Kamera-Kalibration

FH OÖ Studiengänge • Hagenberg • Linz • Steyr • Wels

Inhalt

Einfache Abbildungsgeometrie

- Camera Obscura
  - Koordinatensystem
- Einfache Abbildungsgleichung

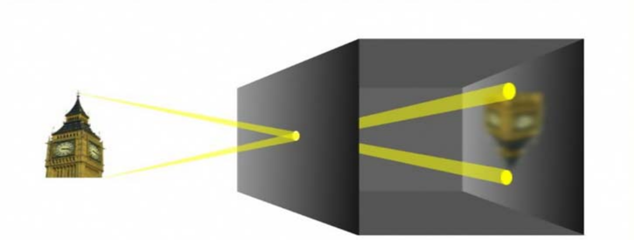
Kamerakalibrierung

- Methode nach Tsai
  - Welt- und Kamerakoordinaten
  - äußere uns innere Parameter

Abbildungssysteme

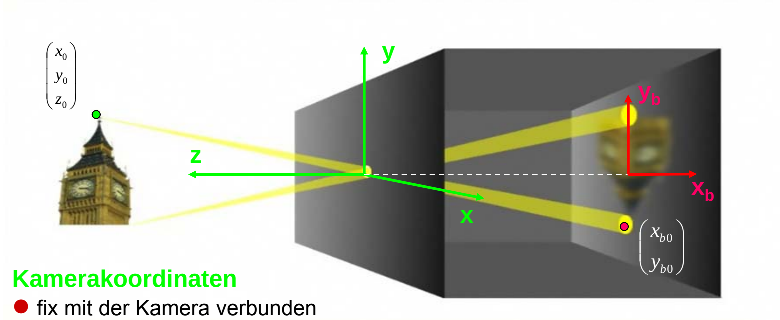
Camera obscura

- einfachstes Abbildungssystem
- bekannt seit dem Altertum
- Licht fällt durch eine Lochblende in das Innere einer abgedunkelten Kammer (camera obscura)
- Objekt wird an der Hinterwand abgebildet



Camera Obscura

Koordinatensysteme



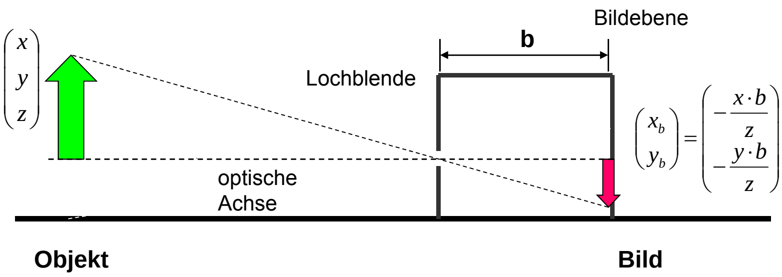
Kamerakoordinaten
- fix mit der Kamera verbunden
- drei dimensional (3D)
- orthonormal
- rechtshändig

Bildkoordinaten
- mit Bildebene verbunden
- zweidimensional
- Projektions-Geometrie

Abbildungssysteme

Camera Obscura

- Abbildung durch Lochblende
- verkehrt und verkleinert



Projektionsgeometrie

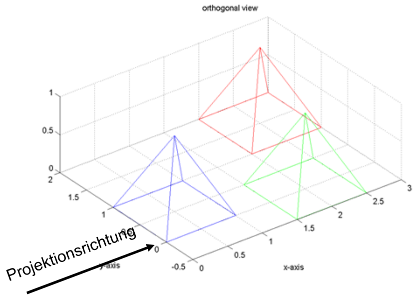
Schlussfolgerungen

$$\begin{pmatrix} -\frac{x \cdot b}{z} \\ -\frac{y \cdot b}{z} \end{pmatrix}$$

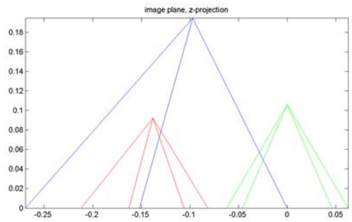
- Projektion eines 3D Raumes auf eine Ebene
- negatives Vorzeichen, Spiegelung (Inversion) an der optischen Achse
- z<b: Objektabstand kleiner als Bildweite
  - Objekt wird vergrößert dargestellt
- z>b: Objektabstand größer als Bildweite
  - Objekt wird verkleinert dargestellt
  - entferntere Objekte werden kleiner abgebildet -> Grundlage für 3D Wahrnehmung

Beispiele

Drei gleich große Pyramiden



Orthoprojektion



Zentralprojektion

Kamera-Kalibration

Abbildungen durch Kamerasysteme (Video, Photo) unterliegen Verzerrungen bedingt durch

- perspektivische Projektion
- Abbildungsfehler der Linsen
  - z.B. achsenferne Strahlen werden stärker gebrochen als achsennahe

Kalibration der Kamera für:

- exakte Datenerhebung aus Abbildungssystemen, z.B. forensische Auswertung von Unfallsphotos
- Normalisierung von Portraitdatenbanken ...

