



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik
Arbeitsbereich SAV/BV (KOGS)

Grundlagen der Signalverarbeitung und Robotik

Teil 1: Grundlagen der Signalverarbeitung

Vorlesung 6: Perspektivische Abbildungen

Benjamin Seppke

Jianwei Zhang

Perspektivische Abbildung

Perspektivische Projektion ist ein wesentlicher Bestandteil von bildgebenden Systemen und Computergrafik: Erzeugen eines 2D-Bildes aus einer 3D-Vorlage.

Für Bildanalyse und Bilddeutung muss der inverse Prozess geleistet werden: Rekonstruktion einer 3D-Szene aus einem 2D-Bild.

Wir behandeln zunächst:

Wo im Bild erscheint ein Punkt einer 3D-Szene?

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \xrightarrow{\text{?}} \begin{bmatrix} x'' \\ y'' \end{bmatrix}$$

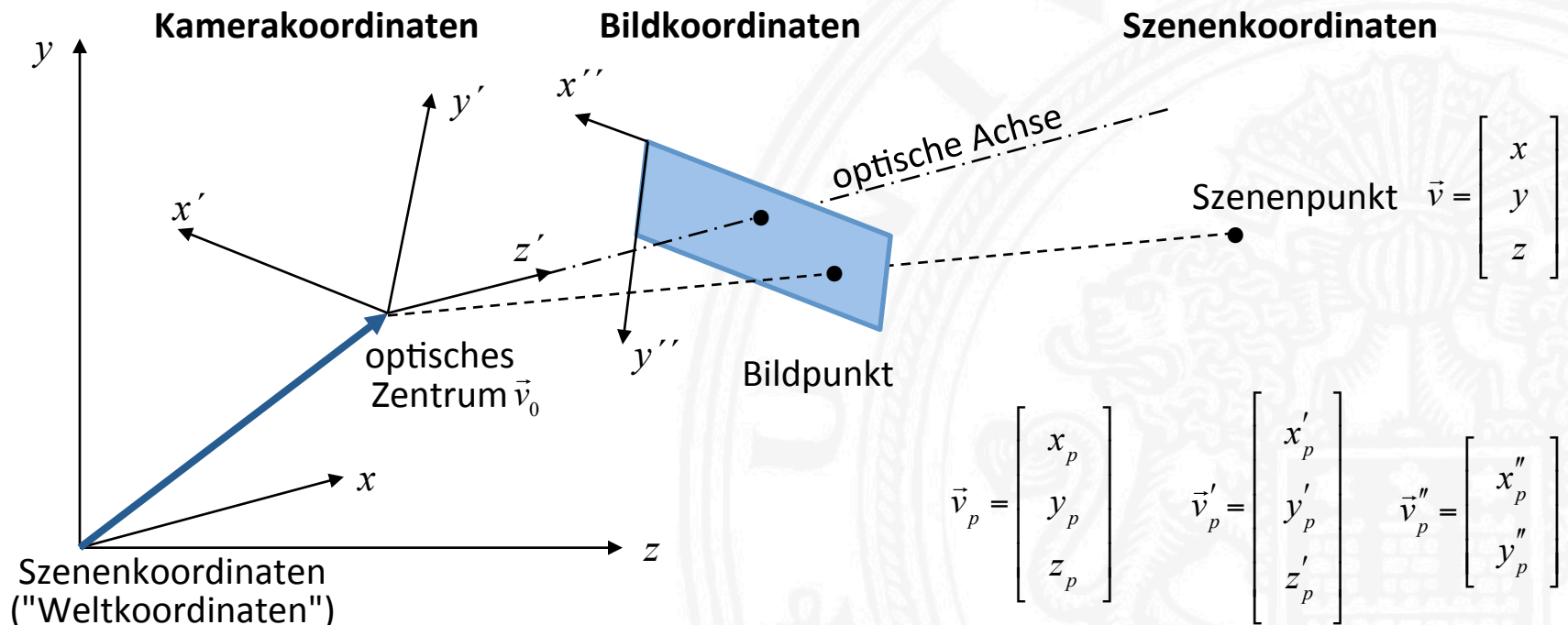
Transformation in 3 Stufen:

1. Szenenkoordinaten \rightarrow Kamerakoordinaten
2. Projektion von Kamerakoordinaten in die Bildebene
3. Kamerakoordinaten \rightarrow Bildkoordinaten

Perspektivische Projektion mit drei unabhängigen Koordinatensystemen

Es ist häufig nützlich, die Realwelt, die Kamerageometrie und die Bildpunkte in getrennten Koordinatensystemen zu beschreiben.

