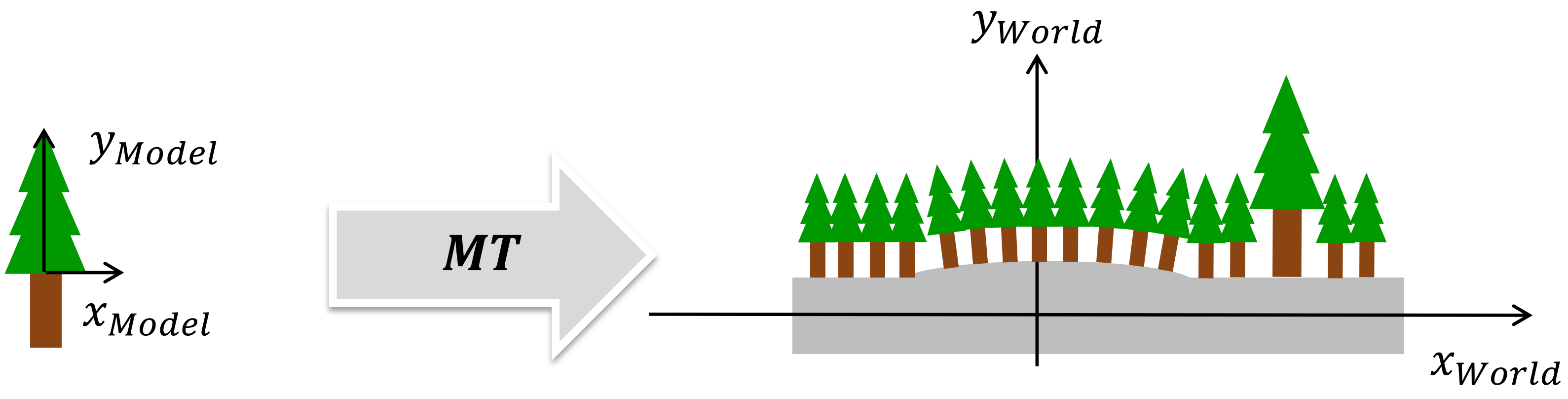
Computergrafik

Universität Osnabrück, Henning Wenke, 2012-05-22

Rückblick

Modeling Transformation



KS-Wechsel der Position & Normale

in

n Vertices	Model Coordinates	
	x	$\mathbb{R}$
	y	$\mathbb{R}$
	z	$\mathbb{R}$
	w	1

⇩

MT

⇩

out

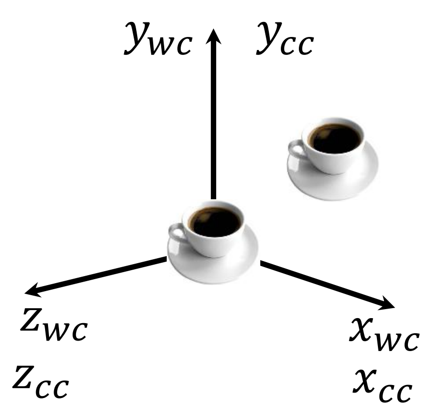
n Vertices	World Coordinates	
	x	$\mathbb{R}$
	y	$\mathbb{R}$
	z	$\mathbb{R}$
	w	1

Kapitel VI:

