

Kognitive Systeme

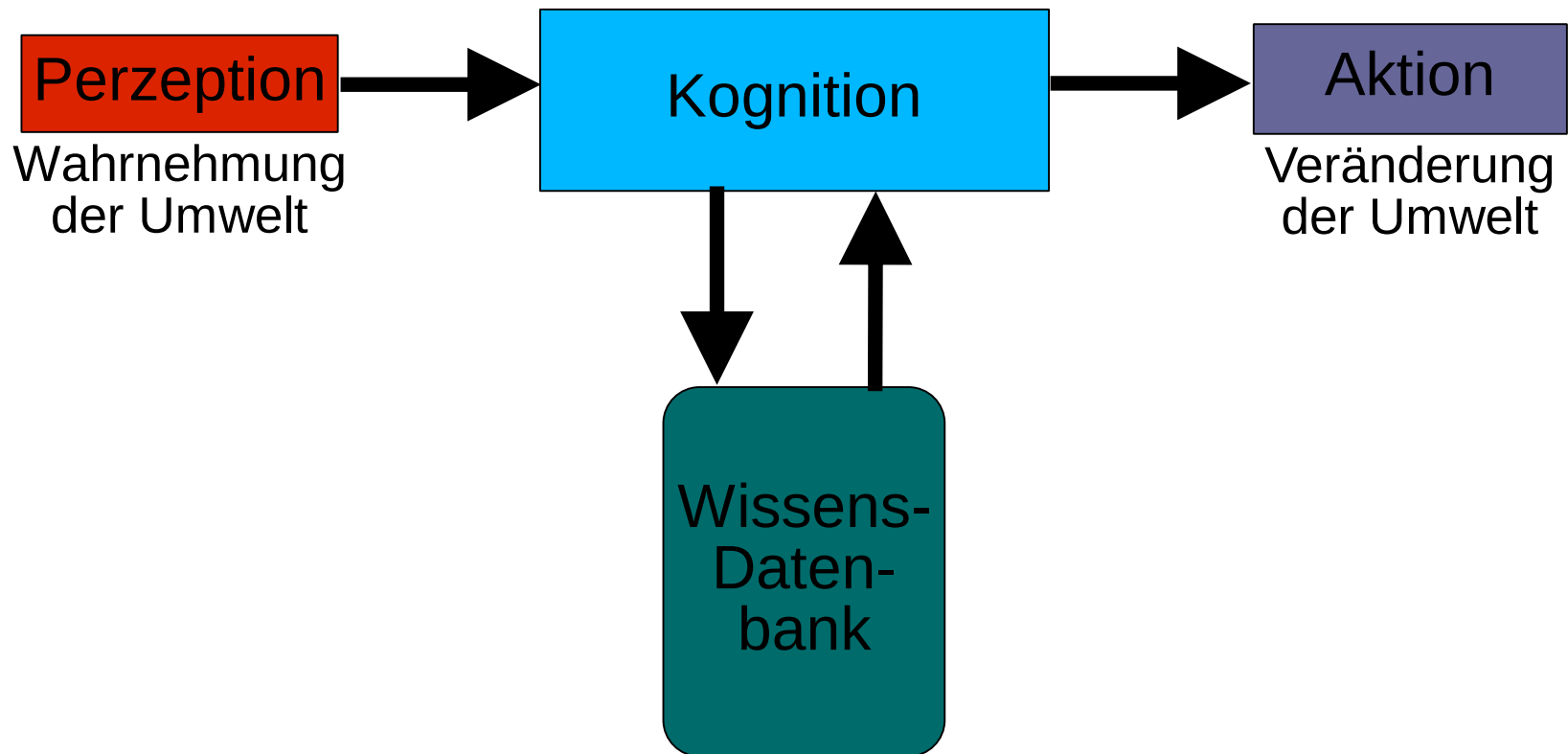
3D-Bildverarbeitung

Montag, 03. Juli 2017

Zunächst: kurze Wiederholung

Elemente eines Kognitiven Systems

Ingenieurmäßiger Ansatz:



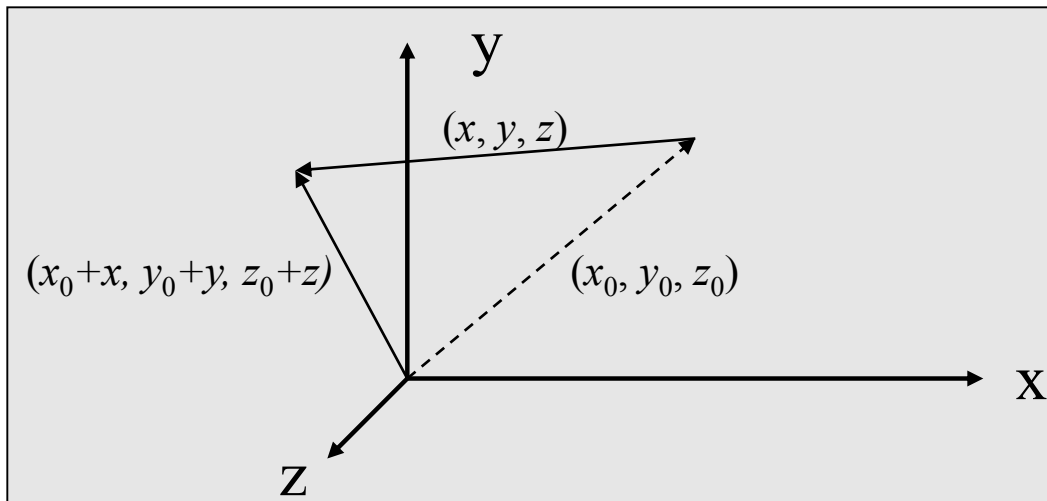
Geometrische 3D-Transformationen

- Grundlage von Sensorik und Aktorik:
Geometrische Beschreibung von Objekten und ihre Relationen (Ort, Orientierung)
 - Ort des messenden Systems im Raum bzw. in einer Szene
 - Ort mehrerer Sensoren zueinander
 - Ort sensorisch erfasster Objekte relativ zum Sensor
 - Ort von Aktoren (Greifer, Werkzeuge etc.) und Relation zu den manipulierenden Objekte
 - Gelenkwinkel des Roboters (innere Zustände)
- Anforderungen an 3D.Transformationen:
 - Geschlossene algebraische Ausdrücke
 - Invertierbarkeit der Transformationen zwischen Konfigurationsräumen
 - Interpolation

- Zwei Darstellungsmethoden sind in der Bildverarbeitung und in der Robotik für 3D – Transformationen üblich:
 - Homogene Transformationsmatrizen und homogene Vektoren (für Translationen und Rotationen)
 - Quaternionendarstellung (nur für Rotationen)

3D-Transformationen: Translation

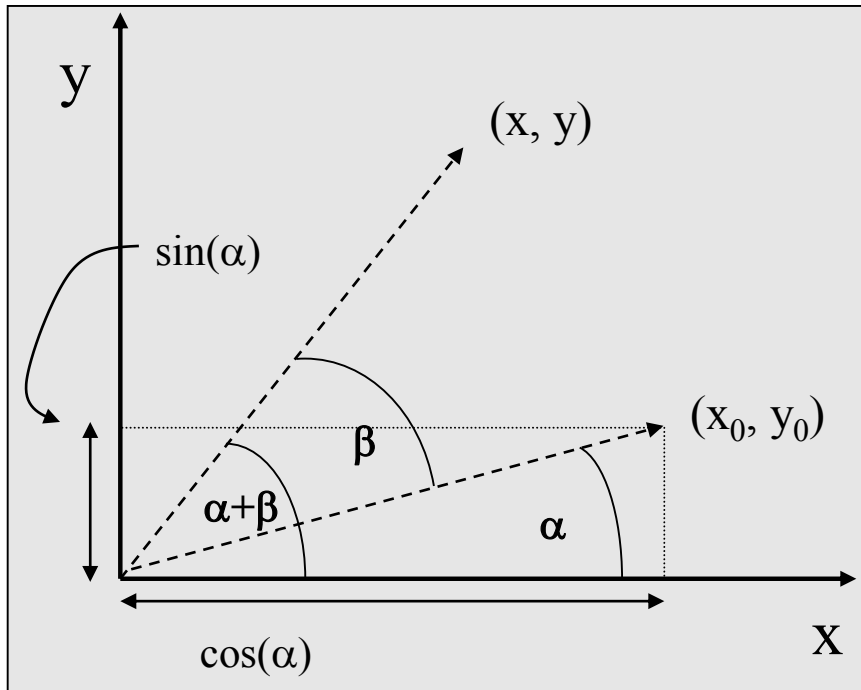
Translation eines 3D-Vektors:



$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_0 + x \\ y_0 + y \\ z_0 + z \end{pmatrix}$$

3D-Transformationen: Rotation

- o.B.d.A. auf Einheitsvektor zurückführbar (Basistransformation)
- Konvention: *Rechtshändiges Koordinatensystem*
- Rotation von (x_0, y_0, z_0) um Winkel β mit Ergebnis (x, y, z_0) :



Aus Additionstheorem:

$$\begin{aligned}x &= \cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta \\y &= \sin(\alpha + \beta) = \sin \beta \cos \alpha + \cos \beta \sin \alpha\end{aligned}$$

oder (mit $(x_0, y_0) = (\cos \alpha, \sin \alpha)$):

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

z_0 invariant, da Rotation um z !

3D-Transformationen: Rotation

3x3 – Rotationsmatrix R:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$$

Rotation um x :

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}$$

Rotation um y :

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

Rotation um z :

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Eigenschaften von Rotationsmatrizen:

- regulär, invertierbar, Determinante = 1
- Jede beliebige Rotation im Raum kann durch drei Variablen beschrieben werden (*Eulers Theorem*)
- Einzelrotationen können als eine Matrix dargestellt werden:

$$R_r(\gamma)R_q(\beta)R_p(\alpha)\begin{pmatrix} x \\ y \\ z \end{pmatrix} \text{ mit } p,q,r \in \{x,y,z\} = R_{pqr}(\alpha,\beta,\gamma)\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

- Damit reicht die Angabe der Euler-Winkel α, β, γ zur Beschreibung der Rotation
- Das ergibt natürlich nur Sinn, wenn eine Konvention für die Zuordnung p, q, r zu den Achsen x, y, z definiert wurde!

- Zwei grundlegend unterschiedliche Rotationstypen:
 - Rotation um mitgedrehte Achsen
 - Rotation um raumfeste Achsen
- Konventionen zur Erstellung von Rotationsmatrizen:
 - Standard-Beispiel für mitgedrehte Achsen:
Zuerst um die raumfeste x-Achse, dann um die mitgedrehte y-Achse, dann um die (zweimal) mitgedrehte z-Achse

$$R_{X'Y'Z'}(\alpha, \beta, \gamma) = R_X(\alpha)R_Y(\beta)R_Z(\gamma)$$

- Interpretation für raumfeste Achsen:
Zuerst um die raumfeste z-Achse, dann um die raumfeste y-Achse, dann um die raumfeste x-Achse

$$R_{ZYX}(\gamma, \beta, \alpha) = R_X(\alpha)R_Y(\beta)R_Z(\gamma)$$

Definition Homogene Koordinaten:

Homogene Koordinaten

$$\mathbf{h} = (h_1, \dots, h_i, h_{i+1})$$

eines Punktes p im R^i mit

$$\mathbf{p} = (p_1, \dots, p_i)$$

sind Zahlen, für die gilt:

$$p_k = \frac{h_k}{h_{i+1}} \quad \forall 1 \leq k \leq i.$$

h_{i+1} ist ein Skalierungsfaktor; üblicherweise 1.

- Transformation definiert durch Rotation R und Translation t :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + t = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

- Darstellung mit Hilfe homogener Koordinaten und einer geschlossenen Transformationsmatrix:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} & & & \\ & R & & t \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ \hline 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{pmatrix} = A \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{pmatrix}$$

- Probleme mit Rotationsmatrizen:
 - Hoch redundant
 - Rechenaufwendig
 - Interpolation schwierig
- Probleme mit Euler-Winkeln
 - Singularitäten

Andere Darstellung von Rotationen?

⇒ *Quaternionen*, Erweiterung der komplexen Zahlen

- schon 1843 von W.R. Hamilton beschrieben
- erst 1995 von Shoemake in der Computergrafik eingeführt

- Erweiterung der komplexen Zahlen ins Vierdimensionale

- **Definition:**

Ein *Quaternion* q ist eine Zahl

$$\begin{aligned} \mathbf{q} &= (q_w, q_x, q_y, q_z)(1, i, j, k)^T = (q_w, \mathbf{q}_v)(1, i, j, k)^T \\ &= q_w + iq_x + jq_y + kq_z \end{aligned}$$

mit

$$\begin{aligned} i^2 &= j^2 = k^2 = -1 \\ ij &= -ji = k \\ jk &= -kj = i \\ ki &= -ik = j. \end{aligned}$$

- q_w ist der **Realteil**, $\mathbf{q}_v = (q_x, q_y, q_z)$ der **Imaginärteil** des Quaternions. Man schreibt einfach (q_w, q_x, q_y, q_z) oder $(q_w, \mathbf{q}_v)$.

Rechenregeln für Quaternionen:

■ Addition:

$$\mathbf{q} + \mathbf{r} = (q_w, \mathbf{q}_v) + (r_w, \mathbf{r}_v) = (q_w + r_w, \mathbf{q}_v + \mathbf{r}_v)$$

■ Multiplikation:

$$\mathbf{qr} = (q_w r_w - \mathbf{q}_v \cdot \mathbf{r}_v, \mathbf{q}_v \times \mathbf{r}_v + q_w \mathbf{r}_v + r_w \mathbf{q}_v)$$

assoziativ, aber *nicht* kommutativ

■ Konjugierter Quaternion:

$$\bar{\mathbf{q}} = (q_w, -\mathbf{q}_v) \text{ für } \mathbf{q} = (q_w, \mathbf{q}_v)$$

Rechenregeln für Quaternionen:

■ Norm:
$$N(\mathbf{q}) = \sqrt{\mathbf{q}\bar{\mathbf{q}}} = \sqrt{\bar{\mathbf{q}}\mathbf{q}} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}$$

Quaternionen $\mathbf{q}$ mit $N(\mathbf{q})=1$ heißen
Einheitsquaternionen.