Die formale Beschreibung der perspektivischen Abbildung beinhaltet die Transformationen zwischen diesen Koordinatensystemen.



3D → 3D Koordinatentransformation (1)

Das neue Koordinatensystem wird durch Translation und Rotation des alten Koordinatensystems bestimmt:

$$\vec{v}' = R (\vec{v} - \vec{v}_0) \quad \begin{array}{l} \vec{v}_0 \text{ Optisches Zentrum} \\ R \text{ Rotationsmatrix} \end{array}$$

R kann in 3 aufeinanderfolgende Rotationen um die jeweils aktuellen Koordinatenachsen zerlegt werden: $R = R_x R_y R_z$

Für Rotationen in dieser Reihenfolge:

- 1) γ = Rotationswinkel um die z-Achse "Kippwinkel"
- 2) β = Rotationswinkel um die (neue) y-Achse "Schwenkwinkel"
- 3) α = Rotationswinkel um die (neue) x-Achse "Neigungswinkel"

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Die hier gezeigte Matrizen beschreiben die Drehung von Koordinaten-Systemen, nicht die Drehung eines Punktes in einem Koordinaten-System!

3D → 3D Koordinatentransformation (2)

Durch Multiplizieren der 3 Matrizen R_x , R_y und R_z erhält man

$$R = \begin{bmatrix} \cos \beta \cos \gamma & \cos \beta \sin \gamma & -\sin \beta \\ \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \cos \beta \\ \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \cos \beta \end{bmatrix}$$

In Formeln und für Ableitungen verwendet man meist

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Eine Rotationsmatrix hat nur 3 Freiheitsgrade, d.h. die 9 Koeffizienten müssen zusätzlichen Bedingungen genügen:

Orthonormalität: $R R^T = I$ (Einheitsmatrix)

Beispiel für eine Koordinatentransformation

Kamerakoordinatensystem :

- Translation um \vec{v}_0
- Rotation um Schwenkwinkel $\beta = -30^\circ$
- Rotation um Neigungswinkel $\alpha = 45^\circ$

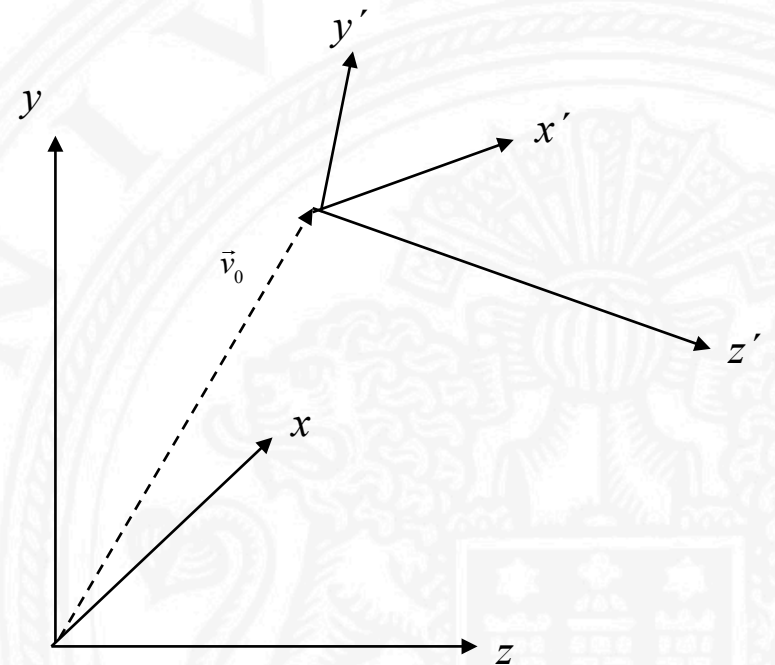
Abbildung von:

$$\vec{v}' = R (\vec{v}' - \vec{v}_0) \quad \text{mit } R = R_x R_y$$

wobei:

$$R_x = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 \\ 0 & \sqrt{2} & \sqrt{2} \\ 0 & -\sqrt{2} & \sqrt{2} \end{bmatrix}$$

