Bildverstehen II

Benkel, Sebastian Bernecker, Tobias Wudenka, Martin

February 21, 2020

1 Merkmalsextraktion

1.1 Maßstabsraum-Repräsentation von Bildern

- Maßstab: n-Meter/Pixel
  - Kleinmaßstäblich(über den Bruch definiert): Objekte werden kleiner dargestellt
  - Großmaßstäblich: Objekte werden größer dargestellt
  - Auch Maß dafür wieviel Information über Details in einer Karte vorhanden sind
  - Auflösung: Fähigkeit feine Strukturen zu unterscheiden(dichte Punkte als solche erkennen)
  - Glättung reduziert Auflösung
  - Glättungsfilter $L(f, t)$: $f(p)$ = Bild, t = Maßstab(scale, quasi Kernelsize)
- Maßstabsraum-Axiome, Gaußfilter
  - Linearität (das Bild ist eine Überlagerung von Signalen)
  - Positionsinvarianz (der betrachtete Ausschnitt der Welt ist irrelevant)
  - Rotationsinvarianz (die Orientierung, in der die Welt betrachtet wird, ist irrelevant)
  - Halbgruppeneigenschaft (Reduktionen der Auflösung kombinieren sich linear)
  - Vernichtung von Maxima mit größeren t : Formalisierung der Reduktion der Auflösung
  - Der Gauß-Filter ist der einzige Filter, der alle Maßstabsraum-Axiome erfüllt
  - Gauß: Maßstab t ist gegeben durch $\sigma = t^2$, Merkregel: Objekte, die kleiner als ungefähr $\sqrt{t}$ sind, werden unterdrückt
  - Diskretes Analogon des Gauß-Filter: Kernel mit entsprechenden (abgetasteten) Werten

1.2 Unterschiedliche Typen von Linienprofilen für unterschiedliche Anwendungen

- Balkenförmiges Linienprofil
  - Bei (näherungsweise) parallele Kanten
  - Bei homogenen Grauwerte innerhalb des Objekts
- Parabolisches Linienprofil
  - Röhrenförmige Objekte die im Durchlicht aufgenommen werden (Objekte werden scharf abgebildet (optisch, Röntgen))
- Gaußsches Linienprofil
  - Röhrenförmige Objekte die im Durchlicht aufgenommen werden (Objekte werden unscharf abgebildet, liegen in streuendem Medium, (optisch, Röntgen))

1.3 Definition und Extraktion von Linienpunkten und Linienbreite

Merkmalsextraktion:

- Kanten
  - Zu extrahierende Merkmale werden häufig über Differentialinvarianten beschrieben und extrahiert
  - Kante $\Leftrightarrow$ Maximum des Gradientenbetrags in Richtung des Gradienten $\Leftrightarrow$ Nullstelle der zweiten Richtungsableitung in Richtung des Gradienten
  - lokales orthonormales Koordinatensystem (u, v), v parallel - u senkrecht zum Gradienten
- Linien
  - Besitzen charakteristisches Grauwertprofil senkrecht zum Linienverlauf
  - Balkenförmiges Linienprofil (Straßen, parallele Kanten und homogene Grauwerte innerhalb des Objekts)
  - Parabolisches Linienprofil (Blutgefäße im Röntgenbild, röhrenförmige Objekte im Durchlicht (scharf))
  - Gaußsches Linienprofil (Blutgefäß im optischen Bild, röhrenförmige Objekte im Durchlicht (unscharf))
  - Asymetrisches Linienprofil: Bsp. Balken:

$$f_b(x) = \begin{cases} 0, & x < -w \\ 1, & \|x\| \leq w \\ a, & x > w \end{cases}$$

  - $w =$ Linienbreite (halber Durchmesser der Linie), $[0, 1] \ni a =$ Differenz Intensität rechte Seite, linke Seite
  - Linienbreiten der drei Modelle auf einer Linie sind nicht vergleichbar, ggf. durch gleiche „Gesamtenergie“ oder auf Bruchteil c des Ziel-Grauwerts
  - für parabolische/Gaußsche Linien gilt: Die Position der Linie ist am lokalen Maxima bzw. Minima des Liniendurchschnitts
  - Bild muss geglättet werden um balkenförmige Linien zu erkennen, bei den anderen zur Rauschunterdrückung
  - sichere Linienselektion für Balkenförmige Linine nur möglich bei $\sigma \geq w/\sqrt{3}$ für Parabol- und Gaußlinien, $\sigma \geq 0$
  - zur Steigerung der Performance wird σ konstant gehalten und w variiert
  - Linienbreite und Mittel

Extraktion der Linienposition:

- Vektor horizontal zur Linie ist duch Eigenvektor zum betragsmäßig größeren Eigenwert der Hessematrix gegeben. Dieser und Maximum bzw. Minimum der Richtungsableitung geben Linienmittelpunkt und Ausrichtung pro Pixel/Auswertung

1.4 Glättung führt inhärent zu verzerrten Extraktionsergebnissen

Maßstabsinvarianz:

- Linienkanten und Mittelpunkt werden proportional verzerrt

1.5 Durch geeignete Modellierung können die Verzerrungen korrigiert werden

- Verzerrung der Breite und der Linienposition kann für ein Bild ermittelt und weitestgehend korrigiert werden
- Dank Maßstabsinvarianz (Fehler skaliert) reicht ermitteln der Invarianz für $\sigma = 1$

1.6 Prinzip der maßstabsraumbasierten Linien- und Kantenextraktion

1.7 Prinzip des Strukturtensors

- Viele Punktextraktoren basieren auf dem Strukturtensor
- Bild f mit Gradient $\nabla f = (f_r, f_c)$
- Strukturtensor $\boldsymbol{S}(r,c) = w(r,c) * \left(\nabla f(r,c)^T \nabla f(r,c)\right) = w(r,c) * \begin{pmatrix} f_r(r,c)^2 & f_r(r,c)f_c(r,c) \\ f_r(r,c)f_c(r,c) & f_c(r,c)^2 \end{pmatrix}$
- Als Gewichtung wird meistens der Gaußfilter verwendet $w(r,c) = \sigma(r,c;\sigma_i)$, wobei σ_i als Integrationsmaßstab bezeichnet wird
- Im Diskreten Ableitungen des Bildes über finite Differenzen, aber oft Ableitungen des Gaußfilters stattdessen verwenden: $(f_r(r,c), f_c(r,c)) := (g_r(r,c;\sigma_g), g_c(r,c;\sigma_g)) * f(r,c)$ (σ_g: lokaler Maßstab)
- Eigenwerte $\lambda_1 \geq \lambda_2 \geq 0$ und Eigenvektoren e_1, e_2 des Strukturtensors liefern Informationen über den Inhalt des Integrationsfensters:
  - ◇ $\lambda_1 \approx 0, \lambda_2 \approx 0 \Rightarrow$ Das Integrationsfenster enthält keine Kanten
  - ◇ $\lambda_1 \gg 0, \lambda_2 \approx 0 \Rightarrow$ Das Integrationsfenster enthält Kanten oder Linien; e_1 ist die Richtung senkrecht zur Kante oder Linie
  - ◇ $\lambda_1 \gg 0, \lambda_2 \gg 0 \Rightarrow$ Das Integrationsfenster enthält eine Ecke oder einen flächenhaften Punkt

