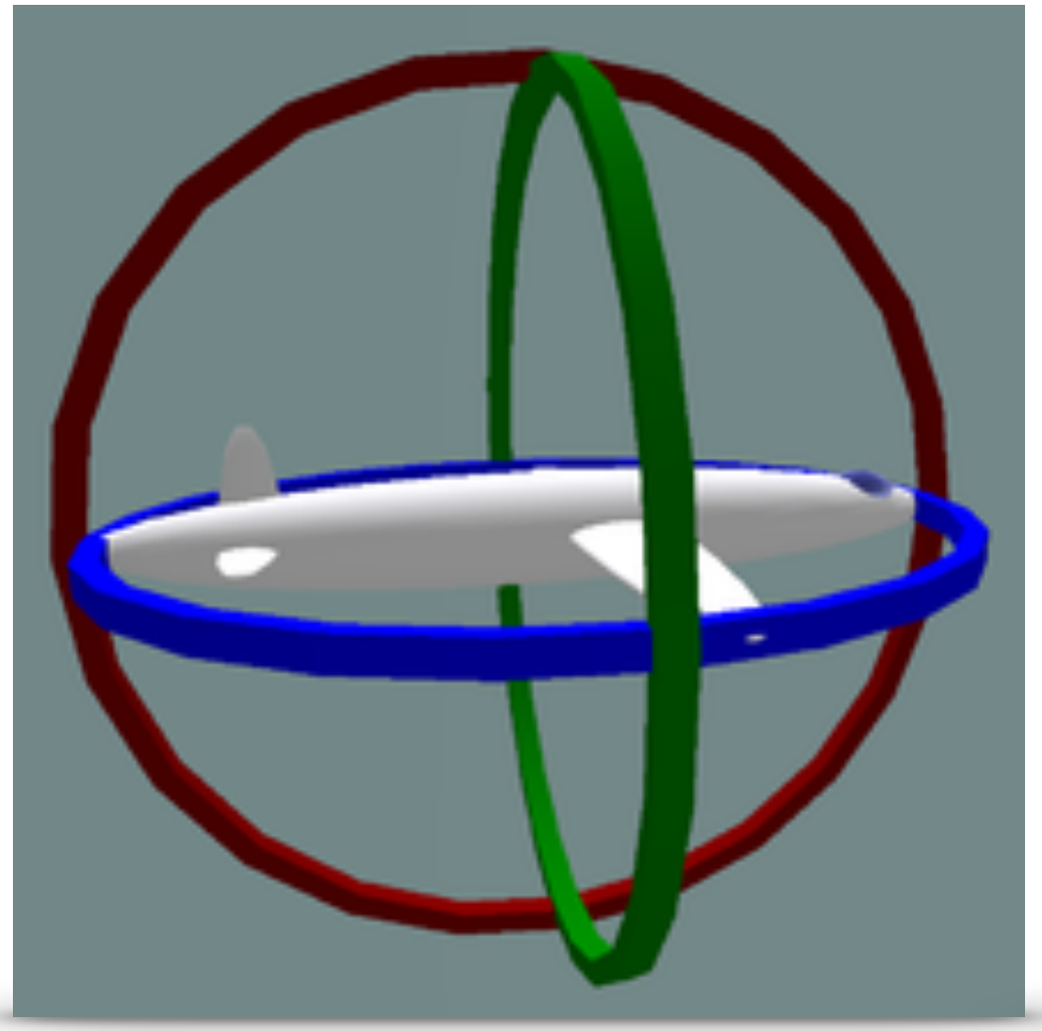
- **Kardanische Blockade** (engl. ***Gimbal Lock***) bezeichnet geometrisches Problem, welches bei hierarchischen Transformationen in Verbindung mit Eulerwinkeln auftreten kann

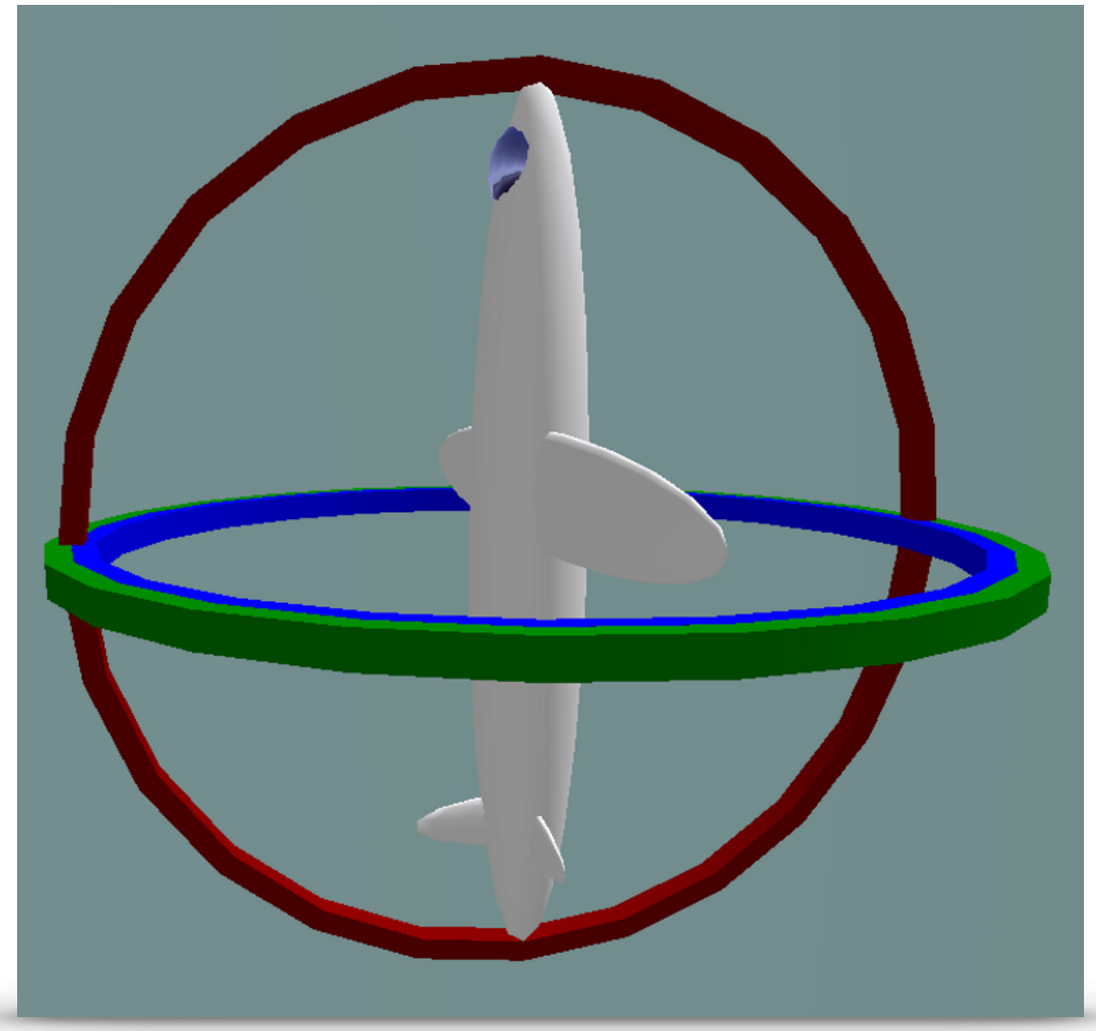
- liegen zwei Kardanringe in gleicher Ebene reduzieren sich Freiheitsgrade