Viewing Transformation

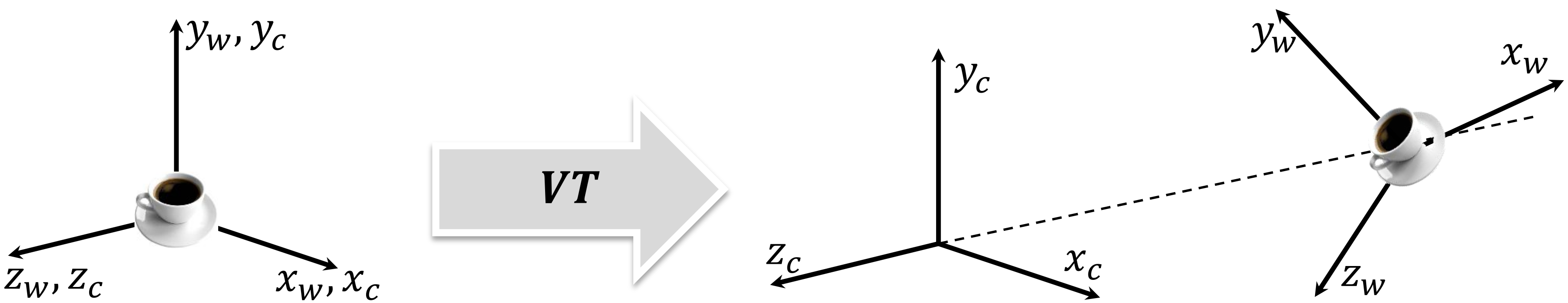
Eingangsinformationen

- Szene, beschrieben in World-Coordinates
- Interessante Objekte befinden sich i.d.R. um den Ursprung verteilt
- Zur Szenenbetrachtung Kamera nötig
  - Definiert durch eigenes KS
- Kamera befindet sich initial ebenfalls im Ursprung des WC-KS und blickt in negative z-Richtung (OpenGL)
  - Warum? Später mehr.
  - Hinweis: Unser Programm hatte bereits Kamera, obwohl nirgends gesetzt!



Ausgangsinformationen

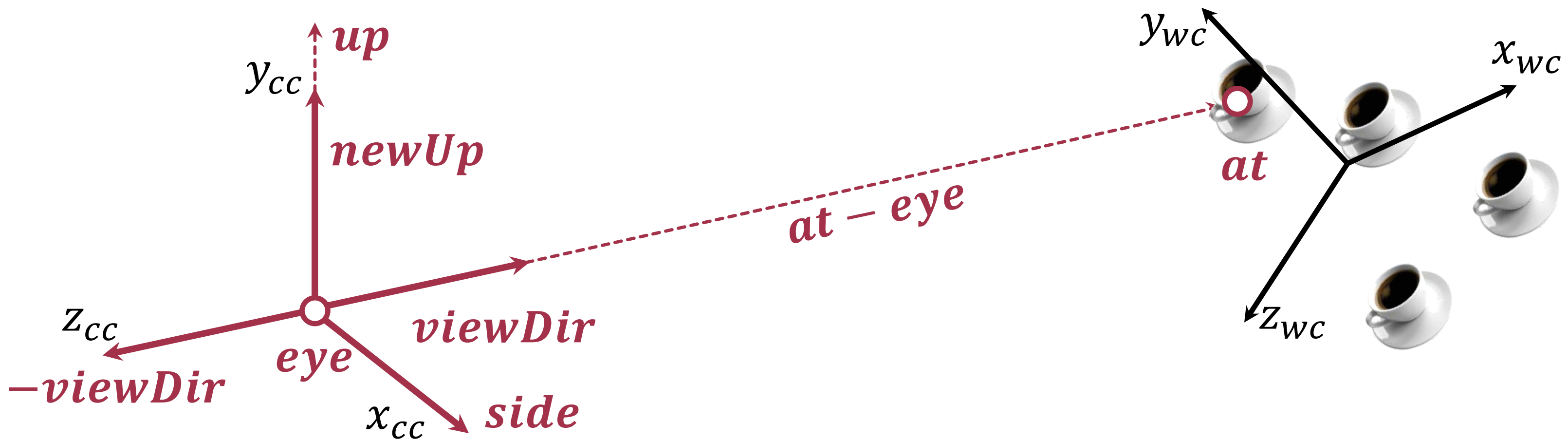
- Ziel: Szenenbetrachtung von außen aus bestimmter Richtung
- Dazu: Wechsel des Koordinatensystems
- Szene dann beschrieben in „Camera-“ oder „Eye Coordinates“
- Interessante Objekte befinden sich i.d.R. in negativer z-Richtung



Ansätze

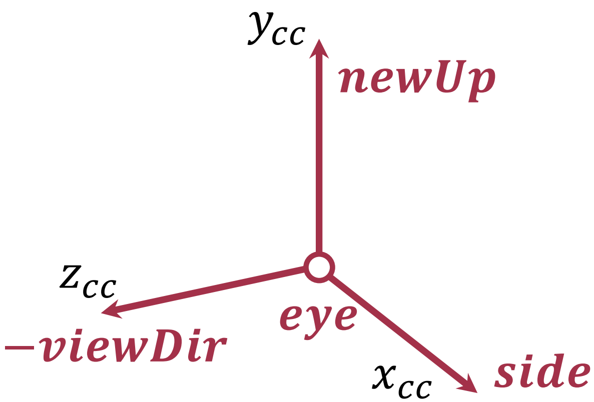
- Manipulation ausgehend vom Ist-Zustand
  - Drehen und verschieben der Szene
  - Oder: Inverse Operationen auf Kamera anwenden
- Alternativ: Gewünschte Ansicht festlegen
  - Ort und Blickrichtung der Kamera festlegen
  - Anschließend Koordinatensystemwechsel
  - Diese Veranstaltung