1.8 Ansatz des Harris-Punktextraktors

- Basiert auf dem Strukturtensor
- Zur Unterdrückung von Kanten und zur Vermeidung der teuren Berechnung der Eigenwerte Verwendung der Antwortfunktion $R = \det(\boldsymbol{S}) - k \operatorname{tr}(\boldsymbol{S})^2$ mit $R > 0$ und R maximal (pixelgenau)
- In der Praxis Verwendung eines höheren Schwellwertes für R und subpixelgenaue Bestimmung der Position der lokalen Maxima

1.9 Ansatz des Förstner-Punktextraktors

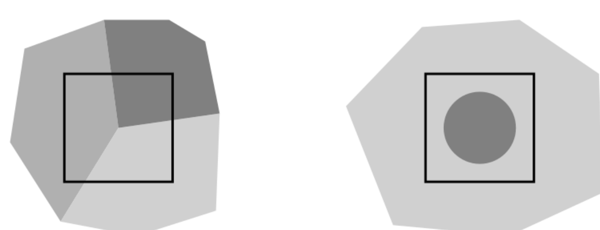
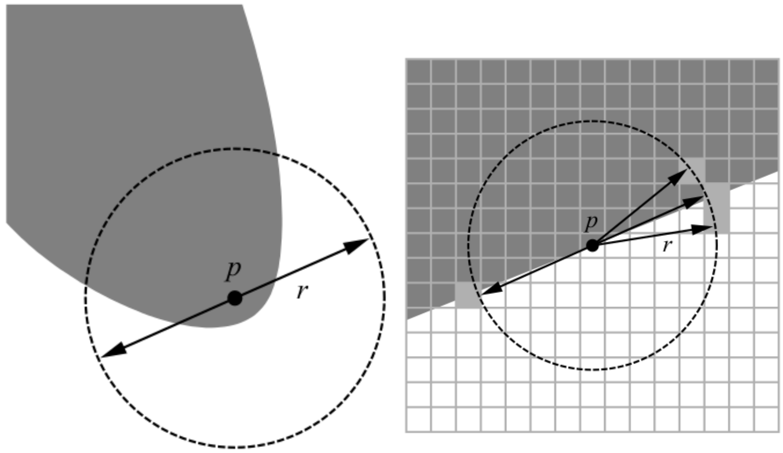


Figure 1: Kreuzungspunkt und Flächenpunkt.

- Basiert auf dem Strukturtensor
- Extrahiert 2D-Merkmale (homogene Regionen), 1D-Merkmale (Kanten, Linien) und 0D-Merkmale (Punkte)
- Unterscheidung zwischen homogenen Regionen H und inhomogenen Regionen (Kanten, Linien, Punkte): $(r, c)^T \in H \Leftrightarrow \lambda_1 + \lambda_2 = \mathrm{tr}(\boldsymbol{S}) < t_h$
- Unterscheidung zwischen Punkten und Kanten/Linien über das Verhältnis der Eigenwerte des Strukturtensors: $(r, c)^T \in P \Leftrightarrow v = \frac{\lambda_2}{\lambda_1} > t_v$
- Aus Geschwindigkeitsgründen Vermeidung der Berechnung der Eigenwerte
- Stattdessen Einführung des Formfaktors $q = \frac{4\det(\boldsymbol{S})}{\mathrm{tr}(\boldsymbol{S})^2} > t_q$
- Umrechnung des Schwellwertes t_v in einen Schwellwert für q: $t_q = \frac{4t_v}{(1+t_v)^2}$
- Weitere Klassifikation der Punktregionen in Kreuzungs- und Flächenpunkte (Figure **??**):
    - In einem idealen Kreuzungspunkt schneiden sich die Kantenrichtungen (Senkrechten der Gradienten) in einem Punkt
    - In einem idealen Flächenpunkt schneiden sich die Gradientenrichtungen in einem Punkt

Berechnung der Punklokalisierungsgüte:

- Verschieben eines Fensters über das Bild
- Berechnung des Abstandes einer Geraden zum Mittelpunkt des Fensters für jeden Punkt des Fensters unter der Annahme, es liege ein Kreuzungspunkt oder ein Flächenpunkt vor
- Summierung der gewichteten Abstände über das Fenster
- Gewichtung der Astände über
    - Stärke des Gradienten (große Gradienten sind sicherere Kanten und tragen viel Information zur Lokalisierung bei)
    - Gauß-Funktion mit Mittelwert am Mittelpunk des Fensters und Maßstab σ_p (weit vom Mittelpunkt des Fensters entfernte Punkte tragen weniger Information zur Lokalisierung bei)
- Bestimmung, welcher Punkttyp vorliegt: statistischer Test auf $s_p = D/D^\perp$ ($D =$ Punktlokalisierungsgüte)
- Klassifikation in:
    - Kreuzungspunkt, falls s_p kleiner als eine untere Signifikanzschwelle
    - Flächenpunkt, falls s_p größer als eine obere Signifikanzschwelle
    - Undefinierter Punkt
- Die Lokalisierung der Punkte erfolgt durch Bestimmung lokaler Minima in V bzw. $V^\perp$ ($V = \frac{D}{w}$ und $V^\perp = \frac{D^\perp}{w}$)



(a) Lepetit-Ansatz im Kontinuierlichen.

(b) Lepetit-Ansatz im Diskreten.

Figure 2: Lepetit-Ansatz.

1.10 Ansatz des Lepetit-Punktextraktors

- Harris- und Förstner-Punktextraktoren sind vergleichsweise aufwendig
- Geringere Genauigkeit als bei Harris- und Förstner-Punktextraktoren
- Für laufzeitkritische Anwendungen (z.B. Objekterkennung)
- Liefert stabile markante Punkte
- Ansatz (Figure **??**): Betrachtung der Grauwerte auf einem Kreis um jeden Punkt im Bild
  $\Rightarrow$ Falls zwei gegenüberliegende Punkte auf dem Kreis ähnliche Grauwerte haben wie der Mittelpunkt, ist der Punkt sicher kein markanter Punkt
  $\Rightarrow$ Falls $|f(\boldsymbol{p}) - f(\boldsymbol{p} + \boldsymbol{d})| \leq t_d$ und $|f(\boldsymbol{p}) - f(\boldsymbol{p} - \boldsymbol{d})| \leq t_d$ ist $\boldsymbol{p}$ kein markanter Punkt ($\boldsymbol{d} = (r\cos\theta, r\sin\theta)^T, \theta \in [0, \pi)$)
- Im Diskreten müssen zur Unterdrücken von Antworten an Kanten nicht nur gegenüberliegende Pixel getestet werden, sondern auch noch benachbarte Pixel
- Die Tests, ob ein Pixel ein potentieller markanter Punkt ist, sind sehr effizient und können nach dem ersten fehlgeschlagenen Test abgebrochen werden
- Nur bei tatsächlichen Kandidaten für markante Punkte wird der gesamte Kreis evaluiert
- Die Tests beim Lepetit-Punktextraktor liefern für jeden markanten Punkt mehrere benachbarte Punkte, die die Tests erfüllen
- Zur eindeutigen Lokalisierung werden die lokalen Maxima der folgenden Antwortfunktion berechnet:

$$l(\boldsymbol{p}) = \sum_{\theta \in [0,\pi)} f(\boldsymbol{p} - \boldsymbol{d}) - f(\boldsymbol{p}) + f(\boldsymbol{p} + \boldsymbol{d})$$