Euler Winkel

Beispiel: Gimbal Lock



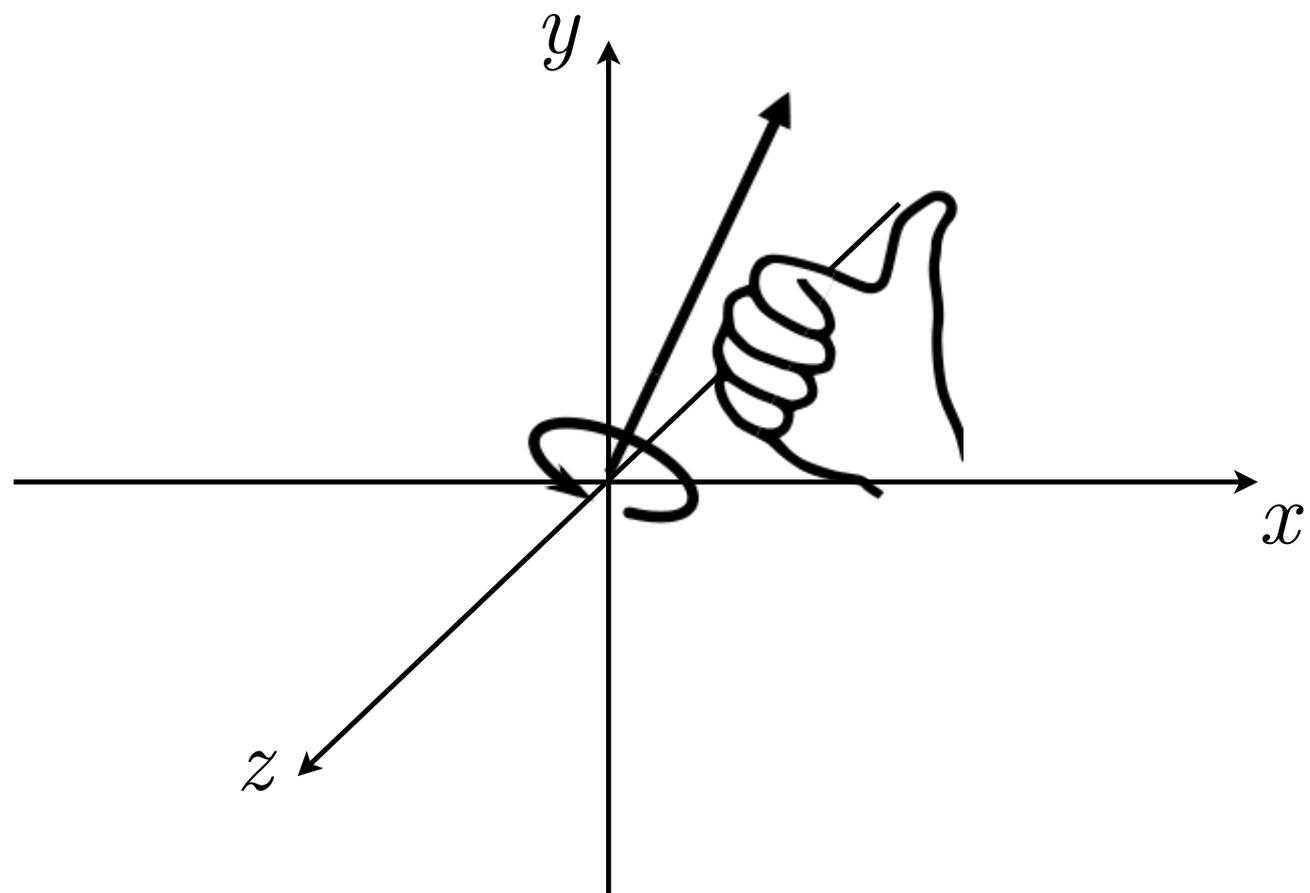


Quaternionen

- **Quaternionen** sind vierdimensionale algebraische Konstrukte $q = (x, y, z, w)$
- alternative Methode um 3D-Orientierung zu spezifizieren

Quaternionen

- jede Orientierung kann durch eine **einzelne Rotation** um eine 3D-Raumachse spezifiziert werden



Quaternionen

- Quaternionen $q = (x, y, z, w)$ repräsentieren **3D-Vektor** $(a_x,\ a_y,\ a_z)$ und **Winkel** θ als

$$q = (a_x \cdot \sin(\theta/2), a_y \cdot \sin(\theta/2), a_z \cdot \sin(\theta/2), \cos(\theta/2))$$

Quaternionen

- Quaternionen $q = (x, y, z, w)$ können in **Rotationsmatrizen** konvertiert werden:

$$R_{quat} = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2wz & 2xz + 2wy & 0 \\ 2xy + 2wz & 1 - 2x^2 - 2z^2 & 2yz - 2wx & 0 \\ 2xz - 2wy & 2yz + 2wx & 1 - 2x^2 - 2y^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Zusammenfassung

Euler-Winkel:

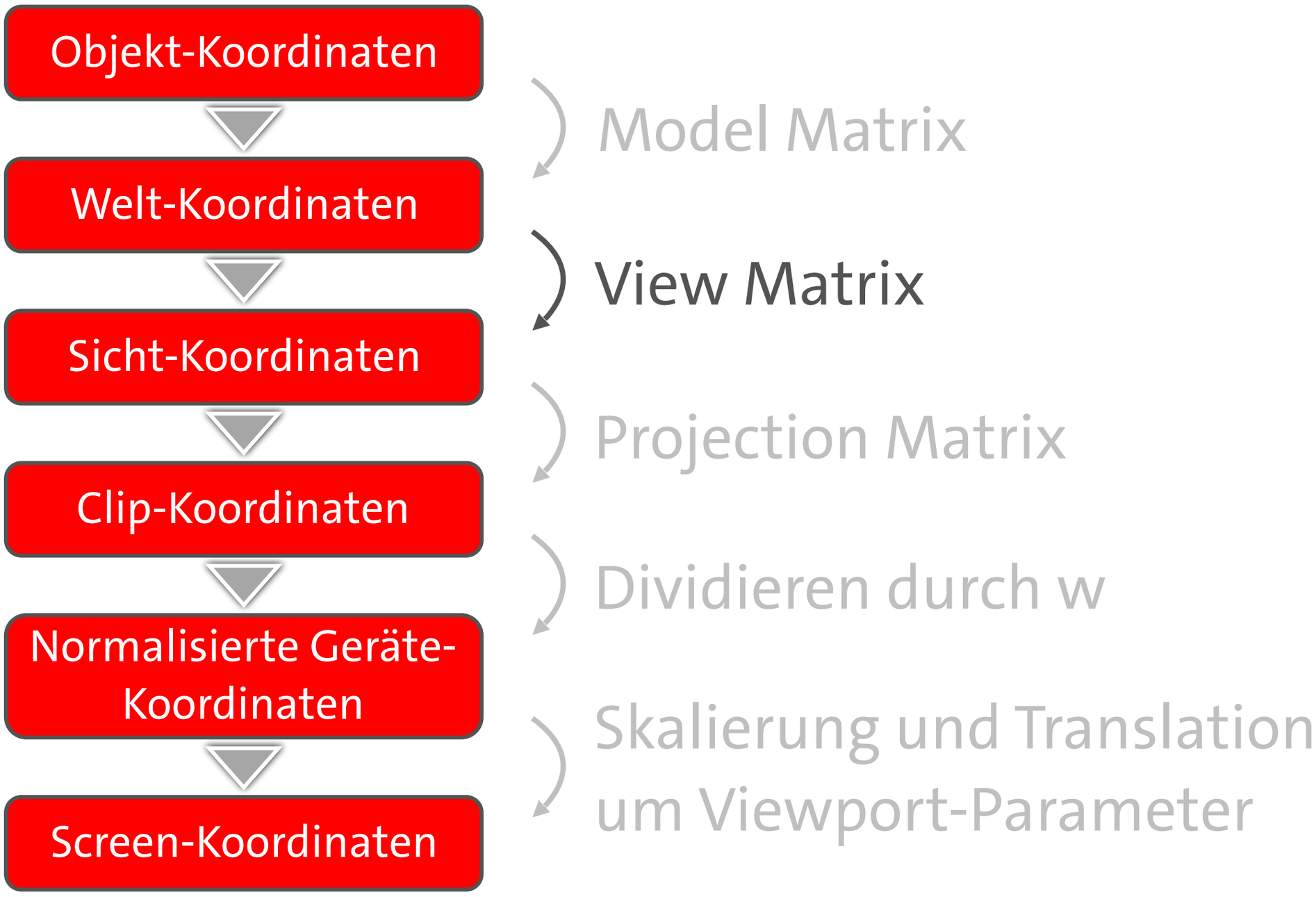
- Werte können leicht geometrisch interpretiert werden
- keine einheitliche Rotationsreihenfolge
- Konvertierung in Matrix teuer
- Gimbal Lock

Quaternionen:

- Werte haben keine direkte geometrische Interpretation
- eine konsistente Repräsentation
- Konvertierung in Matrix günstig