■ Multiplikative Identität:
$$\mathbf{I} = (1, \mathbf{0})$$

■ Multiplikative Inverse:
$$\mathbf{q}^{-1} = \bar{\mathbf{q}} / N^2(\mathbf{q})$$

■ Rotation mit Quaternionen:

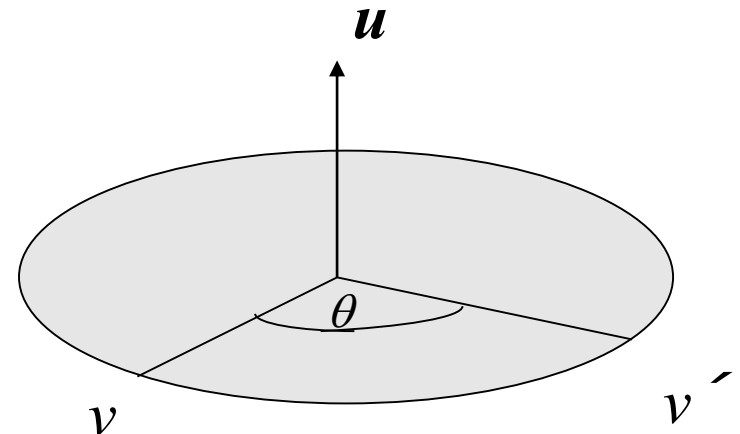
- Einheitsquaternion q ist definiert durch Rotationsachse u mit $|u|=1$ und Winkel θ :

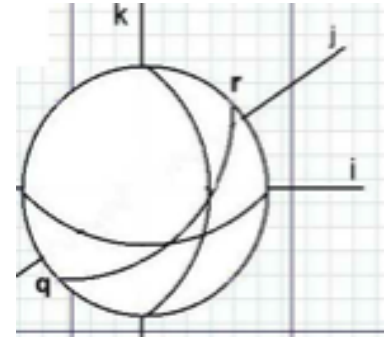
$$q = \left(\cos \frac{\theta}{2}, u \sin \frac{\theta}{2} \right)$$

- Quaternion a ist definiert durch zu rotierenden Vektor v :

$$a = (0, v)$$

- Das Produkt qaq rotiert v um die Achse u mit dem Winkel θ





Interpolation zwischen zwei Quaternionen:

- Sphärische Lineare Interpolation (SLERP)
- Berechnet für $t \in [0,1]$ die kürzeste Verbindung auf der vierdimensionalen Einheitssphäre zwischen q und r

- Analytisch:
$$\text{SLERP}(\mathbf{q}, \mathbf{r}, t) = \mathbf{q}(\mathbf{r}\mathbf{q}^{-1})^t$$

- Numerisch:
$$\text{SLERP}(\mathbf{q}, \mathbf{r}, t) = \mathbf{q} \frac{\sin((1-t)\theta)}{\sin \theta} + \mathbf{r} \frac{\sin(t\theta)}{\sin \theta}$$

mit Winkel θ zwischen $\mathbf{r}$ und $\mathbf{q}$

■ Quaternion $\Rightarrow$ Rotationsmatrix:

$$\mathbf{q} = (q_w, q_x, q_y, q_z) \Rightarrow \mathbf{M}_q = \begin{pmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & 1 - 2(q_x^2 + q_y^2) \end{pmatrix}$$

■ Rotationsmatrix $\Rightarrow$ Quaternion:

$$\begin{aligned} q_w &= \frac{1}{2} \sqrt{1 + \sum_{i=1}^3 m_{ii}} \\ q_x &= (m_{32} - m_{23}) / (4q_w) \\ q_y &= (m_{13} - m_{31}) / (4q_w) \\ q_z &= (m_{21} - m_{12}) / (4q_w) \end{aligned}$$

Vorteile:

- Rotation direkt um gewünschte Drehachse
- Interpolation möglich
- Weniger Rechenaufwand
- Keine Redundanz $\Rightarrow$ numerisch stabiler, weniger Gefahr für Singularitäten

Nachteile:

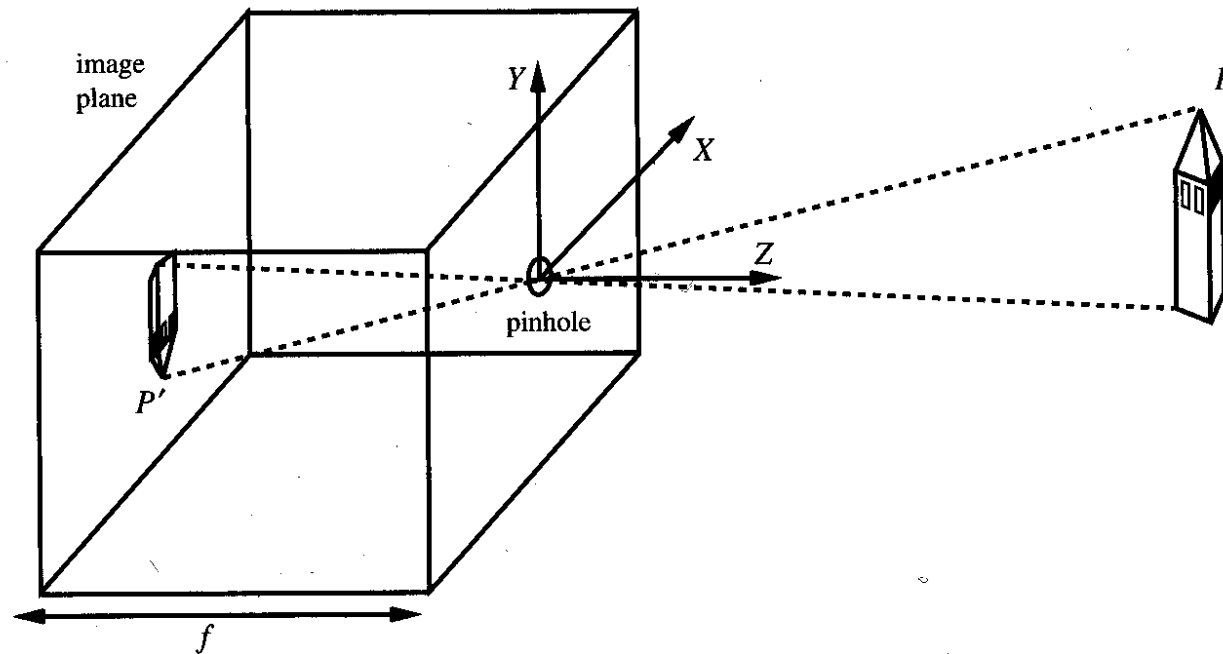
- Nur Rotationen berechenbar $\Rightarrow$ Kombination mit Matrizen nötig $\Rightarrow$ Rechenaufwand für Umwandlungen

Inhalt der heutigen Vorlesung

- Erweitertes Kameramodell
- Kamerakalibrierung
- Stereokamerasysteme
 - Triangulation
 - Epipolargeometrie
- Bewegungserfassung
 - Menschmodell
 - Bildbasiert mit Partikelfilter
- Bewegungserfassung II
 - Auf der Basis von 3D-Punktwolken unter Verwendung des ICP
- Gesichtserkennung

Geometrie und optische Parameter einer Kamera

■ Einfachstes Modell: Lochkameramodell



Interner Parameter:
Brennweite f ("Fokalabstand")

$$\begin{pmatrix} u \\ v \end{pmatrix} = -\frac{f}{Z} \begin{pmatrix} X \\ Y \end{pmatrix}$$

- Lochkameramodell vereinfacht die realen optischen Verhältnisse stark.
- In der Praxis: Erweitertes Lochkameramodell
- Zunächst einige Definitionen:
 - Optische Achse:
Gerade durch das Projektionszentrum, senkrecht zur Bildebene
 - Bildhauptpunkt $C(c_x, c_y)$:
Schnittpunkt der optischen Achse mit der Bildebene

- Koordinatensysteme:

- Bildkoordinatensystem:

2D-Koordinatensystem, Einheit [Pixel], Vereinbarung für die Vorlesung (gilt für die meisten Kamerasreiber): Ursprung in der linken oberen Ecke des Bildes, u -Achse zeigt nach rechts, v -Achse zeigt nach unten

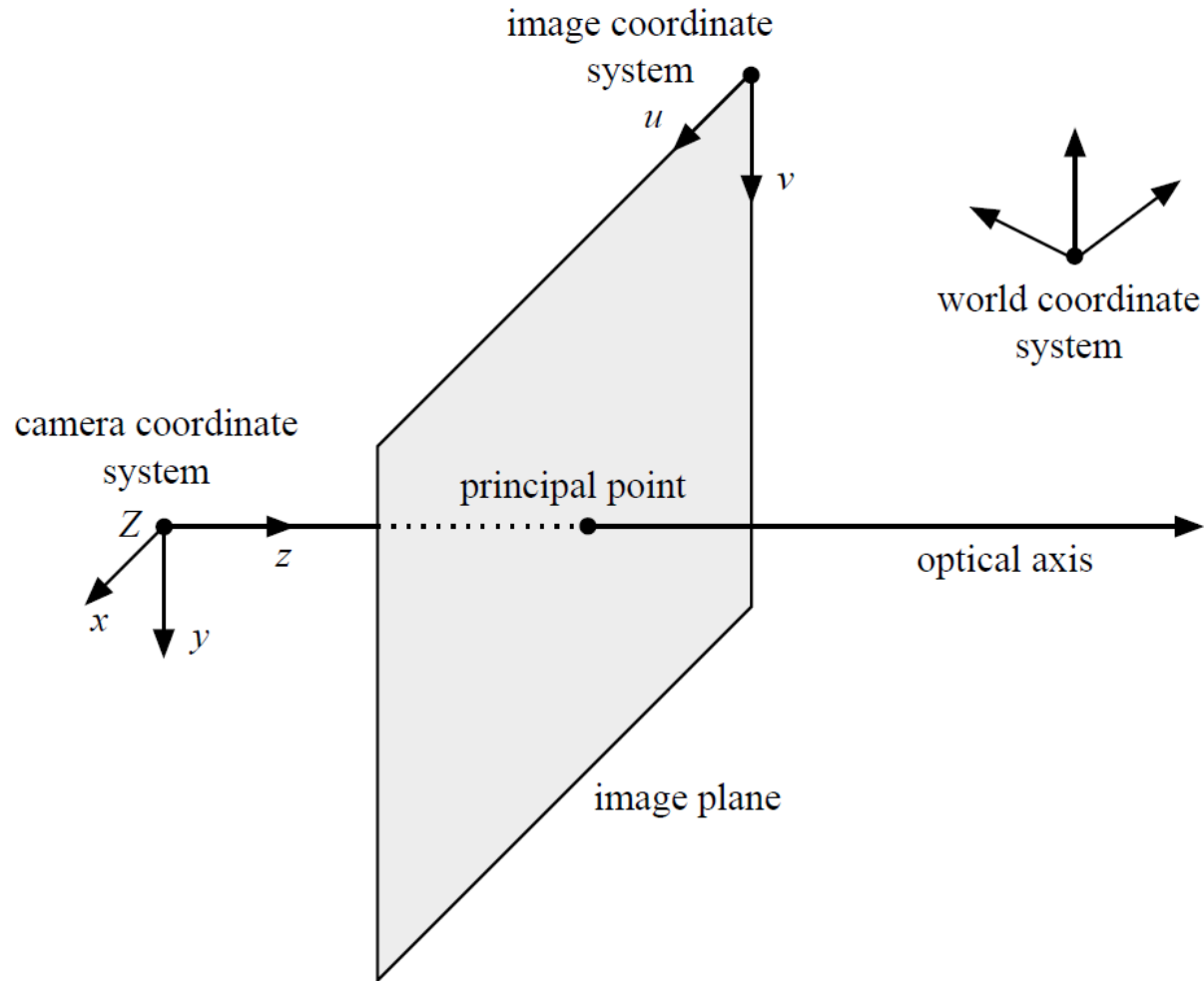
- Kamerakoordinatensystem:

3D-Koordinatensystem, Einheit [mm], Ursprung liegt im Projektionszentrum, Achsen parallel zu den Achsen des Bildkoordinatensystems, d.h. x -Achse nach rechts, y -Achse nach unten, und die z -Achse gemäß der Dreifingerregel für ein rechtshändiges Koordinatensystem nach vorne

- Weltkoordinatensystem:

3D-Koordinatensystem, Einheit [mm],
Basiskoordinatensystem, das beliebig im Raum liegen kann

Erweitertes Kameramodell II



■ Begriffe:

- Intrinsische Kameraparameter: Brennweite, Bildhauptpunkt, Parameter für die Beschreibung radialer/tangentialer Linsenverzerrung; definieren die nicht (eindeutig) umkehrbare Abbildung vom Kamerakoordinatensystem in das Bildkoordinatensystem
- Extrinsische Kameraparameter:
Definieren die Beziehung zwischen dem Kamerakoordinatensystem und dem Weltkoordinatensystem, im Allgemeinen beschrieben durch eine Rotation R und eine Translation t

- Vereinfachungen des Lochkameramodells:
 - Ursprung des Bildkoordinatensystems ist identisch mit dem Bildhauptpunkt
 - Pixel werden als quadratisch angenommen
 - Keinerlei Modellierung der Linsenverzerrung
 - Es existiert kein Weltkoordinatensystem bzw. es ist identisch mit dem Kamerakordinatensystem, d.h. es werden keine extrinsischen Kameraparameter modelliert

■ Brennweite:

- In der Praxis wird die Umrechnung von [mm] nach [Pixel] in die Parameter für die Brennweite mit aufgenommen
- Da Pixel nicht mehr als quadratisch sondern als rechteckig angenommen werden, gibt es deshalb für jede Richtung einen Parameter, also: f_x, f_y
- Die Parameter f_x, f_y sind dann das Produkt aus der tatsächlichen Brennweite mit Einheit [mm] und dem jeweiligen Umrechnungsfaktor mit Einheit [Pixel/mm]
- Die Einheit für die Parameter f_x, f_y ist somit [Pixel]