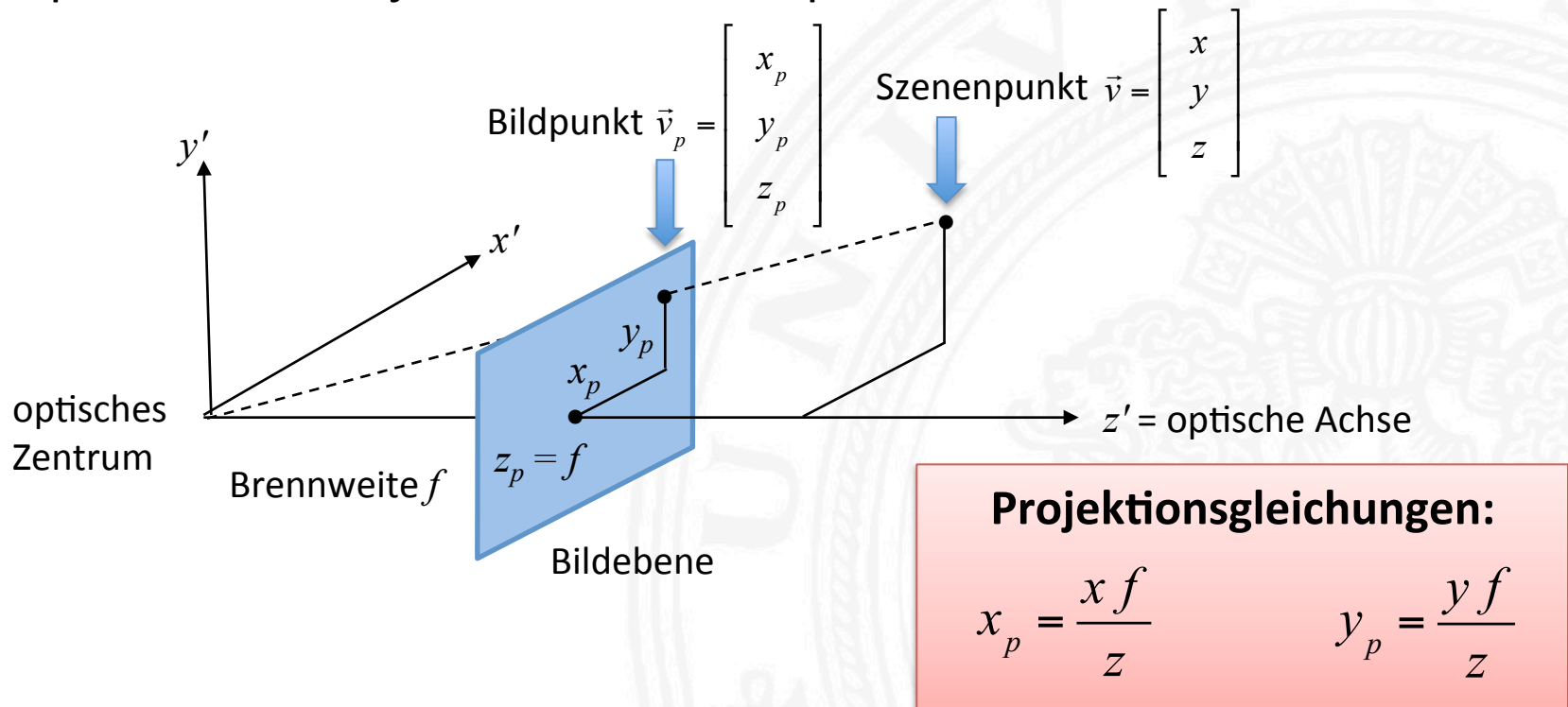
$$R_y = \frac{1}{2} \begin{bmatrix} \sqrt{3} & 0 & 1 \\ 0 & 2 & 0 \\ -1 & 0 & \sqrt{3} \end{bmatrix}$$



Perspektivische Projektion

Perspektivische Projektion bildet einen Szenenpunkt in einen Punkt in der Bildebene ab.

Perspektivische Projektion ist ein adäquates Modell für viele Kameras.



