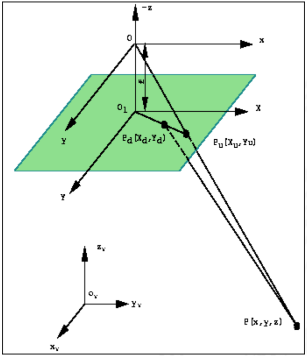
Modell einer Kamera ohne Verzeichnung



Lochkameramodell mit und ohne radiale Linseverzeichnung

Modell einer Kamera ohne Verzeichnung (cont.)

- (x_w, y_w, z_w): 3D-Weltkoordinatensystem mit Ursprung O_w
- (x, y, z): 3D-Koordinatensystem der Kamera mit Ursprung O (optisches Zentrum)
- (X, Y): 2D-Bildkoordinatensystem mit Ursprung O_1
- f: Brennweite der Kamera

Transformation von Welt- in Kamerakoordinaten

- Sei $P(x_w, y_w, z_w)$ ein Punkt im Weltkoordinatensystem
- Seine Projektion auf die Bildebene kann wie folgt ermittelt werden:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + t$$

$$\text{mit} \quad R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad \text{und} \quad t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

- Die Parameter R und t sind die *extrinsischen* Parameter

Projektion von Kamera- auf Bildkoordinaten

- Punkt P wird auf die analoge (ideale) Bildkoordinate (u, v) projiziert
- *Perspektivprojektion* mit Brennweite f:

$$u = f\frac{x}{z} \qquad v = f\frac{y}{z}$$

- Die Bildkoordinate (X, Y) errechnet sich aus (u, v) mit:

$$X = s_u u \qquad Y = s_v v$$

- Die Skalierungsfaktoren s_u und s_v rechnen die analogen Koordinaten von Meter in Pixel um
- s_u, s_v und f sind die *intrinsischen* Kamera-Parameter

Projektion von Welt- auf Bildkoordinaten

- Da nur zwei unabhängige intrinsische Parameter existieren, definiert man:

$$f_x \equiv fs_u \quad \text{und} \quad f_y \equiv fs_v$$

- Diese Gleichungen liefern das verzeichnisfreie Kameramodell:

$$X = f_x \frac{r_1 x_w + r_2 y_w + r_3 z_w + t_x}{r_7 x_w + r_8 y_w + r_9 z_w + t_z}$$

$$Y = f_y \frac{r_4 x_w + r_5 y_w + r_6 z_w + t_y}{r_7 x_w + r_8 y_w + r_9 z_w + t_z}$$

Pixelkoordinaten im Rechner

- Von den im Rechner gespeicherten Bildkoordinaten (X_f, Y_f) wird die Koordinate (C_x, C_y) des Bildmittelpunktes abgezogen
- Damit ergibt sich:

$$X = X_f - C_x$$
$$Y = Y_f - C_y$$

- Die Unsicherheit über den Bildmittelpunkt kann 10-20 Pixel erreichen

Kalibrierung einer Kamera: Grundkonzept

Das Lochkamera-Modell liefert für die Kalibrierung

- die drei unabhängigen extrinsischen Parameter von R
- die drei unabhängigen extrinsischen Parameter von t
- die intrinsischen Parameter f_x, f_y, C_x und C_y

Kalibrierungspunkte

Die Kalibrierung erfolgt mit einer Menge von m Objektpunkten, die

1. bekannte Weltkoordinaten $\{x_{w,i}, y_{w,i}, z_{w,i}\}$, $i = 1, \ldots, m$ in hinreichend genauer Präzision haben
2. innerhalb des Sichtfeldes der Kamera liegen

Diese *Kalibrierungspunkte* werden im Kamerabild mit ihren respektiven Kamerakoordinaten $\{X_i, Y_i\}$ detektiert

Kalibrierung

- Das Problem bei der Kalibrierung einer Kamera ist die Identifikation der unbekannten Koeffizienten des Kameramodells
- Die Bestimmung für das verzeichnisfreie Kameramodell liefert expliziet die Position der Kamera in Weltkoordinaten
- Die grundlegendste Strategie für eine Kamerakalibration ermittelt die Koeffizienten mit Hilfe der *linear-least-squares*-Identifikation der im folgenden vorgestellten *perspektivischen Transformationsmatrix* (engl. *Perspective Transformation Matrix*)

