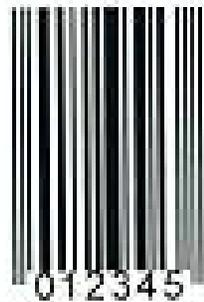


# Scannen von Barcodes - I

Eine technische Lösung zur Erkennung von bekannten, häufig auftauchenden Objekten.

Beispiel: Markierung mit UPC-E und -A Barcodes



*Figure 1—UPC-E bar code*



*Figure 2—UPC-A bar code*

## Scannen von Barcodes - II

Ein Barcode wird mit einer Infrarot-, Laser- oder sichtbaren Lichtquelle beleuchtet. Dunkle Striche absorbieren und Leerfelder reflektieren das Licht.



## Scannen von Barcodes - III

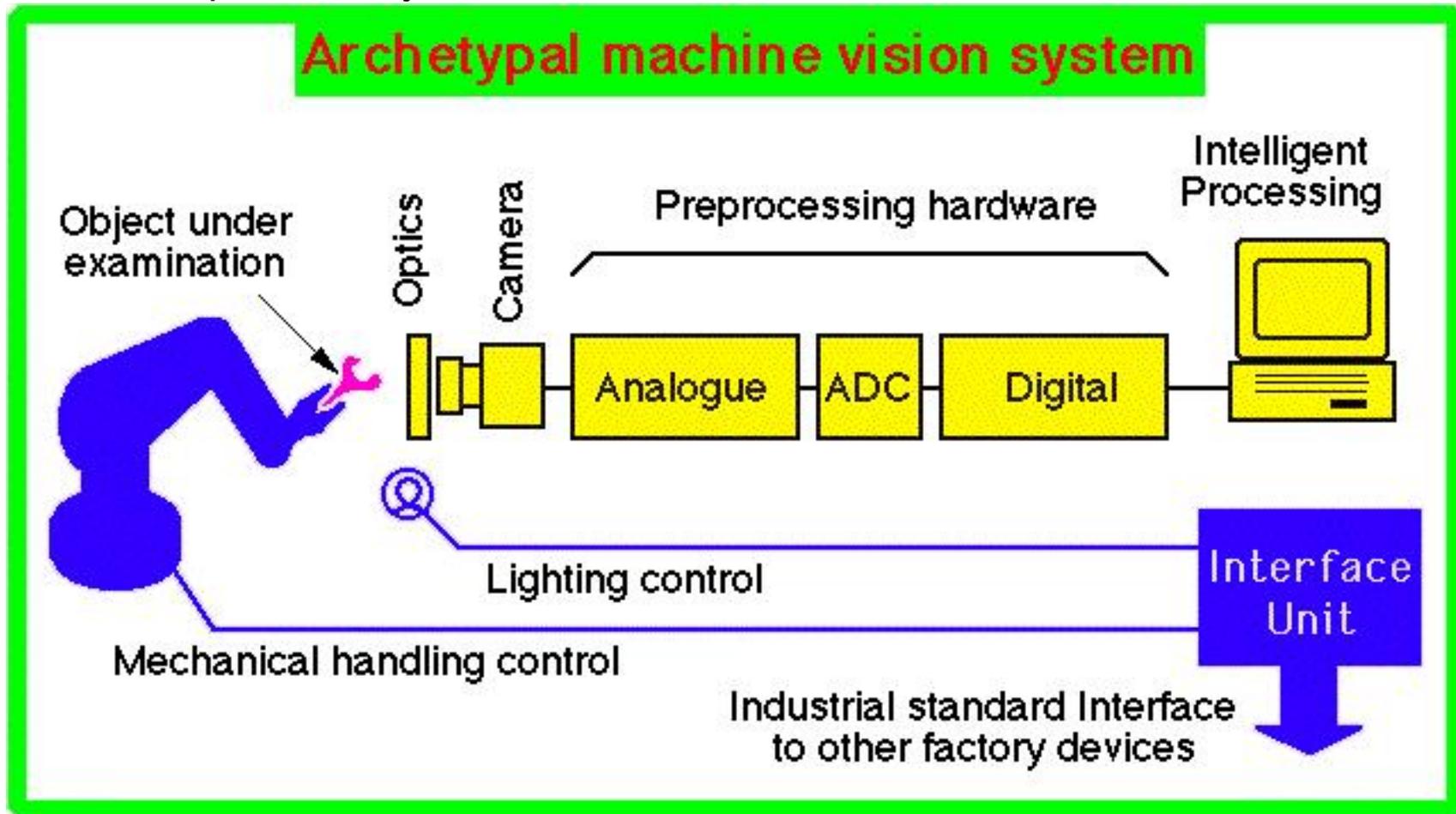
- Der Barcode-Leser wandelt die Lichtschwankungen in elektrische Impulse um. Ein Decoder ermittelt darauf den binären Code.
- **Lichtquellen:** sichtbares oder infrarotes Licht (*light-emitting diode* (LED)), Helium-Neon-Laser oder *solid-state laser diode*. Es wird konstantes Licht oder eine Folge von Lichtimpulsen ausgesandt.
- Einige Lesegeräte brauchen Kontakt mit dem Barcode, andere können aus einer Entfernung von über einem Meter lesen.
- In manchen Barcode-Lesern werden lineare CCD-Kameras (charged-couple device) für die photographische Wahrnehmung eingesetzt.

