

Transformationen

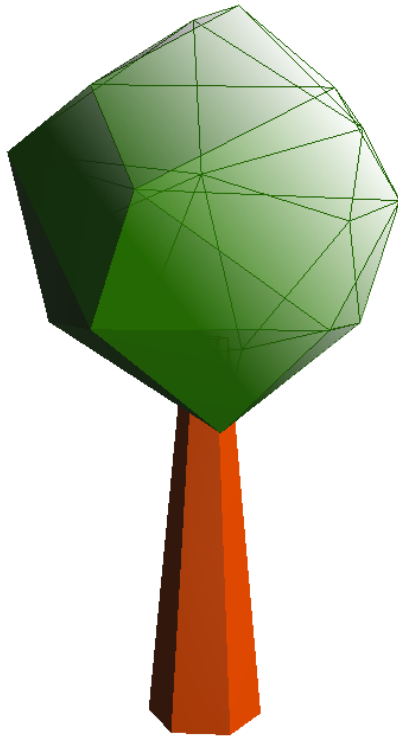
Einführung in die Computergrafik (für Augmented Reality)

Wiederholung

- Dreiecksnetze
- Normalen
- Lokale Beleuchtungsmodelle
- Texturen
- Nachbarschaftsdatenstrukturen
- Glättung
- Grundlagen: Matrizen

Agenda

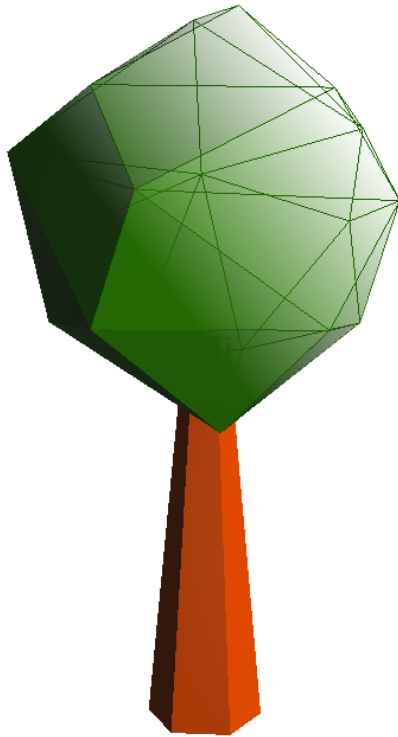
- Transformationen
- Kamera-Transformationen
- Tracking



Koordinatensysteme und Kamera- Transformation

Koordinatensysteme und Kamera-Transformation

- Koordinatensysteme
 - homogene Koordinaten
 - Basisvektoren
 - Matrizen zur Repräsentation von Transformationen
 - Rotationsmatrizen
 - Transformationen zwischen Koordinatensystemen
- Kamera
 - Model-Transformation
 - View-Transformation
 - perspektivische Transformation



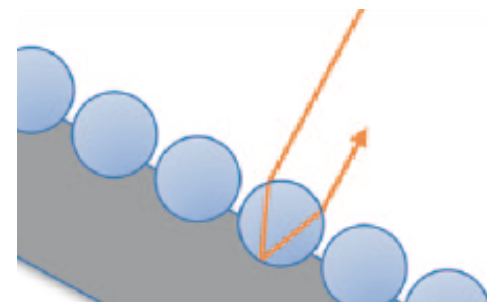
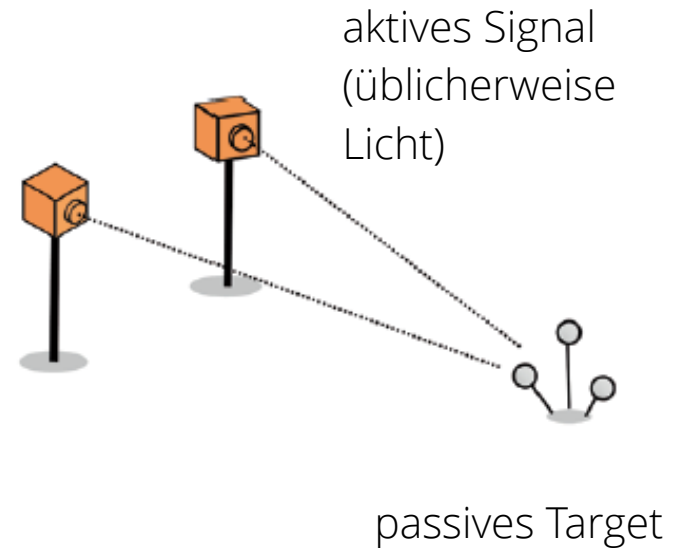
Tracking