Perspektivische und orthographische Projektion

Perspektivische Projektion:

- Projektionsgleichungen

$$x_p = \frac{x f}{z} \quad y_p = \frac{y f}{z} \quad z_p = f \quad (f = \text{Brennweite})$$

- nichtlineare Transformation
- Informationsverlust

Für entfernte Objekte mit geringer Ausdehnung in Tiefenrichtung ist f/z etwa konstant. Die Abbildung kann durch orthographische Projektion beschrieben werden:

$$x_p = s x \quad y_p = s y \quad z_p = f \quad (s = \text{Skalierungsfaktor})$$

Orthographische Projektion kann als Projektion mit parallelen Abbildungsstrahlen modelliert werden, gefolgt von einer Skalierung.

Wegen der Linearität ist orthographische Projektion für formale Analysen beliebt.

Von Kamerakoordinaten zu Bildkoordinaten

Transformation kann erforderlich sein, weil

- die optische Achse die Bildebene nicht am gewünschten Ursprung des Bildkoordinatensystems durchstößt,
- der Übergang zu diskreten Koordinaten mit einer Skalierung verbunden ist.

$$x''_p = (x'_p - x'_{p_0})a$$

a, b Skalierungsparameter

$$y''_p = (y'_p - y'_{p_0})b$$

x'_{p_0}, y'_{p_0} Ursprung des Bildkoordinatensystems

Beispiel:

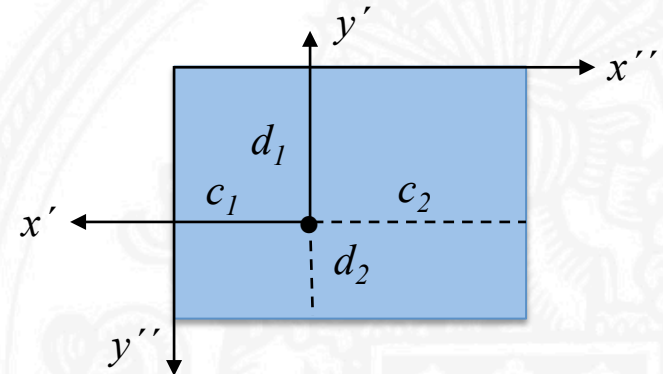
- Bildgrenzen in Kamerakoordinaten:

$$x'_{max} = c_1 \quad x'_{min} = c_2$$

$$y'_{max} = d_1 \quad y'_{min} = d_2$$

- Diskrete Bildkoordinaten:

$$x'' = 0 \dots 511 \quad y'' = 0 \dots 575$$



Transformationsparameter: $x'_{p_0} = c_1, \quad y'_{p_0} = d_1, \quad a = \frac{512}{c_2 - c_1}, \quad b = \frac{576}{d_2 - d_1}$

Vollständige perspektivische Abbildungsgleichungen

Verbindung von 3 Transformationsschritten:

1. Szenenkoordinaten => Kamerakoordinaten
2. Projektion von Kamerakoordinaten in die Bildebene
3. Kamerakoordinaten => Bildkoordinaten

$$x_p'' = \left[\frac{f}{z'} \left(\cos(\beta) \cos(\gamma)(x - x_0) + \cos(\beta) \sin(\gamma)(y - y_0) + \sin(\beta)(z - z_0) \right) - x_{p_0} \right] a$$

$$y_p'' = \left[\frac{f}{z'} \left(\begin{array}{l} (-\sin(\alpha) \sin(\beta) \cos(\gamma) - \cos(\alpha) \sin(\gamma))(x - x_0) \\ (-\sin(\alpha) \sin(\beta) \sin(\gamma) + \cos(\alpha) \cos(\gamma))(y - y_0) \\ + \sin(\alpha) \cos(\beta)(z - z_0) \end{array} \right) - y_{p_0} \right] b$$

$$\text{mit: } z' = \left(\begin{array}{l} (-\cos(\alpha) \sin(\beta) \cos(\gamma) + \sin(\alpha) \sin(\gamma))(x - x_0) \\ + (-\cos(\alpha) \sin(\beta) \sin(\gamma) - \sin(\alpha) \cos(\gamma))(y - y_0) \\ + \cos(\alpha) \cos(\beta)(z - z_0) \end{array} \right)$$

Homogene Koordinaten (1)

