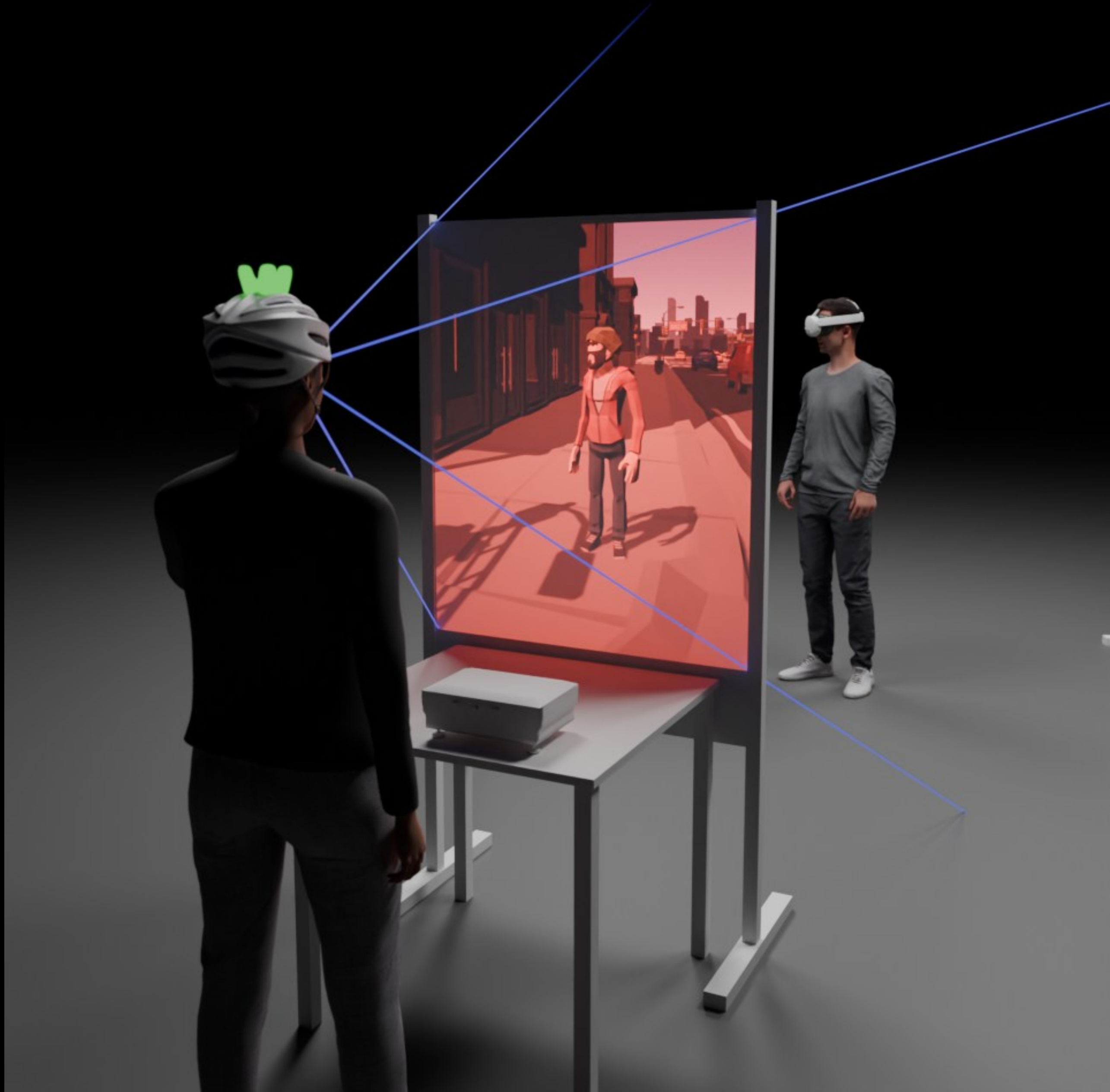
Window into VR

Technical Challenges

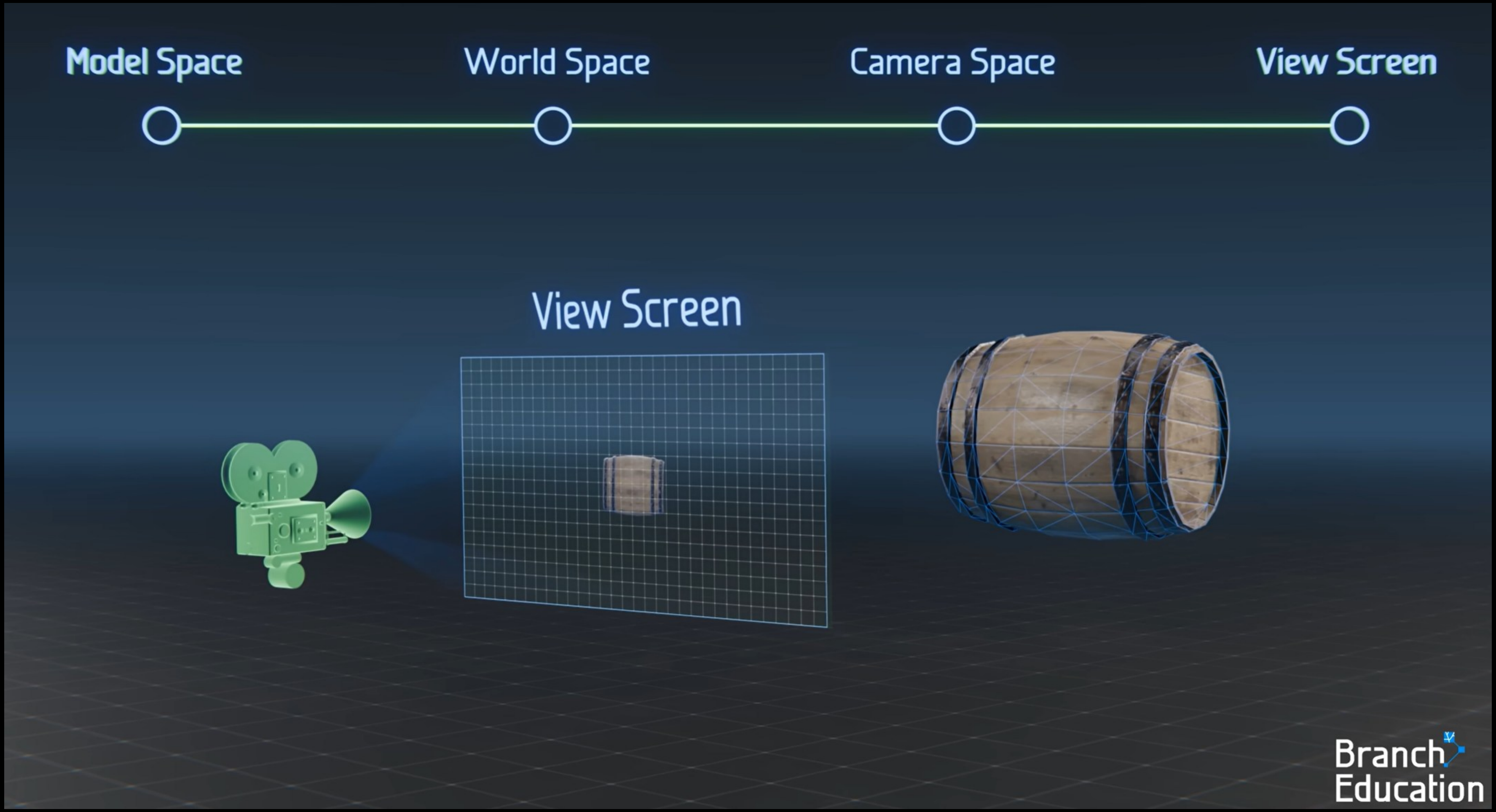
Technical Challenges

- Projektorkalibrierung
- Headtracking
- Off-Axis Rendering