Verzeichnisfreies Kameramodell

Das verzeichnisfreie Kameramodell

$$X = f_x \frac{r_1 x_w + r_2 y_w + r_3 z_w + t_x}{r_7 x_w + r_8 y_w + r_9 z_w + t_z}, \qquad Y = f_y \frac{r_4 x_w + r_5 y_w + r_6 z_w + t_y}{r_7 x_w + r_8 y_w + r_9 z_w + t_z}$$

lässt sich umschreiben zu

$$X = \frac{a_{11} x_w + a_{12} y_w + a_{13} z_w + a_{14}}{a_{31} x_w + a_{32} y_w + a_{33} z_w + a_{34}}$$

$$Y = \frac{a_{21} x_w + a_{22} y_w + a_{23} z_w + a_{24}}{a_{31} x_w + a_{32} y_w + a_{33} z_w + a_{34}}$$

Perspektivische Transformationsmatrix

- Es kann $a_{34} = 1$ gesetzt werden, da eine Skalierung der Koeffizienten $a_{11}, \ldots, a_{34}$ die Werte von X und Y nicht ändert
- Die Koeffizienten $a_{11}, \ldots, a_{34}$ korrespondieren mit der so genannten *perspektivischen Transformationsmatrix*
- Die vorangegangenen beiden Gleichungen können im folgenden Identifikationsmodell zusammengefasst werden:

$$\begin{bmatrix} x_w & y_w & z_w & 1 & 0 & 0 & 0 & 0 & -Xx_w & -Xy_w & -Xz_w \\ 0 & 0 & 0 & 0 & x_w & y_w & z_w & 1 & -Yx_w & -Yy_w & -Yz_w \end{bmatrix} \begin{bmatrix} a_{11} \\ \vdots \\ a_{33} \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix}$$

Least Squares

- Die elf unbekannten Koeffizienten $a_{11}, \ldots, a_{33}$ werden mit Hilfe der Methode der kleinsten Quadrate bestimmt
- Minimal sind **sechs** Kalibrierungspunkte notwendig
- Jedes Paar Datenpunkte $\{(x_{w,i}, y_{w,i}, z_{w,i}), (X_i, Y_i)\}$ liefert zwei algebraische Gleichungen mit den gesuchten Koeffizienten
- Es kann gezeigt werden, dass die Kalibrierungspunkte **nicht koplanar** sein dürfen
- Ist dies nicht der Fall, ist die erste Matrix im Identifikationsmodell singulär, da die Spalten 3 und 4 sowie 7 und 8 linear abhängig sind

Probleme

- Die vorgestellte Lösung ist noch nicht global optimal, da bisher keine Linsenverzeichnung berücksichtigt wurde
- Es ist nicht möglich expliziet die Rotationsmatrix R und den Translationsvektor t zu bestimmen
- Das bedeutet die vorgestellte Kalibrierung ermöglicht **nicht** die Nutzung einer Kamera, die an einem sich bewegenden Roboterarm montiert ist
- Die Herstellung eines präzisen 3D-Kalibrierungsaufbaus ist aufwändiger als eine 2D-Kalibrierungsplatte

Stereo-Vision

- Die bisher vorgestellte Kalibrierungsmethode ermöglicht allerdings eine schnelle, wenn auch unpräzise Messung von Punkten mit einem Stereo-Kamera-Aufbau
- Dazu werden zwei Kameras A und B kalibriert und liefern die Kalbrationsvektoren a^A und a^B
- Dann kann die Koordinate $\{x_w, y_w, z_w\}$ eines jeden Punktes der von beiden Kameras gesehen wird berechnet werden
- Jeder unbekannte Punkt hat die korrespondierenden Bildkoordinaten $\{X^A, Y^A\}$ und $\{X^B, Y^B\}$

Stereo-Vision (cont.)

Mit der Gleichung

$$\begin{bmatrix} a_{11} - a_{31}X & a_{12} - a_{32}X & a_{13} - a_{33}X \\ a_{21} - a_{31}Y & a_{22} - a_{32}Y & a_{23} - a_{33}Y \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} X - a_{14} \\ Y - a_{24} \end{bmatrix}$$

für jede Kamera ensteht ein überbestimmtes Gleichungssystem, welches die Bestimmung der 3D-Koordinate eines Punktes aus den Bildkoordinaten erlaubt

Kameramodell mit Linsenverzeichnung

- Reale Kameras und Linsen verursachen eine Vielzahl von Abbildungsfehlern und genügen nicht dem Modell
- Die Hauptfehlerquellen sind:
  1. Räumliche Auflösung relativ gering, da die Auflösung der Kameras ebenfalls noch gering ist (Aktuelle DV-Kameras: 320x200, 640x480 @30 fps; 800x600, 1024x768 @15 fps; 1280x960 @7.5 fps)
  2. Die meisten (billigen) Linsen sind unsymmetrisch und erzeugen Verzerrungen
  3. Der Zusammenbau der Kamera ist nicht präzise durchführbar (Mittelpunkt des CCD-Chips liegt nicht auf der optischen Achse; Chip liegt nicht parallel zur Linse etc.)
  4. Timing-Fehler zwischen Kamera-Hardware und Grabber-Hardware