4D-Notation für 3D-Koordinaten, mit der nichtlineare 3D-Transformation als lineare 4D-Transformationen ausgedrückt werden können

- Normal (3D): $\vec{v}' = R_{3 \times 3}(\vec{v} - \vec{v}_0)$
- Homogene Koordinaten: $\vec{v}' = R_{4 \times 4} T_{4 \times 4} \vec{v} = A_{4 \times 4} \vec{v}$

$$R_{4 \times 4} T_{4 \times 4} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Übergang zu homogenen Koordinaten:

$$\vec{v}^T = [x \ y \ z] \xrightarrow{\text{affin}} (\vec{v}_4)^T = [wx \ wy \ wz \ w] \quad w \neq 0 \text{ ist beliebige Konstante}$$

Rückkehr zu normalen Koordinaten:

- (i) Komponenten 1 bis 3 durch 4. Komponente dividieren
- (ii) 4. Komponente weglassen

Homogene Koordinaten (2)

Perspektivische Projektion in homogenen Koordinaten:

$$\vec{v}'_p = P_{4 \times 4} \vec{v}' \quad \text{mit} \quad P_{4 \times 4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \quad \text{und} \quad \vec{v}'_4 = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} \quad \text{ergibt:} \quad \vec{v}_{p,4} = \begin{bmatrix} wx \\ wy \\ wz \\ \frac{wz}{f} \end{bmatrix}$$

Rückkehr zu normalen Koordinaten ergibt: $v'_p = \begin{bmatrix} \frac{x_f}{z} \\ \frac{y_f}{z} \\ f \end{bmatrix}$

**Vergleiche mit
vorher angegebenen
Formeln!**

Transformation von Kamera- in Bildkoordinaten:

$$\vec{v}''_{p,4} = B_{4 \times 4} \vec{v}'_{p,4} \quad \text{mit} \quad B_{4 \times 4} = \begin{bmatrix} a & 0 & 0 & -x_0 a \\ 0 & b & 0 & -y_0 b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{und} \quad \vec{v}'_{p,4} = \begin{bmatrix} wx_p \\ wy_p \\ 0 \\ w \end{bmatrix} \quad \text{ergibt:} \quad \vec{v}''_{p,4} = \begin{bmatrix} wa(x_p - x_0) \\ wa(y_p - y_0) \\ 0 \\ w \end{bmatrix}$$

Homogene Koordinaten (3)

Die perspektivische Transformation von Szenenkoordinaten in Bildkoordinaten kann vollständig durch lineare Transformationen in homogenen Koordinaten beschrieben werden:

$$\vec{v}_{p,4}'' = B_{4 \times 4} R_{4 \times 4} R_{4 \times 4} T_{4 \times 4} \vec{v}_4$$

$B_{4 \times 4} R_{4 \times 4} R_{4 \times 4} T_{4 \times 4}$ kann in einer einzigen 4x4-Matrix C kombiniert werden:

$$\vec{v}_{p,4}'' = C_{4 \times 4} \vec{v}_4$$

In der Literatur unterscheiden sich die Parameter von C wegen unterschiedlicher Koordinatensysteme, Folge von Rotationen, Kameramodellen, etc.

Inverse perspektivische Abbildung

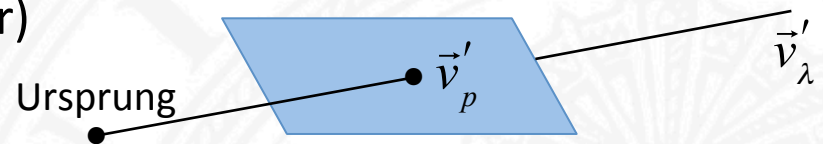
Welcher 3D-Szenenpunkt korrespondiert zu einem Bildpunkt?