- Dies stellt eine Näherung des mit einem Gaußfilter geeigneter Größe geglätteten Laplace-Operators dar
- Zusätzlich zur Schwelle bei der Bestimmung der Kandidaten für markante Punkte kann eine Schwelle für die Punktantwort verwendet werden: $l(\boldsymbol{p}) \geq t_l$

- Für jeden markanten Punkt kann eine Orientierung bestimmt werden:

$$\alpha(\boldsymbol{p}) = \underset{\theta\in[0,2\pi)}{\arg\max}\ |f(\boldsymbol{p}) - f(\boldsymbol{p}+\boldsymbol{d})|$$

1.11 Skalierungsinvariante Punktextraktion

- Harris-, Förstner- und Lepetit-Punktextraktoren nur gegen relativ kleinen Skalierungsbereich invariant
- Um Invarianz gegen großen Skalierungsbereich zu erreichen Extraktion von Punkten in Bildpyramide
- Punkte, die auf verkleinerten Bildern extrahiert werden, müssen in das Koordinatensystem des Originalbildes skaliert werden
- Hierdurch werden u.U. mehrere Punkte in der Bildpyramide für denselben semantischen Punkt extrahiert
- Falls dies unerwünscht ist, können stattdessen sog. skalierungsinvariante markante Punkte extrahiert werden (hier nicht näher behandelt)

2 Klassifikation

2.1 Klassifikation basiert auf Merkmalen

- Muster werden durch Merkmale beschrieben, die in einem Merkmalsvektor zusammengefasst werden
- Ein Merkmal beschreibt eine charakteristische Eigenschaft des Musters
- Beispiele:
  - RGB-Farbwerte eines Pixels
  - Regionenmerkmale einer Region
  - Grauwertmerkmale einer Region
- Merkmale müssen so gewählt werden, dass sich die Klassen unterscheiden lassen

2.2 Prinzip der Bayes-Klassifikation

- $P(\omega_i \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega_i)P(\omega_i)}{P(\boldsymbol{x})}$
- Aus den Trainingsdaten wird die Warscheinlichkeitsverteilung basierend auf den gegebenen Merkmalen erstellt ($\approx$ Histogramm der Merkmalsvektoren + Annahme der Normalverteilung $\Rightarrow$ a-priori-Verteilung)

2.3 Typen von Klassifikatoren

- Es gibt zwei große Gruppen von Klassifikatoren:
  - Klassifikatoren, die die a-posteriori-Wahrscheinlichkeitsverteilungen bzw. über die Bayes-Regel die a-priori-Wahrscheinlichkeiten der einzelnen Klassen konstruieren
  - Klassifikatoren, die explizite Trennflächen zwischen den einzelnen Klassen konstruieren

2.3.1 Schätzen von Wahrscheinlichkeiten

- Um einen Klassifikator zu konstruieren, sind die a-priori-Wahrscheinlichkeitsverteilungen $P(\boldsymbol{x} \mid \omega_i)$ der Klassen sowie die Wahrscheinlichkeiten $P(\omega_i)$ des Auftretens der einzelnen Klassen erforderlich
- Zur Bestimmung dieser Wahrscheinlichkeiten sind Trainingsdaten, d.h. Merkmalsvektoren $\boldsymbol{x}_k$ sowie deren zugehörige Klassen ω_k notwendig
- Möglichkeiten zur Bestimmung von $P(\omega_i)$:
  - ◇ Aus der relativen Häufigkeit des Auftretens der jeweiligen Klasse in den Trainingsdaten
  - ◇ Annahme, dass alle Klassen gleich häufig auftreten, d.h. $P(\omega_i) = \frac{1}{m}$
    $\Rightarrow$ Klassifikation rein nach den a-priori-Wahrscheinlichkeiten: $P(\omega_i \mid \boldsymbol{x}) \propto P(\boldsymbol{x} \mid \omega_i) \to \max$
- Schätzung der Wahrscheinlichkeitsverteilungen wird praktikabel, wenn Form der Verteilung (z.B. Normalverteilung) bekannt ist oder angenommen werden kann
  $\Rightarrow$ Schätzung der Parameter der Verteilung aus den Trainigsdaten z.B. mit Maximum-Likelihood-Schätzung (MLE) nötig

2.3.2 Konstruktion von Trennflächen

- Bayes-Entscheidungsregel zerlegt den Merkmalsraum in die Regionen, in denen die einzelnen Klassen die maximale a-posteriori-Wahrscheinlichkeit besitzt:

$$R_j = \{\boldsymbol{x} \in \mathbb{R}^n \mid P(\omega_j \mid \boldsymbol{x}) > P(\omega_i \mid \boldsymbol{x}) \, \forall i \neq j\}$$

$\Rightarrow$ Es existieren Flächen im $\mathbb{R}^n$, die die Klassen voneinander trennen:

$$P(\omega_i \mid \boldsymbol{x}) - P(\omega_j \mid \boldsymbol{x}) = 0 \, \forall i \neq j$$

2.4 Neuheitserkennung bei der Klassifikation

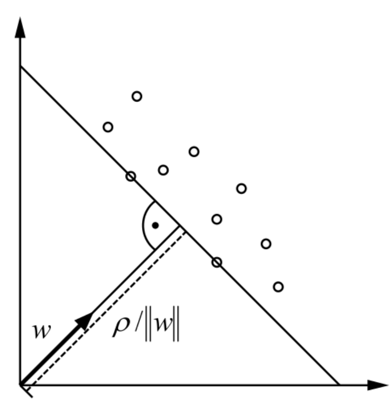


Figure 3: Lepetit-Ansatz im Diskreten.