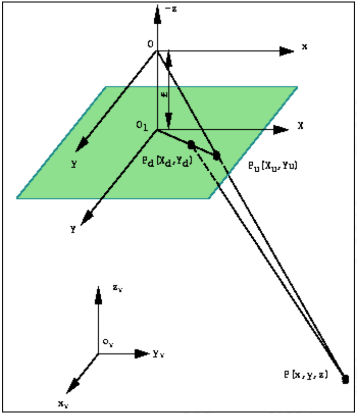
Verzeichnung

- Verzeichnung durch das Linsensystem resultiert in einer geänderten Position der Bildpixel auf der Bildebene
- Das Lochkameramodell wird dem nicht mehr gerecht
- Es wird ersetzt durch folgendes Modell:

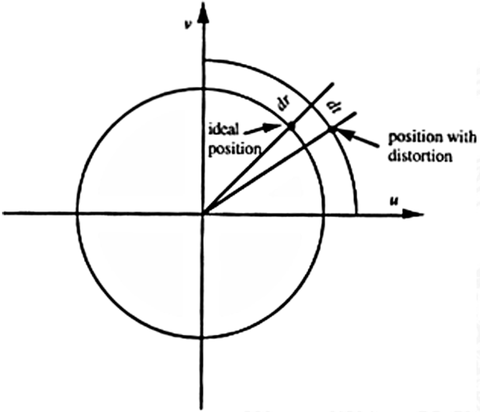
$$u' = u + D_u(u, v)$$
$$v' = v + D_v(u, v)$$

wobei u und v die nicht beobachtbaren, verzeichnisfreien Bildkoordinaten sind und u' und v' die korrespondierenden verzerrten Koordinaten

Verzeichnung (cont.)



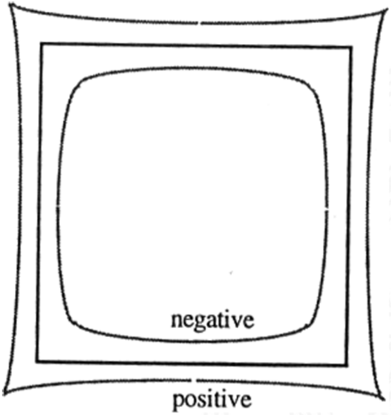
Verzeichnung (cont.)



Arten von Verzeichnungen

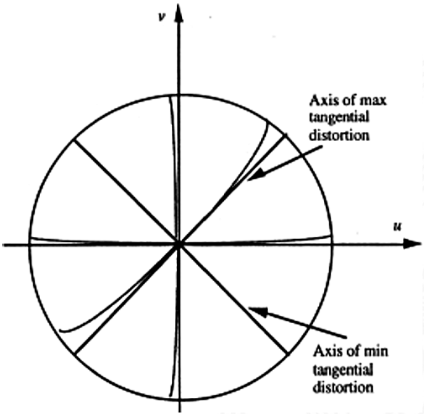
- Es gibt **zwei** Arten von Verzeichnungen:
  - *radial*
  - *tangential*
- Radiale Verzeichnung verursacht einen Versatz der idealen Position nach innen (Tonne) oder außen (Kissen)
- Ursache: fehlerhafte radiale Krümmung der Linse

Radiale Verzeichnung



gerade Linien → keine Verzeichnung

Tangentiale Verzeichnung



gerade Linien → keine Verzeichnung

Modellierung der Linsenverzeichnung

- Nach Weng et. al. (1992) unterscheidet man drei Verzeichnungen, die modelliert werden:

  1. Radiale Linsenverzeichnung (engl. *radial distortion*)
  2. Dezentrierende Verzeichnung (engl. *decentering distortion*)
  3. Verzeichnung des dünnen Prismas (engl: *thin prism distortion*)

- Die dezentrierende Verzeichnung und die Verzeichnung des dünnen Prismas sind sowohl radial als auch tangential
- Bei der dezentrierenden Verzeichnung sind die optischen Zentren der Linsen nicht kolinear