- Die Abbildung vom Kamerakoordinatensystem in das Bildkoordinatensystem, ausschließlich mit den intrinsischen Parametern, ist dann definiert durch:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} c_x \\ c_y \end{pmatrix} + \frac{1}{Z} \cdot \begin{pmatrix} f_x \cdot X \\ f_y \cdot Y \end{pmatrix}$$

Oder als Matrixmultiplikation mit Kalibriermatrix K auf homogenen Koordinaten:

$$\begin{pmatrix} u \cdot w \\ v \cdot w \\ w \end{pmatrix} = K \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

■ Extrinsische Kamerakalibrierung

- Ist definiert durch eine Koordinatentransformation bestehend aus Rotation und Translation
- Koordinatentransformation vom Weltkoordinatensystem in das Kamerakoordinatensystem:

$$\mathbf{x}_c = R \mathbf{x}_w + \mathbf{t}$$

- 3×4 Gesamt-Projektionsmatrix P (intrinsisch und extrinsisch) auf homogenen Koordinaten:

$$\begin{pmatrix} u \cdot w \\ v \cdot w \\ w \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad P = (K R \mid K \mathbf{t})$$

- Die Kalibrierung einer Kamera bedeutet die Bestimmung ihrer Parameter bezüglich eines gewählten Kameramodells
- Die Bestimmung der intrinsischen Parameter ist unabhängig vom Aufbau; solange Zoom und Fokus der Kamera gleich bleiben, verändern sich diese Parameter nicht
- Die Bestimmung der extrinsischen Parameter ist abhängig von der Wahl des Weltkoordinatensystems und ändert sich je nach Aufbau

- Ist die Kamera kalibriert, dann liegt die Abbildungsfunktion f vor, die einen Punkt vom Weltkoordinatensystem eindeutig in das Bildkoordinatensystem abbildet:

$$f : R^3 \rightarrow R^2$$

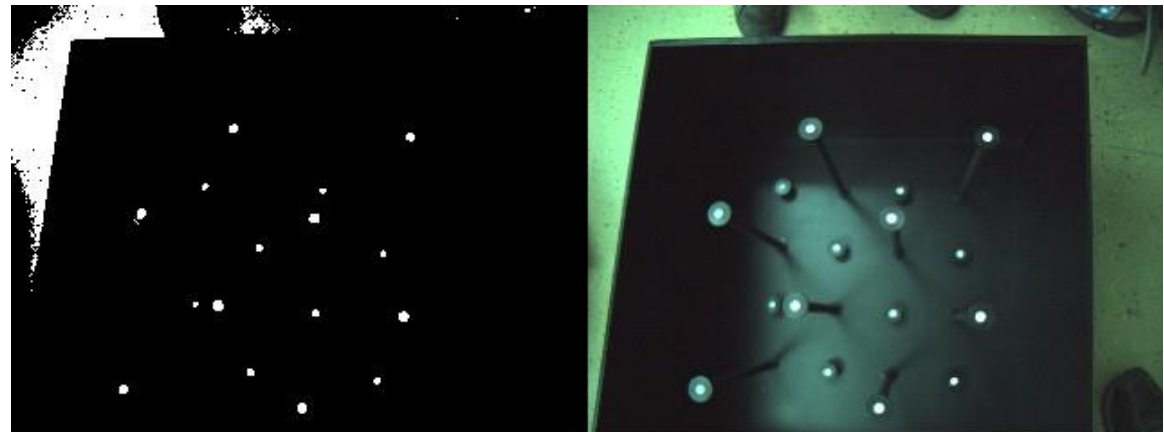
- f ist definiert durch die Projektionsmatrix P und anschließende Transformation der homogenen Koordinaten durch Division durch w
- Die inverse Abbildung bildet einen Punkt im Bildkoordinatensystem auf eine Gerade im Weltkoordinatensystem ab, die durch das Projektionszentrum verläuft

■ Verfahren zur Kamerakalibrierung:

- Roger Y. Tsai: „A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses“, 1987
- Direkte Lineare Transformation (DLT)
- Erweiterungen der DLT, welche Linsenverzerrung modellieren

- Gesucht:
 P ist 3×4 -Matrix, hat also 12 Unbekannte
- Verfahren *Testfeldkalibrierung*:
 - Bestimmung einer Menge von Punktkorrespondenzen: 3D-Punkt in einem gewählten Weltkoordinatensystem und 2D-Punkt im Bildkoordinatensystem
 - 3D-Punkte sind durch Verwendung eines geeigneten Kalibrierobjekts oder -musters a priori bekannt
 - 2D-Punkte werden durch Methoden der Bildverarbeitung berechnet
- Benötigt:
6 bekannte Objektpunkte, da jede Punktkorrespondenz zwei Gleichungen liefert

- Bedingung:
3D-Punkte dürfen nicht koplanar liegen, d.h. sie müssen einen dreidimensionalen Raum aufspannen
- Möglichkeiten:
 - Verwendung eines 2D-Musters, das in mindestens zwei verschiedenen Tiefen präsentiert wird
 - Verwendung eines geeigneten 3D-Kalibrierobjekts, wie unten abgebildet



- Ein Standardverfahren für die Berechnung der Projektionsmatrix P ist die Direkte Lineare Transformation (DLT)

$$\begin{pmatrix} u \cdot w \\ v \cdot w \\ w \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad P = (K \ R \mid K \ t) = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{pmatrix}$$

$$\Rightarrow \begin{aligned} u &= \frac{p_1 X + p_2 Y + p_3 Z + p_4}{p_9 X + p_{10} Y + p_{11} Z + p_{12}} \\ v &= \frac{p_5 X + p_6 Y + p_7 Z + p_8}{p_9 X + p_{10} Y + p_{11} Z + p_{12}} \end{aligned}$$

o.B.d.A. kann ein Parameter normiert werden. Üblicherweise wird $p_{12} = 1$ gewählt.

$$\Rightarrow \begin{aligned} p_1X + p_2Y + p_3Z + p_4 &= up_9X + up_{10}Y + up_{11}Z + u \\ p_5X + p_6Y + p_7Z + p_8 &= vp_9X + vp_{10}Y + vp_{11}Z + v \end{aligned}$$

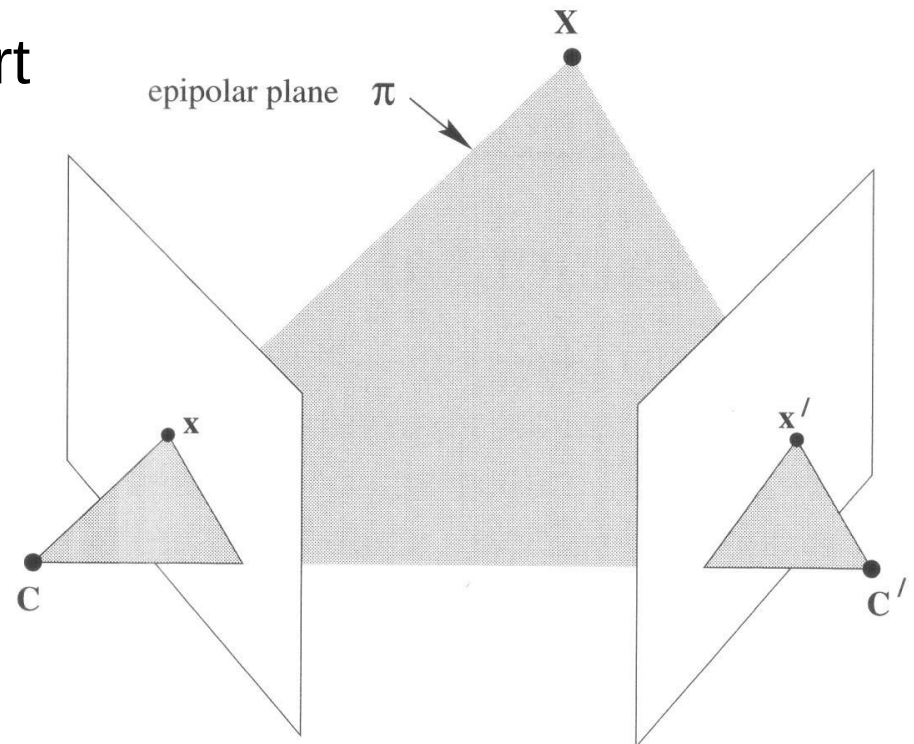
- Formulierung als überbestimmtes LGS $A\mathbf{x} = \mathbf{b}$ mit $n \geq 6$ Punktkorrespondenzen, das beispielsweise mit Hilfe der Normalengleichung gelöst werden kann:

$$A = \begin{pmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} p_1 \\ \vdots \\ p_{11} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{pmatrix}$$

Geometrische Beziehungen zwischen zwei Kameras

Gegeben:

- Zwei Kameras, mit Projektionsmatrizen P und P'
- Zwei Abbilder x und x' des Punktes X
- Dann kann X rekonstruiert werden



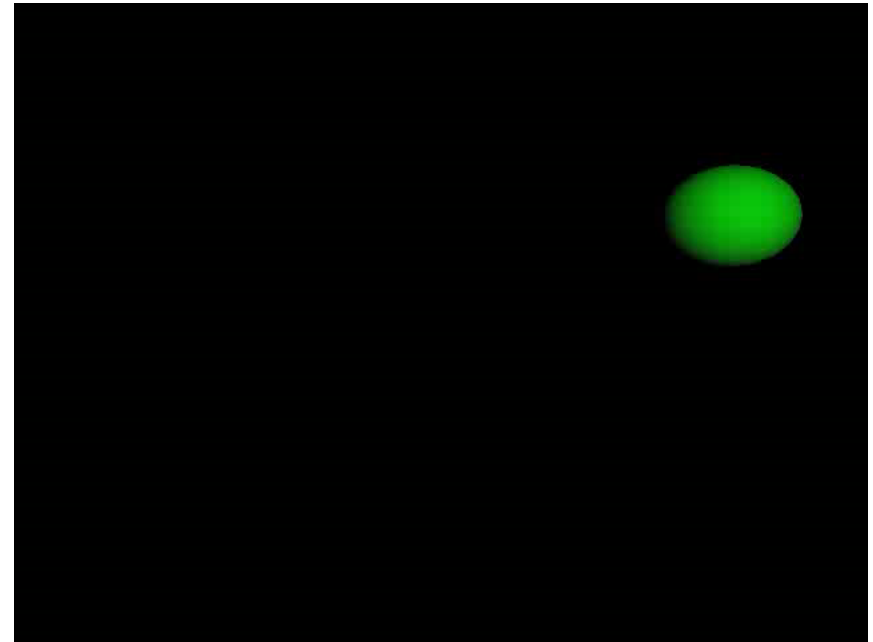
- Triangulation zwischen linker und rechter Kamera möglich durch Kenntnis der Kameraparameter
- Eine Möglichkeit zur Berechnung von 3D-Punkten aus Bildpunkt-Korrespondenzen $\mathbf{x}, \mathbf{x}'$:
 - Aufstellen der beiden Geraden g, g' der möglichen Punkte zu $\mathbf{x}, \mathbf{x}'$ im Weltkoordinatensystem mit Hilfe der Projektionsmatrizen P, P' :
$$g: \mathbf{x} = \mathbf{a} + r \cdot \mathbf{u}$$
$$g': \mathbf{x} = \mathbf{b} + s \cdot \mathbf{v}$$
 - Berechnung des optimalen „Schnittpunktes“ S durch Lösung des überbestimmten LGS $A\mathbf{x} = \mathbf{c}$ mit:

$$A = \begin{pmatrix} u_1 & -v_1 \\ u_2 & -v_2 \\ u_3 & -v_3 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} r^* \\ s^* \end{pmatrix}, \quad \mathbf{c} = \mathbf{b} - \mathbf{a} \quad s = \frac{\mathbf{a} + \mathbf{r}^* \cdot \mathbf{u} + \mathbf{b} + \mathbf{s}^* \cdot \mathbf{v}}{2}$$

- 3D-Tracking von Kopf und Händen auf der Basis von Hautfarbe, Region-Growing und Stereo-Kalibrierung