Bei Unklarheiten (z.B. Begriffen) fragen!

Hintergrund Graphics Pipeline

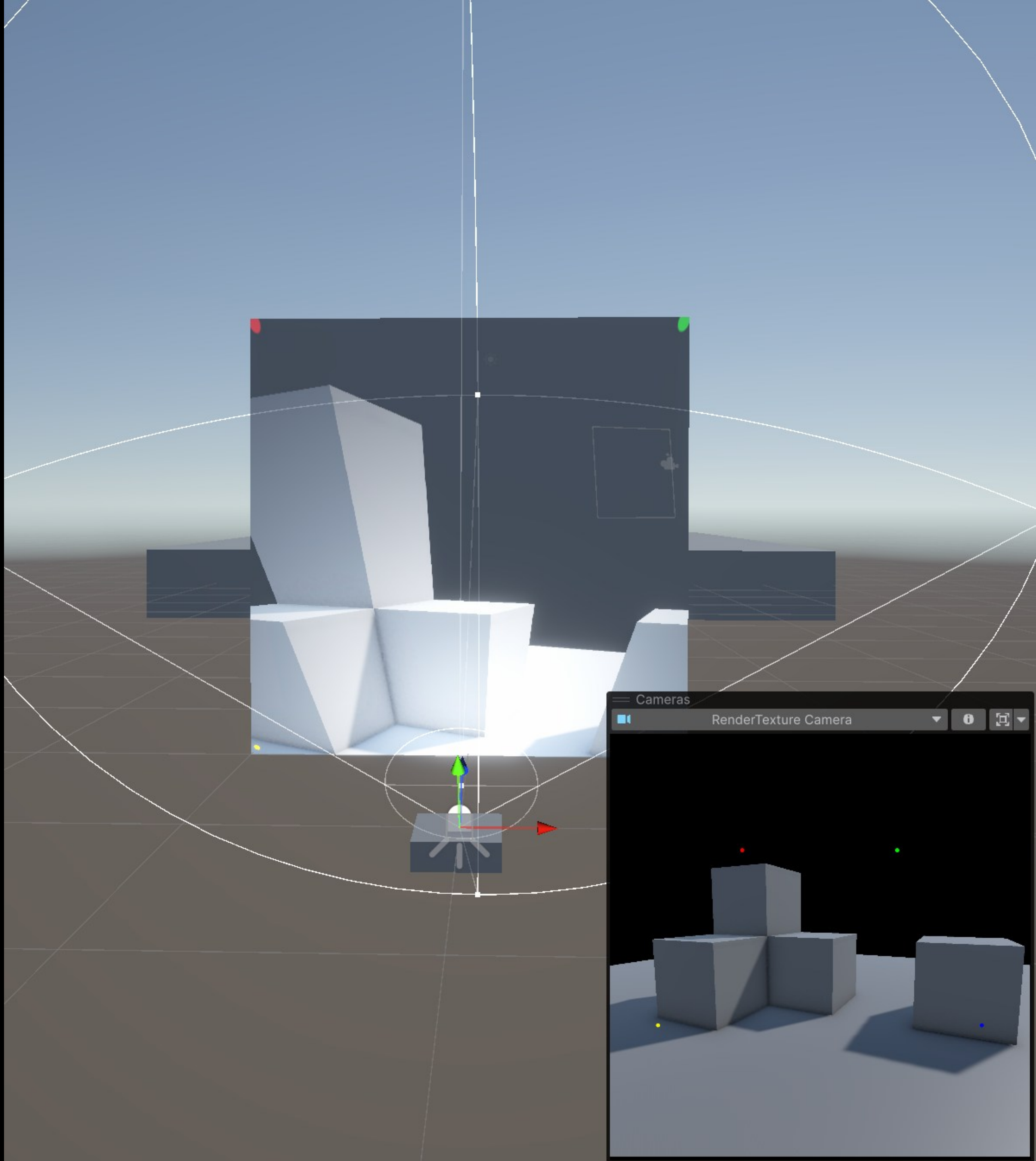


https://youtu.be/C8YtdC8mxTU?t=192

Warum?

Projektionsrichtung ist meist *schief*…

… und das projizierte Bild daher verzerrt.