- Gaußsche Mischmodelle (GMM):
  - ◇ Neuheitserkennung (Daten, die nicht trainiert wurden) über Rückweisung mit k-sigma-Wahrcheinlichkeit:

$$P_{k\sigma}(\boldsymbol{x}) = \frac{\max\limits_{i=1,\ldots,m} P(\omega_i)P(k)}{\max\limits_{i=1,\ldots,m} P(\omega_i)}$$

- ◇ Wahrscheinlichkeit, dass ein Merkmalsvektor außerhalb eines k-sigma-Ellipsoids um den Mittelwert liegt (t als unterer Schwellwert für die Rückweisung: $P_{k\sigma} \geq t$)

- Multilayer Perceptron (MLP):
  - ◇ Aufgrund der Form der Aktivierungsfunktionen sind Perzeptrons nicht zur Neuheitserkennung geeignet (Softmax-Funktion führt dazu, dass die Aktivierungen in einem Sektor des Merkmalsraums (bis unendlich) den Wert 1 haben)
  - ◇ Für Neuheitserkennung Einführung einer separaten Klasse für Merkmalsvektoren, die zu keiner trainierten Klasse gehören
  - ◇ Für diese Rückweisungsklasse können während des Trainings automatisch Trainingsdaten erzeugt werden:
    - ▷ Berechnung des kleinsten umschließenden Hyperquaders der jeweiligen Klasse
    - ▷ Berechnung eriner Quaderschale einer gewissen Dicke und eines gewissen Abstandes zum kleinsten umschließenden Hyperquaders jeder Klasse
    - ▷ Erzeugung von Merkmalsvektoren innerhalb der Quaderschale in den Bereichen, die sich nicht mit den Hyperquadern der anderen Klassen überlappen
  - ◇ Die Neuheitserkennung für CNNs ist ein derzeit noch ungelöstes Problem

- Support Vector Machine (SVM):
  - ◇ Zur Neuheitserkennung Modifikation der SVM
  - ◇ Transformation der Merkmale in den höherdimensionalen Raum
  - ◇ Bestimmung einer Hyperebene (Figure **??**) im höherdimensionalen Raum, die maximalen Abstand vom Ursprung besitzt und den Ursprung von den Trainingsdaten x_i trennt
  - ◇ Parametrisiereung der Hyperebene über $(\boldsymbol{w}, \rho)$
  - ◇ Beachte: Es gibt keine Klassenlabel y_i
  - ◇ Entscheidungsfunktion:

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) - \rho\right)$$

  - ◇ $f(\boldsymbol{x}) = -1 \Rightarrow \boldsymbol{x}$ ist neu
  - ◇ Mit den Gaußschen radialen Basisfunktion kann man jeden Trainingsdatensatz vom Ursprung trennen
  - ◇ Um eventuelle Fehler (Ausreißer) in den Trainingsdaten handhaben zu können, wird auch hier ein Parameter ν eingeführt:
    - ▷ ν ist eine obere Schranke für den Anteil der Ausreißer in den Trainingsdaten
    - ▷ ν ist eine untere Schranke für den Anteil der Stützvektoren unter den Trainingsdaten

2.5 Wichtiges Kriterium für die Klassifikation: Ausreichend viele Trainingsdaten

- Wenn eine große Menge an Trainingsdaten verwendet wird, zeichnet sich keines der Verfahren bei gleichen Eingabedaten gegenüber den anderen Verfahren duch eine signifikant höhere Erkennungsrate (niedrigere Fehlerwahrscheinlichkeit) aus

- In jedem Fall ist eine große Menge an Trainingsdaten die Voraussetzung für eine hohe Erkennungsrate

3 Beschriftungserkennung

- **O**ptical **C**haracter **R**ecognition
  1. Segmentierung der Zeichen
  2. Klassifikation der Zeichen ("Lesen der Zeichen")

  ⇒ OCR ist die Klassifikation der segmentierten Zeichen

3.1 Segmentierungsverfahren bei der OCR zur Auftrennung von verschmolzenen Zeichen

- Schwellwertoperationen: global, dynamisch, automatisch
- Beleuchtungskorrektur
- Regionen-, Grauwertmorphologie
- Alles was situationsspezifisch hilft...
- Industrielle Zeichensätze haben oft konstante Breite → Segmentierung in Quadrate gleicher Breite (konstante Auftrennung)
- Alternativ: Vorgabe der ungefähren Breite und der möglichen Breitenänderung → Suchbereich für Trennlinie → Trennlinie an vertikaler Linie mit den geringsten Punkten und minimalem Abstand zur erwarteten Breite (dynamische Auftrennung)

3.2 Klassifikation bei der OCR basiert auf Regionen- und Grauwertmerkmalen

- Merkmale:
  - ◇ Anisometrie
  - ◇ Verhältnis Breite/Höhe des umschließenden Rechtecks
  - ◇ Anzahl Löcher
  - ◇ Robust kontrastnormierte Grauwerte innerhalb des umschließenden Rechtecks skaliert auf Standardgröße
- Probleme
  - ◇ Ähnlich Zeichen: Cc ,. :;
  - ◇ In der Industrie meist vermieden durch spezielle Zeichensätze

4 Farbbildverarbeitung

4.1 Farbe: Lichtquelle → Reflexion, Transmission (Objekt) → Wahrnehmung (Auge) → Interpretation (Gehirn)

- sichtbares Lichtspektrum: 380 nm – 780 nm
- Röntgen- und Gammastrahlung < Ultraviolettstrahlung < Sichtbares Licht < Infrarotstrahlung < Mikro- und Radiowellen
- Farbspektrum wird am Monitor oder im Druck nicht korrekt widergegeben

- Weißes Licht besteht aus allen sichtbaren Wellenlängen, einzelne Wellenlängen dargestellt als spektrale Leistungsdichte
- Objekte interagieren mit Licht:
  - Transmission (Fensterglas)
  - Absorption (Schwarz)
  - Streuung (Milchglas)
  - Fluoreszenz (umwandeln in andere Wellenlänge: Remission als Fluoreszenz (schnelle Emission) oder Phosphoreszenz (verzögerte Emission))
- (ideale) Schwarze Körper absorbieren komplett und geben nur eine zur Wärme proportionales Leistungsspektrum ab
- CIE hat feste Spektren für Lichtquellen definiert, Normallichtarten typ D liegen nah am Spektrum Schwarzer Körper
- Menschliche Wahrnehmung
  - Stäbchen (rods) $\rightarrow$ Nachtsehen (skotopisches Sehen, Hell-Dunkel) $\approx$ 92 Millionen
  - Zapfen (cones) $\rightarrow$ Tagessehen (photopisches Sehen, Farbwahrnehmung) $\approx$ 4.6 Millionen
  - 'bunte Tannenzapfen seh ich nur am Tag'
  - Stäbchen sind $\approx$ Faktor 100 Lichtsensitiver (Dunkelsicht)
  - L-Zäpfchen maximal effektiv im Rotbereich 566nm (häufigste)
  - M-Zäpfchen maximal effektiv im Grünbereich, 543nm
  - S-Zäpfchen maximal effektiv im Blaubereich 440nm (seltenste)
  - **Metamerie**
    - Eigenschaft spektral unterschiedlicher Farbreize, die gleiche Farbempfindung auszulösen$_{\text{Wikipedia}}$
    - (um Farbreitze zu erzeugen reicht es dem Auge Kompositionen aus R G B zu übermitteln)
    - Rot, Grün und Blau (Monitore)
    - Cyan, Magenta, Gelb (Drucker)
    - Farbe eines Objekts kann je nach Beleuchtung variieren
  - Übertragung ans Gehirn via Helligkeitskanal
  - via Gegenfarbkanäle Rot(+)-Grün(-)-Kanal, Gelb(+)-Blau(-)-Kanal