# Einsatz einer Kamera in der Robotervision - I

- Lineare Kamera
- CCD-Kamera S/W
- CCD-Farb-Kamera (1 Chip oder 3 Chips)
- Kamera + Strukturiertes Licht (Laser, Farb-Streifen, ...)
- Stereo-Kamera
- Digitale Kamera
- High-Dynamic-Range CMOS-Kamera
- “Intelligente Kamera”

# Einsatz einer Kamera in der Robotervision - II

In einem Manipulationssystem:



# Sichtsysteme in industriellen Roboteranwendungen

- Objekt-Greifen
  - Separat-liegende Werkstücke
  - Wahllos durcheinanderliegende Werkstücke: “Bin-Picking”
- Objekt-Handhabung:
  - Veredeln, Abdichten, Entgraten, Schneiden, Binden, Verpacken, ...
  - Inspektion während der Fertigung
- Montage:
  - Punkt- und Bogen-Schweißen, Schrauben, Befestigen, Aufkleben, ...
  - Stecken, Zuführen, ...

# Sichtsysteme in der kognitiven Robotik - I

- Erkennung von
  - Objekten
    - \* statisch: Typen, Suchen, Indizieren, ...
    - \* dynamisch: Bewegungen, Detektion von Fehloperationen, ...
  - Menschen
    - \* Gesicht, Gestik
    - \* Intention
    - \* Operationsvorgang (einzelnen Op., Reihenfolge, ...)
- Weltmodellierung
  - Ortung von Objekten
  - 3-D Rekonstruktion
  - Belegung der Umwelt
  - Lage/Orientierung des Roboters (relativ + absolut, d.h. im Bezug auf verschiedene Koordinatensysteme)

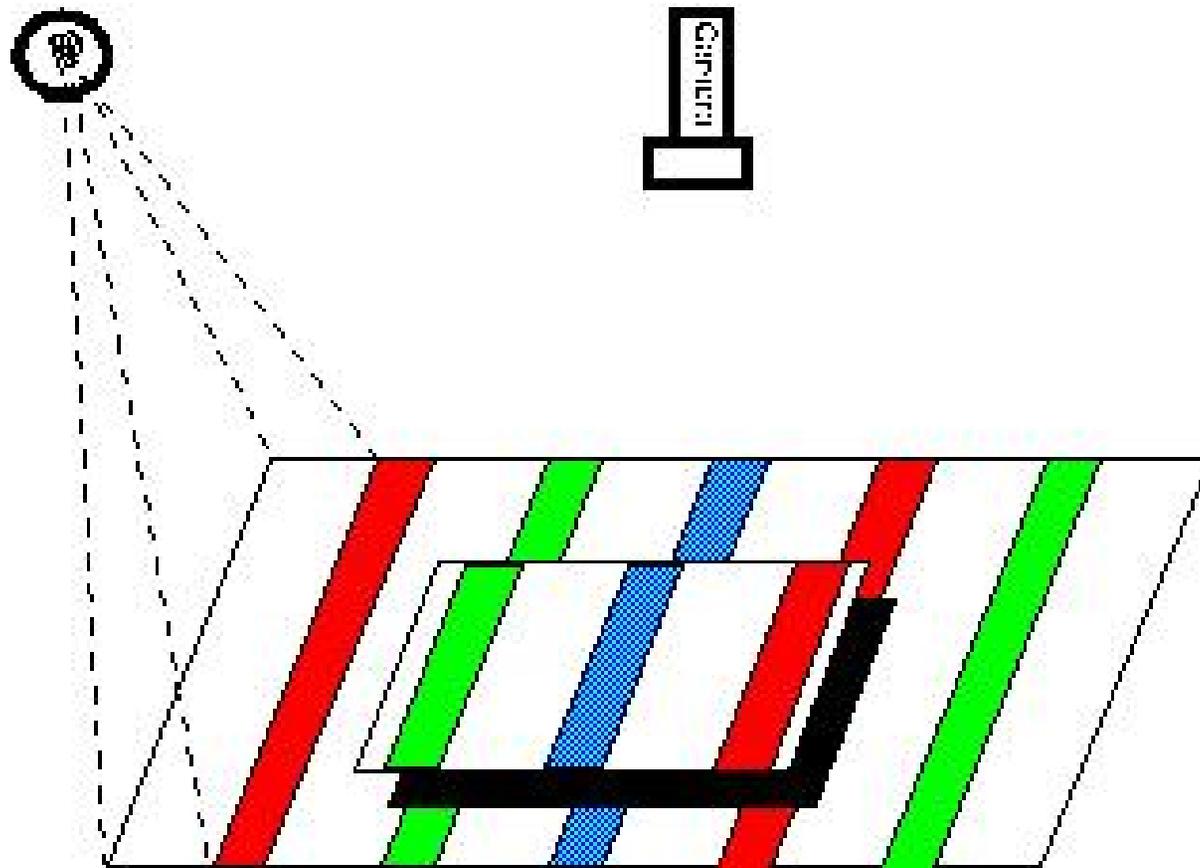
# Sichtsysteme in der kognitiven Robotik - II