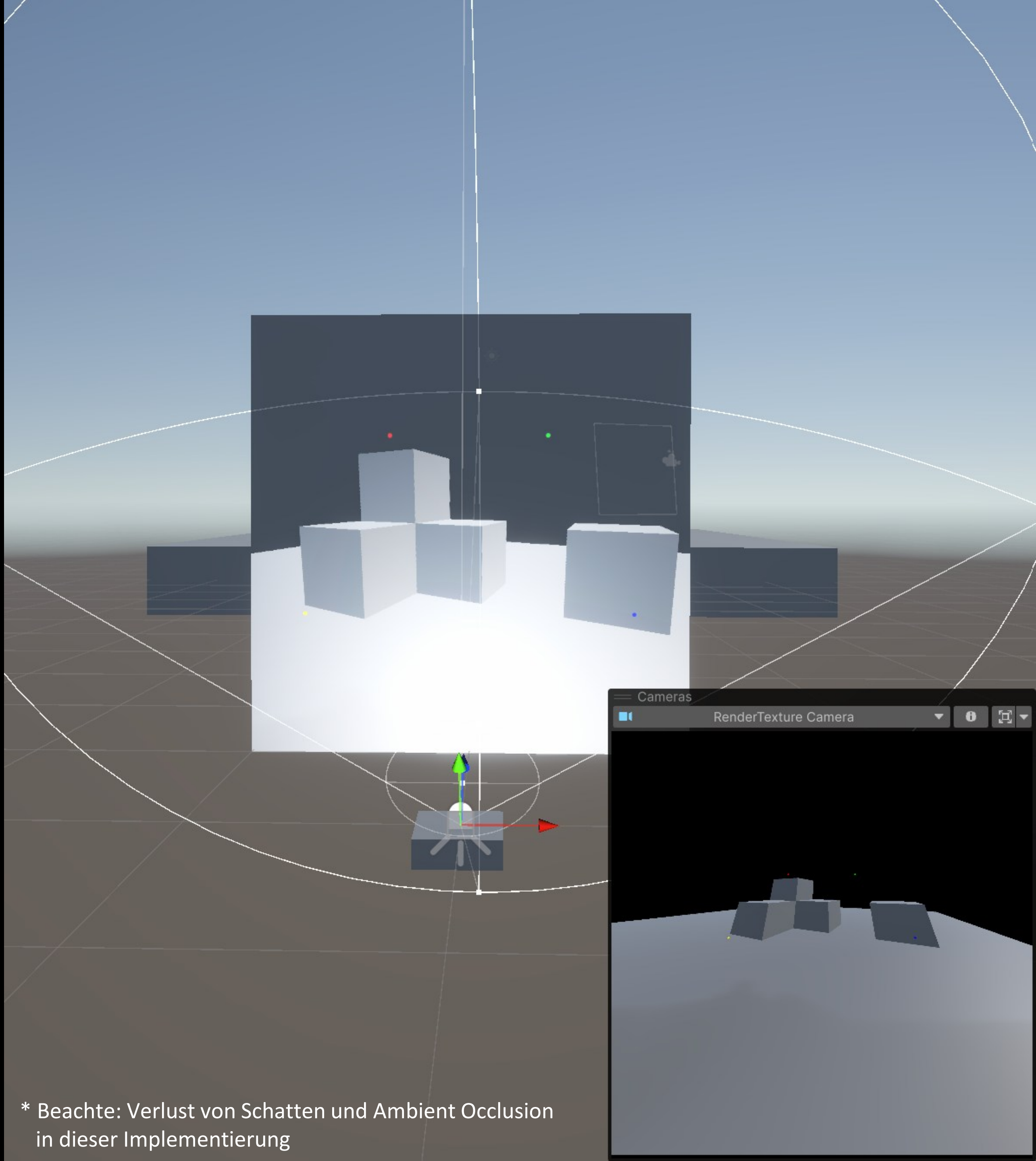
Was?

Lösung:

Verzerrung „invertiert“ anwenden

→ Input Bild verzerrt

→ projiziertes Bild entzerrt

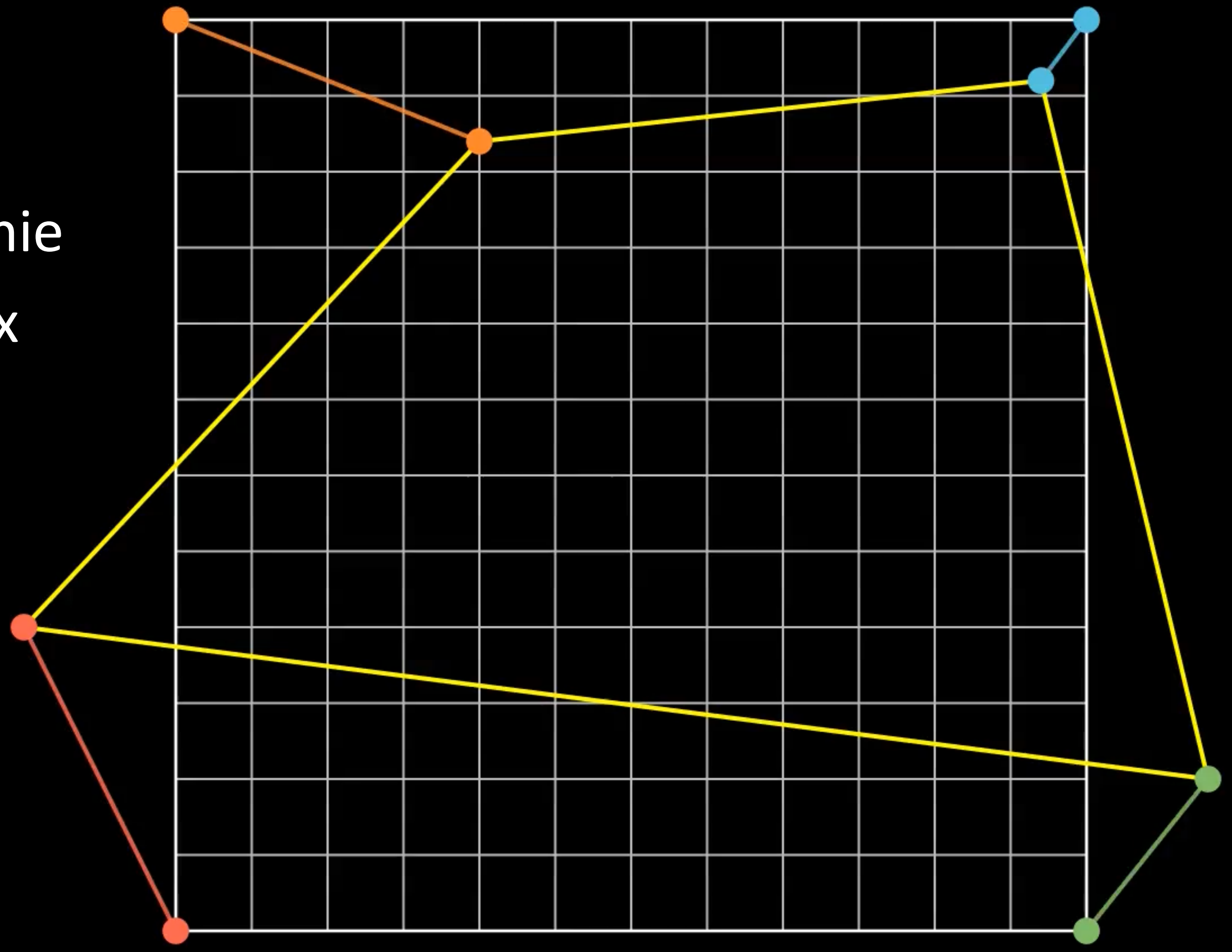


* Beachte: Verlust von Schatten und Ambient Occlusion in dieser Implementierung

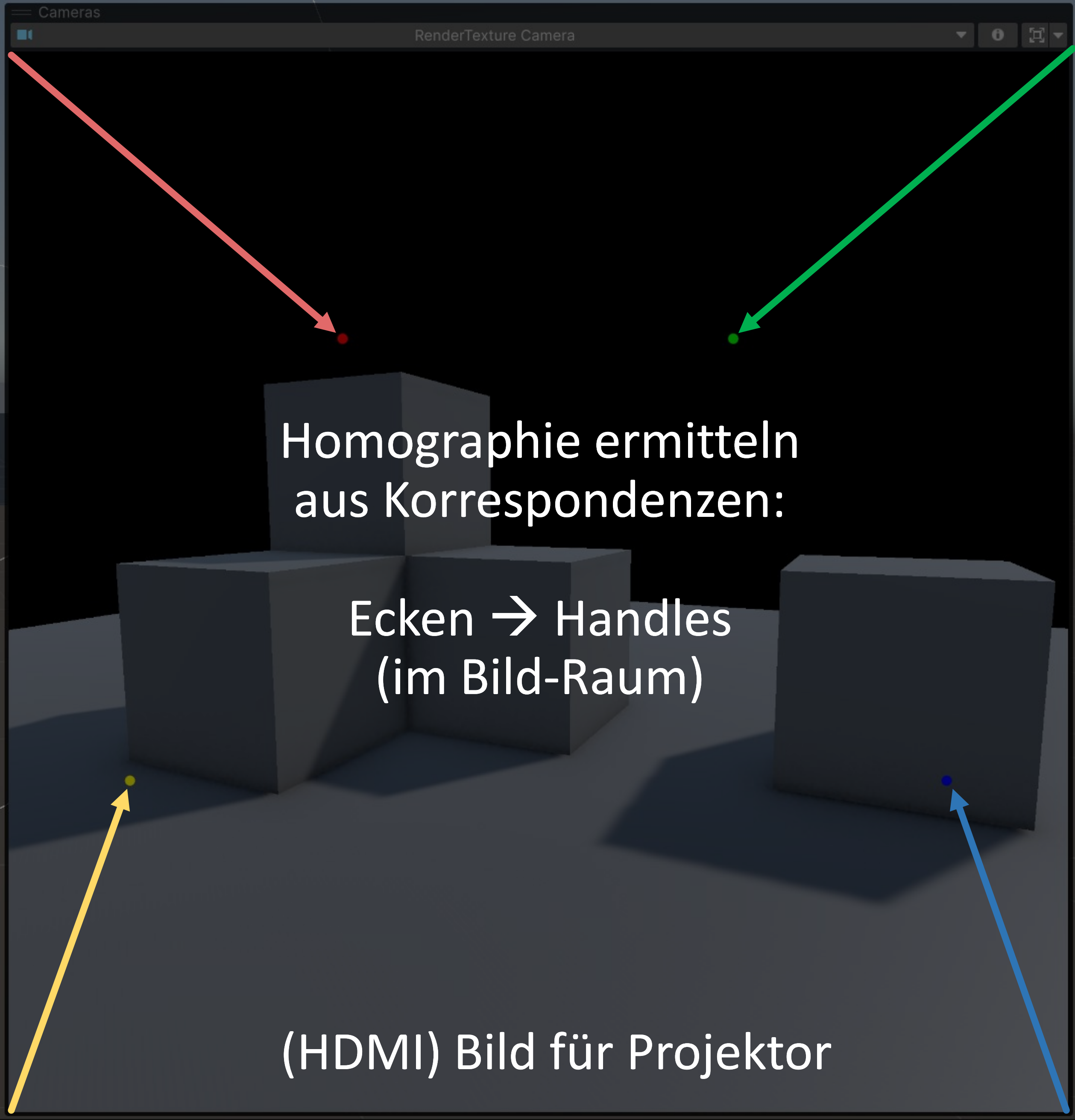
Wie?