Pipeline

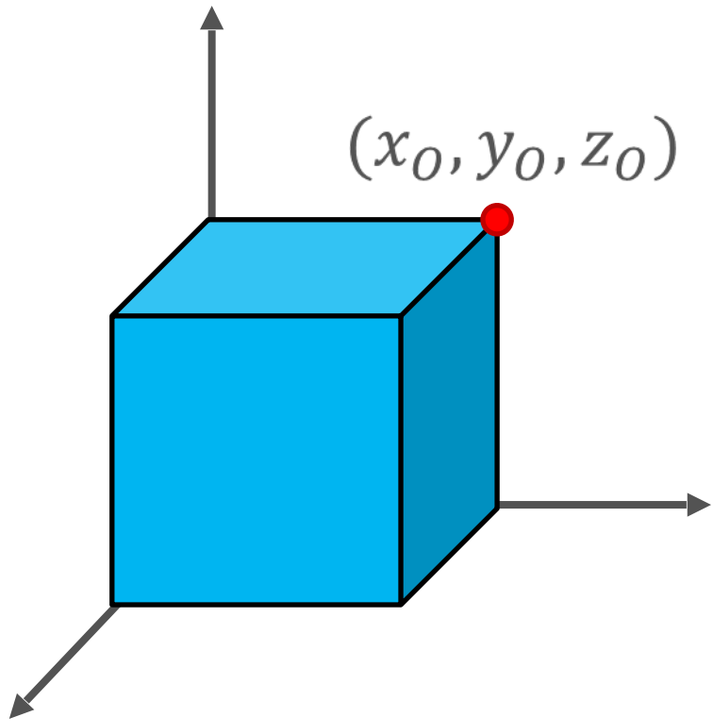
Einordnung



Pipeline

Ausgangszustand

Objekt-Koordinaten

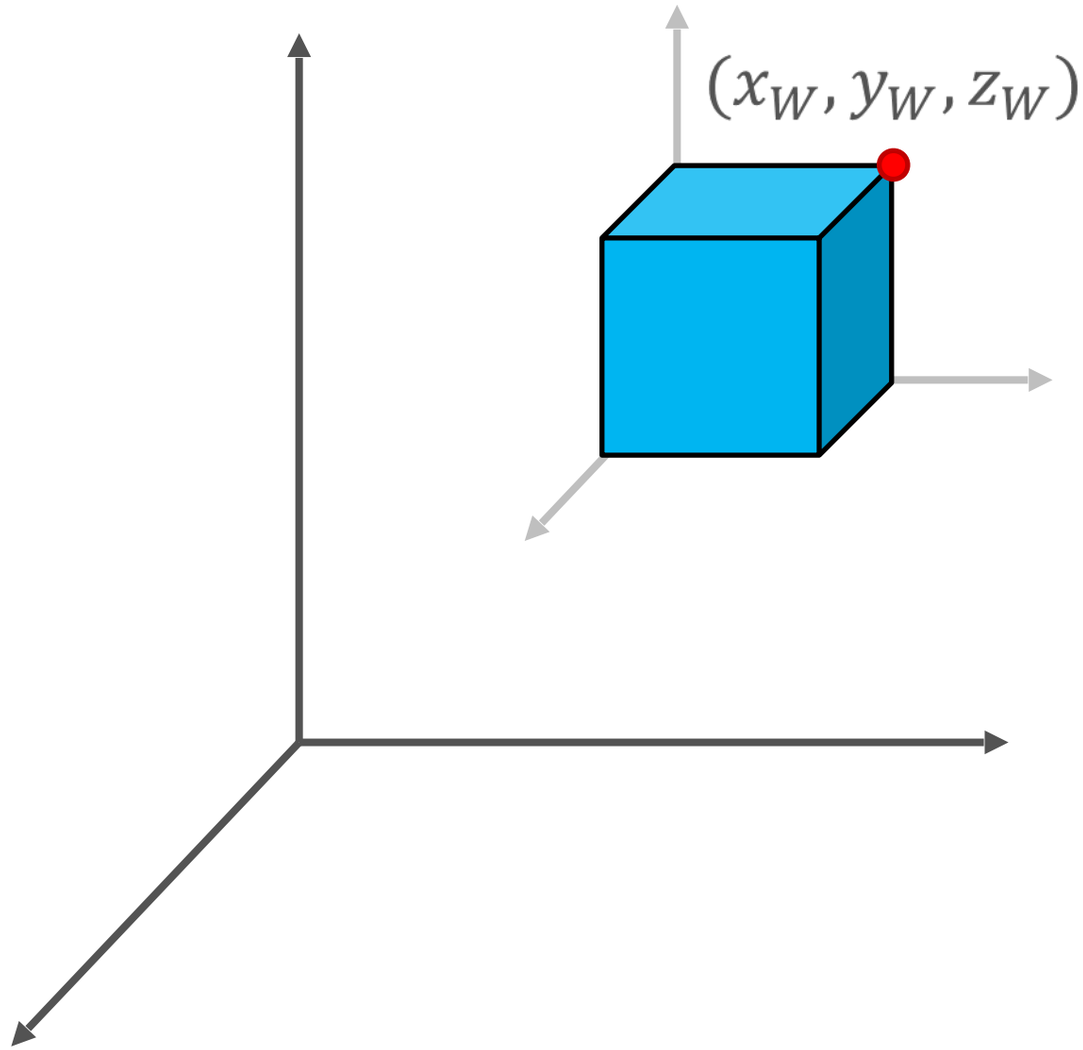


Pipeline

1. Transformation

Objekt-Koordinaten

Welt-Koordinaten

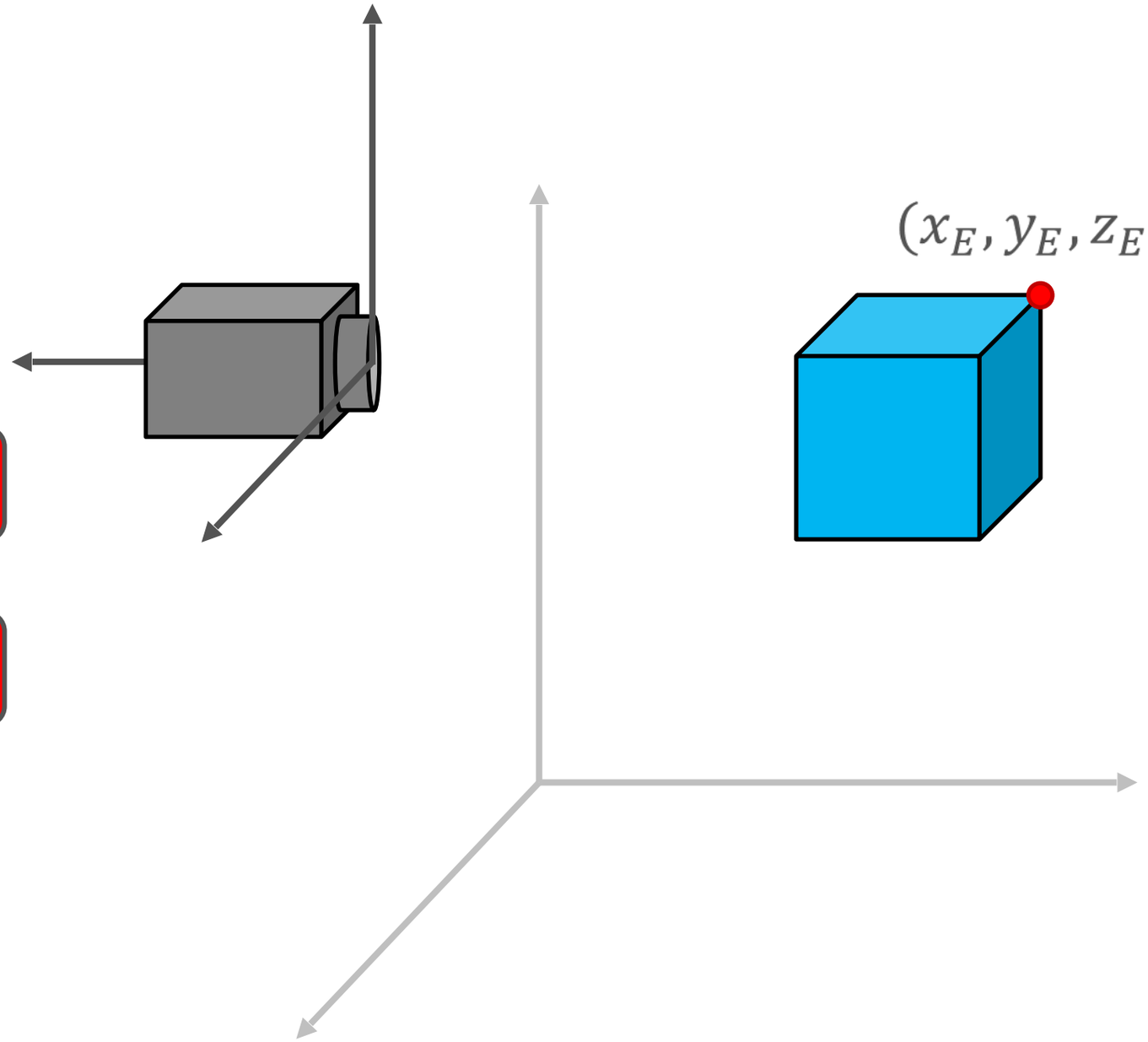