- Visiongeführte Bewegung
  - Visual-Servoing
    - \* Grobe Positionierung
    - \* Verfolgung von beweglichen Objekten
    - \* Feine Positionierung
    - \* Schwingen, Jonglieren, Balancieren, ...
  - Kollisionsvermeidung
    - \* Optical-Flow-basiert
    - \* 3-D Range-Finder
  - Koordination mit anderen Robotern und/oder Menschen
    - \* Intentionsdetektion
    - \* Bewegungsschätzung

# Kamera + Farb-Streifen - I

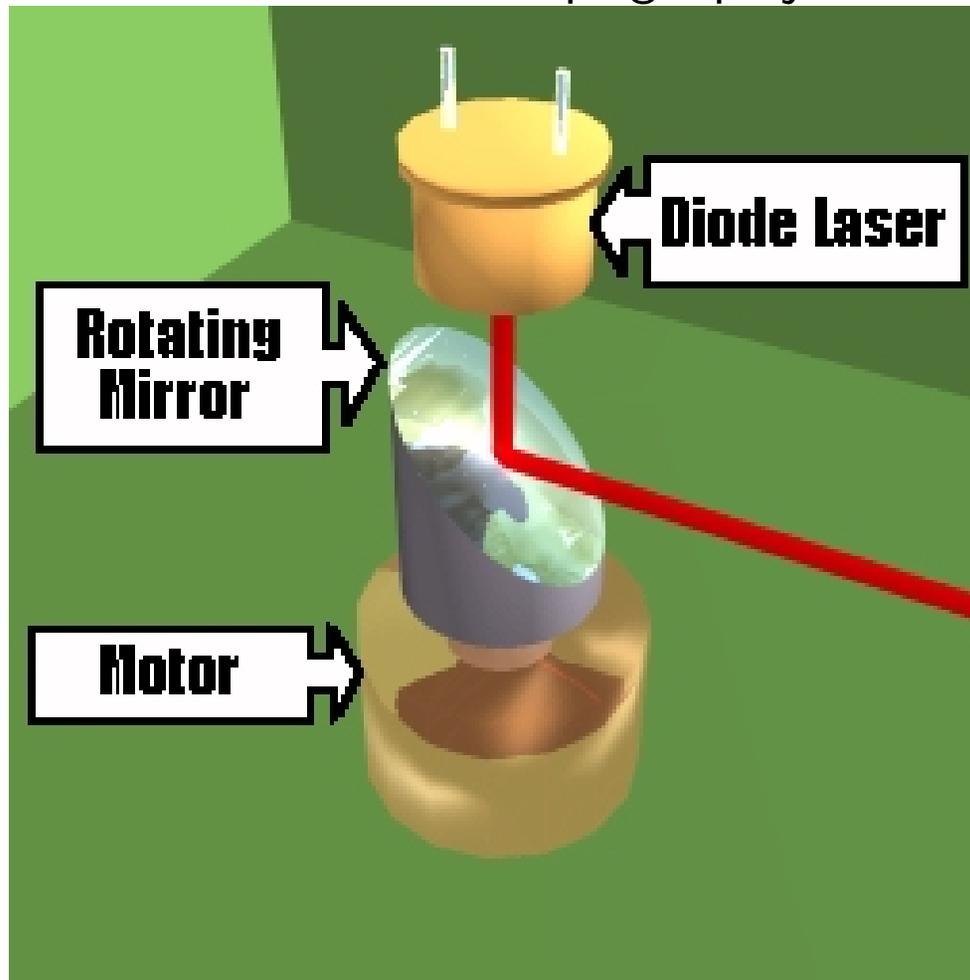
- Wenn parallele Lichtstreifen in einem bestimmten Winkel auf eine Ebene projiziert werden, erscheinen sie auch auf der Ebene parallel.
- Aber wenn die Streifen eine erhobene Fläche beleuchten, erscheinen die Streifen, von oben gesehen, seitwärts versetzt im Vergleich zu denen auf der Ebene.
- Es existiert eine trigonometrische Relation zwischen dem Versatz und der Höhe der Fläche im Vergleich zur Ebene. Diese Höhe kann z.B. mit Hilfe der Triangulation berechnet werden.