Homographie

3x3 Matrix



Projektorkalibrierung



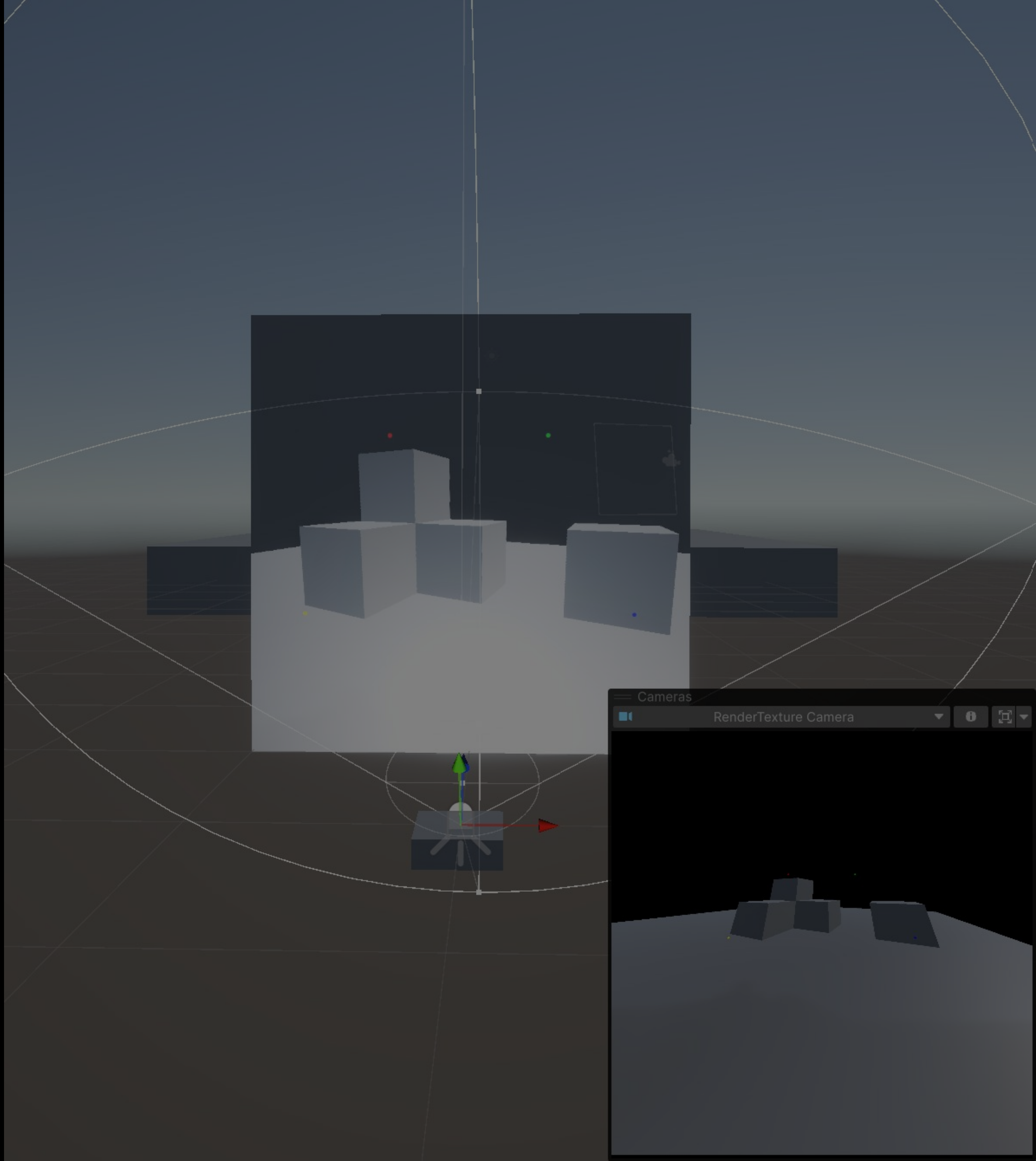
Pipeline Integration

Option A:
Integration in Projection Matrix

- Problem:
  Verändert homogene Koordinate
  & zerstört Shading-Effekte

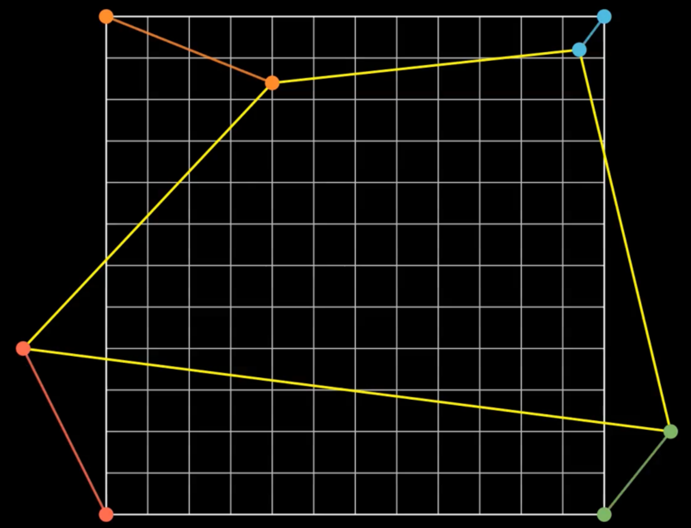
Option B:
1. Render Szene in Buffer (Render Texture)
2. Warp Buffer zu Projektor-Bild

- Nachteil:
  Zusätzlicher Renderpass, Resampling



Ressourcen

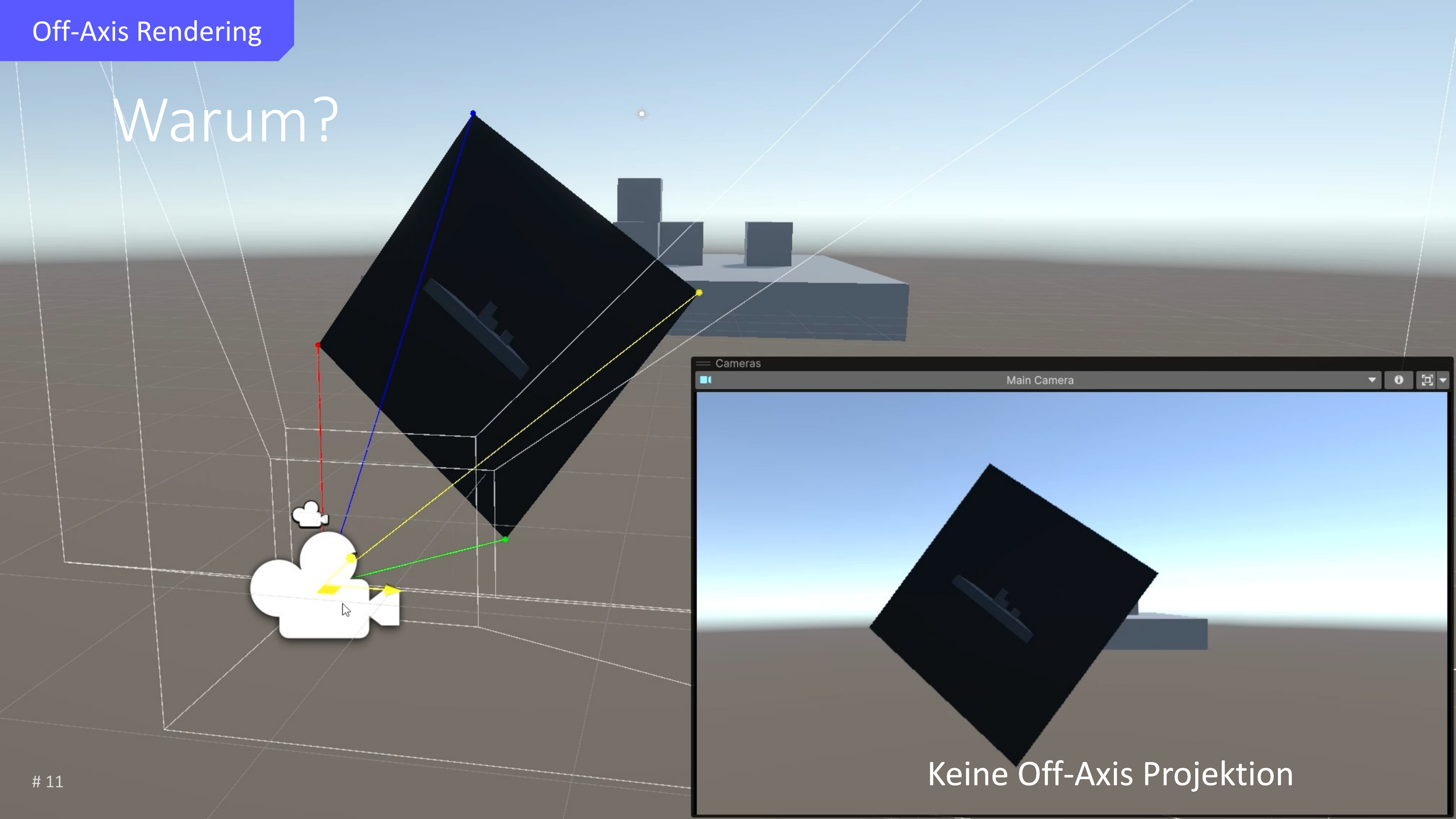
Detailiertes Video zu Homographien aus Computer Vision Sicht:
Youtube Video