Modell: Radiale Verzeichnung

Radiale Verzeichnung

$$D_{ur} = ku(u^2 + v^2) + O[(u, v)^5]$$

$$D_{vr} = kv(u^2 + v^2) + O[(u, v)^5]$$

Modell: Dezentrierende Verzeichnung

Dezentrierende Verzeichnung

$$D_{ud} = p_1(3u^2 + v^2) + 2p_2 uv + O[(u,v)^4]$$

$$D_{vd} = 2p_1 uv + p_2(u^2 + 3v^2) + O[(u,v)^4]$$

Modell: Verzeichnung des dünnen Prismas

Verzeichnung des dünnen Prismas

$$D_{up} = s_1(u^2 + v^2) + O[(u, v)^4]$$

$$D_{vp} = s_2(u^2 + v^2) + O[(u, v)^4]$$

Gesamtmodell der Linsenverzeichnung

Wir ignorieren die Terme mit Ordnung höher als 4 und fassen die vorangegangenen Modelle zusammen:

$$D_u = ku(u^2 + v^2) + (p_1(3u^2 + v^2) + 2p_2 uv) + s_1(u^2 + v^2)$$

$$D_v = kv(u^2 + v^2) + (2p_1 uv + p_2(u^2 + 3v^2)) + s_2(u^2 + v^2)$$

Vereinfachtes Modell

Da die radiale Linsenverzeichnung der dominierende Effekt ist, kann folgendes Gleichungssystem als vereinfachtes Kameramodell verwendet werden:

Vereinfachtes Kameramodell mit Verzeichnung

$$u' = u(1 + k' r'^2)$$
$$v' = v(1 + k' r'^2)$$

mit $r'^2 = u^2 + v^2$

Radialer Verzeichnungskoeffizient

Da u und v unbekannt sind, werden diese durch die messbaren Bildkoordinaten X und Y ersetzt und es gilt

$$r'^2 = (X/s_u)^2 + (Y/s_v)^2$$

Definiert man $k \equiv k' s_v^2$, den *radialen Verzeichniskoeffizienten* (engl. *radial distortion coefficient*), dann folgt

$$\mu \equiv \frac{f_y}{f_x} = \frac{s_v}{s_u}$$

und

$$r^2 \equiv \mu^2 X^2 + Y^2$$

Modell für kleine radiale Verzeichnungen

Mit den oben genannten Modifikationen erhält man folgendes Kameramodell für kleine radiale Verzeichnungen

$$X(1+kr^2) \cong f_x \frac{r_1 x_w + r_2 y_w + r_3 z_w + t_x}{r_7 x_w + r_8 y_w + r_9 z_w + t_z},$$

$$Y(1+kr^2) \cong f_y \frac{r_4 x_w + r_5 y_w + r_6 z_w + t_y}{r_7 x_w + r_8 y_w + r_9 z_w + t_z}$$

Variation

Ein für das *least squares* Verfahren nützlicher Trick ist die Verwendung der folgenden Variation des vorigen Modells

$$\frac{X}{1+kr^2} \cong f_x \frac{r_1 x_w + r_2 y_w + r_3 z_w + t_x}{r_7 x_w + r_8 y_w + r_9 z_w + t_z},$$

$$\frac{Y}{1+kr^2} \cong f_y \frac{r_4 x_w + r_5 y_w + r_6 z_w + t_y}{r_7 x_w + r_8 y_w + r_9 z_w + t_z}$$

und gilt unter der Annahme, dass $kr^2 << 1$ ist

radial alignment constraint

Wenn neben der radialen keine weiteren Verzeichnungen auftreten, erhält man das *radial alignment constraint* (**RAC**)

$$\frac{X}{Y} = \mu^{-1} \frac{r_1 x_w + r_2 y_w + r_3 z_w + t_x}{r_4 x_w + r_5 y_w + r_6 z_w + t_y}$$

bzw:

$$X_d : Y_d = x : y$$

mit $X_d = f_x X$ und $Y_d = f_y Y$

Tsai's RAC-basierte Kamerakalibrierung

- Annahme C_x, C_y und μ sind bekannt
- Ziel ist die Ermittelung der extrinsischen Parameter R und t sowie der intrinsischen Parameter f_x, f_y und k
- Für die Kalibrierung wird eine Menge **koplanarer** Kalibrationspunkte verwendet werden
- Die Kalibrierung beinhaltet zwei Schritte
  1. Ermitteln der Rotationsmatrix R und der Komponenten t_x und t_y des Translationsvektors
  2. Schätzung der übrigen Parameter aufgrund der Ergebnisse des ersten Schrittes

Kamerakalibrierung nach Tsai: Schritt 1

1. Berechnung der Bildkoordinaten (X_i, Y_i)

Sei N die Anzahl der Bildpunkte, dann gilt für $i = 1, 2, \ldots, N$

$$X_i = X_{f,i} - C_x$$
$$Y_i = Y_{f,i} - C_y$$

wobei $X_{f,i}$ und $Y_{f,i}$ die Pixelwerte im Rechner sind

Kamerakalibrierung nach Tsai: Schritt 1 (cont.)