# Kamera + Farb-Streifen - II



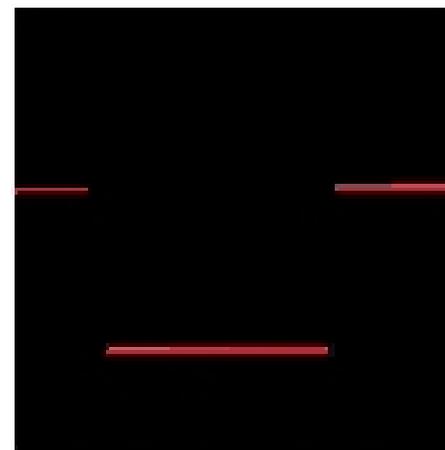
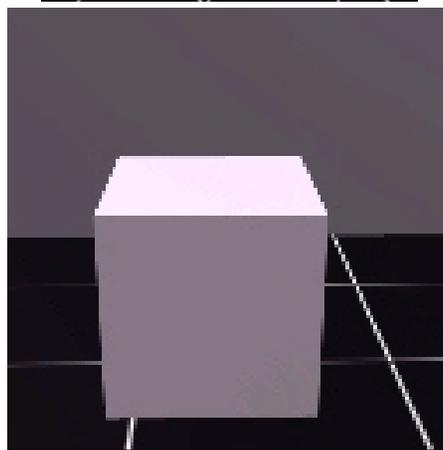
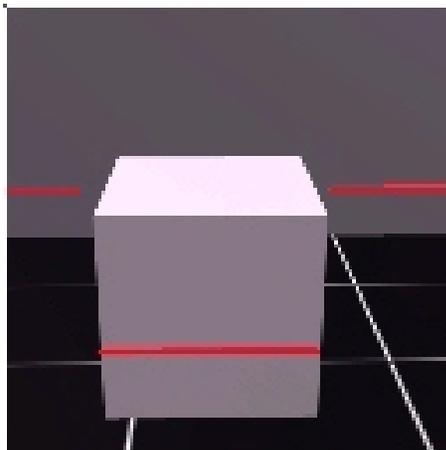
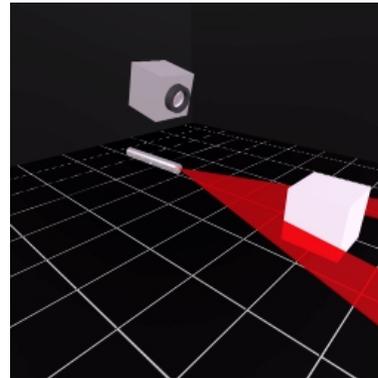
## Kamera + Laser - I

Ein Laser-Licht wird auf einen rotierenden Spiegel projiziert:



# Kamera + Laser - II

Der Laser-Strahl wird vor einer Kamera auf ein Objekt projiziert:

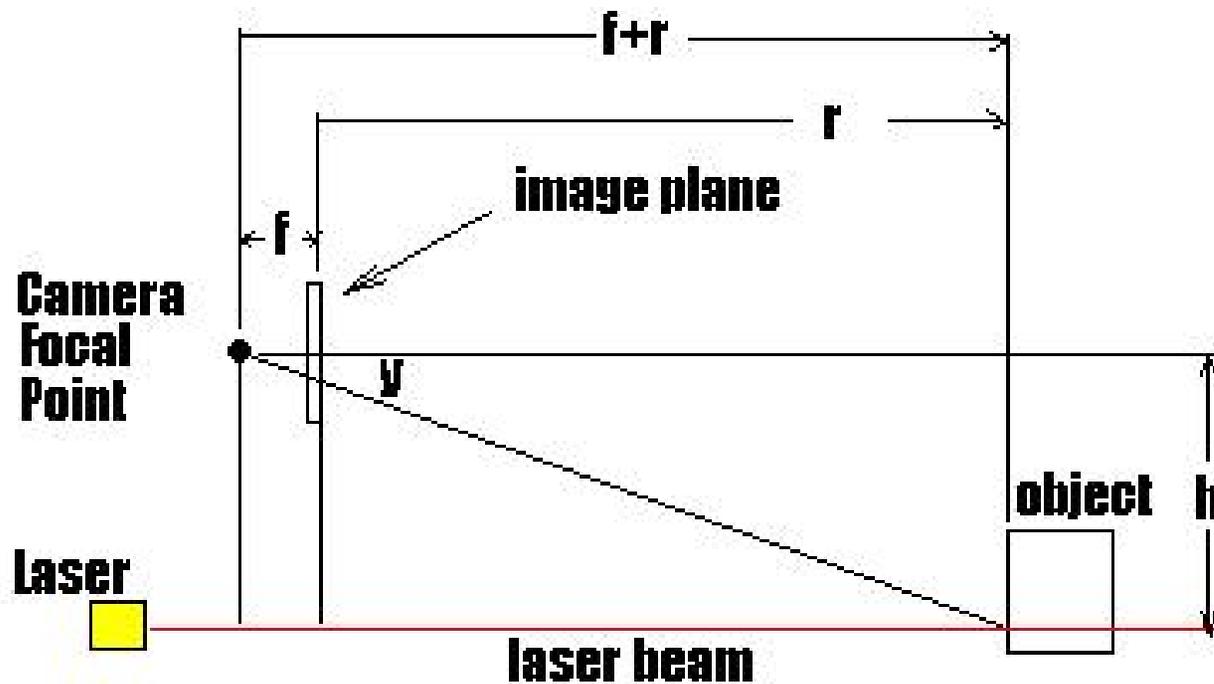


1. Kamerabild mit Laser-Strahl; 2. Kamerabild ohne Laser-Strahl; 3. Differenz-Bild.

# Kamera + Laser - III

Ermittlung der Tiefen-Information über Triangulation:

## SIMPLIFIED PROJECTION



# Binäre Bildverarbeitung

In industriellen Anwendungen gilt es zumeist herauszufinden:

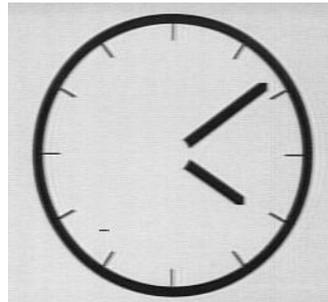
- ob ein Objekt existiert,
- ob es an der richtigen Position liegt,
- ob es die richtigen Orientierung hat,
- ob es das gewünschte Objekt ist, und
- was für ein Objekt es ist.

# Beispiele zur binären Bildverarbeitung - I

In welchem Winkel liegt die Scherer?



Welche Uhrzeit wird gezeigt?



## Beispiele zur binären Bildverarbeitung - II

Erkennung und Ortung von Werkstücken:

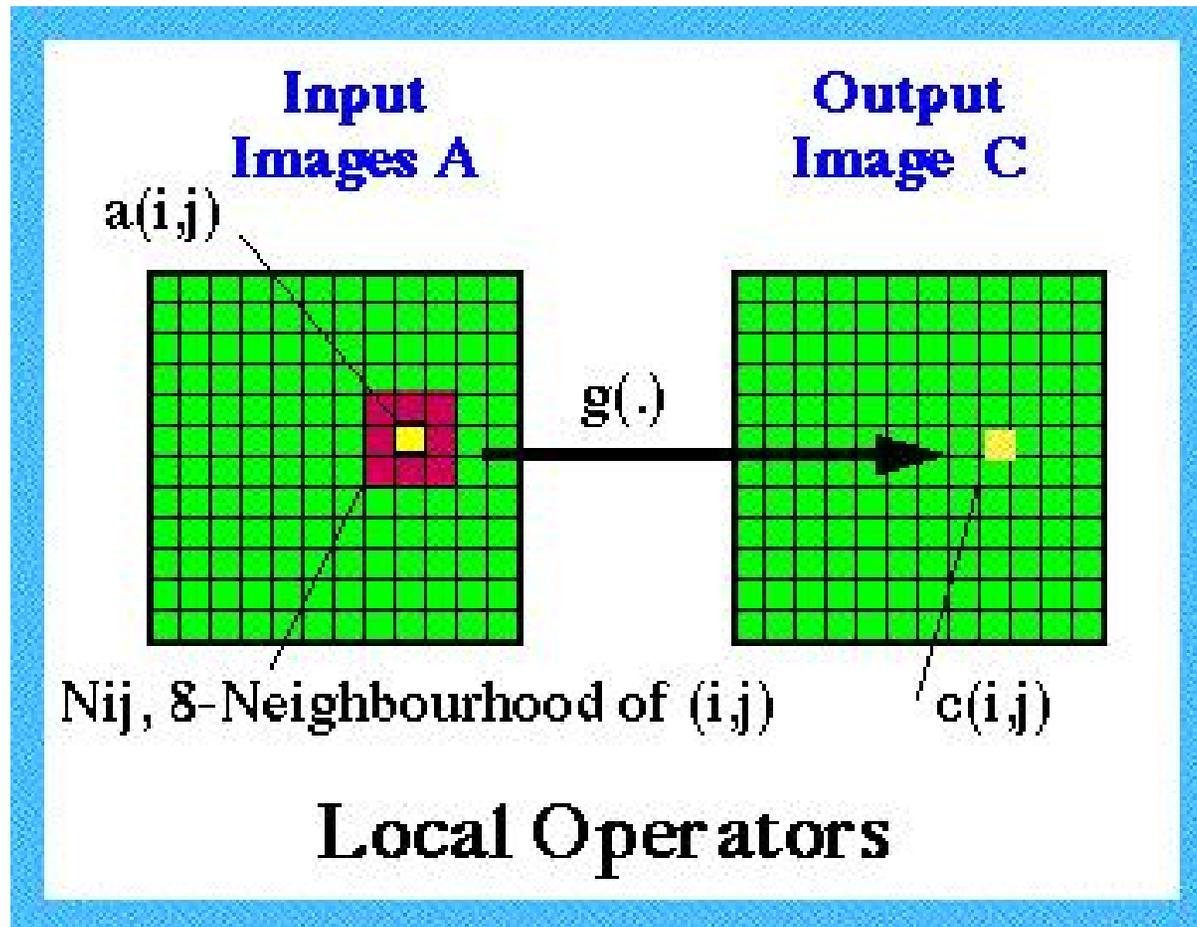


# Merkmale eines Objektes in binärer Darstellung

U. a. werden die folgenden Merkmale verwendet:

- die Massenverteilung ("Distribution of mass") - eine Funktion des Abstands links und rechts vom Schwerpunkt
- der Abstand des weitesten Punktes vom Schwerpunkt
- die Abstände der Grenzpunkte zum Schwerpunkt als eine Funktion der Polarwinkel ("polar coding" oder eine Variante "circular coding")
- Verhältnis  $Fläche / Umfang^2$
- die Anzahl der Löcher
- die Anzahl der Buchten (betrachte das Objekt als eine Insel)
- ...