Algorithmus nach Tsai

Beispiel

Beispiel für Verzerrung durch die Kamera

Ein paralleler Schienenstrang scheint auf einem Photo auf einen gemeinsamen Punkt zusammenzulaufen.

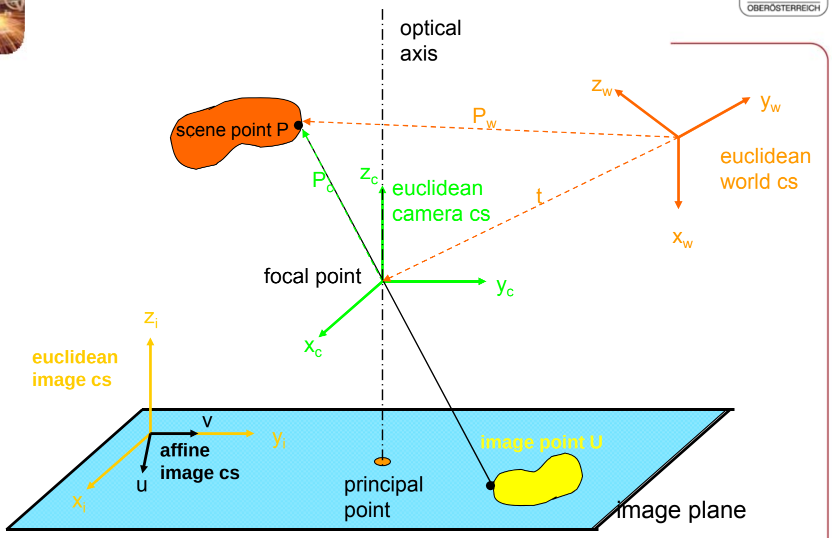
-> Ursache: Zentralprojektion in der Kamera und auch im menschlichen Auge

entfernte Objekte werden kleiner abgebildet.

Zentralprojektion

parallele Linien schneiden sich im Unendlichen

Koordinatensysteme



4 Koordinaten-Systeme

Euklidische Systeme

- Weltkoordinaten (x_w, y_w, z_w)
  - reale Objekte, z.B. Position der Schienen
- Kamerakoordinaten (x_c, y_c, z_c)
  - System fest mit der Blende (focal point) verbunden
- Euklidsche Bildkoordinaten (x_i, y_i, z_i)
  - System bezogen auf die Bildebene (3D)

Affines System

- 2D Bildkoordeinaten (u,v)
  - berücksichtigen Verzerrungen in der Abbildung

Kamera Kalibrierung

Transformationsreihenfolge

- Die Transformation vom Punkt P_w in Weltkoordinaten zum Bildpunkt U in den affinen Bildkoordinaten
  - Weltkoordinaten -> Kamerakoordinaten
  - Kamerakoordinaten -> Euklidische Bildkoordinaten
  - Euklidische Bildkoordinaten –> affine Koordinaten

Operationen zur Koordinatentransformation

- Transformationen in 3D Koordinatensystemen
  - Rotation
  - Translation
  - Skalierung
  - Scherung

Geometrische Transformationen

Koordinaten im Ursprungsbild *(x,y,z)*

Koordinaten im transformierten Bild *(u,v,w)*

Transformation der einzelnen Komponenten

$$u = T_u(x,y,z)$$
$$v = T_v(x,y,z)$$
$$w = T_w(x,y,z)$$

Rücktransformation durch Inversion der Vektorfunktion **T**, d.h. **T** muss eindeutig sein

Zusammenfassen zu einer Vektorfunktion

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \vec{T}(x,y,z)$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \vec{T}^{-1}(u,v,w)$$