4.2 Farbe läßt sich durch drei Farbwerte in einem Farbenraum darstellen

- **Beschreibungen basierend auf Farbmischung**
  - **rgb**
    - Industriestandard bei Digitalkameras, zur direkten Wiedergabe Kalibirierung der Primärfarben des Bildschirms notwendig
    - kann eine Gammakorrektur enthalten
      - $v_{video}^{\gamma} = v_{scene}$
      - bzw.: $v_{video} = v_{scene}^{1/\gamma}$
      - NTSC: $\gamma = 2.2$
      - PAL: $\gamma = 2.8$

- ▷ additive Farbmischung
  - (Red , Green, Blue) additive Farbmischung
  - (0, 0, 0) = Schwarz
- ▷ subtraktive Farbmischung
  - (Cyan , Magenta, Gelb) subtraktive Farbmischung
  - (0 , 0, 0) = Weiß
- ▷ Farbraum ist ein Würfel
- ▷ RGB-Farbenraumdarstellung ist geräteabhängig
- ▷ Farbgamut (Darstellbarer Bereich) im Auge übersteigt den technischer Geräte

- **Beschreibungen basierend auf Farbordnung**
  - ◇ Signal aufgeteilt in:
    - ▷ Kanal für Helligkeit (Farbwürfel zu Quader verzogen, so das Leuchtdichte = z-Position)
    - ▷ Farbkanäle (x-, z-Position innerhalb des Farb-Quaders)
  - ◇ **YIQ**
    - ▷ NTSC-Farbvideonorm
    - ▷ Y = Leuchtdichte (luminance)([0-Positiv])
    - ▷ I,Q = Farbigkeit (chrominance)([Negativ/Positiv])
  - ◇ **YUV**
    - ▷ Y = Leuchtdichte(luminance)([0-Positiv])
    - ▷ U, V = Farbigkeit (chrominance)([Negativ/Positiv])
- **Farbenräume basierend auf Farbordnung**
- Intuitive Beschreibung der Farbe
  - ◇ basierend auf Helligkeit (lightness), Buntton (hue) und Sättigung (saturation) oder Buntheit (chroma)
  - ◇ Vorteile
    - ▷ Buntton und Sättigung (in HLS) sind semantisch sinnvolle Attribute
  - ◇ Nachteile
    - ▷ Helligkeit ist kein semantisch sinnvolles Attribut(reines Blau gefühlt dunkler als reines Grün)
    - ▷ Darstellung ist Geräteabhängig
    - ▷ Farbunterschide sind gefühlt nicht gleichförmig
  - ◇ **HSI**
    - ▷ Intensität I = Diagonale des RGB-Farbraums (alle unbunten Farben)
    - ▷ Polarkoordinatensystem orthogonal zur Intensität
      - Winkel = Buntton (Konvention $0°$ = Rot) (durch Quaderform nicht einheitlich)
      - Abstand = Sättingung (durch Quaderform nicht einheitlich)
      - eine direkte Umrechnung aus RGB-Raum → Farbenraumdarstellung Geräteabhängig
  - ◇ **HLS**
    - ▷ umgerechnet aus HSI
    - ▷ Buntton (Hue) ist einheitlich, Diamantform

        - ▷ Helligkeit (Lightness)
        - ▷ Sättigung (Saturation) (echte Sättigung, Buntheit/Helligkeit ist einheitlich)
        - ▷ Standard in Grafikprogrammen
    - ◇ **HSV**
        - ▷ Buntton (Hue) (reine Farbe hat Helligkeit 1)
        - ▷ Sättigung (Saturation)
        - ▷ Value (Buntheit (Kegel) oder Sättigung (Zylinder))

- **Beschreibungen basierend auf Farbabgleich**
    - ◇ Farbabgleich (color matching) = experimentelles Mischen von 3 Primärvalenzen (Farbiges Licht) um gegebene Farbe zu erzeugen
    - ◇ bestimmte Testfarben nicht durch die drei Primärvalenzen dargestellt werden
    - ◇ Primärvalenzen mit Menschen Ergaben Farbenräume
    - ◇ 'Farbleistung' skaliert und addition und subtraktion von Farben ist möglich)
    - ◇ CIE definiert Farbraum für Normalbeobachter mit 2° Sichtfeld und 10° ohne innere 2° Sichtfeld
    - ◇ repräsentieren die durchschnittliche menschliche Farbwahrnehmung
    - ◇ Die Farbwahrnehmung einer echten Person kann sich erheblich vom Normalbeobachter unterscheiden (Biologisch & Altersbedinkt)

- **CIE-XYZ-Farbwerte**
    - ◇ es gilt X, Y, Z (Farbvalenzen) werden normiert zu x, y, z (Farbwertanteile)
    - ◇ es gilt: $x + y + z = 1$, daher reichen $(x, y) + Y$ zur rücktransformation
    - ◇ (x,y) ergibt eine Hufeisenförmige Visualisierung der Menschen wahrnehmbaren Farbarten
    - ◇ wird oft zur beschreibung der Primärfarben von Ausgabegeräten genutzt
    - ◇ sind Primärfarben des Ausgabegerätes als CIE-XYZ-Farbvalenzen bekannt, kann der jeweilige RGB Wert errechnet werden
    - ◇ erlaubt den Vergleich zweier Farbdarstellungen auf unterschiedlichen Geräten
    - ◇ Farbraum ist nicht gleichförmig, Regionen unterschiedlicher Größe können für Menschen einfarbig aussehen, gerade im oberen Bereich

- **CIELUV**
    - ◇ Farbraum ist näherungsweise gelichförmig (durch projektive Transformation auf wahrgenommene Veränderung angepasst)
    - ◇ ergibt UCS (uniform chromaticity scale) diagram
    - ◇ ist auf CIE-Normlichtart D65 ausgelegt, so das es nur in Kontrollierter Beleuchtung eingesetzt werden sollte