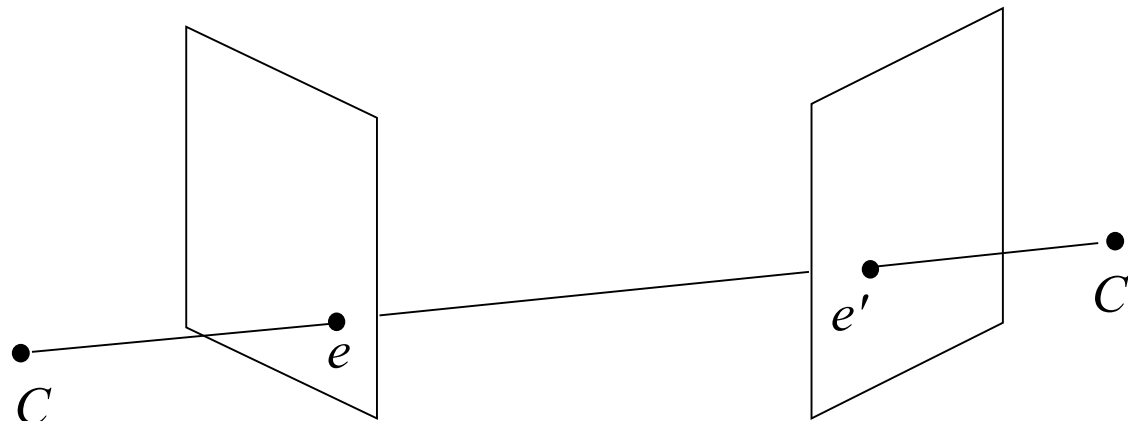


Eingabevideo der linken Kamera

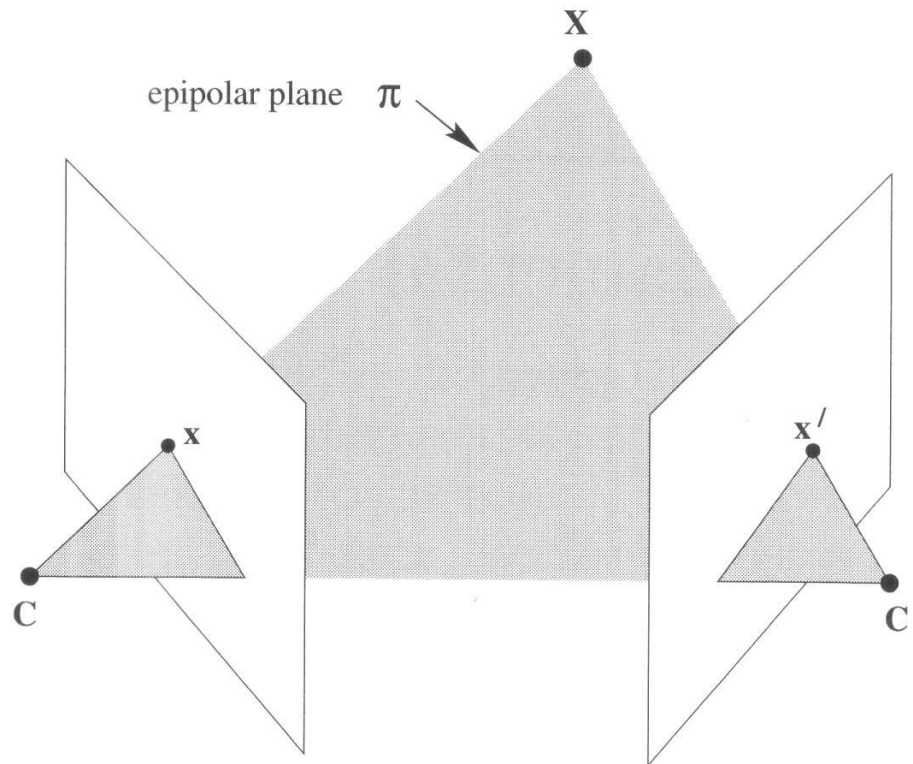


3D-Visualisierung des Ergebnisses

- Zusammenhang zwischen zwei Kameras ist gegeben durch die *Epipolargeometrie*
- Die Schnittpunkte e und e' der Geraden durch die Projektionszentren mit den Bildebenen nennt man *Epipole*

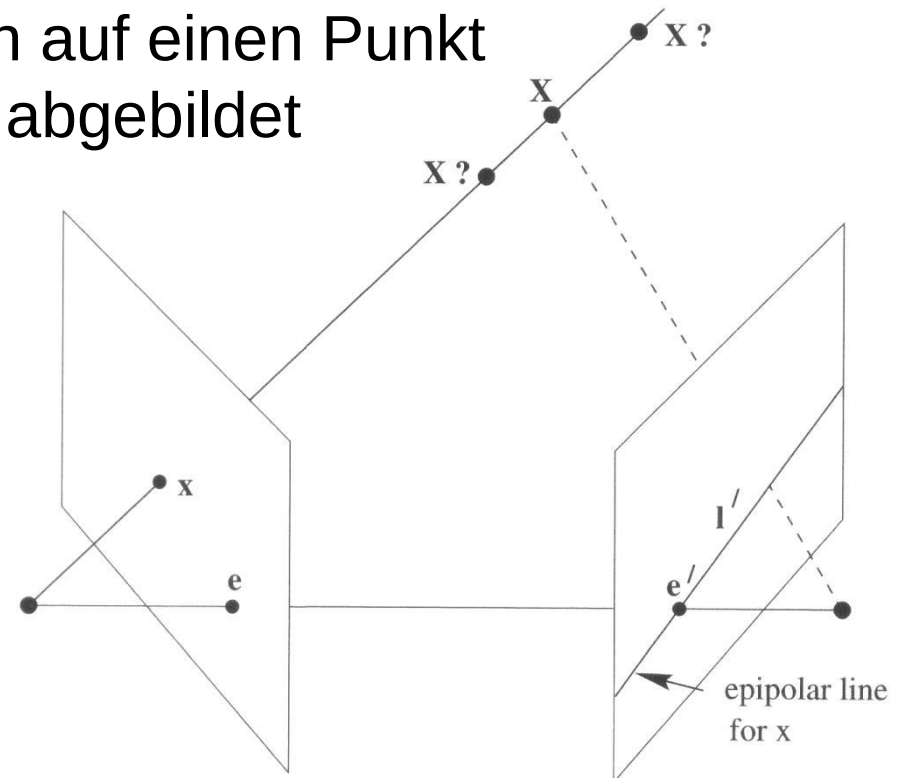


- *Epipolarebene $\pi(X)$:*
Ebene, die durch C , C' und Szenenpunkt X aufgespannt wird



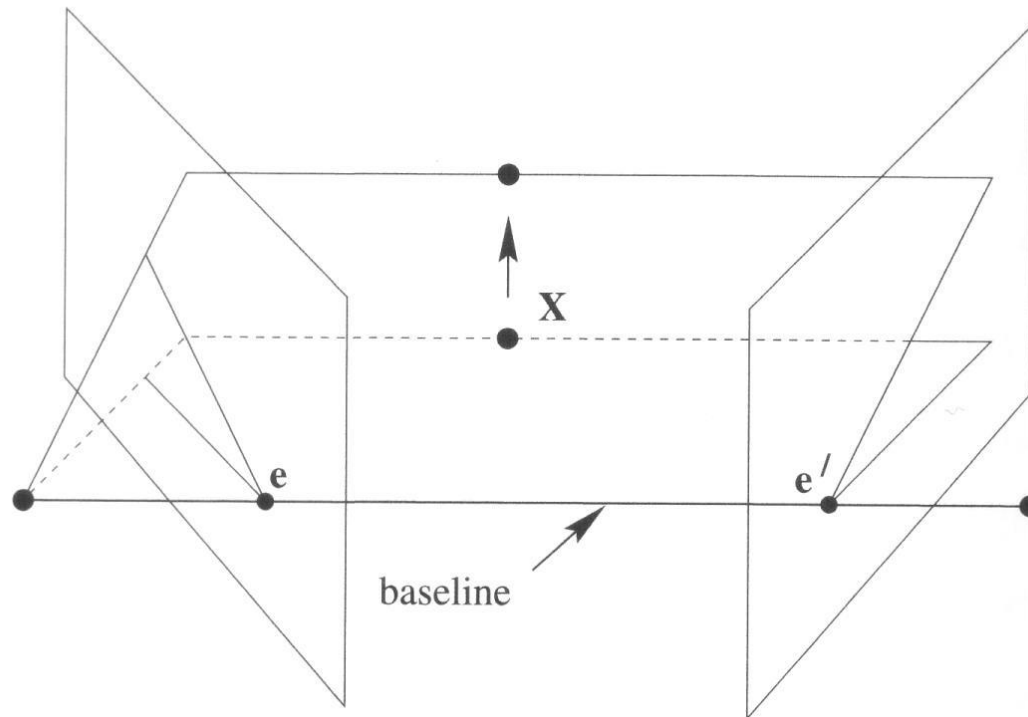
Epipolargeometrie III

- Epipolarlinie $l'(x)$: Schnittgerade von $\pi(X)$ mit Bildebene
- Alle Punkte X , die auf x in Kamera 1 abgebildet werden, werden auf einen Punkt der Linie $l'(x)$ in Kamera 2 abgebildet



Epipolargeometrie IV

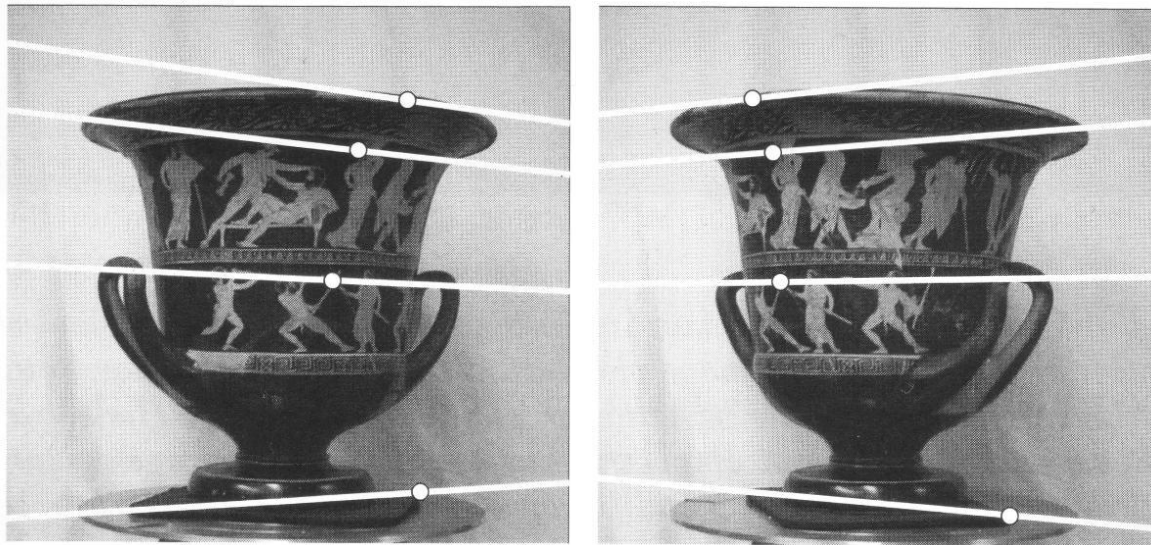
- Alle Epipolarlinien eines Kamerasystems schneiden sich in den Epipolen e und e'



Nutzen:

Einschränkung des Korrespondenzproblems von zwei Dimensionen auf eine Dimension, da nach entsprechenden Merkmalen nur noch entlang der Epipolarlinie gesucht werden muss:

- Höhere Robustheit (weniger falsche Korrespondenzen)
- Höhere Effizienz



- Mathematische Beschreibung der Epipolargeometrie erfolgt durch die Fundamentalmatrix
- Eigenschaften der Fundamentalmatrix F :
 - Ist 3×3-Matrix
 - Hat Rang 2
 - Für alle Korrespondenzen $\mathbf{x}, \mathbf{x}'$ gilt: $\mathbf{x}'^T F \mathbf{x} = 0$
 $\mathbf{x}$ und $\mathbf{x}'$ sind Bildpunkte in homogenen Koordinaten mit $w = 1$

- Mit der Fundamentalmatrix lassen sich die Epipolarlinien berechnen
- Epipolarlinien:
 - $l = F^T x'$
 - $l' = Fx$
- Für die Epipole gilt:
 - $Fe = 0$
 - $F^T e' = 0$
- Hinweis: l (bzw. l') definieren eine 2D-Gerade wie folgt: $l \cdot x = 0$ für alle Bildpunkte x (in homogenen Koordinaten mit $w = 1$), die auf dieser Geraden liegen

- Die Fundamentalmatrix lässt sich auf mehrere Arten berechnen:
 - Über Bildpunkt-Korrespondenzen in der linken und rechten Kamera
 - Bei bekannter intrinsischer und extrinsischer Kalibrierung der Kameras direkt über die Kalibriermatrizen K , K' und die Essentialmatrix E , die durch die extrinsischen Parameter definiert ist

Fundamentalmatrix IV

Berechnung über Bildpunkt-Korrespondenzen:

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0, \quad \mathbf{x}' = (x', y', z'), \quad \mathbf{x} = (x, y, z)$$

$$\Rightarrow \begin{aligned} &x'x f_{11} + x'y f_{12} + x' f_{13} \\ &+ y'x f_{21} + y'y f_{22} + y' f_{23} \\ &+ x f_{31} + y f_{32} + f_{33} = 0 \end{aligned}$$

Für $n \geq 7$ Korrespondenzen $\mathbf{x}, \mathbf{x}'$:

$$\underbrace{\begin{pmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} f_{11} \\ f_{12} \\ \vdots \\ f_{33} \end{pmatrix}}_{\mathbf{f}} = 0$$

$\mathbf{A} \mathbf{f} = 0$ lösen z.B. mit Singulärwertzerlegung (SVD)

- Berechnung der Fundamentalmatrix über Essentialmatrix
- Essentialmatrix lässt sich durch die extrinsischen Parameter berechnen:
 - Gegeben:
 - Kamera 1 mit $(I | \mathbf{0})$ als Transformation (Identität)
 - Kamera 2 mit $(R | \mathbf{t})$ als Transformation
 - Essentialmatrix E lässt sich berechnen zu:

$$E = [\mathbf{t}]_{\times} R = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix} R$$

Für die Epipole gilt:

$$\mathbf{e} = -KR^T \mathbf{t}$$

$$\mathbf{e}' = K' \mathbf{t}$$

- Hat man die Essentialmatrix (z.B. berechnet über die extrinsischen Parameter) und die intrinsischen Parameter, d.h. die Kalibriermatrizen K, K' , so lässt sich die Fundamentalmatrix berechnen zu:

$$F = K'^{-T}EK^{-1}$$

- Hat man umgekehrt die Fundamentalmatrix (z.B. bestimmt über Bildpunkt-Korrespondenzen) und die intrinsischen Parameter, d.h. die Kalibriermatrizen K, K' , so lässt sich die Essentialmatrix berechnen zu:

$$E = K'^TFK$$

- Weitere Eigenschaften der Fundamentalmatrix:
 - Mit ihr lassen sich die Eingabebilder *rektifizieren*
 - Nach Rektifizierung verlaufen alle Epipolarlinien horizontal mit derselben v -Koordinate wie der Bildpunkt im anderen Kamerabild
 - Nach Korrespondenzen muss nur noch horizontal (in eine Richtung) gesucht werden
 - Mit Hilfe der Essentialmatrix lassen sich die Projektionsmatrizen bis auf Skalierung genau rekonstruieren, mit Hilfe der Fundamentalmatrix bis auf Skalierung und Projektion genau

- Rektifizierte Bilder haben den Vorteil, dass sich optimierte Korrelations-Algorithmen für die Lösung des Korrespondenzproblems verwenden lassen
 - ⇒ Laufzeit unabhängig von der Fenstergröße
- Nachteile:
 - Interpolation notwendig für die Berechnung der rektifizierten Bilder
 - ⇒ Qualitätsverlust
 - Bilder je nach Aufbau stark verzerrt