Matrixdarstellung

Lineare geometrische Transformation

$$\vec{T}(a\vec{x}+b\vec{y}) = a\vec{T}(\vec{x})+b\vec{T}(\vec{y})$$

$$\vec{x} = (x_1, x_2, x_3)$$

Darstellung als Matrix

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Berechnung:

$$v = t_{21} \cdot x + t_{22} \cdot y + t_{23} \cdot z$$

Eigenschaften

vertauschen i.A. nicht

$$T_1 \cdot T_2 \cdot \vec{x} \neq T_2 \cdot T_1 \cdot \vec{x}$$

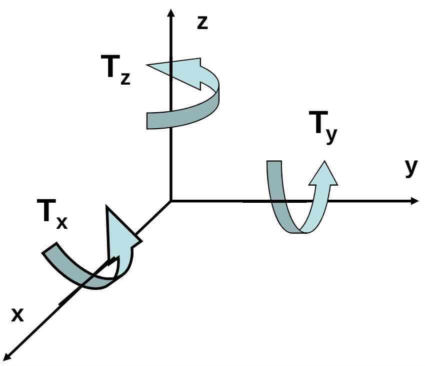
Hintereinanderausführung

$$(T_1 \cdot T_2) \cdot T_3 \cdot \vec{x} = T_1 \cdot (T_2 \cdot T_3) \cdot \vec{x}$$

z.B. Zusammensetzung einer Rotation in 3D aus Rotationen um die Koordinatenachsen

Rotation in 3D

Rotationsmatrix T wird als Hintereinanderausführung der Rotationen um die einzelnen Koordinatenachsen T_x,T_y, T_z zusammengesetz

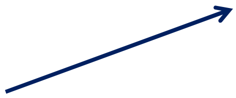


Transformationen - Vektoren

Ein Vektor ist eine gerichtete Größe, Beispiele dafür sind:

- Der Wind wird durch Windstärke und –richtung beschrieben.
- Geschwindigkeit besitzt Größe und Richtung
- Weg zwischen zwei Punkten, besitzt Richtung und Länge des Weges

- Eine vektorielle Größe wird durch eine Pfeil dargestellt.
  - Die Richtung ist durch die Orientierung des Vektorpfeils gegeben.
  - Die Länge des Pfeils gibt den Betrag der Vektors an.



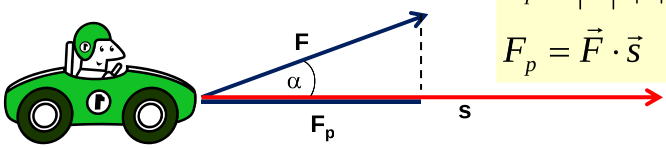
Vektoren - Skalarprodukt

Arbeit = Kraft x Weg

- Wenn die Kraft jedoch nicht in Richtung des Weges weist, so wird nur jene Komponente des Kraftvektors **F** in Richtung des Verschiebevektors **s** zur Berechnung herangezogen.
- F_p entspricht der Projektion des Vektors **F** auf den Vektor **s**.
- Die Arbeit ergibt sich aus dem Produkt dieser Projektion F_p und dem Betrag des Vektors |**s**|.
- Das entspricht dem skalaren (inneren) Produkt der Vektoren **F** und **s**.

$$F_p = |\vec{F}| \cdot |\vec{s}| \cos(\alpha)$$

$$F_p = \vec{F} \cdot \vec{s}$$

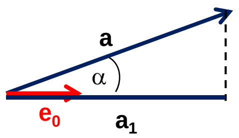


Vektoren - Skalarprodukt

- Ist der Vektor auf den projiziert wird, ein Einheitsvektor, d.h. seine Länge (Betrag) ist gleich eins, so ergibt das innere Produkt die Komponente des Vektors **a** in Richtung des Vektors $\mathbf{e}_0$.

$$a_1 = |\vec{a}| \cdot |\vec{e}_0| \cos(\alpha)$$

$$a_1 = |\vec{a}| \cdot 1 \cdot \cos(\alpha)$$



Vektoren - Koordinaten

Ein Vektor ist allein durch seine Richtung und seinen Betrag bestimmt. Er ist unabhängig vom Koordinatensystem.

Die Beschreibung eines Vektors in einem Koordinatensystem erfolgt durch die Projektion auf die Achsen. In diesem Zusammenhang beschränkt sich die Diskussion auf orthonormale Koordinatensysteme.

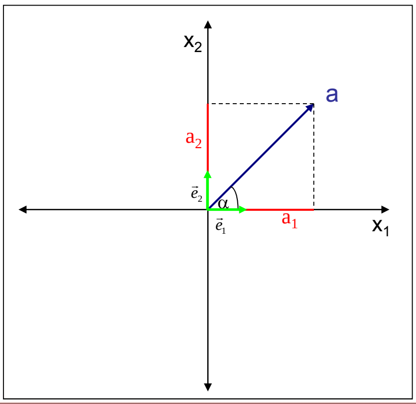
Ein Koordinatensystem wird durch seine Basisvektoren definiert. Sie zeigen vom Ursprung in Richtung der Koordinatenachsen.