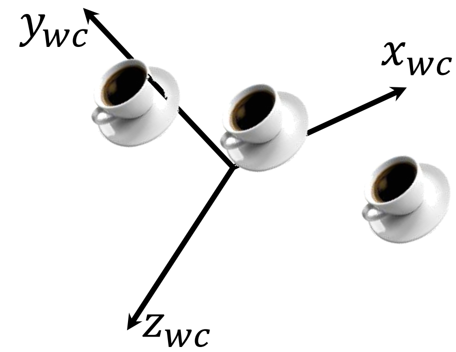
Koordinatensystemwechsel mit „LookAt“



- Definiere Betrachtung der Szene durch
  - $\boldsymbol{at}$: Anvisierter Punkt
  - $\boldsymbol{eye}$: Betrachterstandpunkt
  - $\boldsymbol{up}$: Vektor zur Orientierung der Szene nach oben
- Berechne anschließend
  - $\boldsymbol{-viewDir = -\frac{at-eye}{\|at-eye\|}}$ Vektor gegen Sichtrichtung (norm.), neue z-Achse
  - $\boldsymbol{side = \frac{viewDir \times up}{\|viewDir \times up\|}}$ Vektor zur Seite (normiert), neue x-Achse
  - $\boldsymbol{newUp = \frac{side \times viewDir}{\|side \times viewDir\|}}$ Vektor nach oben (normiert), neue y-Achse

Matrix für „LookAt“





- $$\boldsymbol{M}_{\boldsymbol{Camera}\rightarrow\boldsymbol{World}} = \begin{pmatrix} side_x & newUp_x & -viewDir_x & eye_x \\ side_y & newUp_y & -viewDir_y & eye_y \\ side_z & newUp_z & -viewDir_z & eye_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- $$\boldsymbol{M}_{\boldsymbol{World}\rightarrow\boldsymbol{Camera}} = \boldsymbol{M}^{-1}_{\boldsymbol{Camera}\rightarrow\boldsymbol{World}} =$$

$$\boxed{\begin{pmatrix} side_x & side_y & side_z & -dot(\boldsymbol{eye}, \boldsymbol{side}) \\ newUp_x & newUp_y & newUp_z & -dot(\boldsymbol{eye}, \boldsymbol{newUp}) \\ -viewDir_x & -viewDir_y & -viewDir_z & -dot(\boldsymbol{eye}, -\boldsymbol{viewDir}) \\ 0 & 0 & 0 & 1 \end{pmatrix} = \boldsymbol{M}_{\boldsymbol{V}}}$$

Viewing Transformation im VS

```
#version 330 core
in vec3 normalMC;
in vec4 posMC;

uniform mat4 mc2wc_Pos;
uniform mat3 mc2wc_Normal;

// Viewing Transformation: World Coords -> Camera Coords
uniform mat4 view;
uniform vec3 inverseLightDir;

out float brightness;

void main(){
  // Objekte erst in Szene anordnen, dann Szenenbetrachtung festlegen
  gl_Position    = view * mc2wc_Pos * posMC;
  gl_Position    = mc2wc_Pos * posMC;
  vec3 normalWC = mc2wc_Normal * normalMC;
  brightness = max(dot(normalWC, inverseLightDir), 0.0);
}
```

KS-Wechsel der Position

in

n Vertices	World Coordinates	
	x	$\mathbb{R}$
	y	$\mathbb{R}$
	z	$\mathbb{R}$
	w	1