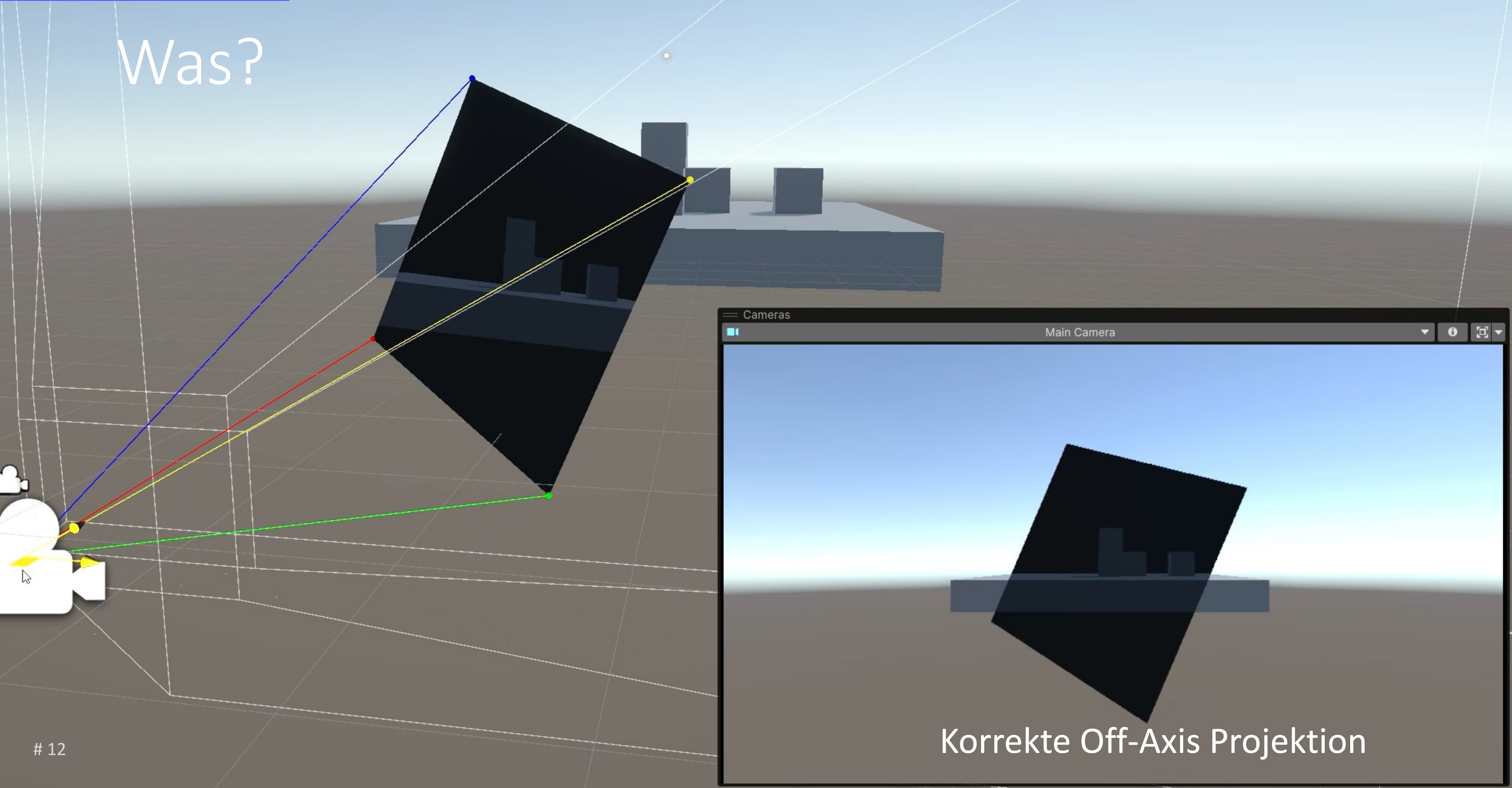
Vorlesungsfolien zu u.a. Image Warping
Slides (Image Warping starts on #52)

Homography Berechnungscode in C#:
Github Gist

Warum?



Was?