Stehen die Basisvektoren im rechten Winkel zueinander und haben die Länge eins, so spricht man von einem Orthonormalsystem.

Das kartesische System ist ein Beispiel für ein Orthonormalsystem.

Koordinaten

Kartesische Koordinaten in der Ebene. Die Komponenten $\mathbf{a_1}$ und $\mathbf{a_2}$ des Vektors a entsprechen jeweils dem inneren Produkt mit den Basisvektoren $\mathbf{e_1}$ und $\mathbf{e_2}$.

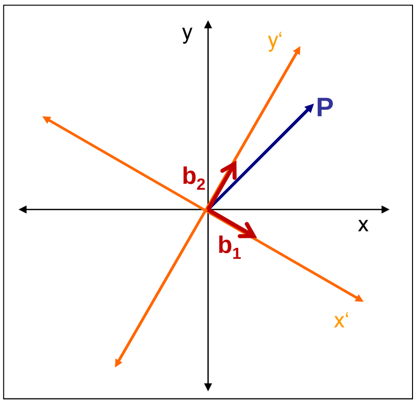


$$a_i = \vec{a} \cdot \vec{e}_i = |\vec{a}| \cdot |\vec{e}_i| \cos(\alpha)$$

Der Vektor **a** ergibt sich aus der Vektoraddition seiner Komponenten

$$\vec{a} = a_1 \cdot \vec{e}_1 + a_2 \cdot \vec{e}_2 = \sum_{i=1}^{2} a_i \cdot \vec{e}_i$$

Koordinaten Transformation



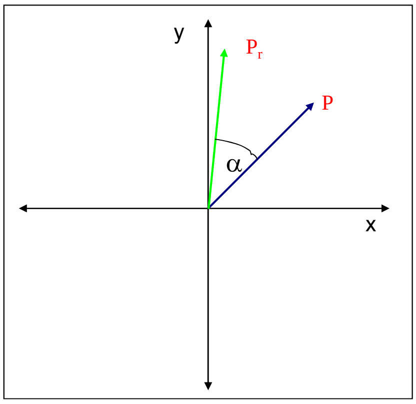
Die Transformation des Punktes aus der Basis {x,y} in die Basis {x',y'}.
Die Koordinaten von P bezüglich der Basis {x',y'} werden durch die inneren Produkte mit den Basisvektoren $\mathbf{b_1}$ und $\mathbf{b_2}$ gebildet (beide Vektoren sind in der {x,y} Darstellung.

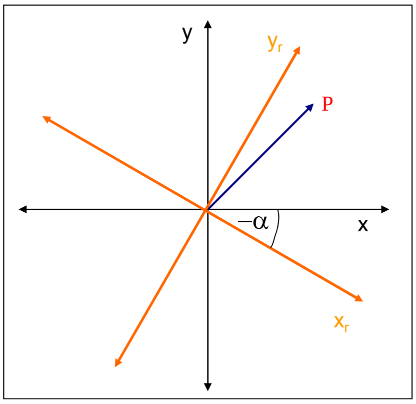
$$\vec{p}' = U^T \cdot \vec{p}$$

In Matrizenschreibweise läßt sich diese Transformation kompakt ausdrücken.
U ist eine Matrix, deren Spalten die neuen Basisvektoren enthält.
$\mathbf{U}^T$ ist die Transponierte dieser Matrix.

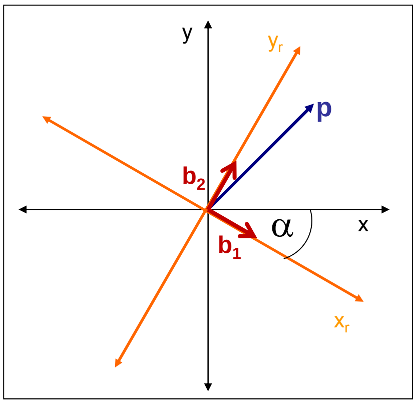
Rotation eines Punktes in der Ebene

Ein Punkt wird gegen den Uhrzeigersinn (mathematische Orientierung) um den Winkel α gedreht. **Äquivalent:** Transformation des Punktes in ein gegengleich gedrehten Koordinatensystem $\{x_r, y_r\}$