Tracking

- Target-basiertes Tracking
- Marker-basiertes Tracking
- markerloses Tracking

Target-basiertes Tracking

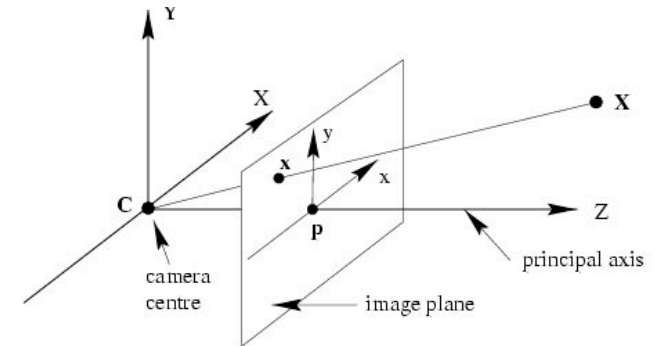
- bekannte passive Targets
- aktive Sender
 - z.B. Infrarot
- Target reflektiert Signal zurück zum Sender
- Sender ist gleichzeitig Empfänger
 - Aufnahme von Bild mit Reflexionen
 - Positionen der Reflexionen = (x,y) Bildpunkte



Reflexion beim Target

Kamera-Model

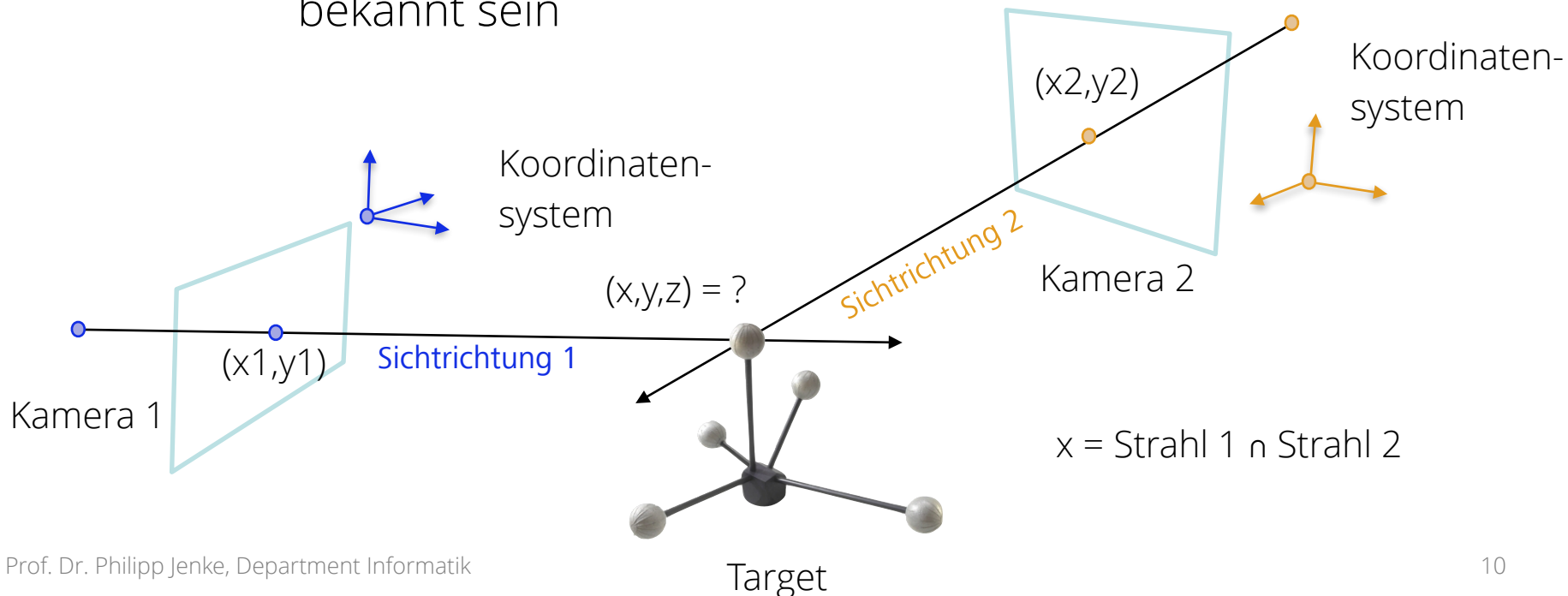
- Lochkamera-Modell
- Hauptachse = Sichtlinie
- Entfernung Auge (C) - Bildebene = focal length
- extrinsische Kameraparameter
 - Position (t_x, t_y, t_z)
 - Orientierung (Euler-Winkel)
 - extrinsische Kameramatrix
- intrinsische Kameraparameter (einfache Variante)
 - focal length ($f_x = f_y = f$)
 - Auflösung: $w \times h$
 - principal point (c_x, c_y) = ($w/2, h/2$)
 - intrinsische Kameramatrix:



$$\mathbf{K} = \begin{pmatrix} f & 0 & 0 & w/2 \\ 0 & f & 0 & h/2 \end{pmatrix}$$

Target-basiertes Tracking

- Rekonstruktion der Target-Position aus Reflexionsbildern
- mindestens zwei Kameras notwendig
 - je mehr desto besser
 - System muss kalibriert sein
 - intrinsische and extrinsische Kameraparameter müssen bekannt sein

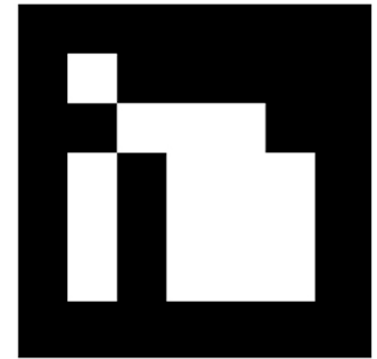
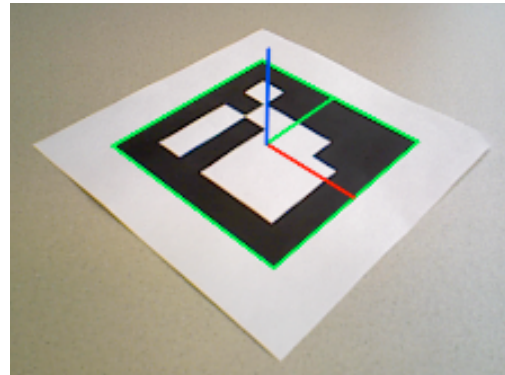


Target Position

- Kamera 1:
$$\begin{pmatrix} x_1 \\ y_1 \\ w_1 \end{pmatrix} = K_1 \cdot T_1 \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$
- Kamera 2:
$$\begin{pmatrix} x_2 \\ y_2 \\ w_2 \end{pmatrix} = K_2 \cdot T_2 \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$
- Auflösen nach der unbekannten Position (x,y,z)

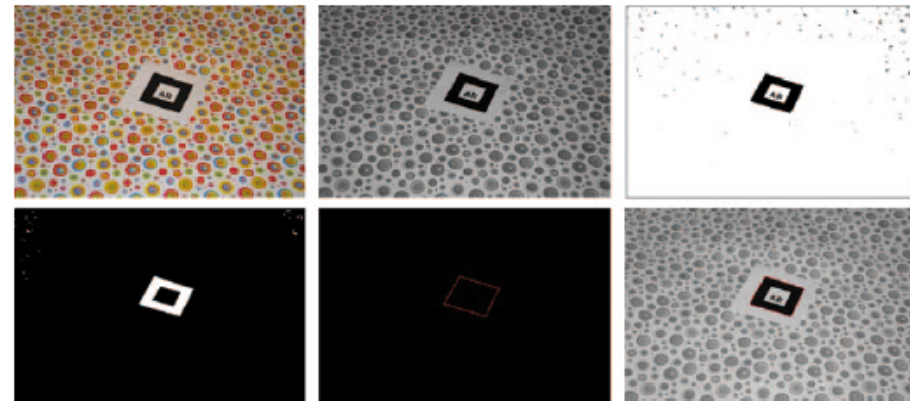
Marker-basiertes Tracking

- Idee: positioniere Marker in der Szene
- Marker repräsentiert 6-DOF Koordinatensystem
 - Verschiebung (3-DOF)
 - Orientierung (3-DOF)