# Lokale Operatoren



# Kamera-Kalibrierung - I

## Was versteht man unter Kamera-Kalibrierung?

Kamera-Kalibrierung im Kontext der drei-dimensionalen maschinellen Bildverarbeitung ist die Bestimmung der *intrinsischen* und/oder *extrinsischen* Kamera-Parameter.

- **intrinsische Parameter:** interner geometrischer Aufbau und optische Eigenschaften der Kamera
- **extrinsische Parameter:** drei-dimensionale Position und Orientierung des Koordinatensystems der Kamera relativ zu einem Welt-Koordinatensystem

## Welche Information erhält man?

Um 3D-Objekte aus zwei oder mehr Bildern zu generieren, ist es notwendig die Beziehung zwischen dem Koordinatensystem des 2D-Bildes und des 3D-Objekts zu kennen.

# Kamera-Kalibrierung - II

Die Beziehung zwischen 2D und 3D kann durch zwei Transformationen beschrieben werden:

## (1) **Perspektivische Projektion eines 3D-Punktes auf einen 2D-Bildpunkt**

Mit der Schätzung eines 3D-Objektpunktes und der dazugehörigen Fehler-Kovarianzmatrix kann die Projektion auf ein Bild vorhergesagt werden.

## (2) **Rück-Projektion eines 2D-Bildpunktes auf einen 3D-Strahl**

Sei ein 2D-Bildpunkt gegeben, dann gibt es einen Strahl im drei-dimensionalen Raum auf dem der korrespondierende 3D-Objektpunkt liegt. Gibt es zwei (oder mehr) Ansichten des 3D-Punktes, kann dessen Koordinate mit Hilfe der *Triangulation* bestimmt werden.