Rotation in der Ebene



$$b_1 = \begin{pmatrix} \cos(\alpha) \\ -\sin(\alpha) \end{pmatrix} \quad b_2 = \begin{pmatrix} \sin(\alpha) \\ \cos(\alpha) \end{pmatrix}$$

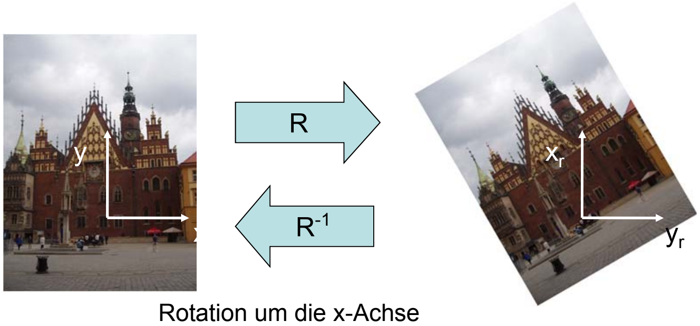
$$\mathrm{U} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

$$U^T = R = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

$$\vec{p}_r = R \cdot \vec{p}$$

Die Rotationsmatrix **R** entspricht der Transformationsmatrix in die gedrehten Koordinaten.

Rotation in 3D



Rotation um die x-Achse

$$\begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\vec{x}_r = T_x \cdot \vec{x}$$

Rotationen um y- und z-Achse

$$\vec{x}_r = T_y \cdot \vec{x}$$

$$\begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

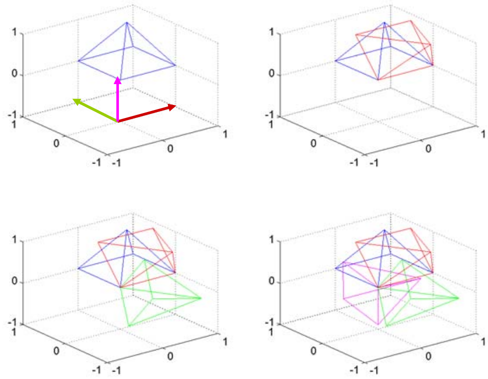
$$\vec{x}_r = T_z \cdot \vec{x}$$

$$\begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Rotation um mehrere ortsfeste Achsen (Weltkoordinaten)

$$\vec{x}_r = \underbrace{T_x T_y T_z}_{\text{Matrizen-Multiplikation}} \cdot \vec{x}$$

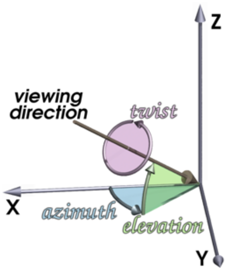
Beispiele

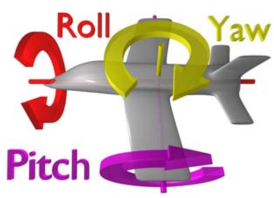


Rotation: original, x-Achse (45°), y-Achse (30°), z-Achse (70°)

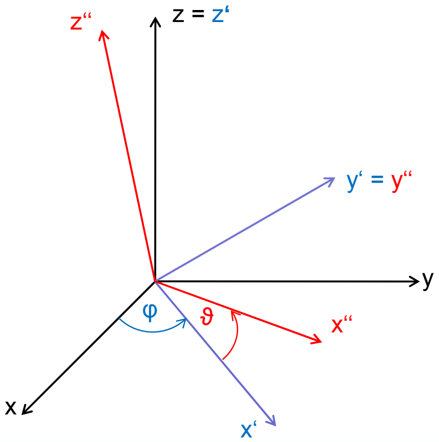
Rotation um objektbezogene Achsen

In vielen Fällen ist die Beschreibung einer zusammengesetzten Rotation in Bezug auf ein fixes Weltkoordinatensystem nicht sehr anschaulich.
Intuitiver erscheint die Rotation bezüglich eines sich mit dem Objekt mitbewegenden Koordinatensystems.
Beispiele sind die Winkel aus der Fliegerei: Pitch, Yaw, Roll
oder die Eulerwinkel: Azimut, Elevation, Spin





Objektbezogene Rotation

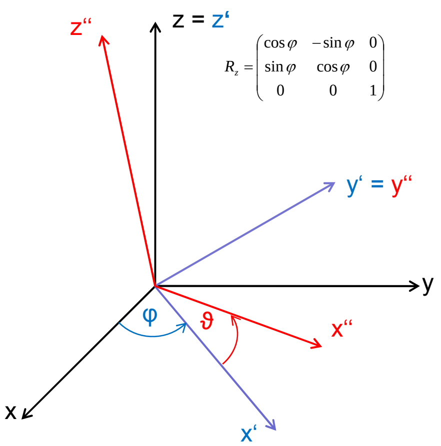


In diesem Fall wird eine Rotation um die Eulerwinkel Azimut φ und Elevation ϑ durchgeführt. Es handelt sich dabei im die Eulerwinkel in der *zy‘* Konvention.