Jeder Bildpunkt definiert einen Abbildungsstrahl, auf dem der Szenenpunkt liegen muss (zur besseren Klarheit in Kamerakoordinaten beschrieben):

$$\vec{v}'_p \rightarrow \vec{v}'_\lambda = \lambda \vec{v}'_p \quad (\lambda \text{ ist freier Parameter})$$

$$\vec{v} = \vec{v}_0 + R^T \lambda \vec{v}'_p$$

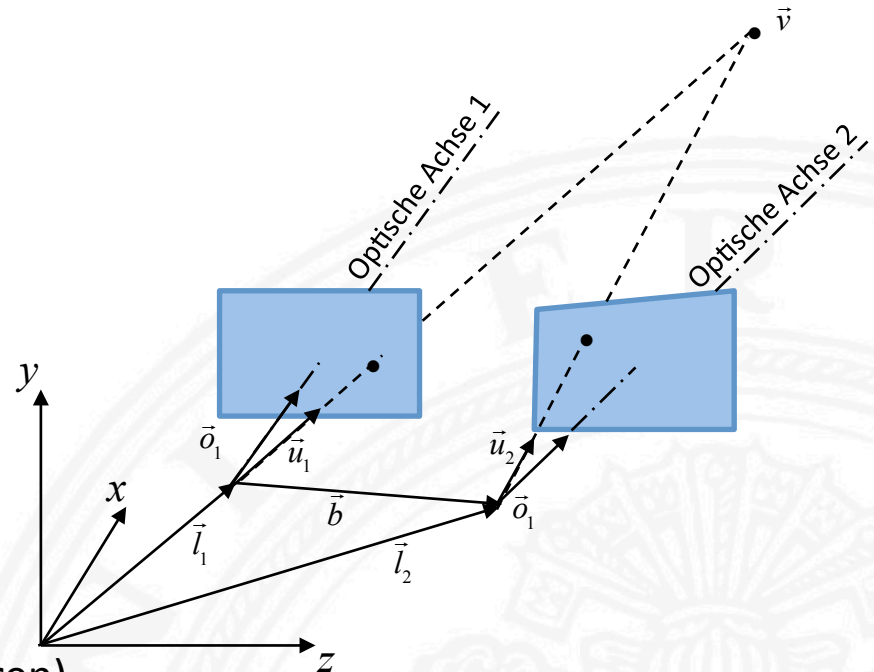


Ergebnis: 3 Gleichungen mit den 4 Unbekannten x , y , z , λ und Kameraparametern R und \vec{v}_0

Anwendungen der inversen perspektivischen Abbildung:

- Abstandsmessung
- Binokularstereo, Bewegungstereo
- Kamerakalibrierung

Binokularstereo (1)



\vec{l}_1, \vec{l}_2 Kamerapositionen (optische Zentren)

\vec{b} Stereobasis

\vec{o}_1, \vec{o}_2 Kameraorientierungen (Einheitsvektoren)

\vec{f}_1, \vec{f}_2 Brennweiten

\vec{v} Szenenpunkt

\vec{u}_1, \vec{u}_2 Abbildungsstrahlen des Szenenpunktes (Einheitsvektoren)

Binokularstereo (2)

Der Abstand zu \vec{v} wird durch Messen von \vec{u}_1 und \vec{u}_2 bestimmt

$$\text{Formell: } \alpha \vec{u}_1 = \vec{b} + \beta \vec{u}_2 \quad \Rightarrow \quad \vec{v} = \alpha \vec{u}_1 + \vec{l}_1$$

α und β spezifizieren die Position des Szenenpunktes of den Abbildungsstrahlen, sind aber durch die Vektorgleichung überbestimmt.

Da Messungen in der Praxis ungenau sind, können die Gleichungen inkonsistent sein und keine Lösung ergeben.

Besserer Ansatz: Bestimme den Punkt der dichtesten Annäherung beider Abbildungsstrahlen:

$$\vec{v} = \frac{\alpha_0 \vec{u}_1 + (\vec{b} + \beta_0 \vec{u}_2)}{2} + \vec{l}_1 \quad \rightarrow \quad \text{minimiere: } \left\| \alpha_0 \vec{u}_1 + (\vec{b} + \beta_0 \vec{u}_2) \right\|^2$$

Lösung:

$$\alpha_0 = \frac{\vec{u}_1^T \vec{b} - (\vec{u}_1^T \vec{u}_2)(\vec{u}_2^T \vec{b})}{1 - (\vec{u}_1^T \vec{u}_2)} \quad \beta_0 = \frac{(\vec{u}_1^T \vec{u}_2)(\vec{u}_1^T \vec{b}) - (\vec{u}_2^T \vec{b})}{1 - (\vec{u}_1^T \vec{u}_2)^2}$$