Pipeline

2. Transformation

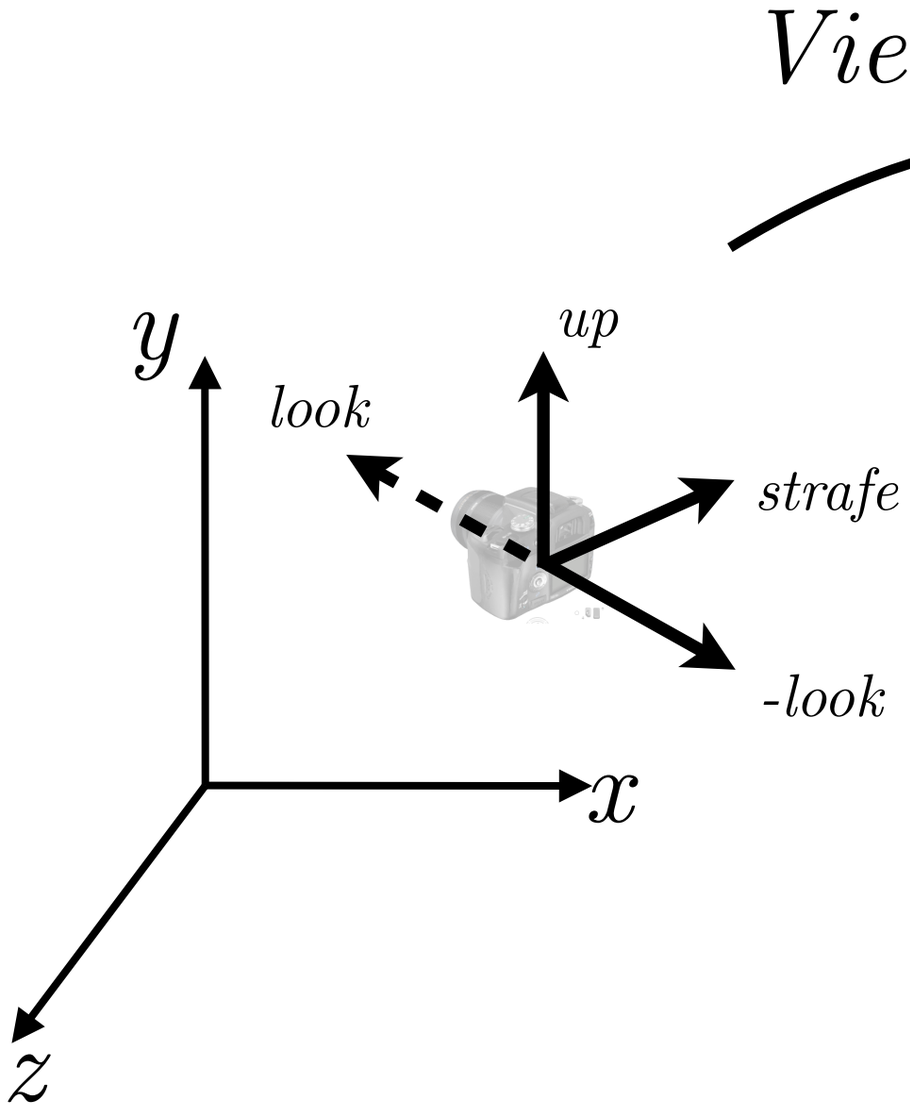
Welt-Koordinaten

Sicht-Koordinaten

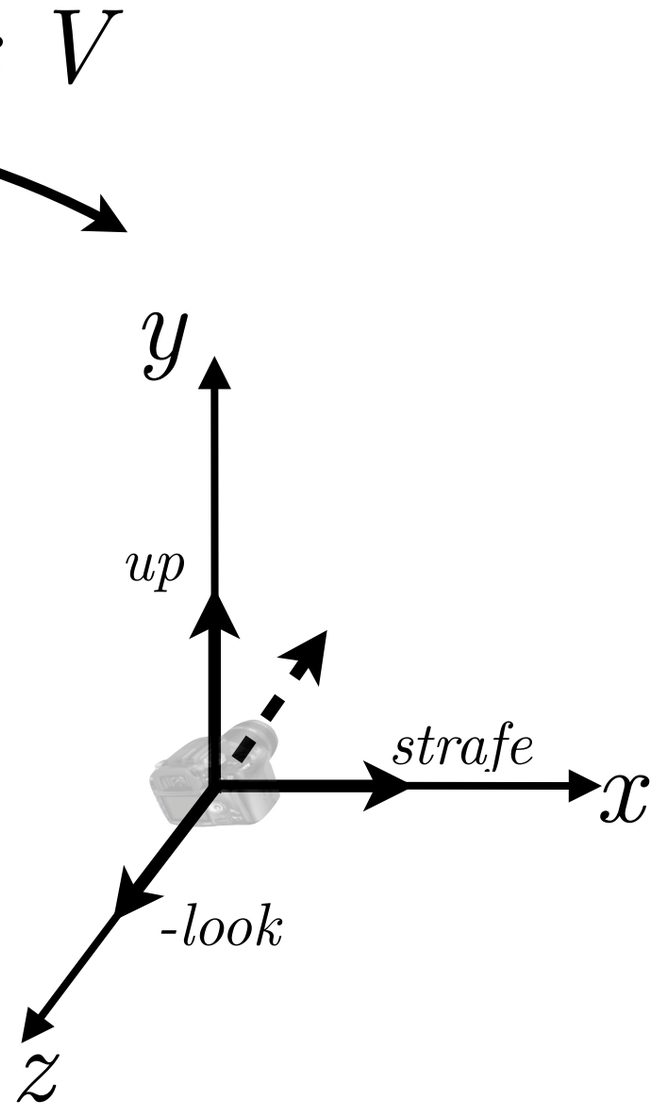


Pipeline

Sichtkoordinaten



$View\text{-}Matrix\ V$

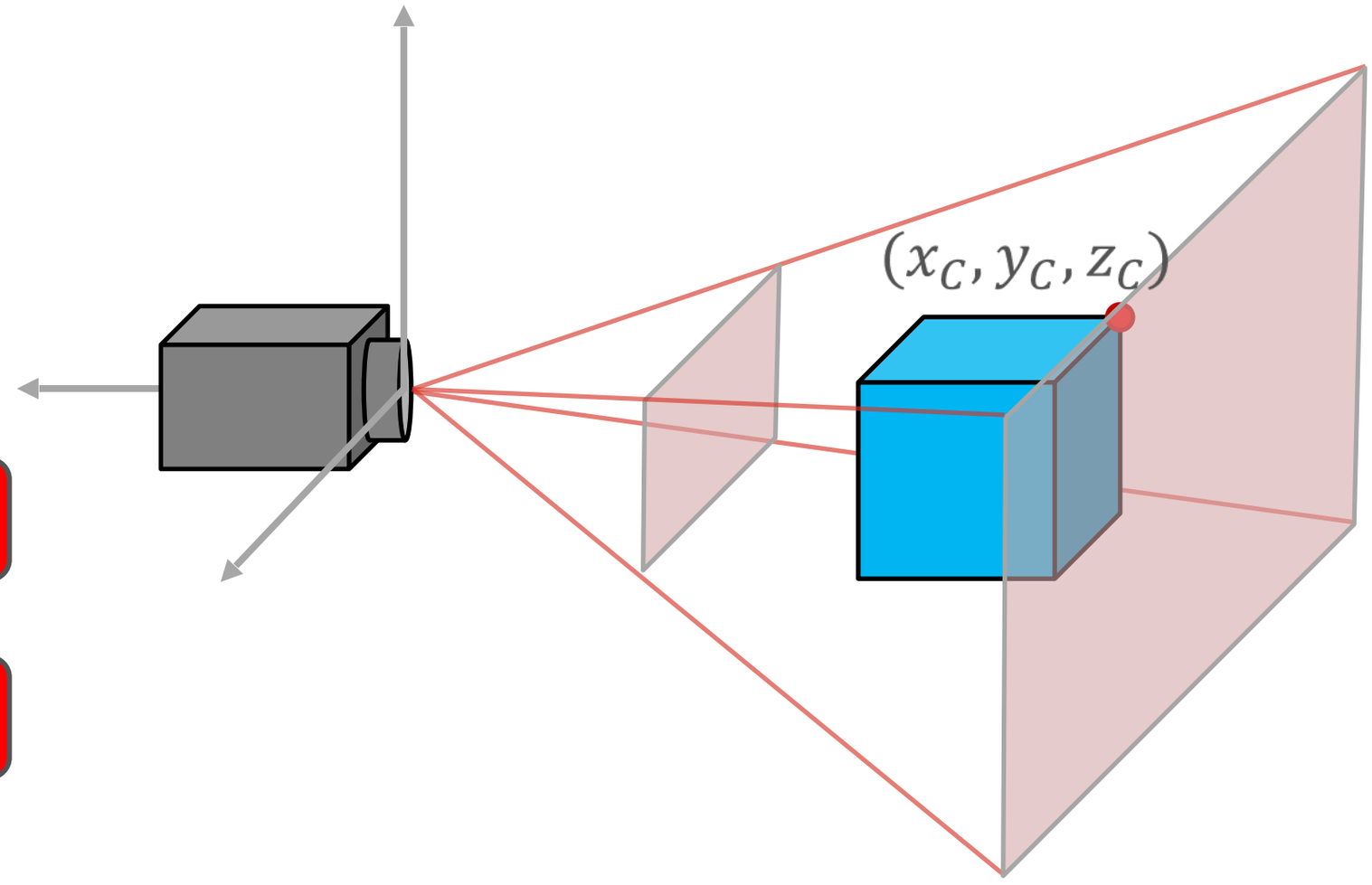