Es werden folgende Drehungen durchgeführt:

- Rotation um z-Achse = Trafo in blaue gestrichene Koordinaten
- Rotation um die y‘-Achse = Trafo in rote zweigestrichene Koordinaten.

Objektbezogene Rotation



$$R_z = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\vec{p}' = R_z \cdot \vec{p}$$

$$\vec{p}'' = R_{y'} \cdot \vec{p}' = R_{y'} R_z \cdot \vec{p}$$

Die erste Drehung erfolgt um $\mathbf{R_z}$.
Die zweite Drehung erfolgt um $\mathbf{R_{y'}}$ im gestrichenen System. Die Darstellung von $\mathbf{R_{y'}}$ im ungestrichenen System ist wie folgt:

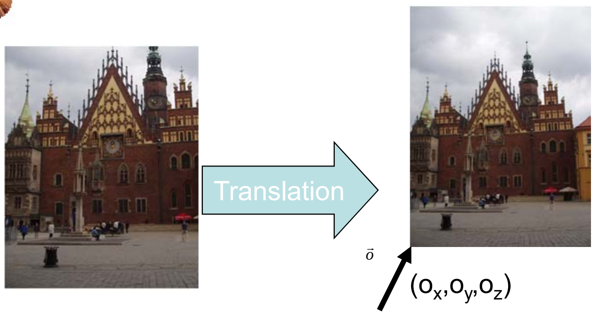
$$R_{y'} = R_z R_y R_z^{\ T}$$

$$R = R_z R_y R_z^{\ T} R_z$$

$$R = R_z R_y$$

Die Rotation in Objektkoordinaten erfolgt durch **umgekehrte Anwendung** der Matrizen.

Translation



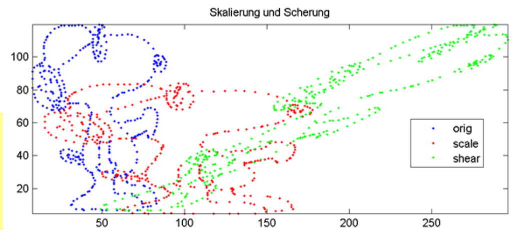
Jedem Punkt wird komponentenweise der Translationsvektor $\vec{o}$ addiert.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} o_x \\ o_y \\ o_z \end{pmatrix}$$

$$\vec{u} = \vec{x} + \vec{o}$$

Skalierung und Scherung

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} s_x & t \\ 0 & s_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$



Parameter

- sx, sy Skalierung in x und y Richtung
  - <1 Verkleinerung, >1 Vergrößerung
- t Scherung
  - <0 nach links, >0 nach rechts

Homogene Koordinaten

Alle geometrischen Operation werden durch Matrizenoperationen ausgedrückt.

Ausnahme: Translation!

Durch Einführen einer Hilfsdimension kann die Translation integriert werden

Übergang zu homogenen Koordinaten

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} r_{11} & r_{12} & r_{13} & o_x \\ r_{21} & r_{22} & r_{23} & o_y \\ r_{31} & r_{32} & r_{33} & o_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Kamera Kalibrierung (IDEE!!)

Transformationsreihenfolge

- Die Transformation vom Punkt P_w in Weltkoordinaten zum Bildpunkt U in den affinen Bildkoordinaten
  - Weltkoordinaten -> Kamerakoordinaten
  - Kamerakoordinaten -> Euklidische Bildkoordinaten
  - Euklidische Bildkoordinaten –> affine Koordinaten

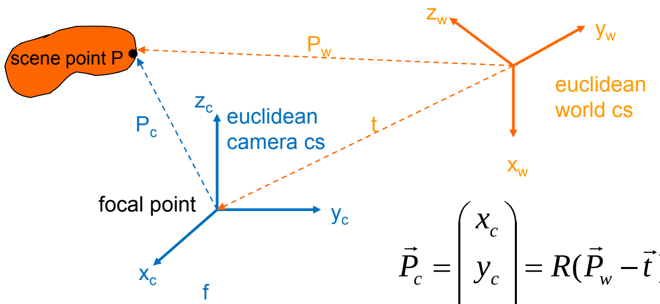
Operationen zur Koordinatentransformation

- Transformationen in 3D Koordinatensystemen
  - Rotation
  - Translation
  - Skalierung
  - Scherung

Kamera Kalibrierung I

Welt -> Kamerakoordinaten

- Durch eine einfache Rotation und Translation wird ein Punkt aus den Weltkoordinaten in Kamerakoordinaten umgerechnet.