Aruco Marker

- Tracking
 - Bildaufnahme
 - Kanten- und Eckenerkennung
 - Markererkennung
 - Schätzen der Pose



Kanten- und Eckenerkennung

- Computer Vision-Problem
- Canny-Algorithmus:
 - Vorverarbeitung: Bildfaltung mit Matrix M

$$M = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

- Anwenden des Sobel-Operators S
 - partielle Ableitung
 - x- und y-Richtung

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

*

- Kantenrichtung schätzen
 - Kantenstärke schätzen
 - Non-Maximum-Suppression: nur lokale Maxima behalten
 - Hysterese: Verfolgung von Startpixel, Schwellwert

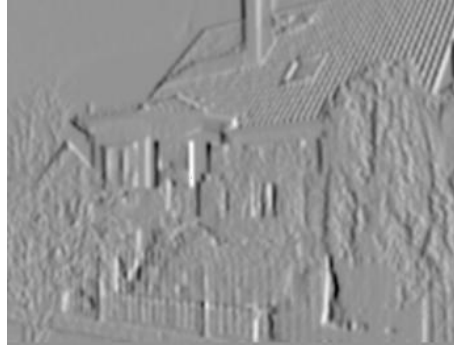
$$\theta = \text{atan}_2(g_y, g_x)$$

$$G(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2}$$

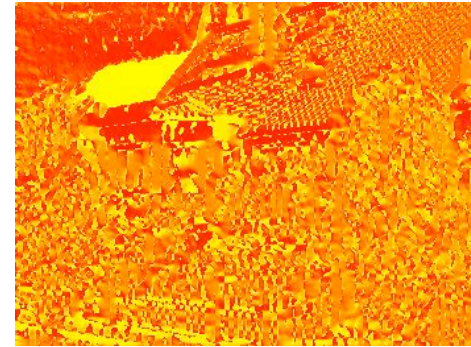
Canny Kantenerkennung



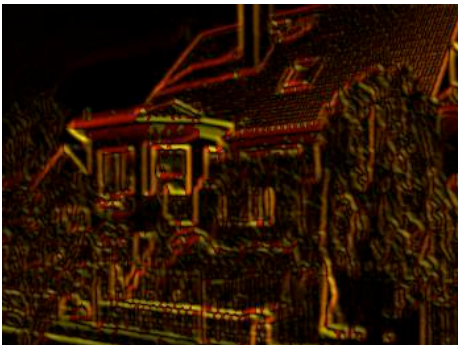
Vorverarbeitung:
Bildfaltung



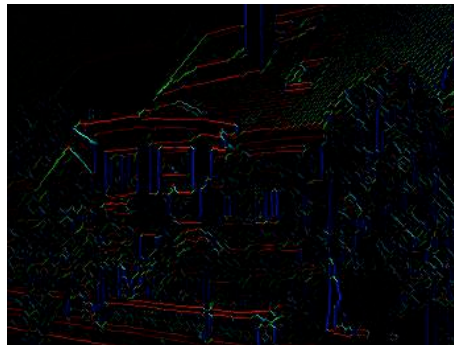
Sobel-Operator



Richtung



Kantenstärke



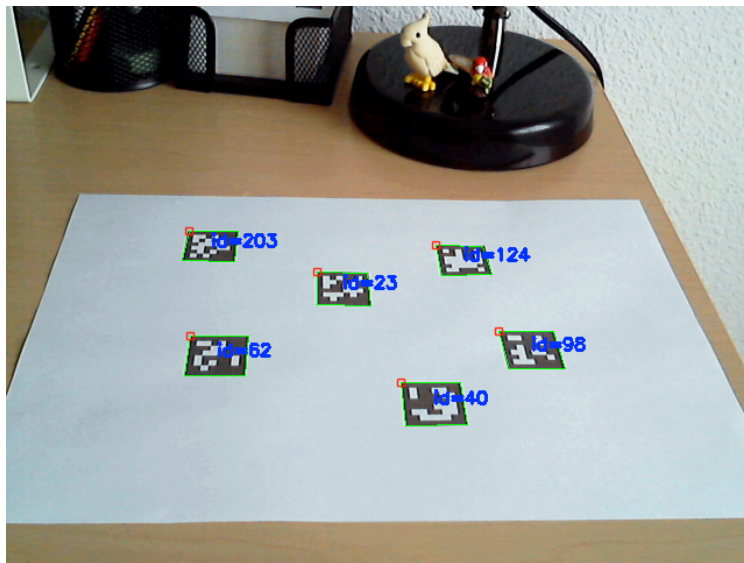
Non-Maximum-
Suppression



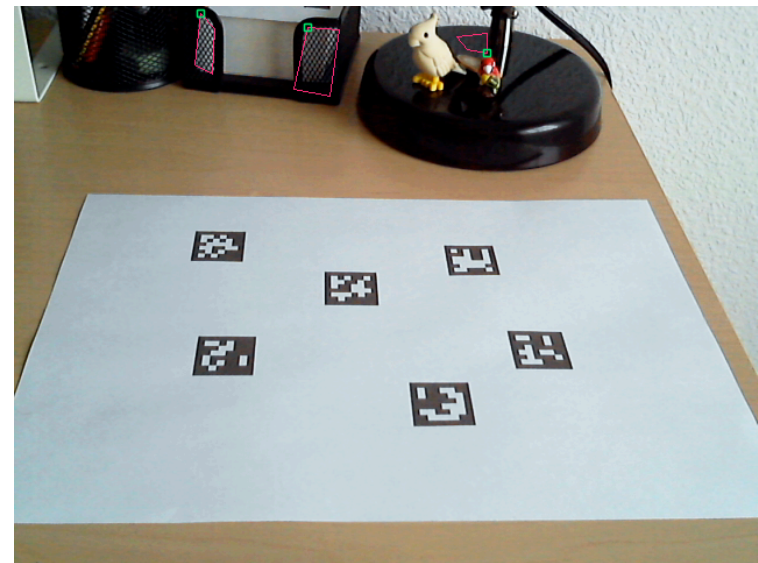
Hysterese

Erkennen von Vierecken

- Schnitt von Kanten: Ecken
- 4 durch Kanten verbundene Ecken: Kandidat für einen Marker