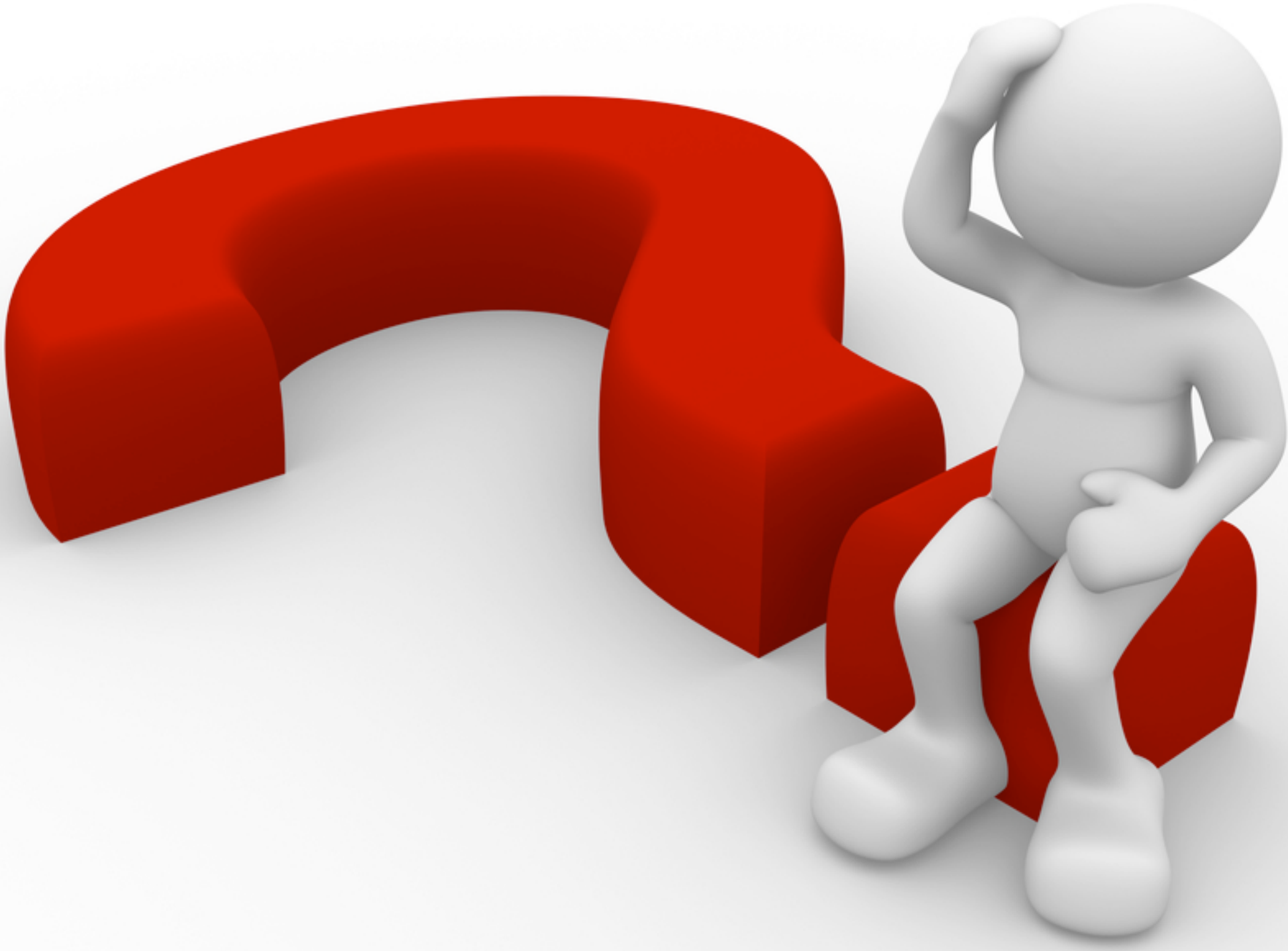
Pipeline

3. Transformation

Sicht-Koordinaten

Clip-Koordinaten





Interaktive Computergrafik

Kapitel Kameras

Sichtbarkeitsermittlung

Sichtbarkeit

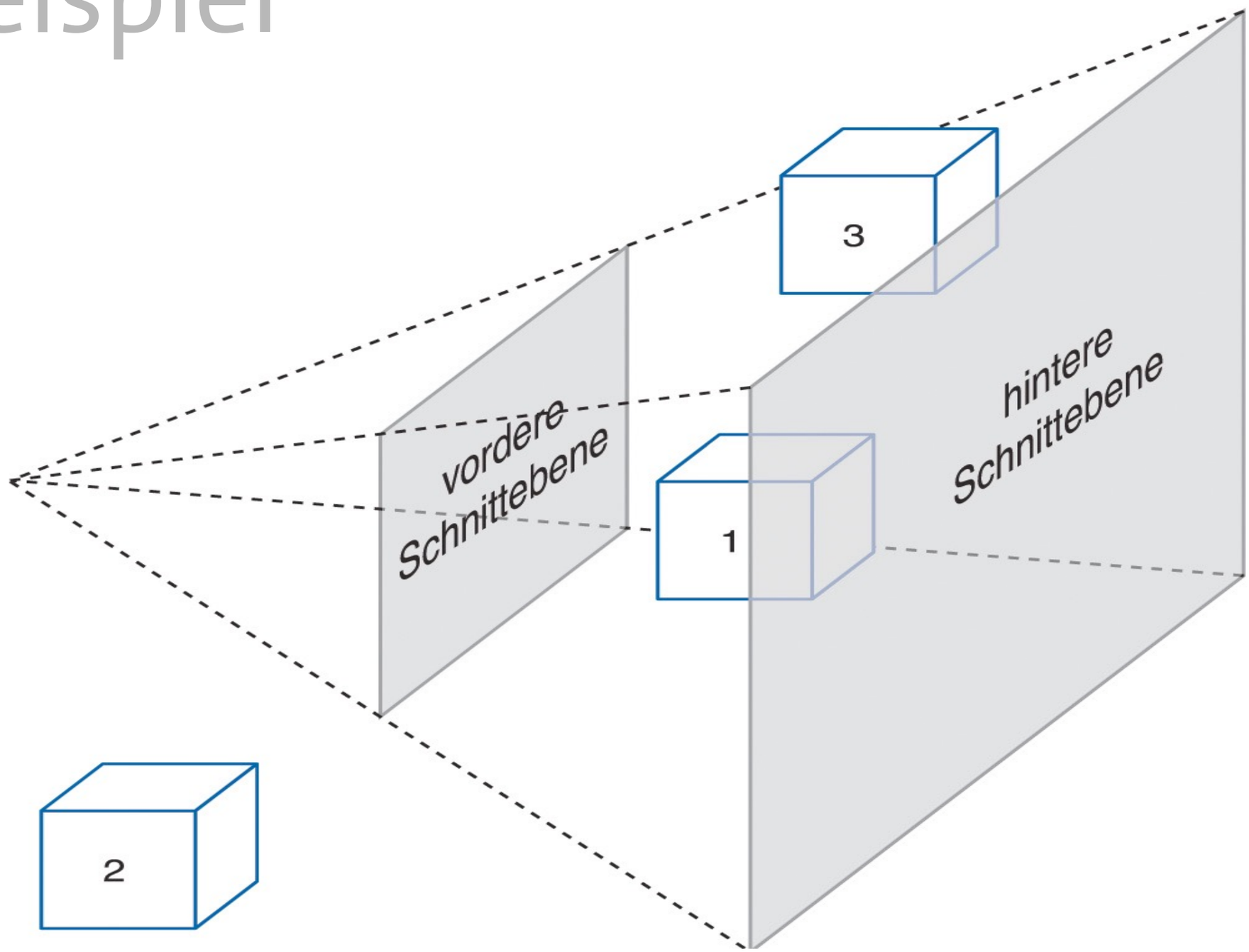
- **Sichtbarkeitsermittlung** erlaubt solche Polygone der Szene auf Bildschirm darzustellen, die tatsächlich im Sichtvolumen liegen

View Frustum Culling

- beim Culling gegen Sichtvolumen (engl. ***View Frustum Culling***) können Polygone ausgeschlossen werden, die später nicht im Bild zu sehen sein werden