Korrekte Off-Axis Projektion

Wie?

l & r und b & t müssen nicht symmetrisch sein, daher auch für Off-Axis Projektion geeignet

```
// gather position and direction vectors
Vector3 pa = window.BottomLeft;
Vector3 pb = ...

Vector3 pe = cam.transform.position;

// direction vectors of virtual screen plane (RIGHT, UP,
NORMAL)
Vector3 vr, vu, vn = ...

// distance from eye to virtual screen PLANE
float d = Vector3.Dot(vn, (pe - pa));
float near, far = ...

// projection of vertical and horizontal frustum limits onto
near-plane
float l, r, b, t = ...
```

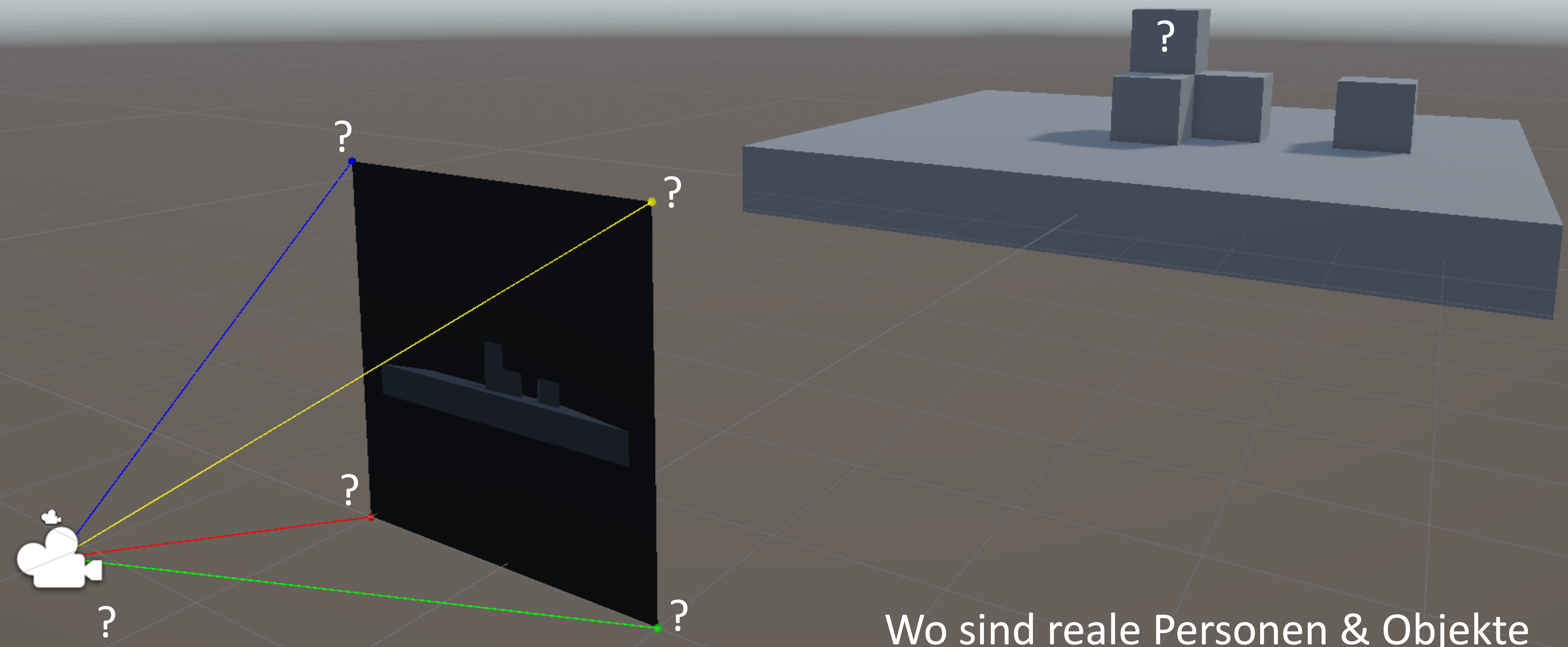
```
// use built-in function for frustum calculation
Matrix4x4 P = Matrix4x4.Frustum(l, r, b, t, near, far);
// setting the new projection matrix for rendering:
cam.projectionMatrix = P;


// translation of camera
Matrix4x4 T_cam = ...

// rotation of window (with flip of handedness)
Matrix4x4 R_window = ...

// setting the new view matrix for rendering:
cam.worldToCameraMatrix = R_window * T_cam;
```

Ressourcen

- Artikel mit Code von Michel de Brisis zu „Off-axis projection in Unity“
- Unity Forum Post
- Blogartikel (nicht in Unity)
- Artikel zu Stereoscopic Rendering von Paul Bourke
- Projection Stage der Rendering Pipeline