$\Downarrow$

$$\boldsymbol{M_V}$$

$\Downarrow$

out

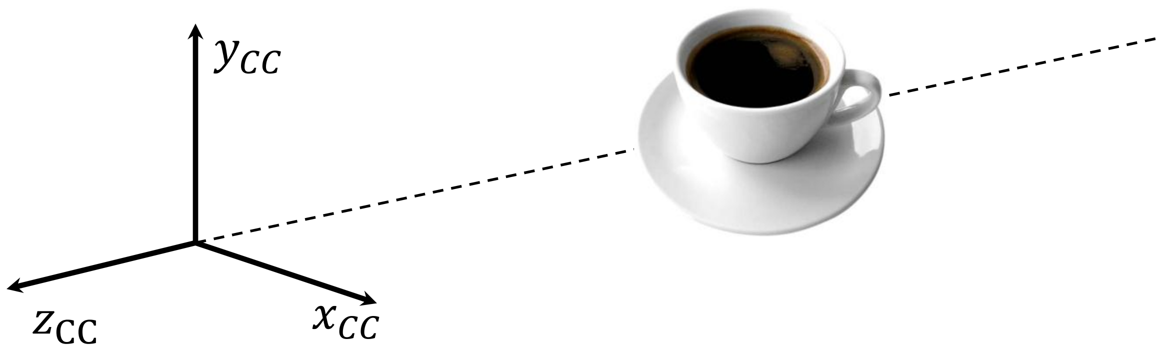
n Vertices	Camera Coordinates	
	x	$\mathbb{R}$
	y	$\mathbb{R}$
	z	$\mathbb{R}$
	w	1

Kapitel VII:

Projection Transformation

Überblick

- Eingangsinformationen
  - Szene, beschrieben in Camera-Coordinates
  - Interessante Objekte befinden sich i.d.R. in negativer z-Richtung