- Beispiel für Rektifizierung mit einem Standard-Stereo-Setup $\Rightarrow$ relativ geringe Verzerrungen

Originalbilder
links / rechts



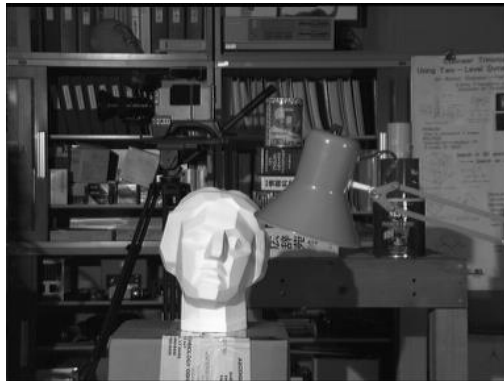
Rektifizierte
Bilder
links / rechts



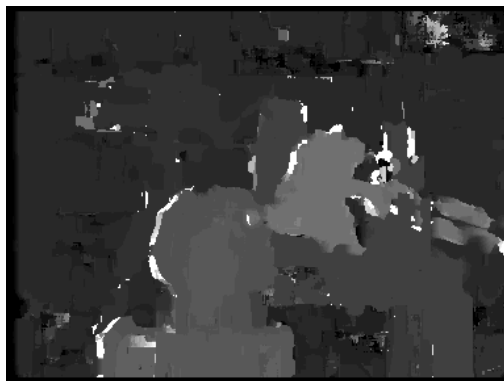
- Nach Lösung des Korrespondenzproblems können:
 - Punktwolken berechnet werden durch Triangulation, wie zuvor erläutert
 - Tiefenbilder erzeugt werden durch Eintrag der Disparitäten (Differenz der u -Koordinaten für gefundene Korrespondenzen in den rektifizierten Bildern) in ein Graustufenbild:
 - ⇒ Je höher der Grauwert, desto näher befindet sich der entsprechende 3D-Punkt zur Kamera

- Beispiel für ein Standard-Benchmark Bildpaar „Tsukuba“

Linkes
Bild



Rechtes
Bild



Tiefenbild

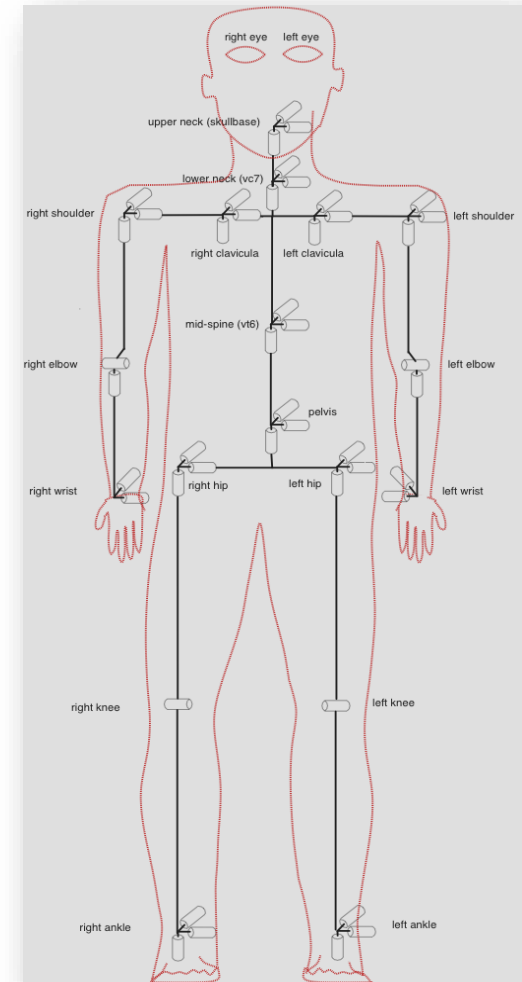
Human Motion Capture: Erfassung von Bewegungen und Handlungen des Menschen

- Problemstellung Human Motion Capture (HMC):
 - Eingabe:
Sequenz von Bildern bzw. Bildpaaren oder Bildtupeln, oder Punktwolken
 - Ausgabe:
Geschätzte Konfiguration des Menschen(Gelenkwinkel) für jedes Frame bezüglich eines zuvor definierten Menschmodells
 - Schwierigkeit:
Hohe Dimensionalität des Suchraumes
 - Eingeschränkte Beobachtbarkeit

- Menschmodell für Human Motion Capture (HMC) setzt sich zusammen aus:
 - Kinematischem Modell
 - Geometrischem Modell
 - Meist aus Festkörpern bestehend
 - Optional deformierbares Muskel- und Hautmodell
- Aus Gründen der Rechenzeit werden vereinfachte Modelle verwendet

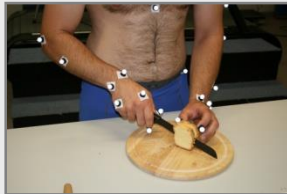
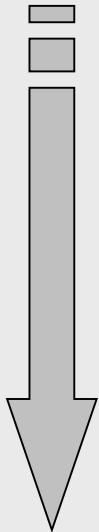
Kinematisches Modell des Menschen

- Definiert die Anzahl und Art der Gelenke
- Definiert die Segmentlängen zwischen den Gelenken
- Für die Erfassung wird die Schulter meist durch ein einzelnes Kugelgelenk modelliert

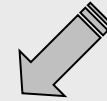
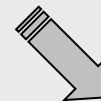


Bewegungs- und Handlungserfassung

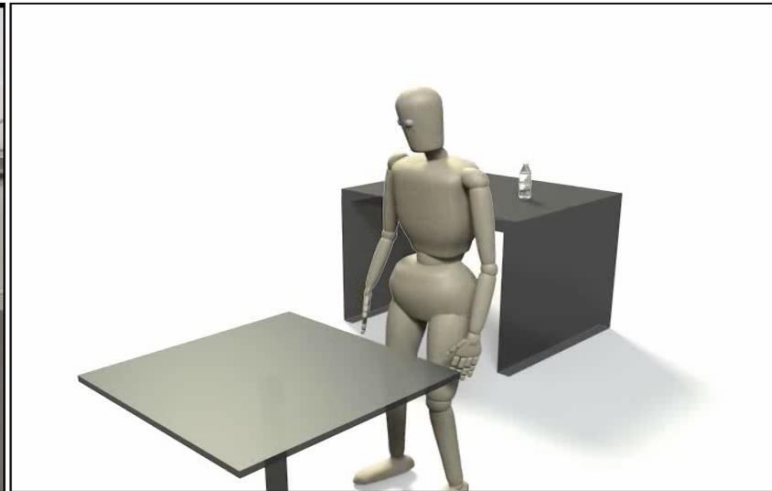
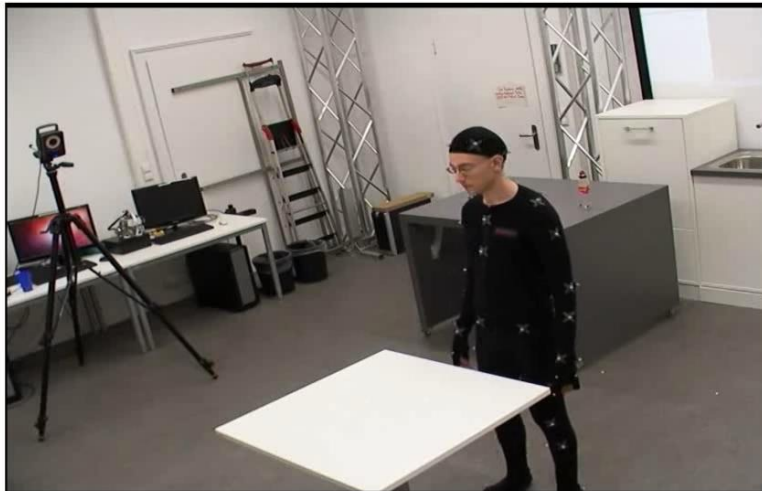
Sequential Motion



Parallel and coordinated Motion



KIT whole-body human motion database



Conversion of Human and Object Motions with the MMM Framework

<https://motion-database.humanoids.kit.edu/>

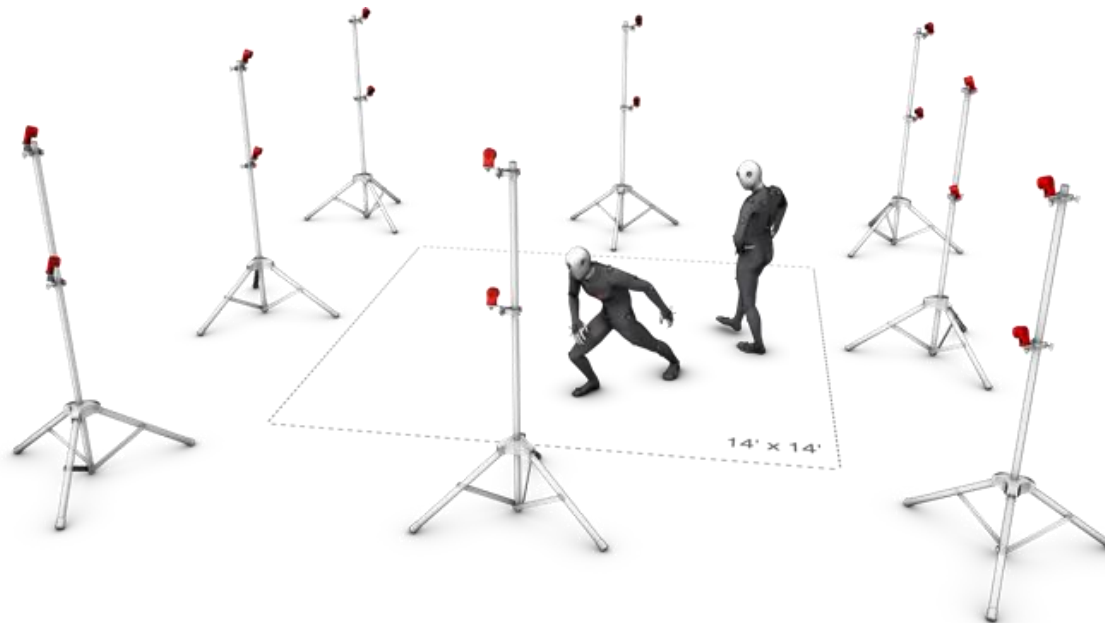
Action representation

- Hidden Markov Models (HMM) Humanoids 2006, IJHR 2008
 - Extract key points (KP) in the demonstration
 - Determine key points that are common in multiple demonstrations (common key points: CKP)
 - Reproduction through interpolation between CKPs
- Dynamic movement primitives (DMP) ICRA 2009, T-RO 2010
 - Ijspeert, Nakanishi & Schaal, 2002
 - Trajectory formulation using canonical systems of differential equations
 - Parameters are estimated using locally weighted regression
- Spline-based representations Humanoids 2007
 - fifth order splines that correspond to minimum jerk trajectories to encode the trajectories
 - Time normalize the example trajectories
 - Determine common knot points so that all example trajectories are properly approximated. Similar to via-point, key-points calculation.