- **CIELCh$_{\mathbf{uv}}$**
    - ◇ CIELUV in Polarkoordinaten

- **CIELAB**
    - ◇ wie CIELUV mit anderer projektiver Transformation

- **CIELCh$_{\mathbf{ab}}$**
    - ◇ CIELAB in Polarkoordinaten

4.3 Prinzip des Farbmanagements

- Falls eine Transformation der RGB-Rarbwerte der Kamera in CIE-XYZ-Farbwerte (mit möglichst kleinem Fehler) bestimmt worden ist, ist die Kamera farbkalibriert ($\Rightarrow$ Kamera unterliegt somit einem Farbmanagement)
- Das Farbmanagement dient dazu, die menschliche Wahrnehmung möglichst gut nachzubilden
- Je nach spektralem Empfindlichkeitsgrad der Kamerasensoren können bestimmte Farben mehr oder weniger korrekt beschrieben werden
- Die unterschiedliche Bahandlung von Metamerie bei Mensch und Kamera kann durch Farbmanagement nicht nachträglich korrigiert werden
- Zur Bestimmung der Transformation von RGB zu CIE-XYZ wird eine Farbkalibriertafel (besitzt eine bestimmte Anzahl von Feldern, für die die CIE-XYZ-Farbwerte (oder CIE-Yxy-Farbwerte) bekannt sind) verwendet
- Vorab muss radiometrische Kalibrierung der Kamera durchgeführt werden, um eventuelle Nichtlinearitäten der RGB-Farbwerte zu entfernen
- Die Farbkalibrierung erfolgt durch Minimierung der Fehler der transformierten linearen RGB-Farbwerte und der XYZ-Farbwerte der Rarbkalibriertafel
- Transformation oft als linear angenommen, ansonsten quadratische Transformation

4.4 Prinzip der chromatischen Adaption

- kalibrierte Farbwerte sind nur für die Beleuchtung gültig in der sie kalibriert wurden
- Veränderung des Spektrums abhängig von
  - Tageszeit, Standort, Bewölkung
- Auswirkung:
  - Farbwerte ändern sich
  - für Menschen erscheinen unbunte Farben schnell bunt
- für hohe Genauigkeit muss eine erneute Kalibrierung durchgeführt werden
- für geringe Genauigkeit reicht eine chromatic adaptation transformation (CAT)
  - Hauptziel ist die Wiederherstellung der Unbuntheit der unbunten Farben, daher der Beiname **Weißabgleich**
  - Annahme: Unbunte Farben führen zu identischen Aktivierungen der Zapfen ($L_a = M_a = S_a$)
  - Annahme:Empfindlichkeit der Zapfen(Farbsicht) passt sich der Beleuchtung linear an
  - wird typischerweise auf CIEXYZ Farbraum durchgeführt, kann bei bekannten Transformationsmatrizen aus anderen Farbräumen transformiert werden und durch Inversion der Matrix wieder zurück
  - nichtlineare Kodierung der Farbwerte muss vorher entfent werden(Gammakorrektur etc.)
  - Druchführung:
    - ggf. inverse Gammakorrektur
    - Farbwerte aus einem geräteunabhängigen Farbraum transformieren

        - ▷ Vehältniss der Echten und transformierten CIEXYZ-Farbwerte berechnen
        - ▷ Bild transformieren
    - ◇ bei unbekanntem Beleuchtungsspektrum(Normalfall) kann die chromatische Adaption basierend auf einer weißen Region im Bild durchgeführt werden
    - ◇ Automatischer Weißabgleich wie in Kameras kann durch einige Approximationen algorythmisch automatisiert werden:
        - ▷ Die Welt ist im Mittel grau: die Mittelung aller Farben im Bild ergibt ein Grau mit 18% Helligkeit
        - ▷ Die hellste Farbe im Bild ist weiß
        - ▷ Analyse des Farbgamuts des aufgenommenen Bildes und statistischer Vergleich mit Farbgamutdaten aus einer Datenbank

4.5 Prinzip des Demosaicking

- CCD- und CMOS-Sensoren können verschiedene Farben (also unterschiedliche Wellenlängen) nicht unterscheiden
- Um Farbbilder aufzunehmen, existieren zwei verschiedene Technologien:
    - ◇ Drei-Chip-Kameras
        - ▷ verwenden ein Prisma als Strahlenteiler, um das einfallende Licht auf drei Sensoren umzulenken
        - ▷ Vorteil: Das Bild hat die volle Auflösung in jeder Farbe
        - ▷ Nachteil: Deutlich höherer Preis
    - ◇ Ein-Chip-Kameras
        - ▷ besitzen nur einen Sensor mit derselben Auflösung wie ein Schwarzweiß-Sensor
        - ▷ Hier werden die Filter für die drei Grundfarben in einer Mosaikanordnung vor dem Sensor angebracht
        - ▷ Nachteil: Die Auflösung der Farbkanäle ist unterschiedlich und geringer als die Auflösung eines Schwarzweiß-Sensors
          $\Rightarrow$ Die Farbinformation muss interpoliert werden (sog. Demosaicking)

4.6 Methoden zur Segmentierung von Farb- und Mehrkanalbildern

-

4.7 Prinzip der Kanten-, Linien- und Punktextraktion über den metrischen Tensor des Farb- oder Mehrkanalbildes

-

5 Hand-Auge-Kalibrierung

5.1 Koordinatensysteme und –transformationen

- beschrieben durch **Ursprung** und n **orthogonale Achsen** (Einheitsvektoren)
- **rechts**- und linkshändige
- Transformation von einem zum anderen: **starre Abbildung im n-dimensionalen Raum**

- Ursprung von einem KS ist Punkt, der in anderem dargestellt werden kann (Translationsvektor)
- Rotationsmatrix: Achsen von einem KS in anderem ausgedrückt:
- $x_1^{\{0\}} = \boldsymbol{R}_1^0 x_1^{\{1\}} \Rightarrow \boldsymbol{R}_1^0 = \begin{bmatrix} x_1^T x_0 & y_1^T x_0 \\ x_1^T y_0 & y_1^T y_0 \end{bmatrix}$ (alle Vektoren in KS $\{0\}$)
- $\boldsymbol{R}_0^1 = \begin{bmatrix} x_0^T x_1 & y_0^T x_1 \\ x_0^T y_1 & y_0^T y_1 \end{bmatrix}$ (alle Vektoren in KS $\{0\}$)
- $\boldsymbol{R}_0^1 = (\boldsymbol{R}_1^0)^T = (\boldsymbol{R}_1^0)^{-1}$
- orthogonale Matrizen: a) $det(\boldsymbol{R}) = 1$ Rotationen, b) $det(\boldsymbol{R}) = -1$ Spiegelungen
- $p^{\{0\}} = \begin{bmatrix} p^{\{1\}T} x_0^{\{1\}} & p^{\{1\}T} y_0^{\{1\}} & p^{\{1\}T} z_0^{\{1\}} \end{bmatrix}^T = \boldsymbol{R}_1^0 p^{\{1\}}$
- Drehung Objekt von p_a zu p_b: $p_b^{\{0\}} = \boldsymbol{R}_1^0 p_b^{\{1\}} = \boldsymbol{R}_1^0 p_a^{\{0\}}$
- $\boldsymbol{R}_2^0 = \boldsymbol{R}_1^0 \boldsymbol{R}_2^1$
- starre Abbildung: $p^{\{0\}} = \boldsymbol{R}_1^0 p^{\{1\}} + t_1^0 = \boldsymbol{R}_1^0 \boldsymbol{R}_2^1 p^{\{2\}} + \boldsymbol{R}_1^0 t_2^1 + t_1^0$
- homogene Koordinaten: $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{R} & t \\ \boldsymbol{0}^T & 1 \end{bmatrix}$, $p_{hom} = \begin{bmatrix} p \\ 1 \end{bmatrix}$, $\boldsymbol{H}^{-1} = \begin{bmatrix} \boldsymbol{R}^T & -\boldsymbol{R}^T t \\ \boldsymbol{0}^T & 1 \end{bmatrix}$
- 2D Rotationen $\boldsymbol{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, $z = x + iy = r(\cos\theta + i\sin\theta) = re^{i\theta}$