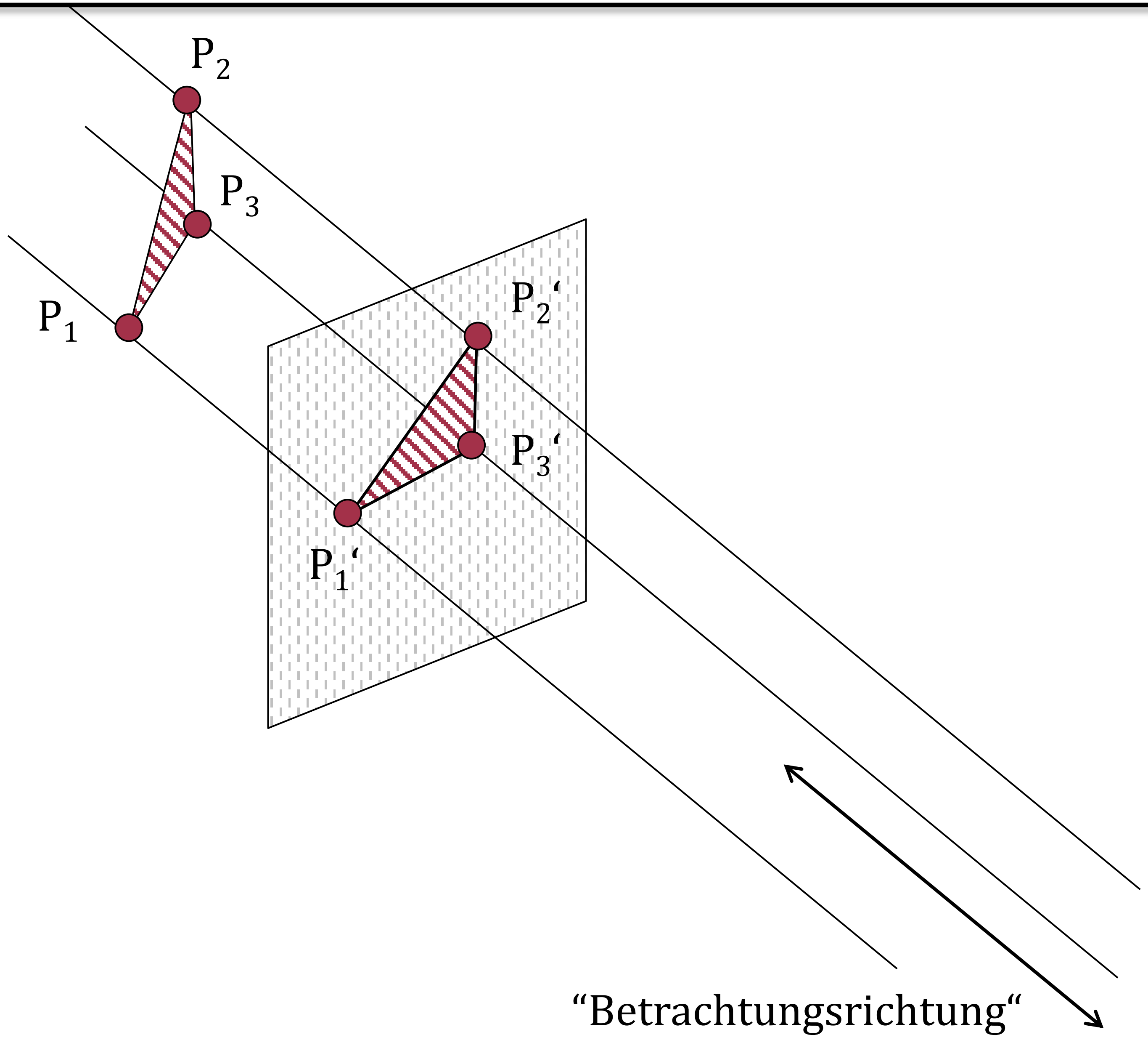


- Ausgabe
  - Szene, beschrieben in
  - „Normalized Device Coordinates“ (Orthogonale Projektion)
  - „Clip Coordinates“ (Perspektivische Projektion)

VII.1

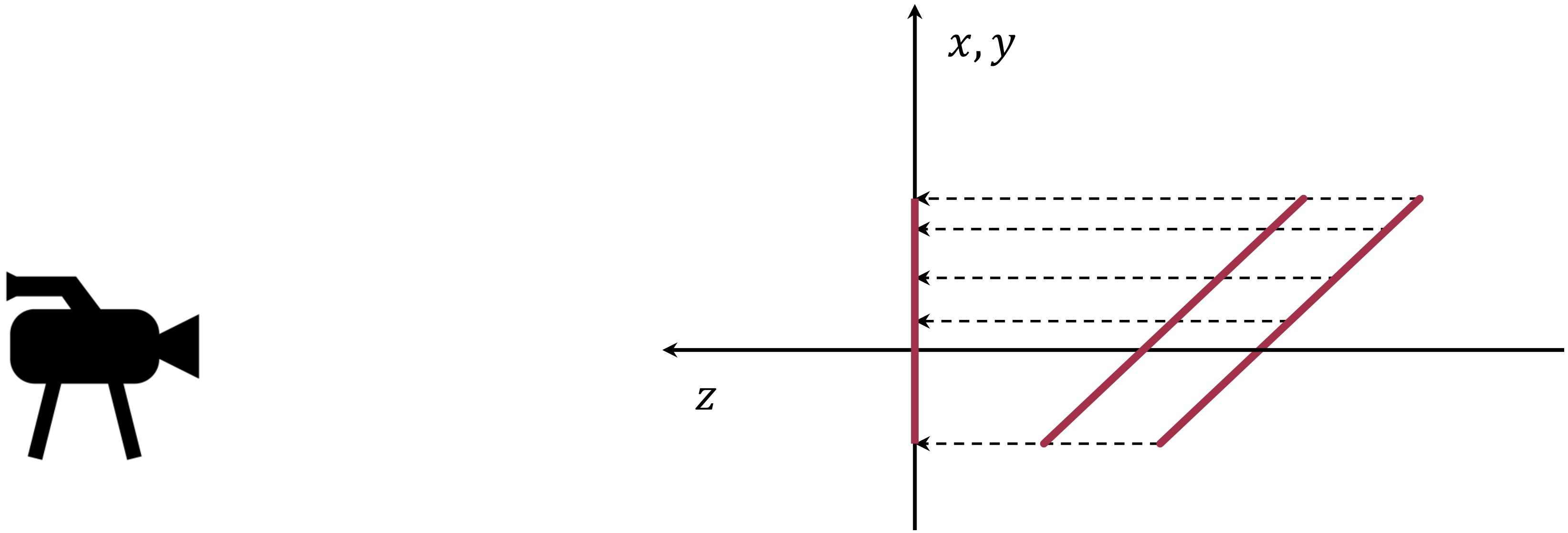
Parallel Projection (“klassischer“ Ansatz)