Perception

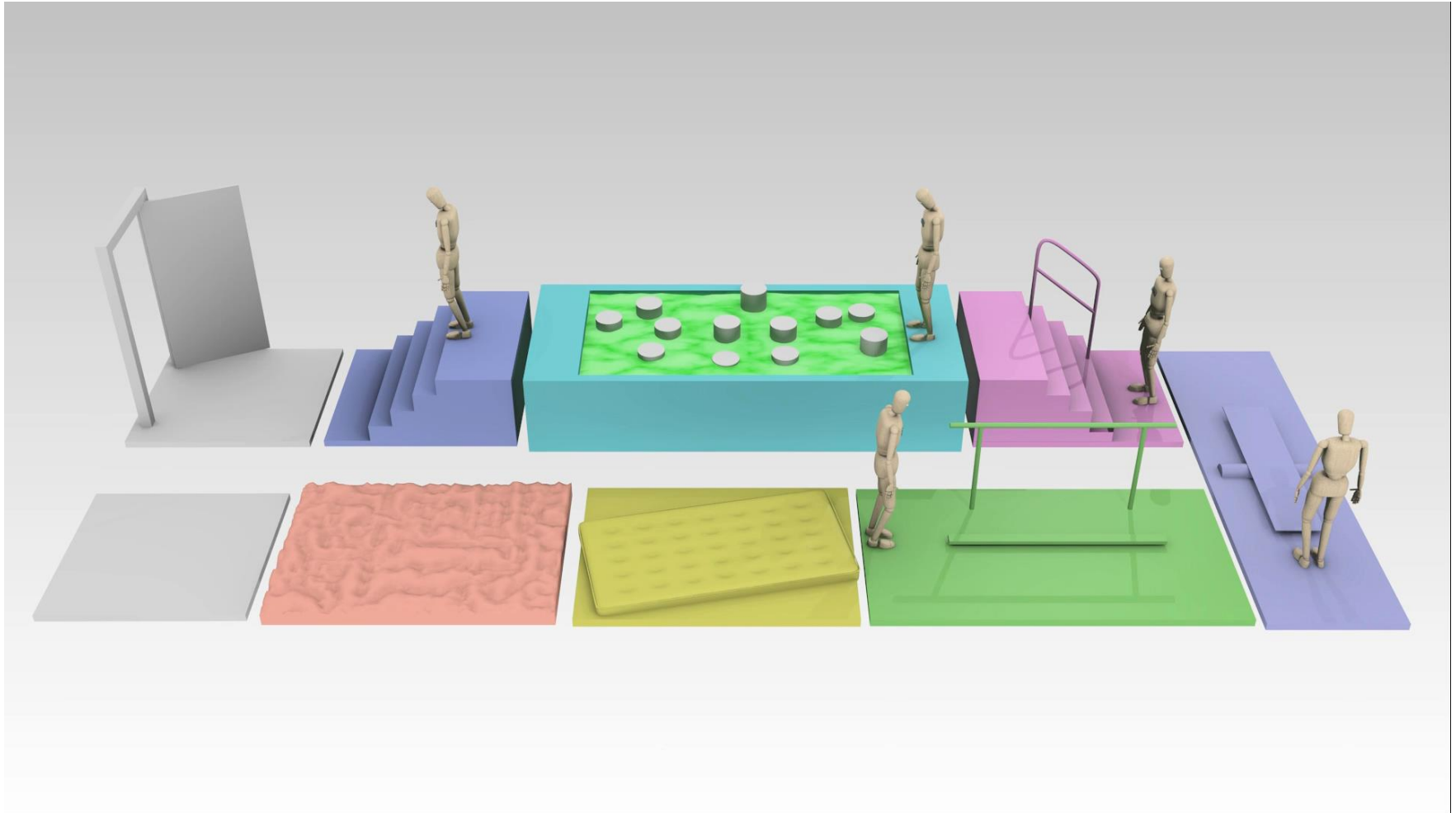
Motion Capture: Optitrack

- Motion tracking system as ground truth reference

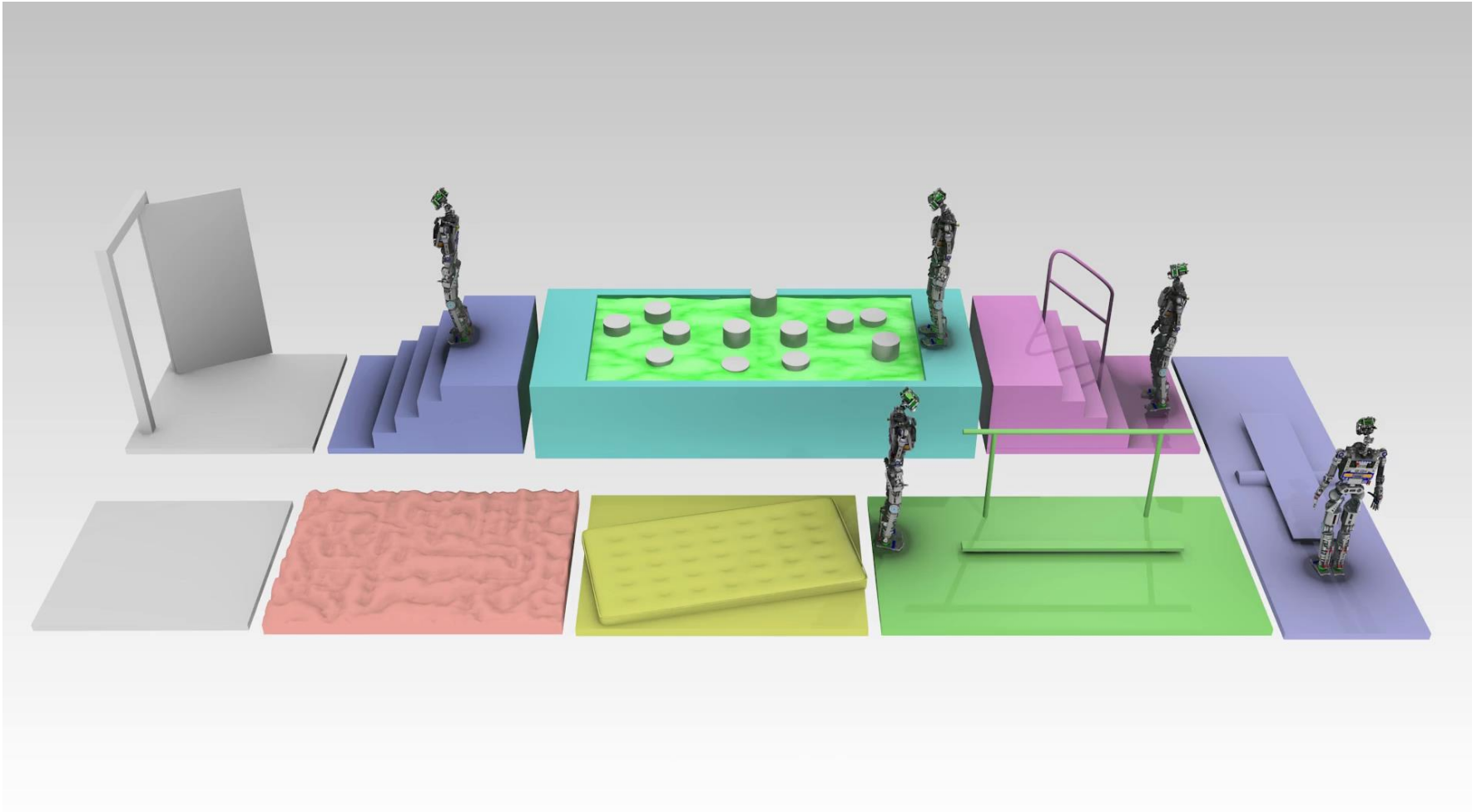


- Cameras: Flex13:
 - 1.3 million pixels, 120 FPS, 56° FOV
- Positional error < 0.3mm, rotational error < 0.05°

Human → MMM

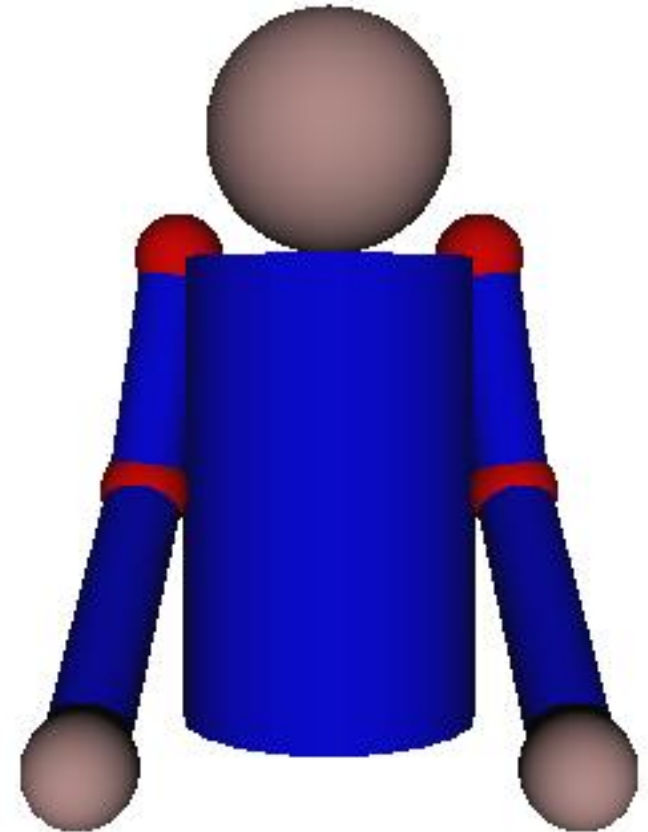


MMM → ARMAR-4



■ Geometrisches Modell

- Definiert die 3D-Form der einzelnen Segmente
- Übliche 3D-Primitive:
 - Zylinder
 - Kegelausschnitte
 - Kreisförmiger Querschnitt
 - Ellipsenförmiger Querschnitt



- Berechnung der projizierten Kontur $\overline{P_1P_2}$ und $\overline{P_3P_4}$ eines Kegelausschnitts mit kreisförmigem Querschnitt
- Gegeben (Z ist Projektionszentrum = Ursprung):
 - Fußpunkt c
 - Richtung n
 - Länge L , Radien r, R

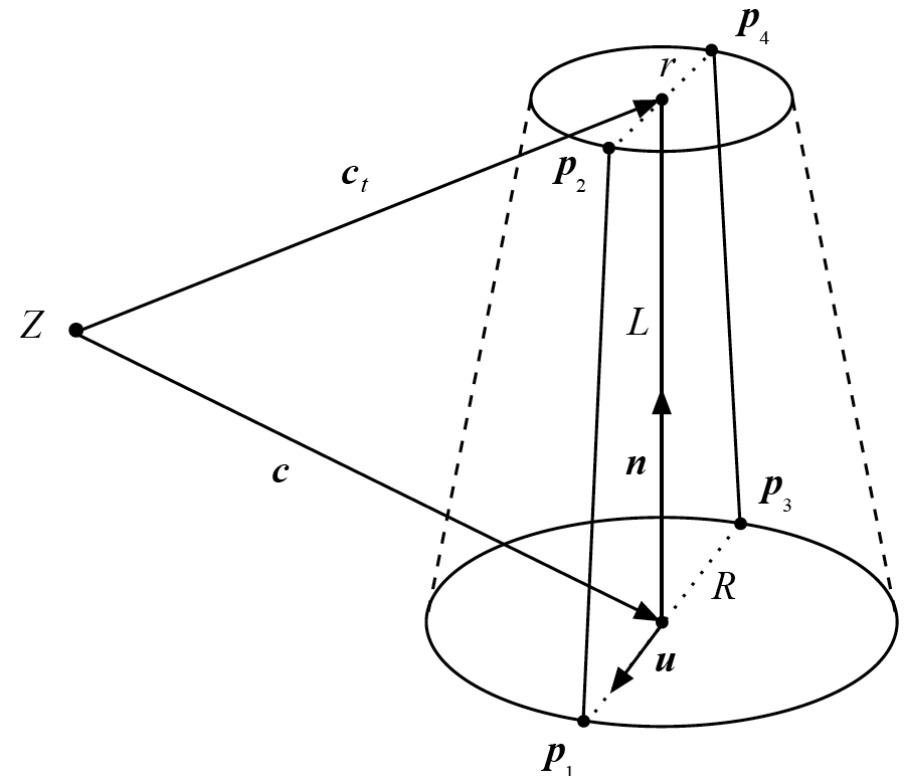
- Berechnung:

$$u = \frac{n \times c}{|n \times c|}$$

$$c_t = c + L \cdot \frac{n}{|n|}$$

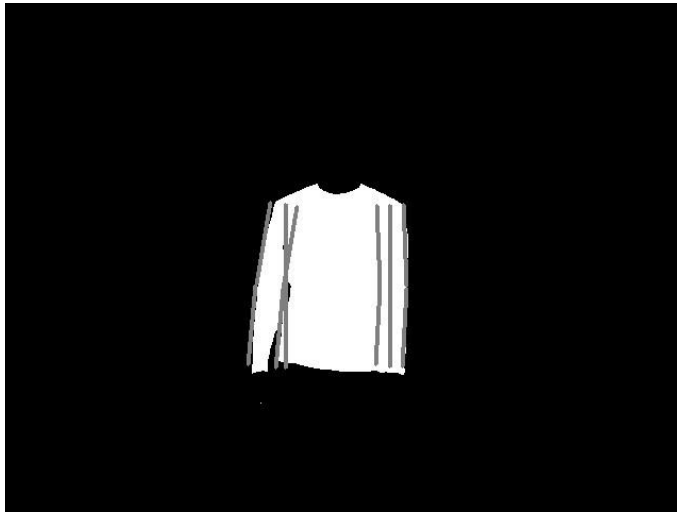
$$p_{1,3} = c \pm R \cdot u$$

$$p_{2,4} = c_t \pm r \cdot u$$

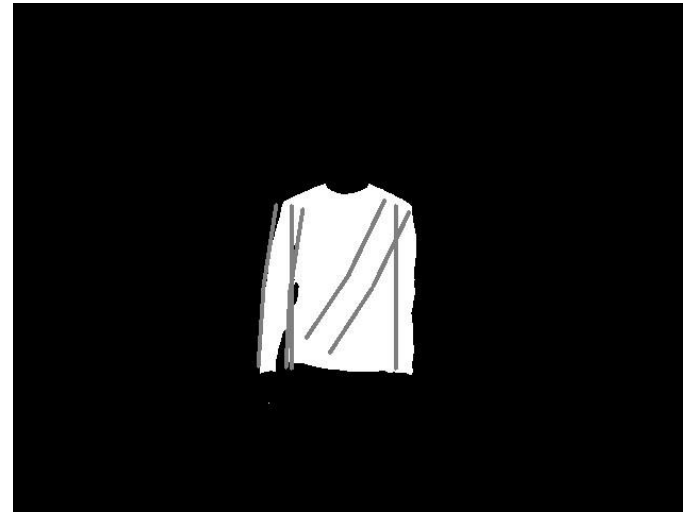


- Den Kern bildet eine Wahrscheinlichkeitsfunktion, welche bewertet, wie gut eine gegebene Konfiguration des Menschmodells zu den aktuellen Beobachtungen (Bilddaten) passt
- Hinweise (engl. *Cues*) für die Bewertung, die aus den Bilddaten gewonnen werden können, sind:
 - Region Cue [Deutscher et al., 2000]
 - Kanten Cue [Deutscher et al., 2000]
 - Distanz Cue [Azad et al., 2006]

- Beispiel einer Mehrdeutigkeit der Region Cue:
Beide Konfigurationen produzieren die gleiche Bewertung
für dasselbe Segmentierungsergebnis



Korrekte Konfiguration

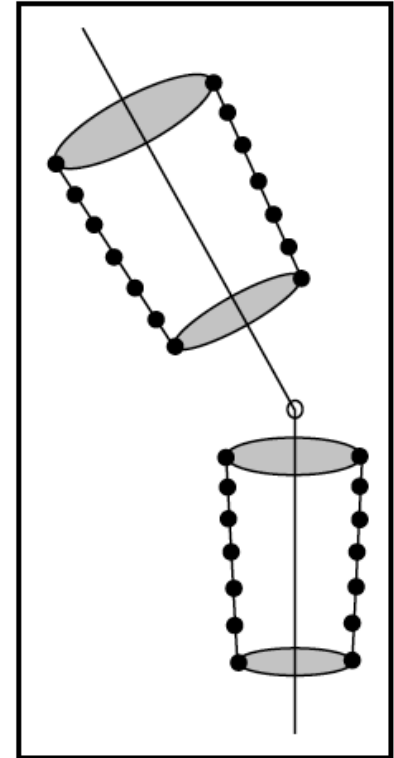


Falsche Konfiguration

■ Kanten Cue

- Benötigt Segmentierung der zur Person gehörenden Kanten
- Bewertet den Abgleich des Kanten-Bildes mit der Projektion der Kontur der Körpersegmente
- Hierzu werden Punkte der projizierten Kontur überprüft
- Bewertungsfunktion:
(Menge der gerasterten Punkte P ,
Gradientenbild $I_g, I_g(u,v) \in [0, 1]$)