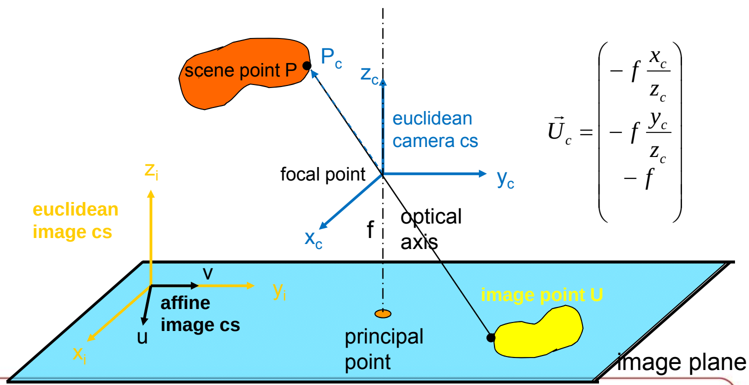


$$\vec{P}_c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = R(\vec{P}_w - \vec{t})$$

Kamera Kalibrierung II

Kamerakoordinaten -> Bildkoordinaten

- Der Punkt P_c aus den Kamerakoordinaten wird in den Bildraum U_c projiziert. Abbildung anhand ähnlicher Dreiecke.



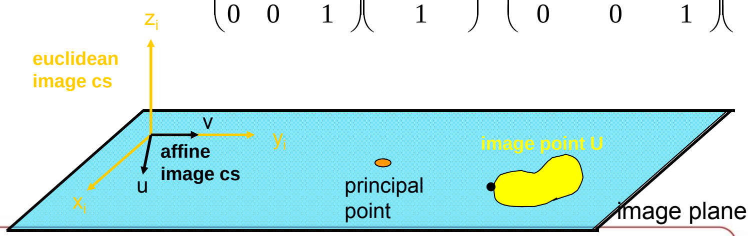
$$\vec{U}_c = \begin{pmatrix} -f\dfrac{x_c}{z_c} \\ -f\dfrac{y_c}{z_c} \\ -f \end{pmatrix}$$

Kamera Kalibrierung III

Bildkoordinaten -> affine Bildkoordinaten

- Transformation des Ursprungs vom „principal point“ (u_0,v_0) zum LO Bildrand
- Pixel beinhalten: Skalierung (a,c) und Scherung (b)

$$\tilde{u} = \begin{pmatrix} a & b & -u_0 \\ 0 & c & -v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -fx_c/z_c \\ -fy_c/z_c \\ 1 \end{pmatrix} = \begin{pmatrix} -fa & -fb & -u_0 \\ 0 & -fc & -v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c/z_c \\ y_c/z_c \\ 1 \end{pmatrix}$$



Kamera Kalibrierung IV

Intrinsische Parameter

- Annahme Welt- und Kamerakoordinaten sind identisch X_w=X_c
- Abbildung von fünf inneren Parametern abhängig

$$\tilde{u} = \begin{pmatrix} -fa & -fb & -u_0 \\ 0 & -fc & -v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c/z_c \\ y_c/z_c \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \alpha_u \dfrac{x_c}{z_c} & + & \alpha_s \dfrac{y_c}{z_c} & - & u_0 \\ & & \alpha_v \dfrac{y_c}{z_c} & - & v_0 \end{pmatrix}$$

- u_0, v_0 offset zum principal point
- α_s Scherung
- α_u, α_v Skalierung der Bildpunkte im Bild

Kamera Kalibrierung V

Extrinsische Parameter

- Weltkoordinaten X_w werden auf Kamerakoordinaten X_c und weiter auf affine Bildkoordinaten u abgebildet.

$$\tilde{u} = K\,\underbrace{R(\vec{X}_w - \vec{t}\,)}_{\vec{X}_c} = (KR - KR\vec{t}\,)\begin{pmatrix}\vec{X}_w \\ 1\end{pmatrix} = M\tilde{X}_w$$