2. Bestimmung der Zwischenparameter $\{v_1, v_2, v_3, v_4, v_5\}$

- Da RAC unabhängig vom k und f ist, können R, t_x und t_y berechnet werden
- Wir definieren

$$\{v_1, v_2, v_3, v_4, v_5\} \equiv \{r_1 t_y^{-1}, r_2 t_y^{-1}, t_x t_y^{-1}, r_4 t_y^{-1}, r_5 t_y^{-1}\}$$

Kamerakalibrierung nach Tsai: Schritt 1 (cont.)

- Wenn man für den *i*-ten Kalibrationspunkt beide Seiten der RAC-Gleichung durch t_y teilt und den entstehenden Ausdruck umarrangiert, erhält man

$$\begin{bmatrix} x_{w,i}Y_i & y_{w,i}Y_i & Y_i & -x_{w,i}\mu X_i & -y_{w,i}\mu X_i \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} = \mu X_i$$

wobei $x_{x,i}$ und $y_{w,i}$ die x und y Koordinaten des *i*-ten Kalibrationspunktes sind

Kamerakalibrierung nach Tsai: Schritt 1 (cont.)

- Die minimale Anzahl an notwendigen **nicht kolinearen** Kalibrationspunkten ist $N = 5$
- In der Praxis sollte $N > 5$ gewählt werden
- **Bemerkung:** Falls $t_y = 0$ kann obige Gleichung auch in Abhängigkeit von t_x formuliert werden
- Erhält man $t_x = t_y = 0$, so ist der gewählte Kameraaufbau in geeigneter Form abzuändern

Kamerakalibrierung nach Tsai: Schritt 1 (cont.)

3. Berechnung von R, t_x und t_y

- Definiere $C \equiv \begin{bmatrix} v_1 & v_2 \\ v_4 & v_5 \end{bmatrix}$
- Wenn **keine** Zeile oder Spalte zu null wird, gilt:

$$t_y^2 = \frac{S_r - \sqrt{S_r^2 - 4(v_1 v_5 - v_4 v_2)^2}}{2(v_1 v_5 - v_4 v_2)} \qquad \text{mit} \qquad S_r \equiv v_1^2 + v_2^2 + v_4^2 + v_5^2$$

- Andernfalls gilt

$$t_y^2 = (v_i^2 + v_j^2)^{-1}$$

wobei v_i und v_j die Elemente aus C sind, die nicht null sind

Kamerakalibrierung nach Tsai: Schritt 1 (cont.)

- Physikalisch sollten die Vorzeichen von x und X sowie y und Y gleich sein
- Dies wird genutzt, um das Vorzeichen von t_y zu bestimmen
- Annahme: $t_y > 0$
- Berechnung von

$$
\begin{aligned}
r_1 &= v_1 t_y \\
r_2 &= v_2 t_y \\
r_4 &= v_4 t_y \\
r_5 &= v_5 t_y \\
t_x &= v_3 t_y
\end{aligned}
$$

Kamerakalibrierung nach Tsai: Schritt 1 (cont.)

- Mit einem beliebigen Kalibrationspunkt lassen sich die folgenden Koordinaten bestimmen:

$$x = r_1 x_w + r_2 y_w + t_x$$
$$y = r_4 x_w + r_5 y_w + t_y$$

- Gilt $sign(x) = sign(X)$ und $sign(y) = sign(Y)$, dann gilt die Annahme $sign(t_y) = 1$ und wir behalten r_1, r_2, r_4, r_5 und t_x
- Andernfalls setzen wir $sign(t_y) = -1$ und drehen die Vorzeichen von r_1, r_2, r_4, r_5 und t_y entsprechend um

Kamerakalibrierung nach Tsai: Schritt 1 (cont.)

- Es gibt zwei mögliche Lösungen für die Rotationsmatrix R, wenn eine 2×2-Teilmatrix bekannt ist
- Diese Lösungen sind f_x mit positivem und mit negativem Vorzeichen
- R kann wie folgt berechnet werden

$$
\begin{aligned}
r_3 &= \pm(1 - r_1^2 - r_2^2)^{1/2} \\
r_6 &= \mp sign(r_1 r_4 + r_2 r_5)(1 - r_4^2 - r_5^2)^{1/2} \\
[r_7 \ r_8 \ r_9]^T &= [r_1 \ r_2 \ r_3]^T \times [r_4 \ r_5 \ r_6]^T
\end{aligned}
$$

- Eine der beiden Lösungen führt zu einem positiven f_x in *Schritt 2*

Kamerakalibrierung nach Tsai: Schritt 1 (cont.)