Parallele Abbildung auf Bildebene



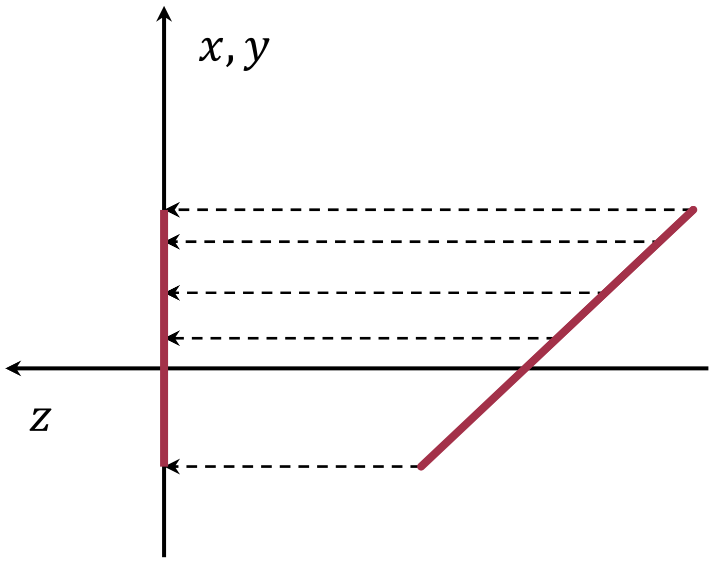
Orthogonale (Parallel) Projektion I

- Betrachter befindet sich im Unendlichen
- Keine Verjüngung der Szene nach hinten
- Hier: Rechtwinklige Parallel Projektion



Orthogonale (Parallel) Projektion II

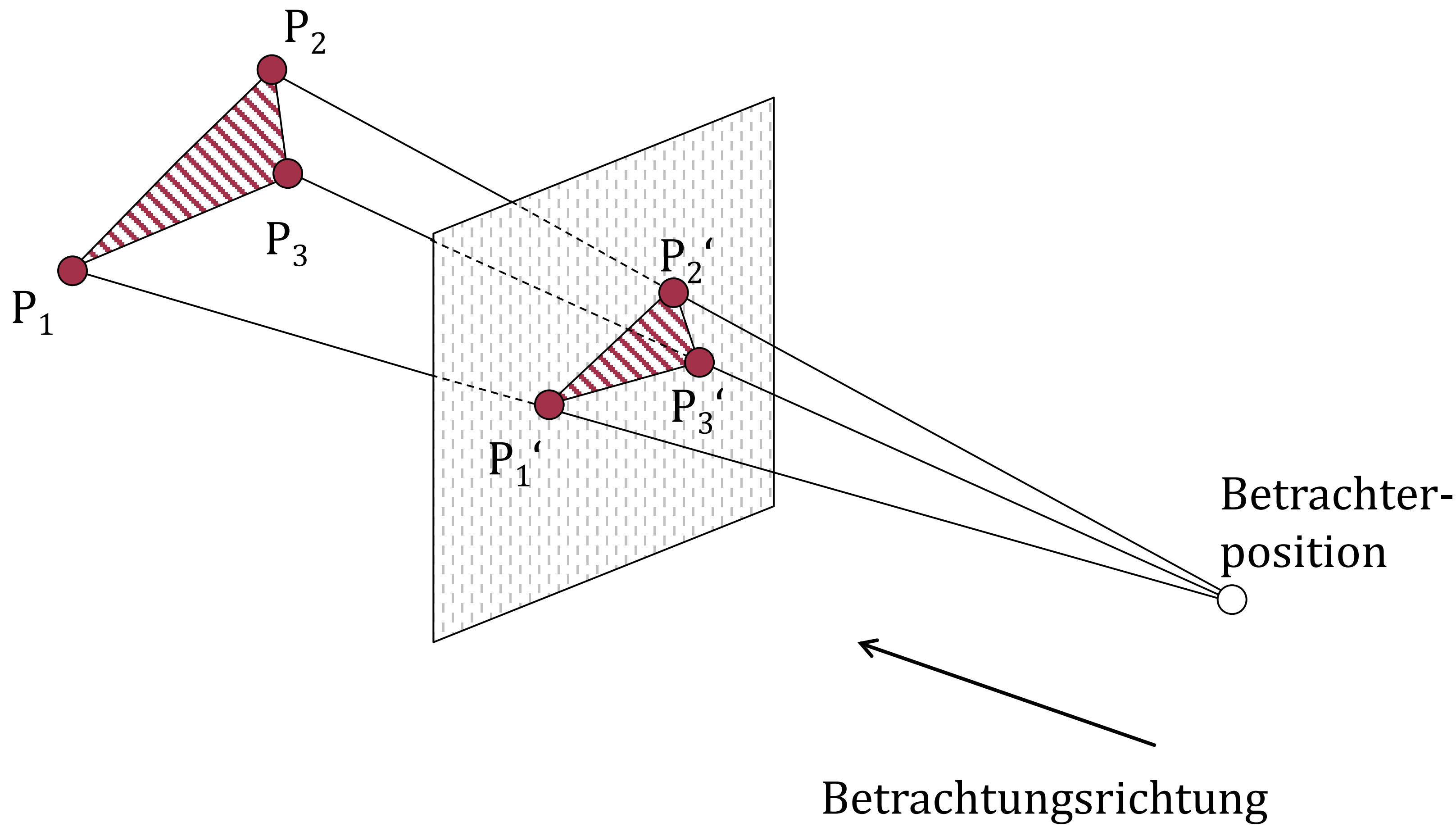
- Parallelprojektion von P ergibt P', mit:
  - $p_x' := p_x$
  - $p_y' := p_y$
  - $p_z' := 0$
  - $p_w' := p_w = 1$
- Matrix für orthogonale Parallelprojektion:
  - $\boldsymbol{P}_o := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$