View Frustum Culling

Beispiel

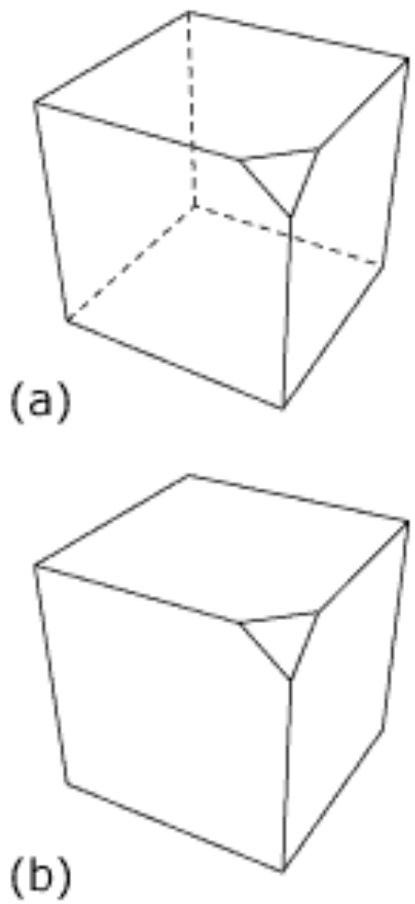


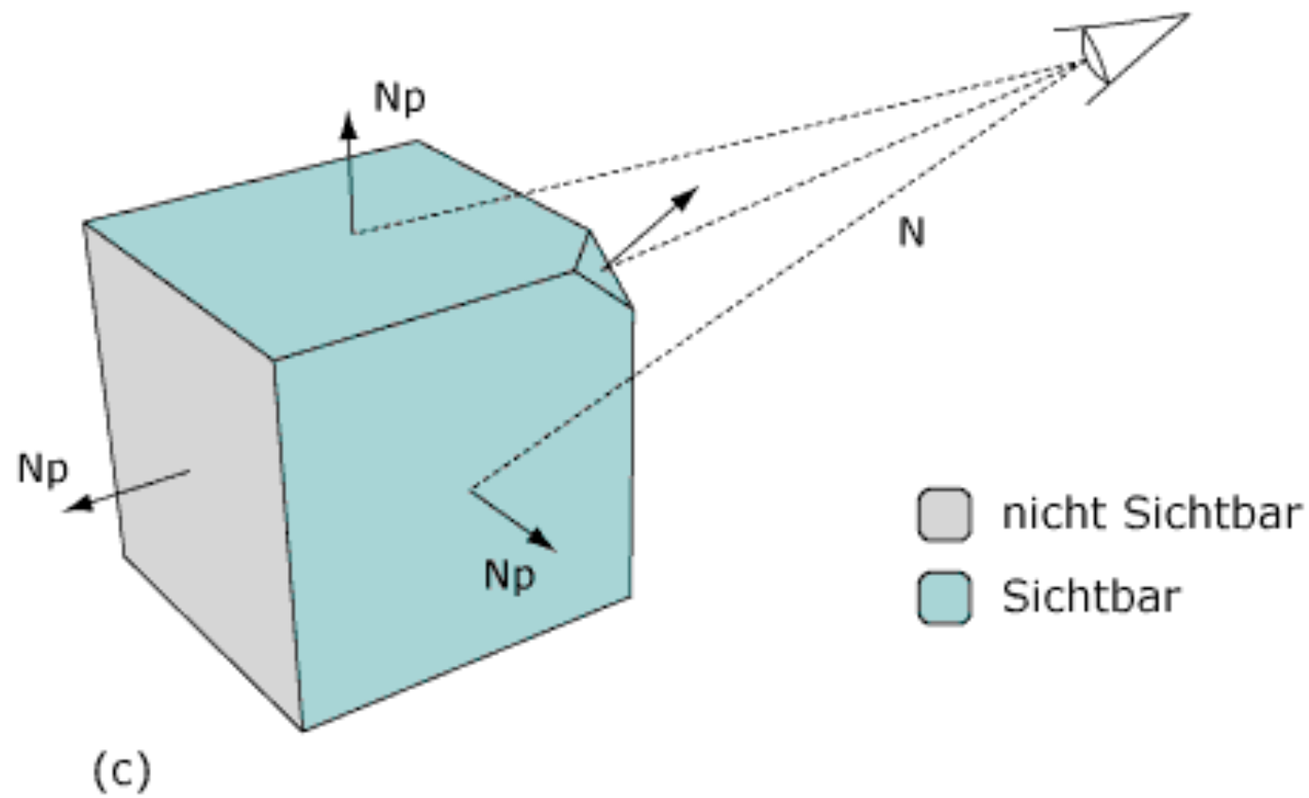
Back Face Culling

- **Back Face Culling** ist Algorithmus, der Polygone verwirft, deren Normale von Kamera weg zeigen
- bei geschlossenen Polygonnetzen sind diese Polygone auf Rückseite von Objekten und werden daher nicht gesehen

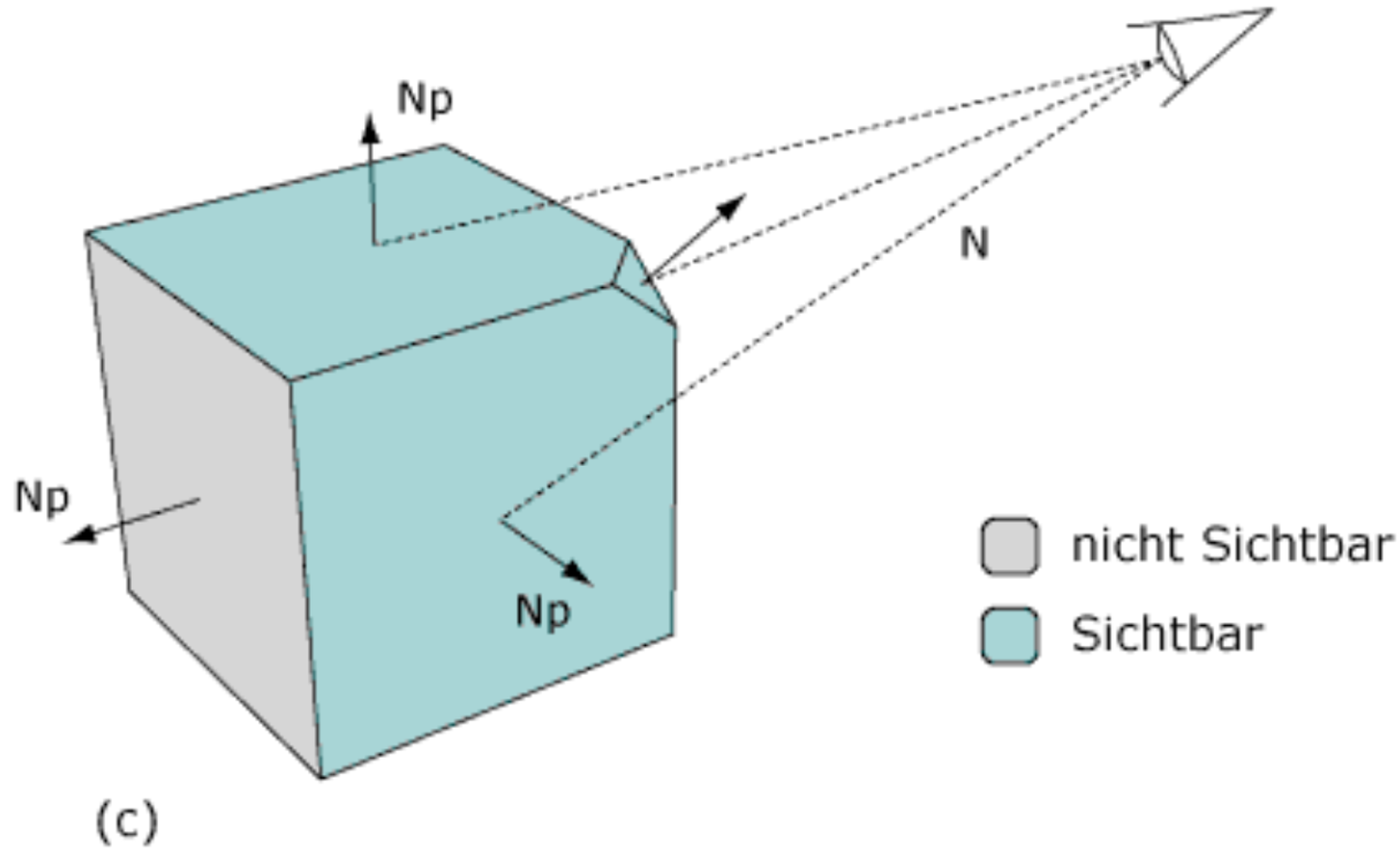
Back Face Culling

Beispiel





Diskussion



Wie lässt sich erkennen, ob Polygon von Kamera gesehen wird?

Skalarprodukt

Geometrische Interpretation

- Winkel α zwischen zwei Vektoren x und y

$$\cos\alpha = \left(\frac{x \cdot y}{|x| \cdot |y|} \right)$$