Anmerkungen:

- Die sich ergebende Matrix R ist möglicherweise nicht orthonormal
- Es müssen daher noch Orthonormalisierungsschritte durchgeführt werden, die hier nicht weiter erläutert werden

Kamerakalibrierung nach Tsai: Schritt 2

Bestimmung der Parameter t_z, k, f_x **und** f_y

- Wenn R, t_x und t_y bekannt sind, können die übrigen Parameter mit folgender Gleichung für den i-ten Kalibrationspunkt bestimmt werden:

$$\begin{bmatrix} -X_i & x_i & -x_i r_i^2 \end{bmatrix} \begin{bmatrix} t_z \\ f_x \\ kf_x \end{bmatrix} = X_i w_i$$

mit

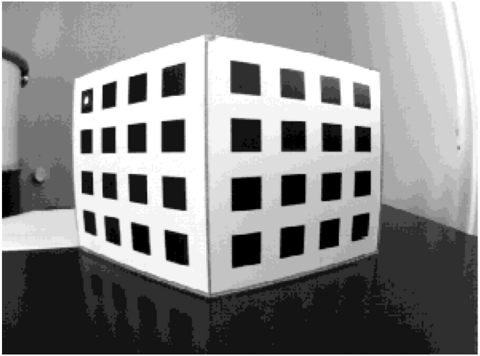
$$x_i \equiv r_1 x_{w,i} + r_2 y_{w,i} + t_x$$
$$w_i \equiv r_7 x_{w,i} + r_8 y_{w,i}$$

Kamerakalibrierung nach Tsai: Schritt 2 (cont.)

- Immer wenn mehr als drei Kalibrationspunkte benutzt werden, entsteht ein überbestimmtes Gleichungssystem
- Die Lösung mit Hilfe des *least-squares*-Verfahrens liefert k, t_z und f_x
- Mit f_x lassen sich die übrigen Parameter errechnen:

$$f_y = f_x \mu$$
$$k = (kf_x) f_x^{-1}$$

3D-Kalibrierungsaufbau



Typischer 3D-Kalibrierungsaufbau

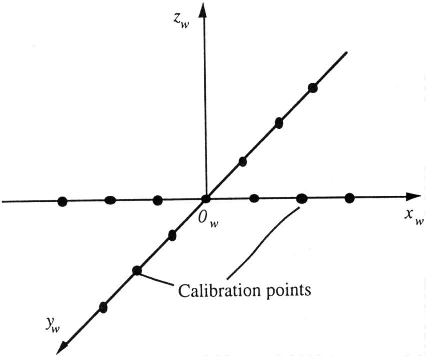
Schnelle RAC-basierte Kalibrierung

- Wenn für den erste Schritt des Tsai-Algorithmus nur Kalibrationspunkte auf der x- und y-Achse des Weltkoordinatensystems verwendet werden, vereinfacht sich die RAC-Gleichung
- Üblicherweise: Die mittlere Reihe und mittlere Spalte einer typischen Kalibrierungsplatte definieren dann die x_w- und y_w-Achse
- Beim Tsai-Algorithmus wird das *linear-least-squares*-Verfahren in Schritt eins auf fünf, in Schritt zwei auf drei Variablen angewandt
- Mit obiger Vereinfachung muss das *least-squares*-Verfahren dreimal für zwei Variablen angewandt werden

Schnelle RAC-basierte Kalibrierung (cont.)

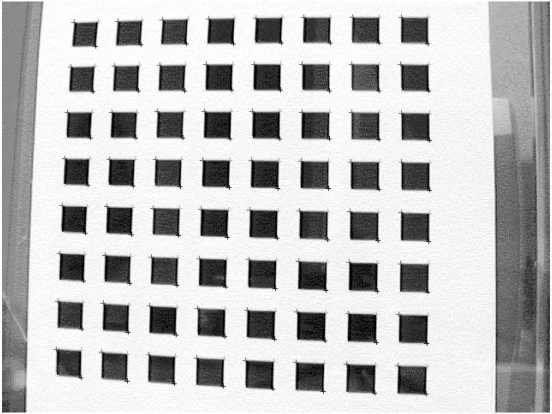
- Da es hierfür eine geschlossene Lösung gibt, reduziert sich die für die Kalibrierung notwendige Rechenzeit signifikant
- Voraussetzung für die schnelle Variante des Tsai-Algorithmus ist, das μ sowie $C_x$und C_y *a priori* bekannt sind
- Wie bei Tsai's Kalibrierung sind zwei Schritte notwendig:
  1. Verwendung von Kalibrationspunkten auf der x_w- und y_w-Achse und einer vereinfachten RAC-Gleichung, um R, t_x und t_y zu bestimmen
  2. Bestimmung der übrigen Parameter mit allen sichtbaren Kalibrationspunkten

Schnelle RAC-basierte Kalibrierung (cont.)