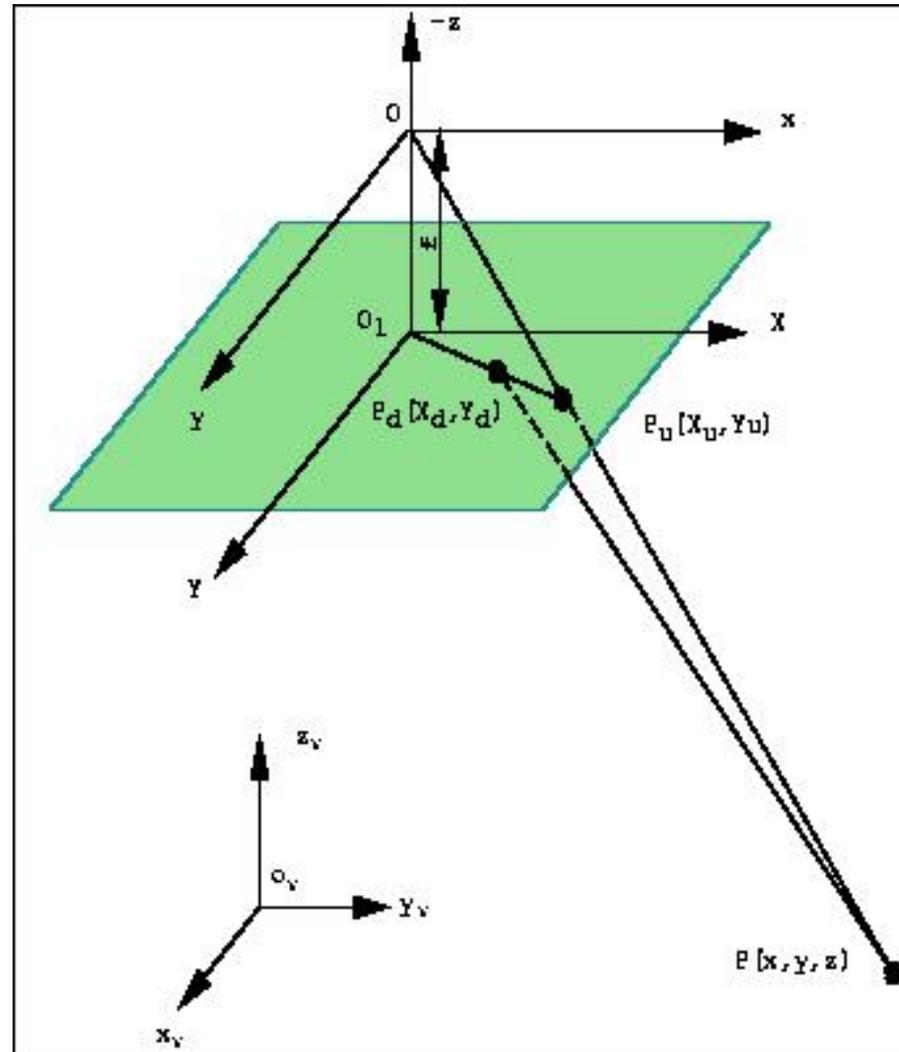
# Kamera-Kalibrierung - III

- Die erste Transformation ist nützlich um den Suchraum beim Merkmalsvergleich oder bei der Hypothesen-Verifikation in der Szenenanalyse zu verringern.
- Die zweite Transformation ist hilfreich um 3D-Information aus Merkmalen in 2D-Bildern abzuleiten.
- Die obigen Transformationen finden sich in verschiedenen Anwendungen wieder:
  - Automatische Montage
  - 3D-Messtechnik
  - Roboter-Kalibration
  - *Tracking*
  - Trajektorien-Analyse
  - automatische Fahrzeugführung.
- Die Kamera-Kalibrierung kann *on-line* oder *off-line* durchgeführt werden.

# Kamera-Kalibrierungsmethoden

- Mit Kalibrierungsplatte
  - Identifikation von Kamera-Parametern
  - Direkte Erstellung der Koordinaten-Transformation zwischen Kamera und Welt
- Selbst-Kalibrierung
- Lernen

# Das Lochkameramodell mit Linsenverzeichnung - I



# Das Lochkameramodell mit Linsenverzeichnung - II

Das vorangegangene Bild zeigt die Kamera-Geometrie mit Perspektivprojektion und radialer Linsenverzeichnung.

- $(x_w, y_w, z_w)$  ist das 3D-Weltkoordinatensystem mit dem Ursprung  $O_w$ .
- $(x, y, z)$  ist das 3D-Koordinatensystem der Kamera mit dem Ursprung  $O$ , das optische Zentrum.
- $(X, Y)$  ist das 2D-Bildkoordinatensystem mit dem Ursprung  $O_1$ .
- $f$  ist die Brennweite.

# Transformation vom Welt- zum Kamera-Koordinatensystem

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

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

wobei  $R$  eine  $3 \times 3$  Rotationsmatrix und  $T$  ein Translationsvektor ist.

Die Parameter in  $R$  und  $T$  sind die *extrinsischen* Parameter ( “*extrinsic camera parameters*”).

# Perspektivprojektion vom Kamera-KS zum Bild-KS

Punkt  $P$  wird zur idealen Bildkoordinaten  $(X_u, Y_v)$  projiziert. Die Projektion erfolgt durch Perspektivprojektion mit der Brennweite  $f$ :

$$X_u = f \frac{x}{z}$$

$$Y_v = f \frac{y}{z}$$

# Linsenverzeichnung

Der projizierte Punkt wird zu einer neuen Position  $(X_d, Y_d)$  verschoben:

$$X_d + D_x = X_u$$

$$Y_d + D_y = Y_u$$

wobei

$$D_x = X_d(\kappa_1 r^2 + \kappa_2 r^4 + \dots)$$

$$D_y = Y_d(\kappa_1 r^2 + \kappa_2 r^4 + \dots)$$

$$r = \sqrt{X_d^2 + Y_d^2}$$

# Modellierung der Linsenverzeichnung - I

Im Folgenden: Die Bildkoordinaten  $X_u$  und  $Y_u$  des Lochkammermodells werden notiert als  $u$  und  $v$ .

Die drei Haupttypen der geometrischen Linsenverzeichnung können wie folgt modelliert werden (Weng '92, pp. 965-969):

1. Radiale Verzeichnung ("radial distortion"):

$$D_{ur} = k_1 u(u^2 + v^2) + O[(u, v)^5]$$

$$D_{vr} = k_1 v(u^2 + v^2) + O[(u, v)^5]$$

# Die Modellierung von drei Typen von Linsenverzeichnung - II

2. Dezentrierende Verzeichnung (“decentering distortion”):

$$D_{ud} = p_1(3u^2 + v^2) + 2p_2uv + O[(u, v)^4]$$

$$D_{vd} = 2p_1uv + p_2(u^2 + 3v^2) + O[(u, v)^4]$$

3. Verzeichnung des dünnen Prismas (“thin prism distortion”):

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

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

# Gesamtmodell zur Linsenverzeichnung

Wir ignorieren die Terme mit Ordnung höher als 4 und fassen 1,2 und 3 zusammen:

$$D_u = k_1 u(u^2 + v^2) + (p_1(3u^2 + v^2) + 2p_2 uv) + s_1(u^2 + v^2)$$

$$D_v = k_1 v(u^2 + v^2) + (2p_1 uv + p_2(u^2 + 3v^2)) + s_2(u^2 + v^2)$$

# Pixel-Koordinaten im Rechner

$$X_f = S_x X_d + C_x$$

$$Y_f = S_y Y_d + C_y$$

wobei  $S_x$  und  $S_y$  jeweils ein Skalierungsfaktor ist.