5.1.1 Eulerwinkel

- Aufsplittung in 3 Rotation um orthogonale Achsen
- 12 versch. Sequenzen möglich
- $\boldsymbol{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$, $\boldsymbol{R}_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$, $\boldsymbol{R}_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

5.1.2 Rotationsachse und Rotationswinkel

- Rotationsachse n, Rotationswinkel θ, $||n|| = 1$
- Rodrigues Formel: $\boldsymbol{R} = I + \sin\theta [n]_\times + (1 - \cos\theta)[n]_\times^2$ ($[n]_\times = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$)
- von Rotationsmatrix zu Rotationsachse: Eigenvektor zum Eigenwert 1, θ aus $e^{i\theta}$ oder $e^{-i\theta}$

5.1.3 Quaternionen

- 3fach komplexe Zahlen (i, j, k), $\boldsymbol{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T = \begin{bmatrix} q_0 & q_v^T \end{bmatrix}^T$
- $\boldsymbol{pq} = \begin{bmatrix} p_0 q_0 - p_v^T q_v & (p_0 q_v + q_0 p_v + p_v \times q_v)^T \end{bmatrix}^T$
- $\bar{\boldsymbol{q}} = \begin{bmatrix} q_0 & -q_v^T \end{bmatrix}^T$

- $\boldsymbol{pq} \neq \boldsymbol{qp}$
- $\boldsymbol{q}^{-1} = \frac{\bar{\boldsymbol{q}}}{\boldsymbol{q}\bar{\boldsymbol{q}}}$
- von Rotationsvektor zu Quaternion: $\boldsymbol{q} = \begin{bmatrix}\cos(\frac{1}{2}\theta) & \sin(\frac{1}{2}\theta)n^T\end{bmatrix}^T$
- Punkt rotieren: $\boldsymbol{p} = \begin{bmatrix}0 & p^T\end{bmatrix}^T$, $\boldsymbol{p}_{rot} = \boldsymbol{qp}\bar{\boldsymbol{q}}$
- $\boldsymbol{q}$ und $-\boldsymbol{q}$ beschreiben gleiche Rotation

5.1.4 Schraubungen

- Gerade $g = p + \lambda(q - p) = p + \lambda l$
- Plückerkoordinaten: $l = q - p$ und $m = p \times l = p \times q$, l normieren
- homogene Plückerkoordinaten: $l_{ij} = -l_{ji} = p_i q_j - p_j q_i$, $l = \begin{bmatrix}-l_{14} & l_{42} & -l_{34}\end{bmatrix}^T$ and $m = \begin{bmatrix}l_{23} & -l_{13} & l_{12}\end{bmatrix}^T$
- von $\boldsymbol{R}, t$ auf Schraubung: $l, \theta =$ Rotationsachse und -winkel von $\boldsymbol{R}$, $d = t^T l$, $m = \frac{1}{2}(t \times l + (t - dl)\cot(\frac{1}{2}\theta))$
- von Schraubung zu $\boldsymbol{R}, t$: R - Rodrigues Formel, $t = (I - R)(l \times m) + dl$
- Punkt auf g mit kürzesten Abstand zu Ursprung: $l \times m$

5.1.5 Duale Quaternionen

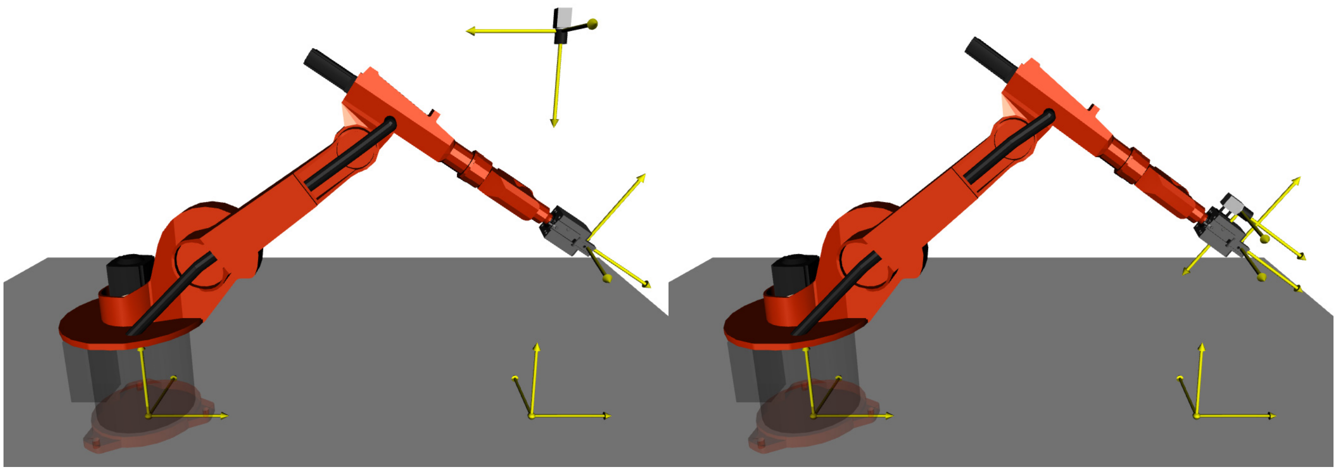
- $\hat{\boldsymbol{q}} = \begin{bmatrix}\hat{q}_0 & \hat{q}_1 & \hat{q}_2 & \hat{q}_3\end{bmatrix}^T = \begin{bmatrix}\hat{q}_0 & \hat{q}_v^T\end{bmatrix}^T$
- $\hat{\boldsymbol{p}}\hat{\boldsymbol{q}} = \begin{bmatrix}\hat{p}_0\hat{q}_0 - \hat{p}_v^T\hat{q}_v & (\hat{p}_0\hat{q}_v + \hat{q}_0\hat{p}_v + \hat{p}_v \times \hat{q}_v)^T\end{bmatrix}^T$
- $\bar{\hat{\boldsymbol{q}}} = \begin{bmatrix}\hat{q}_0 & -\hat{q}_v^T\end{bmatrix}^T$
- Geraden in Plückerkoordinaten durch duale Einheitsquaternionen: $\hat{l} = \begin{bmatrix}0 & \hat{l}_v^T\end{bmatrix}^T = \begin{bmatrix}0 & l^T\end{bmatrix}^T + \epsilon\begin{bmatrix}0 & m^T\end{bmatrix}^T$
- von Schraubung zu duales Einheitsquaternion: $\hat{q} = \begin{bmatrix}\cos(\frac{1}{2}\hat{\theta}) & \hat{n}^T\sin(\frac{1}{2}\hat{\theta})\end{bmatrix}^T$, $\hat{n} = n + \epsilon o$, $||n|| = 1$, $n^T o = 0$, $\hat{\theta} = \theta + \epsilon d$