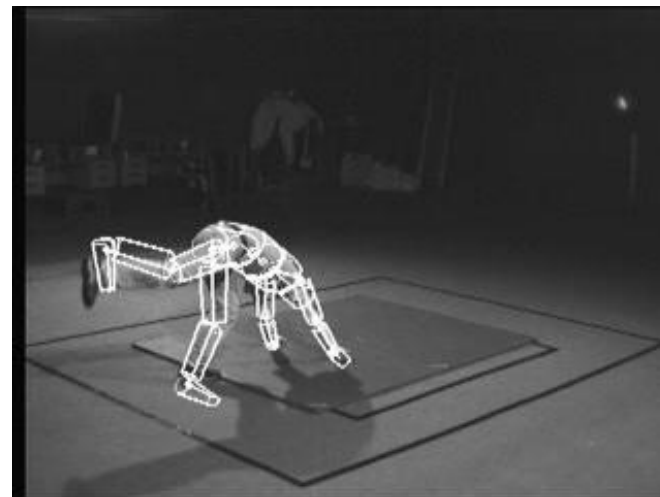
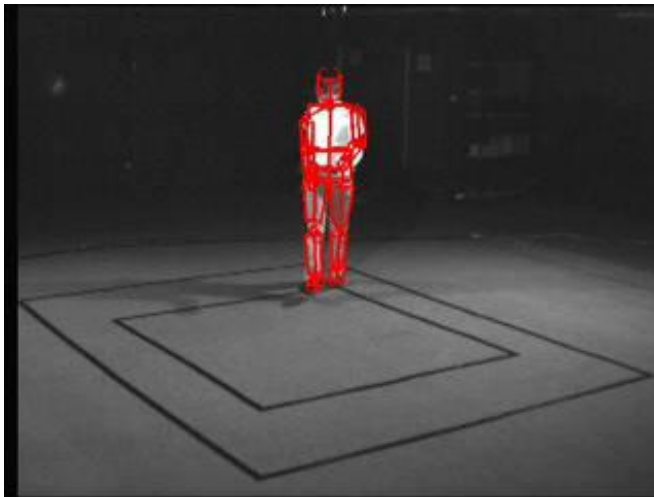
$$w_g(I_g, P) = \frac{1}{|P|} \sum_{i=1}^{|P|} (1 - I_g(\mathbf{p}_i))^2$$



(Aus [Deutscher et al., 2000])

■ System aus [Deutscher et al., 2000]

- Fusion der Kanten und Region Cue durch Multiplikation der Wahrscheinlichkeiten
- Drei Kameras in ca. 120 Grad Abstand auf die Szene gerichtet
- Verarbeitungsrate: 15 Sekunden pro Frame (1,5 GHz CPU)
- Kann komplexe Bewegungen erfassen



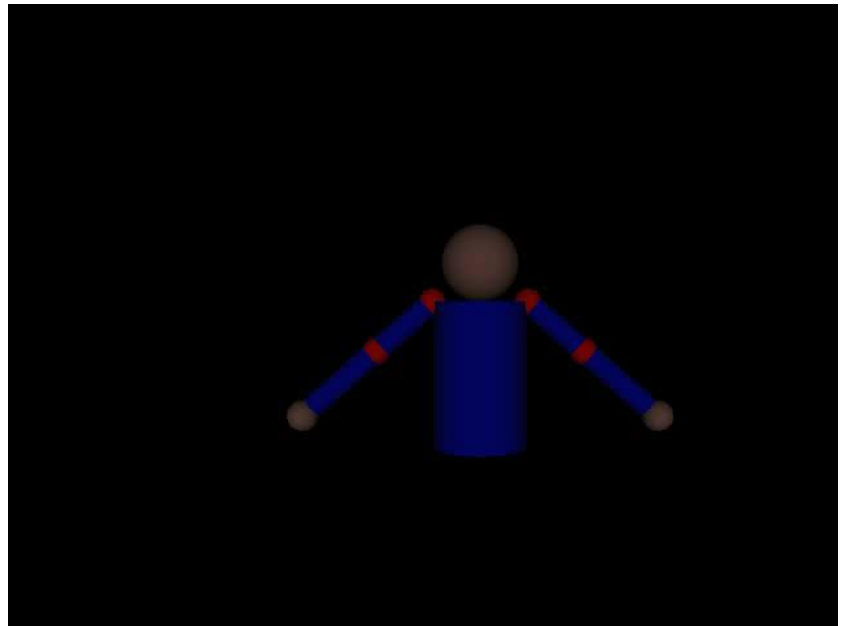
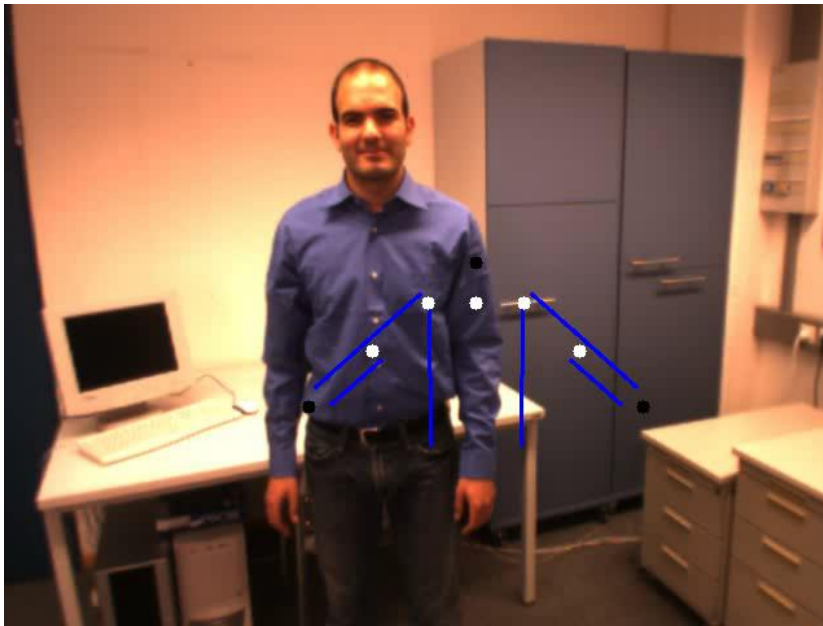
- Probleme bei der Anwendung des Ansatzes auf dem Stereokopf eines humanoiden Roboters:
 - Nur zwei Kameras im Augenabstand (70–100 mm)
 - Geringere Tiefenauflösung
 - Vollständige Okklusionen (Verdeckungen)
 - Roboterkopf nicht statisch
 - Echtzeitanforderung (optimalerweise 30 Hz oder höher)
- Ansatz:
 - Separates Tracking der Hände und des Kopfes
 - Eingabe der bestimmten Positionen in eine Distanz Cue

■ Distanz Cue

- Benötigt 3D Hand/Kopf-Tracking
- Bewertet die Distanz zwischen den erfassten Positionen mittels Hand/Kopf-Tracking und den entsprechenden durch die zu bewertende Konfiguration vorgegebenen Positionen (über Vorwärtskinematik)
- Bewertungsfunktion:
(Menge der zu verfolgenden Modellpunkte P ; zu p_i korrespondierender verfolgter Punkt $p'_i(I_d)$)

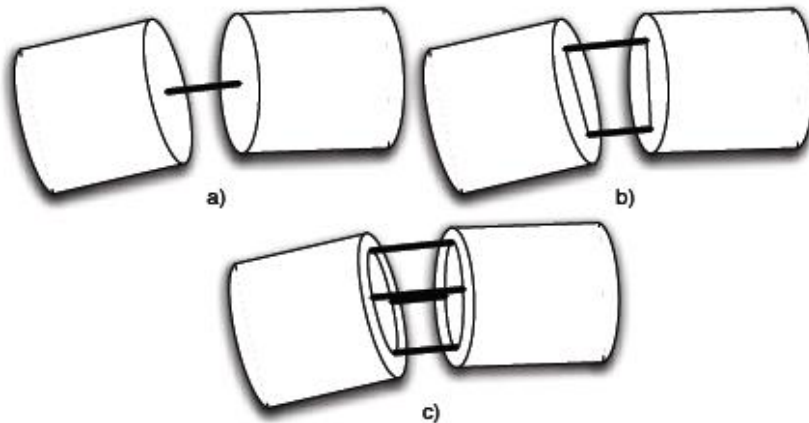
$$w_d(I_d, P) = \sum_{i=1}^{|P|} |p_i - p'_i(I_d)|^2$$

- System aus [Azad et al., 2008]
 - Stereokamerasystem mit 90 mm Augenabstand
 - Distanz Cue, spezielle Fusion, inverse Kinematik
 - Verarbeitungsrate von 40 Hz auf einem Core 2 Duo 3 GHz



■ [Knoop et al., 2006]

- Als 3D-Sensor wird der Swiss Ranger verwendet
- Gelenkwinkelbeziehungen werden über künstliche Punktkorrespondenzen einbezogen

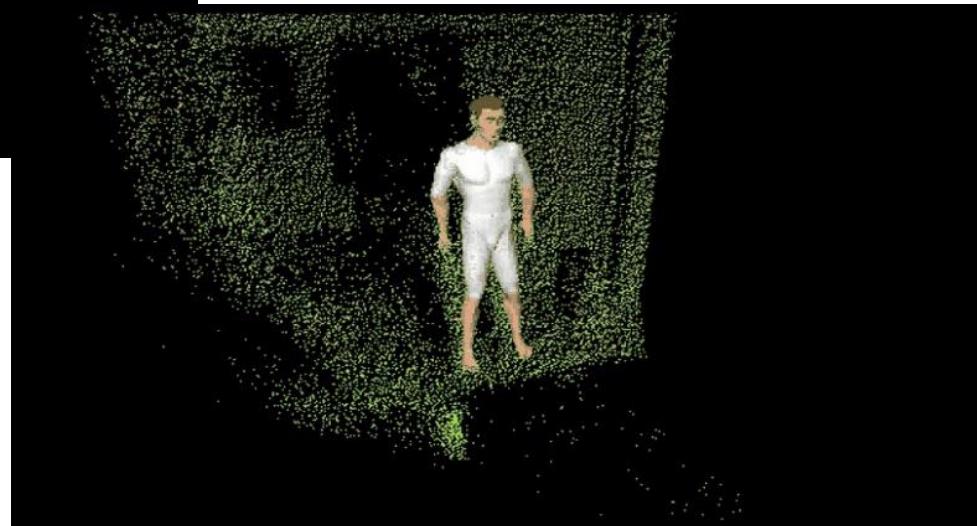
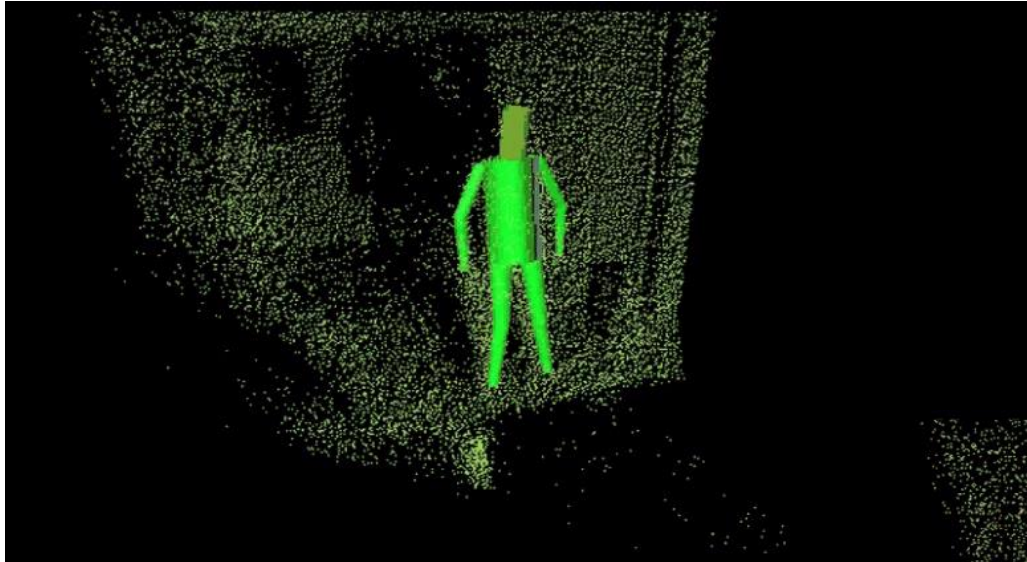


- a) Universales Gelenk mit 3 DoF
- b) Scharniergelenk mit 2 eingeschränkten DoF
- c) Elliptisches Gelenk mit 3 eingeschränkten DoF



Swiss Ranger SR3000
[<http://mesa-imaging.ch>]

■ Beispielanwendung



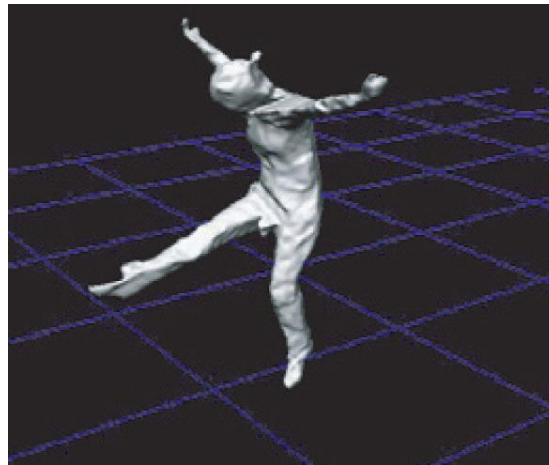
■ [Ogawara et al., 2007]

- Dichte 3D-Punktwolken und -Netze durch Volumenschnittverfahren
- Deformierbares Hautmodell
- Zweistufiges Verfahren:
 - Hierarchische Registrierung jedes einzelnen Segments (erst Torso, dann Oberarme, usw.)
 - Berücksichtigung von Gelenkwinkelbeziehungen durch Optimierungsverfahren auf der Basis von 3D-3D Punktkorrespondenzen
- Sehr rechenaufwändig → Offline-Anwendung