Insgesamt werden die Kamera-Parameter in zwei Gruppen klassifiziert:

- extrinsische Parameter:  $R, T$
- intrinsische Parameter:
  - $f$ : effektive Brennweite,
  - $\kappa_i$ : Koeffizienten der Linsenverzeichnung,
  - $S_x, S_y$ : die Skalierungsfaktoren,
  - $(C_x, C_y)$ : Pixelkoordinate im Rechner zum Ursprung der Bildebene.

# Kalibration einer Kamera ohne Linsenverzeichnung - I

Wenn die expliziten Kamera-Parameter und die Brennweite kombiniert werden, dann kann ein Kamera-Modell ohne Linsenverzeichnung wie folgt beschrieben werden (Zhuang/Roth pp. 13-14):

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

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

# Kalibration einer Kamera ohne Linsenverzeichnung - II

Wenn  $a_{34}$  in obiger Gleichung zu eins gesetzt wird, dann entstehen 11 Unbekannte. Die folgende Gleichung beschreibt das Identifikationsmodell der Unbekannten:

$$\begin{bmatrix} x_w & y_w & z_w & 1 & \cdots & -X_u x_w & -X_u y_w & -X_u z_w \\ 0 & 0 & 0 & 0 & \cdots & -Y_u x_w & -Y_u y_w & -Y_u z_w \end{bmatrix} \begin{bmatrix} a_{11} \\ \cdot \\ \cdot \\ \cdot \\ a_{33} \end{bmatrix} = \begin{bmatrix} X_u \\ Y_u \end{bmatrix}$$

Es werden mindestens sechs (nicht koplanare) Kalibrationspunkte benötigt, um die Gleichungen mit der Methode der kleinsten Quadrate (*least squares*) zu lösen.

# Kalibration einer Kamera ohne Linsenverzeichnung - III

Nachteile:

- Kamerastellung (extrinsische Parameter) kann daraus nicht abgeleitet werden,
- 3D statt 2D Kalibrierungspunkte müssen zur Verfügung gestellt werden,
- nur suboptimales Modell ohne Korrektur der Linsenverzeichnung.

Vorteil:

Wenn zwei Kameras  $A$  und  $B$  kalibriert werden ( $\mathbf{a}^A, \mathbf{a}^B$  sind die Ergebnisse), dann kann die Koordinate eines beliebigen unbekanntes 3D-Punktes  $(x_w, y_w, z_w)$  berechnet werden, solange dieser in beiden Kamerabildern zu sehen ist.

# Korrektur radialer Linsenverzeichnung nach Tsai - I

In der Praxis reicht oft die Betrachtung der radialen Verzeichnung ohne die Terme höherer Ordnung. Man erhält folgende Gleichungen (*radial alignment constraint (RAC)*):

$$P_{bild} = R \cdot P_{welt} + T$$

$$\Rightarrow \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\Rightarrow \frac{X_u}{Y_u} = \frac{r_1 x_w + r_2 y_w + r_3 z_w + t_x}{r_4 x_w + r_5 y_w + r_6 z_w + t_y}$$

wobei  $r_1, \dots, r_6$  die Elemente der ersten zwei Zeile der Rotationsmatrix  $R$  sind, und  $t_x$  und  $t_y$  die ersten zwei Translationselemente sind.

# Korrektur radialer Linsenverzeichnung nach Tsai - II

Verfahren von Tsai:

- **Erster Schritt:** Bestimmung von  $r_1, \dots, r_6$  und  $t_x, t_y$ . Die Berechnung ist dabei gegenüber der radialen Linsenverzeichnung invariant.
- **Zweiter Schritt:** Bestimmung von  $t_z, f$  und  $\kappa$ .

Dadurch werden zwei Gleichungssysteme gelöst. Jede Gleichung ist gut konditioniert. Verhältnismässig gute Ergebnisse können geliefert werden.

# Das implizite Verfahren - I

- Kalibration unter Berücksichtigung aller Verzeichnungen ( $\rightarrow$  Gesamtmodell)
- extrinsische Kameraparameter werden nicht bestimmt
- Mit Hilfe zweier Kalibrationsebenen  $\pi_1$  und  $\pi_2$  können die Transformationskoeffizienten bestimmt werden.
- Jeder Bildpunkt ist auf zwei Ebenen projizierbar und die wirkliche Koordinate liegt auf der Geraden durch diese beiden Punkte.
- Notation:
  - $(u^i, v^i)$  Pixelkoordinate im Bild
  - $(x_i, y_i)$  Weltkoordinate von Punkt  $P$  auf Ebene  $\pi_i$

## Das implizite Verfahren - II

Die Rückprojektion, vom Pixel-KS  $(u_i, v_i)$  zum Welt-KS  $(X_i, Y_i)$  auf Ebene  $\pi_1$  wird wie folgt formuliert:

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

$$Y_1 = \frac{\sum_{0 \leq i+j \leq 3} a_{ij}^{(1)} u_1^i v_1^j}{\sum_{0 \leq i+j \leq 3} a_{ij}^{(3)} u_1^i v_1^j}$$

wobei  $\{a^{(k)}_{ij}\}$  die Transformationskoeffizienten sind.