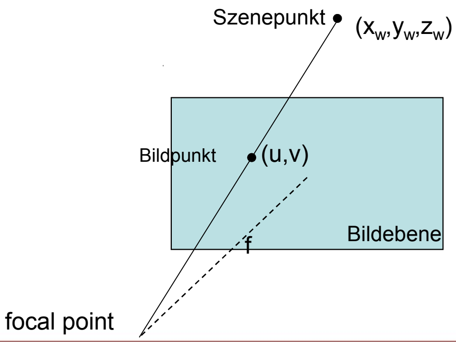
- Welt- und Kamerakoordinaten durch 6 äußere Parameter miteinander verknüpft
- *R* Rotationsmatrix bestimmt durch drei Winkel
- *t* Translationsvektor mit drei Komonenten

Homogene Koordinaten erlauben kompakte Schreibweise mittels *Kameramatrix M*. Kalibrierung bestimmt Koeffizienten von *M*.

Kamera Kalibrierung V

Kalibrierung mit bekannter Szenerie

- Methode benötigt Kalibrierungsobjekte
- mindestens 6 Paare aus Bildpunkten und zugehörigen Weltpunkten müssen bekannt sein.



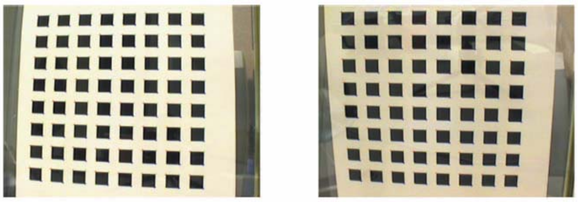
•Jedes Paar aus Bild- und Szeneriepunkten definiert 2 Gleichungen.
•Kameramatrix (3x4) hat 12 Unbekannte =>6 Punktpaare

⇒Üblicherweise werden mehr Punktpaare verwendet. Parameter werden durch ein Ausgleichsverfahren (Regression) bestimmt

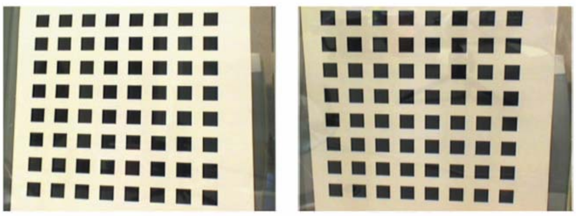
Testmuster

- Verwendete Testmuster zur Kalibrierung
- Punkte liegen nicht in derselben Ebene
- Punkte werden segmentiert und die Mittelpunkte und Konturen zur Kalibrierung verwendet

Resultate der perspektivischen Entzerrung



originale Bilder

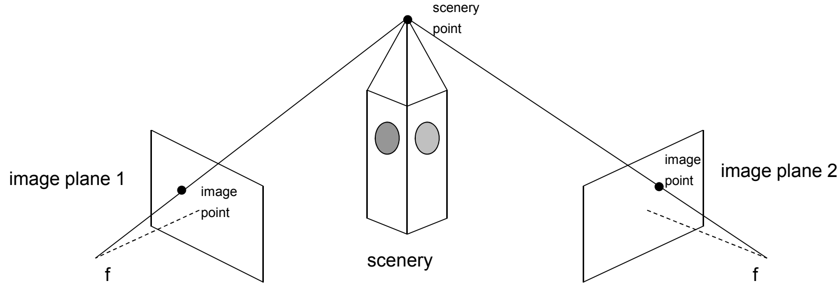


nach Korrektur der Verzerrung

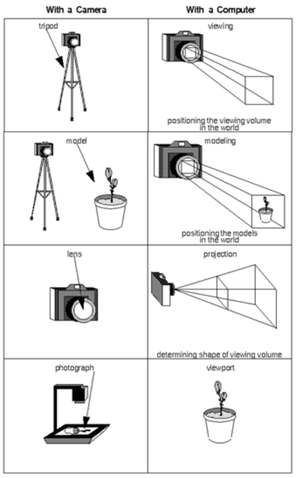
Rekonstruktion einer Szene in 3D

Rekonstruktion mittels Epipolargeometrie

- Eindeutige Bestimmung eines Szenerie-Punktes durch den Schnitt zweier Epipolarlinien
- Bildpunkt und Fokus definieren eine „Leitlinie“ durch die Szenerie, die den Objekt Punkt enthält



Dualität Kamera-Computergraphik



Kamera	OpenGL
Welt → KameraKoordinaten	Modelview
Welt → Kamerakoordinaten	Modelview
Kamera → Euklid‘sche Bildkoordinaten	Projection-Transform
Euklid‘sche → affine Bildkoordinaten	Viewport Transformation