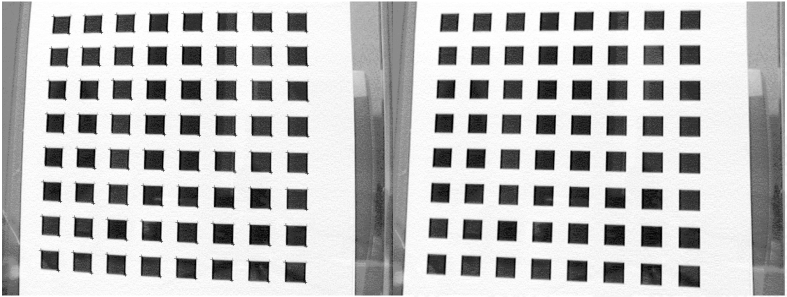
Kalibrationspunkte für die erste Phase der schnellen RAC-basierten Kalibrierung

Schnelle RAC-basierte Kalibrierung (cont.)



Typische Kalibrierungsplatte

Schnelle RAC-basierte Kalibrierung (cont.)



Mit einer kalibrierten Kamera lässt sich das Bild entzerren

Implizite Kamerakalibrierung

- Die *implizite Kamerakalibrierung* berücksichtigt alle Linsenverzeichnungen (Gesamtmodell)
- Wie bei der Kalibrierung ohne Linsenverzeichnung werden die Parameter nicht explizit bestimmt
- **Notation:**

(u^i, v^i)	Pixelkoordinaten im Bild
(x_i, y_i)	Weltkoordinaten von Punkt P auf Ebene π_i

Implizite Kamerakalibrierung (cont.)

- Projektion vom Pixel- auf das Weltkoordinatensystem auf eine Ebene π_1 ergibt sich aus dem Gesamtmodell wie folgt:

$$X_1 = \frac{\sum_{0 \leq i+j \leq 3} a_{ij}^{(1)} u_1^i v_1^j}{\sum_{0 \leq i+j \leq 3} a_{ij}^{(3)} u_1^i v_1^j}$$

$$Y_1 = \frac{\sum_{0 \leq i+j \leq 3} a_{ij}^{(2)} u_1^i v_1^j}{\sum_{0 \leq i+j \leq 3} a_{ij}^{(3)} u_1^i v_1^j}$$

wobei $a_{ij}^{(k)}$ die Transformationskoeffizienten sind

Implizite Kamerakalibrierung (cont.)

- Mit Hilfe zweier Kalibrationsebenen π_1 und π_2 können die Transformationskoeffizienten zu den beiden Ebenen bestimmt werden
- Somit ist jeder Bildpunkt auf zwei Ebenen projezierbar
- Die wirkliche Koordinate liegt auf der Verbindungsgeraden durch die beiden Punkte

Bestimmung einer Zeigerichtung: Motivation

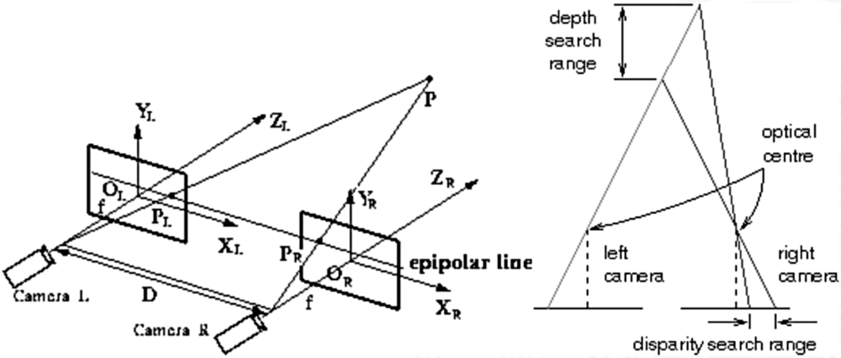
Motivation:

- Die Erkennung von Handgesten kann im Rahmen der Mensch-Maschine-Kommunikation genutzt werden
- Anwendungen im Bereich der virtuellen Realität, von Multimedia oder Roboter-Instruktion und -Teleoperation

Lösungen:

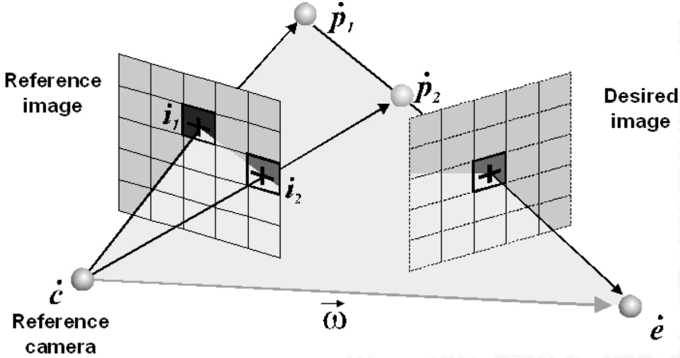
- Sensoren an der Hand (z.B. Daten-Handschuh)
- Stereo-Vision (kalibriert)
- Stereo-Vision (unkalibriert)

Bestimmung einer Zeigerichtung: Stereo-Vision



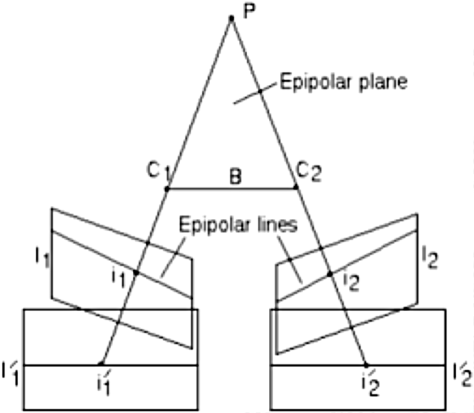
Prinzipieller Stereoaufbau mit parallelen optischen Achsen

Bestimmung einer Zeigerichtung: Epipolarlinien



Der mit einem Punkt aus Bild 1 korrespondierende Punkt kann auf der zugehörigen Epipolarlinie in Bild 2 gefunden werden

Bestimmung einer Zeigerichtung: Epipolarlinien



Bei parallelen optischen Achsen sind die Epipolarlinien horizontale Linien

Unkalibrierte Stereo-Vision

- Cipolla et. al. (1994) präsentieren eine unkalibriertes Stereo-System zur Erkennung von Zeigegesten
- **Annahme:** Lochkameramodel mit Blick auf eine Ebene
- Die Beziehung zwischen Ebenenkoordinatensystem (X, Y) und Bildkoordinatensystem (u, v) lautet:

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = T \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

wobei $T_{3\times 3}$ eine homogene Matrix mit $t_{33} = 1$ ist

Unkalibrierte Stereo-Vision (cont.)

- Um T zu bestimmen müssen mindestens vier Punkte beobachtet werden
- Man definiert die Grenzen der Arbeitsebene mit $(0,0)$,$(0,1)$,$(1,0)$ und $(1,1)$
- Für die beiden Kameras werden die Transformationen T und T' bestimmt

Bestimmung des Zeigepunktes

Notation:

l_w: Längsachse des Zeigers in der Welt

l_i: Projektion von l_w auf die Bildebene

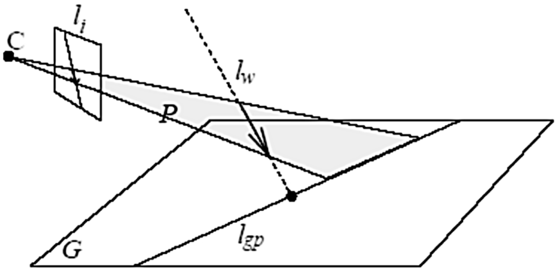
l_{gp}: Projektion von l_w auf Ebene G

Verfahren:

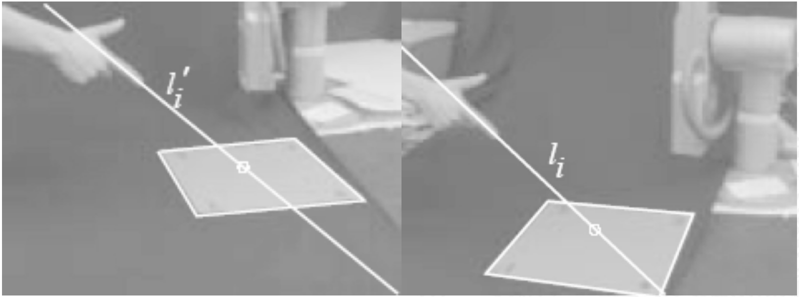
- Mit dem Bild der zweiten Kamera erhält man eine Projektion l'_{gp} deren Schnittpunkt mit l_{gp} der Zeigepunkt ist
- l_i ist das Bild von l_{gp}, d.h. $l_i = Tl_{gp}$
- Daraus folgt

$$l_{gp} = T^{-1} l_i \qquad \text{und} \qquad l'_{gp} = T'^{-1} l_i$$

Bestimmung des Zeigepunktes (cont.)



Bestimmung des Zeigepunktes (cont.)



Bestimmung des Zeigepunktes (cont.)