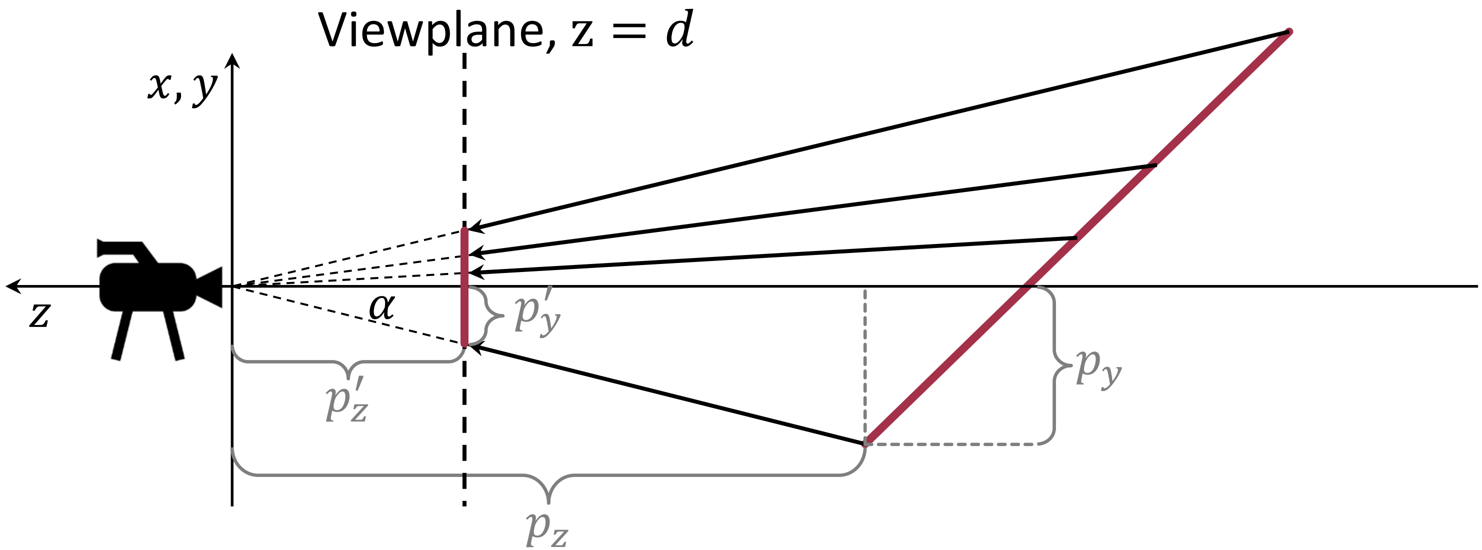
VII.2

Perspective Projection (“klassischer“ Ansatz)