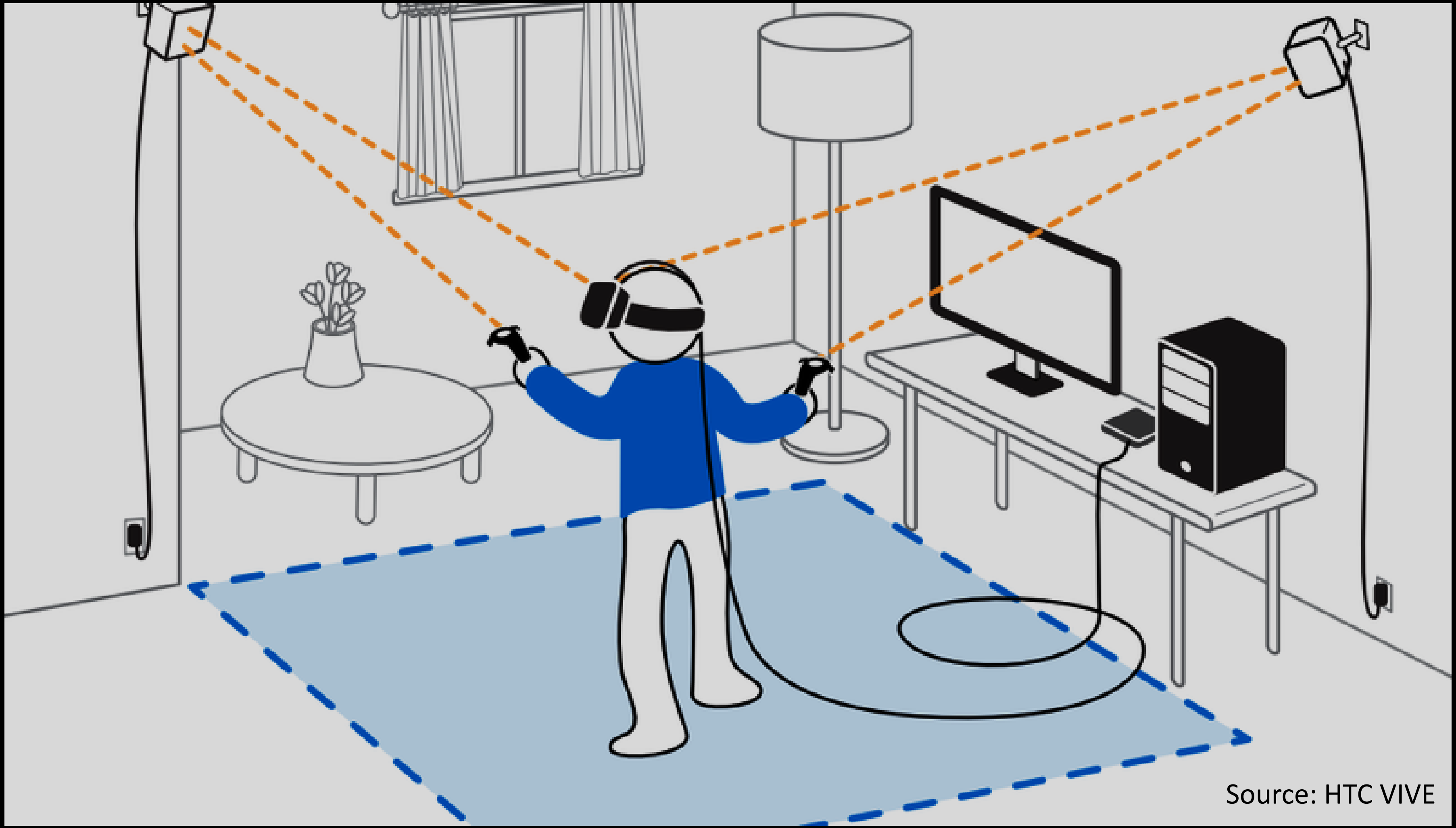
© Michel de Brisis

Warum & Was?



Wo sind reale Personen & Objekte
in der virtuellen Welt?

Wie?



Source: HTC VIVE

Wie?

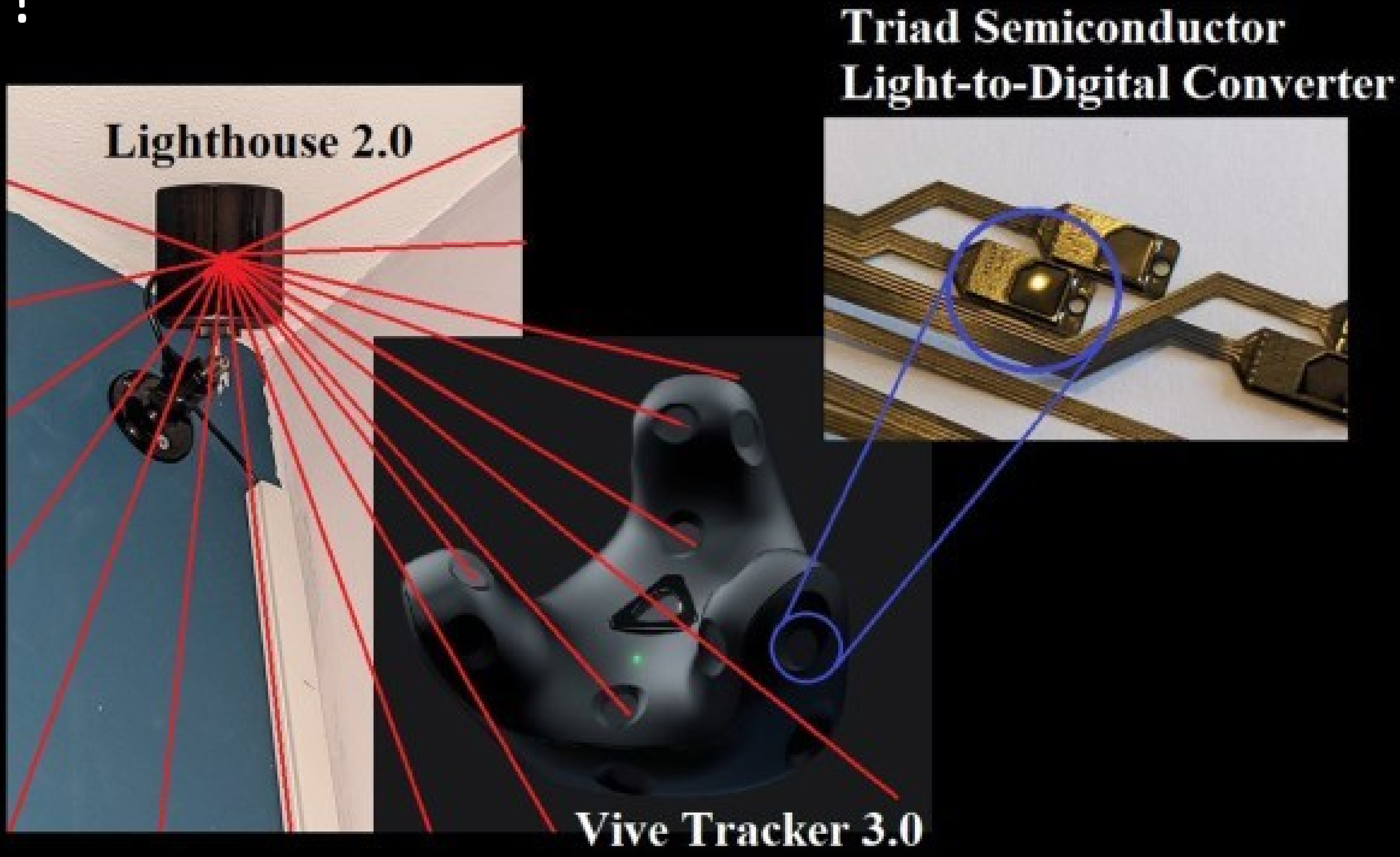


Image by Rob Cole
https://skarredghost.com/2022/12/02/vive-trackers-full-body-vrchat-2/

Ressourcen

- Official VIVE Tracker Guidelines
- Official VIVE Tutorial for Unity

Next steps

- Heute: Themen Verteilung

- Bis nächste Woche:
  - Ziele festlegen, Meilensteine planen
  - In Technical Challenges einlesen
  - Versuchen Lösung in Tasks aufzuteilen

- [optional]: Hands-on Einführung in Unity

Themen Verteilung

- Projektorkalibrierung: 2er Team
- Headtracking: 5er Team
- Off-Axis Rendering: 4er Team