5.2 Koordinatensysteme an Robotern

5.3 Problemdefinition (stationäre und bewegte Kamera)

5.3.1 stationäre Kamera

- außerhalb Roboter
- beobachtet Arbeitsraum von Roboter
- Problem: bestimme Lage Kamerakoordinatesnsystem relativ zu Basiskoordinatensystem
- Kalibrierkörper wird in n unterschiedlichen Posen bewegt
- gesucht: Basis → Kamera, Tool → Endeffektor



(a) stationäre Kamera (b) bewegte Kamera

5.3.2 bewegte Kamera = Auge-in-Hand-Konfiguration

- am Endeffektor montiert
- Problem: bestimme Lage Kamerakoordinatesnsystem relativ zu Werkzeugkoordinatensystem
- Kamera wird in n unterschiedliche Posen bewegt
- gesucht: Kamera $\rightarrow$ Endeffektor, Kalibrierkörper $\rightarrow$ Basis

5.4 Lösungsmethoden

5.4.1 Ansätze, die Gleichungen der Form $AX = XB$lösen

5.4.2 Ansätze, die Gleichungen der Form $AX = YB$lösen

5.4.3 Optimale Hand-Auge-Kalibrierung

6 Objekterkennung

6.1 Prinzip des deskriptorbasierten Matchings

- Homographie (projektive Transformation) H: $p'_i = \lambda_i H p_i$ (λ - unbekannter Skalierungsfaktor) $\Rightarrow$ $p'_i \times H p_i = 0$
- direct linear transformation (DLT)
- mindestens 4 Punkte benötigt, numerisch instabil (leichte Verbesserung durch Normalisierung)
- besser: bundle adjustment (Bündelausgleich), mehr Rechenaufwand
- Punkt-Gerade-Korrespondenz: $l^T H^{-1} p' = 0 \rightarrow$ analog zu Punkten, aber min. 8 Korrespondenzen
- Weltkoordinaten schätzen: bekannt: interne Kameraparameter + Sehstrahlen durch Merkmale, zusätzlich benötigt: Entfernung Modellebene oder Kalibrierkörper
- Kameramatrix $K = \begin{bmatrix} \frac{f}{s_x} & 0 & c_x \\ 0 & \frac{f}{s_y} & c_y \\ 0 & 0 & 1 \end{bmatrix}$, Kamerapose $P = \begin{bmatrix} R & t \end{bmatrix}$, $R = K^{-1}H$

- Modellpose berechnen: Distanzen zwischen projizierten Geraden und korrespondierten Kantenpunkten minimieren
- langes Training, schnelle Suche
- benötigt viel Textur
- keine repetetive Texturen

6.1.1 Extraktion von markanten Punkten

- im Modell und Suchbild markante Punkte extrahieren, wiederfinden mit Deskriptoren
- stabil unter Lageänderung, Skalierung, Rotation, Beleuchtung
- Punktextraktion mittels Harris, Lepetit

6.1.2 Klassifikation der Punkte im Suchbild

- Korrespondenzsuche ist Klassifikationsproblem (jeder Modellpunkt eigene Klasse), Lösung: randomisierte Bäume
- zum Training Daten generieren (skalieren, kippen, rotieren)
- randomized Trees: an jedem Knoten werden 2 zufällige Punkte aus der Umgebung verglichen

6.1.3 Hypothesengenerierung und –validierung über RANSAC

- wähle s (genug um Modellparameter zu schätzen) zufällige Datenpunkte aus -¿ Modelparameter
- Konsensmenge berechnen (Punkte die max. Abstand d zu Modell haben)
- wiederholen
- Modellparameter mit größter Konsensmenge verwenden

6.2 Prinzip des deformierbaren Matchings

- zerlege Objekt in mehrere Teile (Clustering)
- Erzeuge Hypothesen auf oberster Pyramidenstufe durch template matching
- Match durch Pyramide tracken, Hypothese verfeinern
- auf unterster Stufe Homographie, Pose oder Deformation bestimmen
- auch als Tracking verwendbar
- Deformation durch Vektorfeld (über Punktkorrespondenzen)
- gut für Ojekte mit Kanten
- Training schnell
- Suche relativ schnell
- nicht geeignet für kleine Objekte oder dünne Linien (verschwinden aus Bildpyramide)

6.3 Prinzip des ansichtenbasierten Matchings

- für Objekte mit wenig Textur, aber Kanten
- 3D-CAD Modell verfügbar
- virtuelle Kameras auf Schale um Objekt platzieren
- Kanten mit renderings matchen - Pose von Rendering verwenden mit 2D Matching-Parameter verfeinern
- sehr rechenaufwendig, geringe Genauigkeit, nur orthographische Projektion (Objekt in Mitte)

6.4 Prinzip des kantenbasierten 3D-Matchings

- wie ansichtenbasiert
- Suchraumeinschränkung
- erst Überabtastung (Kanten im virtuellen benachbarten Bildern liegen max. 1 Pixel auseinander)
- Ausdünnen benachbarter Ansichten, falls Ähnlichkeit bei Templatematching groß, Verwendung Bildpyramiden
- Representation in Baumstruktur
- robustes Templatematching entlang Baumstruktur
- robuster: 3-Kanal Bild: Modellflächen erhalten Grauwerte je nach Normalenvektor
- Tracking geeignet

6.5 3D Aufnahme

- Stereo-Setup
- Flächenlaser
- strukturiertes Licht (Kinect)
- Radar, Lidar, Time-of-Flight
- Iterative-Closest-Point-Algorithmus: Lage 2er Punktwolken abgleichen aufgrund von Korrespondenzen und Abstandsminimierung
- Korrespondenzfindung über k-d-Bäume, Voxel, vollständige Suche
- 3D Deskriptoren: Spin-Images, Punktpaarhistogramme

6.6 Prinzip des oberflächenbasierten 3D-Matchings

- Datenbank von Punktpaaren: Abstand, Winkel zw. Normalen, Winkel zw. Normalen und Differenzvektor
- diskretisierter Merkmalsvektor als Hashschlüssel
- pro: finden in konstanter Zeit, invariant gegenüber starren Abbildungen, 2 Korrespondenzen reichen aus

- Größe ist quadratisch zur Zahl der abgetasteten Punkte
- für jeden Referenzpunkt in Szene Hough-Transformation
- Referenzpunkt mit allen anderen Punkten als Paar in DB gesucht
- jeder Referenzpunkt gibt Lagehypothese: Non-Maximum-Suppression
- Verfeinerung mit ICP