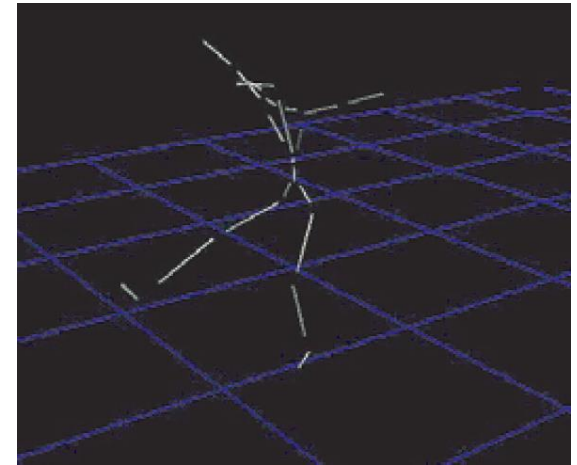
- Beispielbilder aus [Ogawara et al., 2007]



Eingabebild einer Kamera



Ergebnis des
Volumenschnitts



Berechnete Lage der
Körpersegmente

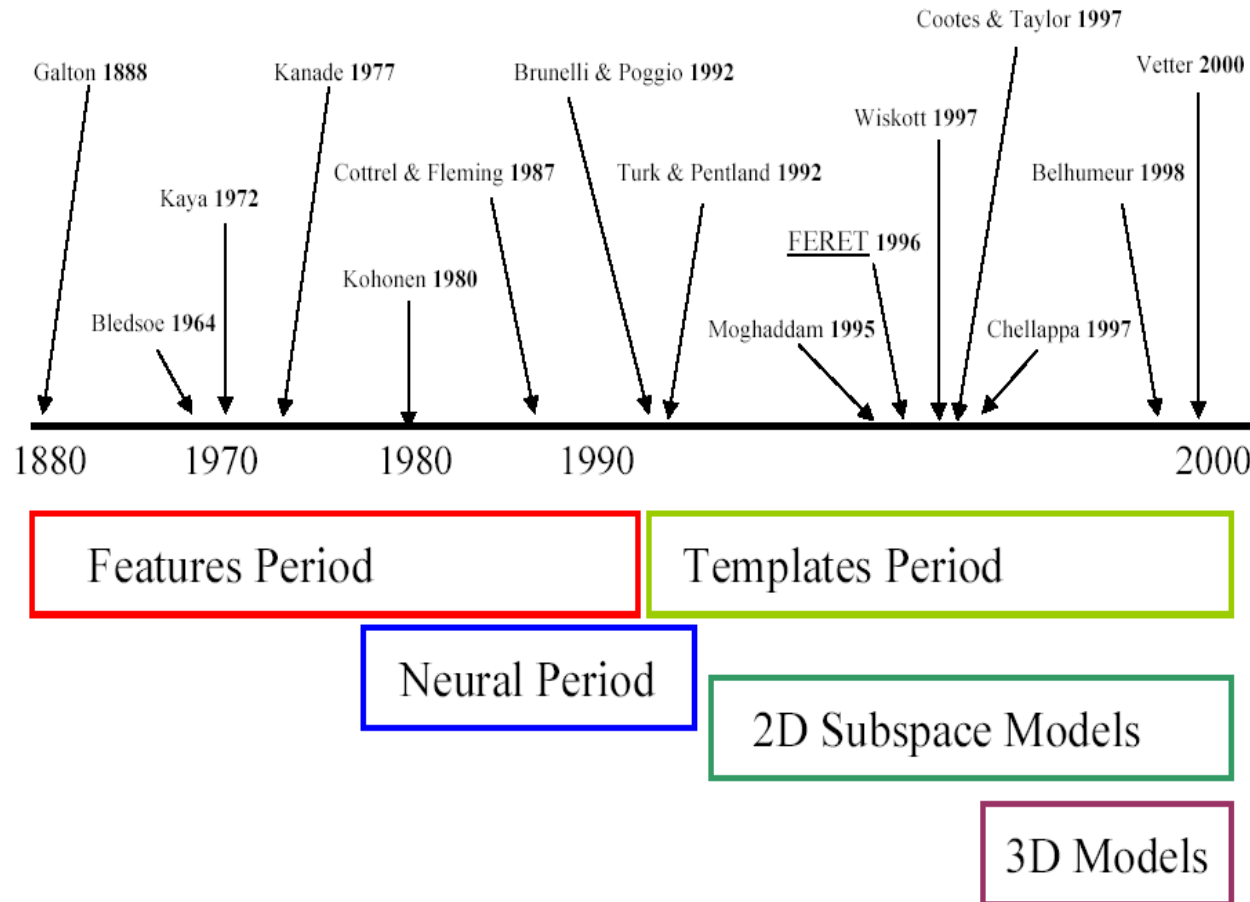
Gesichtserkennung

- Zu unterscheiden sind:
 - Gesichtsdetektion: Finden von Gesichtern (engl. *face detection*) (z.B. [Viola & Jones, 2001])
 - Identität der Person wird nicht bestimmt

- Im Folgenden:
Gesichtserkennung = Identifikation von Gesichtern

(die nachfolgenden Folien wurden von Prof. Stiefelhagen und Dr. Ekenel zur Verfügung gestellt)

Historischer Überblick



Gesichtserkennung vs. Objekterkennung

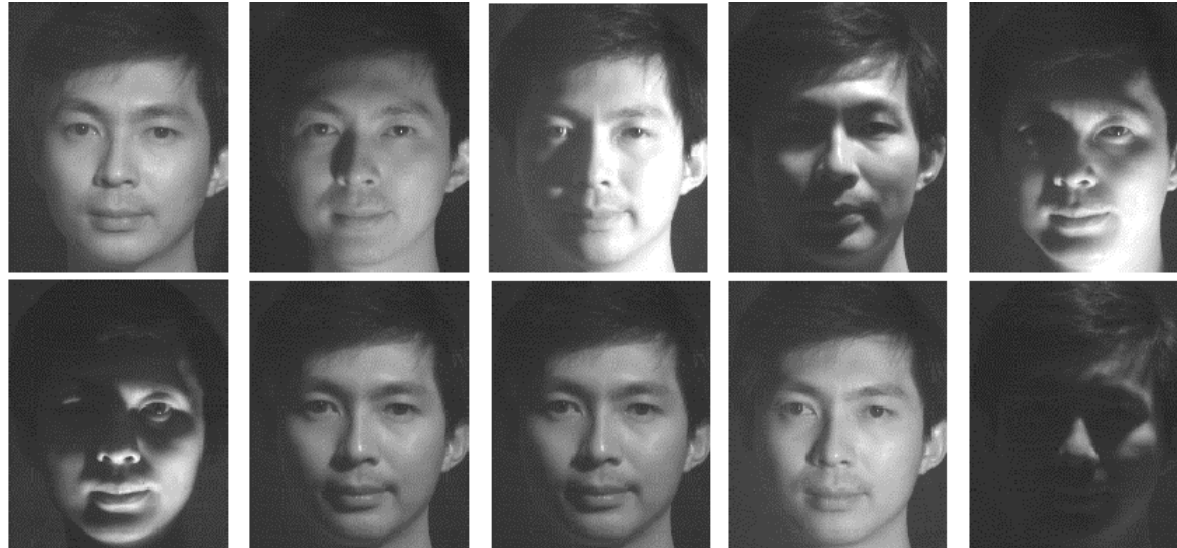
- Erkennung grundlegend unterschiedlicher Objekte und Objektklassen



- Erkennung verschiedener „Objekte“ innerhalb einer Klasse „Gesichter“



Hauptproblem

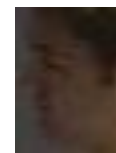


The variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity.

-- Moses, Adini, Ullman, ECCV'94

Ursachen für Variationen

- Sicht / Blickrichtung
- Beleuchtung
- Verdeckung
- Brille
- Bart
- Makeup
- Altersunterschied



1+ Jahre Unterschied

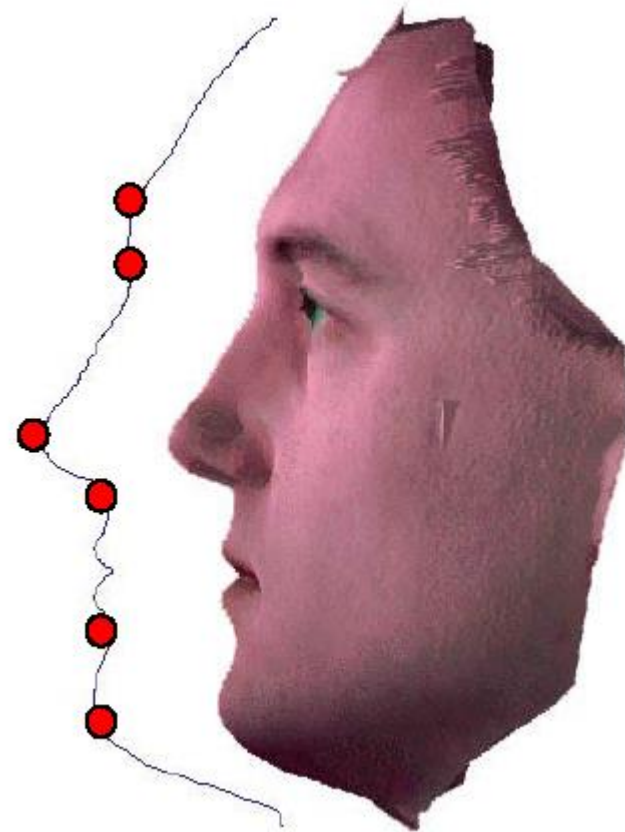
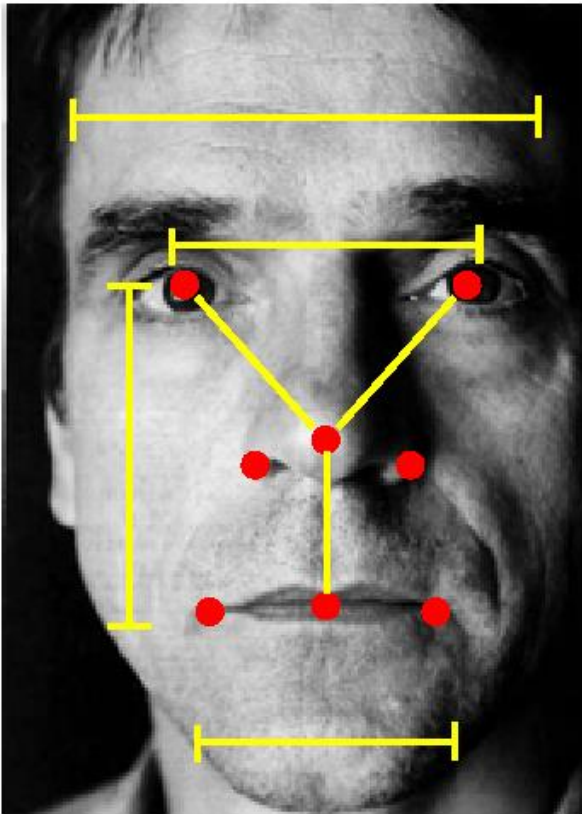
■ Merkmalsbasiert

- Markante Punkte (engl. *fiducial points*)
- Abstände, Winkel, Flächen, etc.
- Geometrische Merkmale

■ Ansichtsbasiert

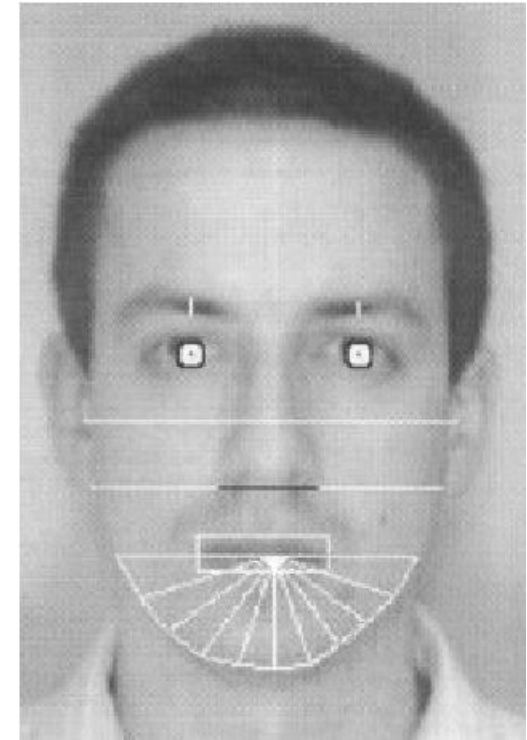
- Holistisch (global)
- Markante Regionen (lokal) (engl. *fiducial regions*)
- Statistisch

Merkmale: Frontal & Profil



Merkmalsbasierte Gesichtserkennung

- Dicke der Augenbrauen und vertikale Position relativ zum Auge
- Bogen der Augenbrauen
- Vertikale Position und Breite der Nase
- Vertikale Position des Mundes
- Breite und Höhe der Lippen
- Feste Anzahl (11) von Radien, welche die Form des Kinns beschreiben
- Breite des Gesichts auf Höhe der Nase
- Breite des Gesichts auf der mittleren Höhe zwischen Augen und Nase



R. Brunelli, T. Poggio, "Face Recognition: Features versus Templates", *IEEE Trans. on PAMI*, Vol. 15, No. 10, pp. 1042-1052, Oct. 1993.

- Nächster Nachbar-Klassifikator mit der Mahalanobis-Distanz als Distanzmetrik:

$$\Delta_j(\mathbf{x}) = (\mathbf{x} - \mathbf{m}_j)^T \Sigma^{-1} (\mathbf{x} - \mathbf{m}_j)$$

$\mathbf{x}$: Anfragebild (Gesicht)

$\mathbf{m}_j$: Durchschnittsvektor der j -ten Person.

Σ : Kovarianzmatrix

- Eine Person wird nur durch ihren durchschnittlichen Merkmalsvektor beschrieben
- Für jede Anfrage (Gesicht) werden die Merkmalsvektoren aller in der Datenbank abgespeicherten Personen überprüft

Beispielanwendung Gesichtserkennung

- System von Hazim Ekenel und Kai Nickel
(CV:HCI Prof. Stiefelhagen)



Nächste Vorlesung:

Mittwoch, 05. Juli 2017