Perspektivische Abbildung auf Bildebene



Koordinatentransformation



- Betrachter befinde sich im Ursprung
- Dann gehen Sichtstrahlen durch diesen
- Gesucht: Schnittpunkte v. Sichtstrahlen und Viewplane
- Es gilt: $\frac{p'_y}{p_y} = \frac{p'_z}{p_z}$, da $\tan(\alpha) = \frac{p'_y}{p'_z}$ und $\tan(\alpha) = \frac{p_y}{p_z}$
  - $p'_z \coloneqq d$
  - $p'_y \coloneqq d\frac{p_y}{z}$
  - $p'_x \coloneqq d\frac{p_x}{z}$ (analog)

Matrix

- Perspektivische Projektion von P ergibt P', mit:
  - $\mathrm{x}' := \frac{d \cdot x}{z}$
  - $\mathrm{y}' := \frac{d \cdot y}{z}$
  - $z' := d$
  - $w' := w = 1$
- Für die perspektivische Projektionsmatrix gilt:

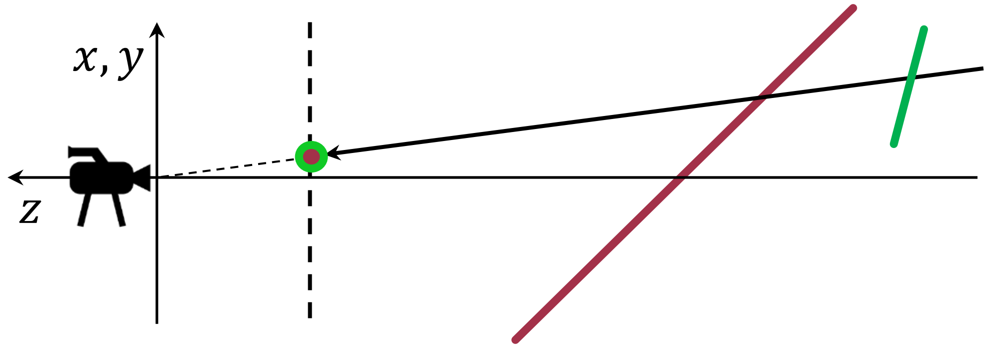
$$\boldsymbol{P_p} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ \frac{z}{d} \end{pmatrix} \rightarrow \begin{pmatrix} d \cdot x/z \\ d \cdot y/z \\ d \\ 1 \end{pmatrix}$$

$$\boldsymbol{P_p} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & {}^{1}/_{d} & 0 \end{pmatrix}$$

Einschränkungen

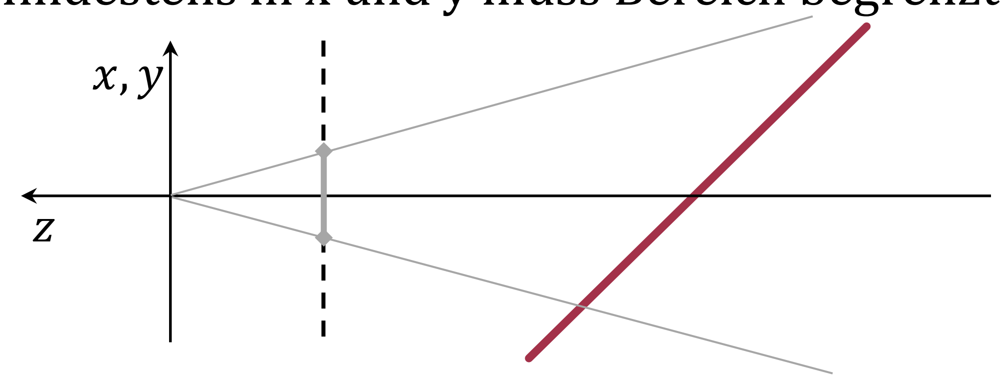
1. Punkte eines Sichtstrahls w. a. identischen Punkt d. Bildebene abgebildet

   ⟹ Tiefenordnung muss festgelegt werden



2. Ausgabemedien haben endliche Ausdehnung

   ⟹ Mindestens in x und y muss Bereich begrenzt werden



Versionshistorie

- 2012-05-22
  - Folie 8 & 9: Üblich ist anstelle des ursprünglich genannten Vektors „$left$“ ein Vektor „$side$“ der $-left$ entspricht und die x-Achse des Kamera Koordinatensystems darstellt. Dadurch ändert sich die Händigkeit des KS bei dieser Transformation an dieser Stelle noch NICHT.
  - Folie 23 und danach: Entfernt. Wird überarbeitet und am 29.5 erneut vorgestellt