Marker



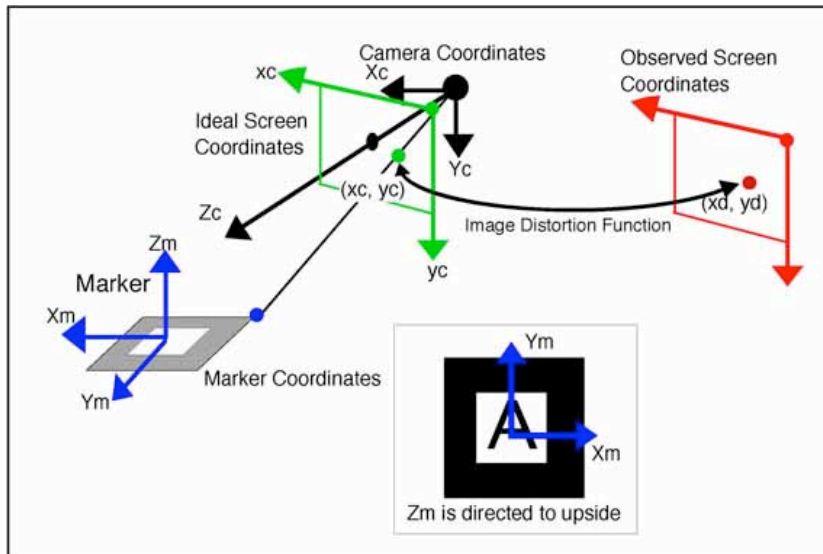
verworfenne Vierecke

OpenCV Aruco Bibliothek

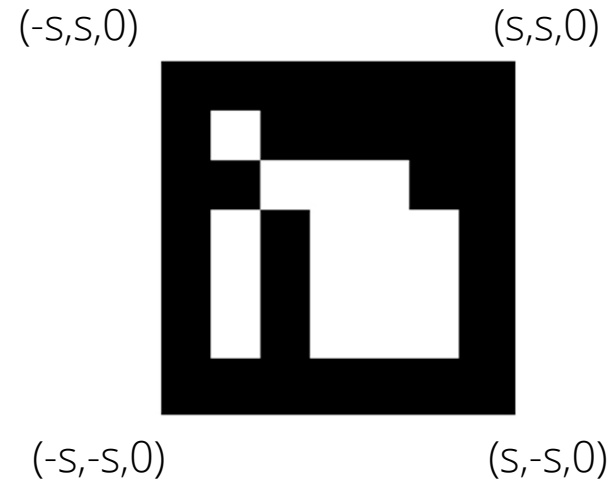
Registrierung

- gegeben:
 - Marker-Ecken (x_d, y_d)
 - intrinsische Kameraparameter K
 - Kandidaten für Marker-Ecken
- unbekannt: Marker-Pose T (Position + Orientierung)

$$p_{\text{pixel}} = KPT^{-1}p$$



Kamera- und Marker Koordinaten-
system (Quelle: ARToolkit)



Markergröße: $2s$

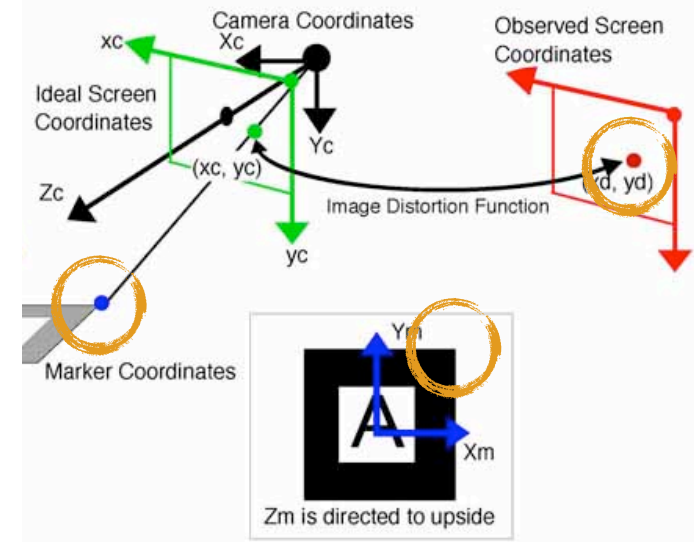
Registrierung

- Bestimmen der Marker-Pose
- Fehlerfunktion f

$$f = \sum_{i=1}^4 (x_i - \mathbf{KPT}^{-1}c_i)^2$$

- (x_d^i, y_d^i) : Pixel für Ecke i
- (c_x^i, c_y^i, c_z^i) : Eckpunkt im Marker Koordinatensystem
- Lösung: Finden des Minimums der Fehlerfunktion für Parameter
 - Position: t_x, t_y, t_z
 - Orientierung: α, β, γ

$$\arg \min_{t_x, t_y, t_z, \alpha, \beta, \gamma} f$$



Markerloses Tracking

- Informationen zum getrackten Objekt muss vorher bekannt sein
- Finden von markanten Punkten auf dem Objekt (z.B. SIFT, SURF, ...)
- Finden identischer markanter Punkte im Bild, Festlegen von Korrespondenzen
- weitere Details in der einschlägigen Literatur, z.B. [1]

Zusammenfassung

- Transformationen
- Kamera-Transformationen
- Tracking

Literatur

- [1] Schmalstieg, Dieter und Höllerer, Tobias. (2016): Augmented Reality - Principles and Practice, Addison Wesley, 1. Auflage, 2016