

1. Allgemeines

Vorverarbeitung - Merkmalsextraktion - Erkennung - Training

1.1. 2D-Fouriertransformation

$$G_c(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x_1, x_2) e^{-j \cdot (\omega_1 x_1 + \omega_2 x_2)} dx_1 dx_2$$

$g(x_1, x_2)$	$G_c(\omega_1, \omega_2)$
$g(a x_1, b x_2), (a, b \text{ reel})$	$\frac{1}{ ab } G_c(\frac{\omega_1}{a}, \frac{\omega_2}{b})$
$g(x_1 - a, x_2 - b), (a, b \text{ reel})$	$e^{-j \cdot (\omega_1 a + \omega_2 b)} G_c(\omega_1, \omega_2)$
$e^{-j \cdot (a x_1 + b x_2)} g(x_1, x_2)$	$G_c(\omega_1 - a, \omega_2 - b)$
$\frac{\partial^m}{\partial x_1^m} \frac{\partial^n}{\partial x_2^n} g(x_1, x_2)$	$(j\omega_1)^m (j\omega_2)^n G_c(\omega_1, \omega_2)$
$(-j\omega_1)^m (-j\omega_2)^n g(x_1, x_2)$	$\frac{\partial^m}{\partial \omega_1^m} \frac{\partial^n}{\partial \omega_2^n} G_c(\omega_1, \omega_2)$
$G_c(x_1, x_2)$	$(2\pi)^2 g(-\omega_1, -\omega_2)$
$\delta(x_1 - a, x_2 - b) = \delta(x_1 - a) \delta(x_2 - b)$	$e^{-j \cdot (\omega_1 a + \omega_2 b)}$
$\frac{1}{4\pi\sqrt{ab}} e^{-(\frac{x_1^2}{4a} + \frac{x_2^2}{4b})}, (a, b > 0)$	$e^{-j \cdot (\omega_1^2 a + \omega_2^2 b)}$
$\begin{cases} 1 & \text{für } x_1 < a, x_2 < b \\ 0 & \text{sonst} \end{cases}$	$4 \frac{\sin(a\omega_1) \sin(b\omega_2)}{\omega_1 \omega_2}$
$\begin{cases} 1 & \text{für } x_1 < a \\ 0 & \text{sonst} \end{cases}$	$2 \frac{\sin(a\omega_1)}{\omega_1} \delta(\omega_2)$

1.2. Klassifikation

Erkennung	X vorhanden	X nicht vorhanden
X erkannt	a) richtig positiv	b) falsch positiv
X nicht erkannt	c) falsch negativ	d) richtig negativ

1.3. Farbräume

1.3.1 RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.140 \\ 1 & -0.395 & -0.581 \\ 1 & 2.033 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

1.3.2 YUV

$$Y: \text{Chrominanz (Helligkeit)} \quad U = 0.492 \cdot (B - Y) \\ U, V: \text{Luminanz (Farbe)} \quad V = 0.877 \cdot (R - Y)$$

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

1.3.3 HSV (nicht blau, weiß und schwarz!!!)

H: Farbton (hue)
S: Sättigung (saturation)
V: Farbwert (value)

$$V = \max\{R, G, B\}, S = \begin{cases} 0 & \text{für } V = 0 \\ \frac{V - \min\{R, G, B\}}{V} & \text{sonst.} \end{cases}$$

$$H = \begin{cases} \frac{G - B}{V - \min\{R, G, B\}} & \text{für } V = R, S \neq 0 \\ 2 + \frac{B - R}{V - \min\{R, G, B\}} & \text{für } V = G, S \neq 0 \\ 4 + \frac{R - G}{V - \min\{R, G, B\}} & \text{für } V = B, S \neq 0 \\ 0 & \text{sonst.} \end{cases}$$

1.4. Hauptachsentransformation / PCA

Zweck: Reduktion der Merkmale durch Verwendung der repräsentativsten Hauptachsen, geringer Rechenaufwand

Berechnung
1. Mittelwert der Punkte: $\underline{a} = \frac{1}{M} \sum_{i=1}^M \underline{a}_i$
2. Mittelwertbefreites Ensemble: $\underline{\Psi} = [(\underline{a}_1 - \underline{a}), \dots, (\underline{a}_M - \underline{a})]$
3. Kovarianzmatrix: $\underline{\Phi} = \frac{1}{M} \underline{\Psi} \underline{\Psi}^T$
4. Eigenwerte: $\det(\underline{\Phi} - \lambda \cdot \underline{I}) = 0$
5. Eigenvektoren: $\underline{\Phi} \cdot \underline{u}_k = \underline{u}_k \cdot \lambda_k \Rightarrow (\underline{\Phi} - \lambda_k \cdot \underline{I}) \underline{u}_k = 0$
6. Eigenvektoren normieren \Rightarrow Eigenvektoren sind orthogonal

2. Handschrifterkennung

2.1. Vorverarbeitung

Eingabemethoden	
<ul style="list-style-type: none">● freie Eingabe (hohe Vorverarbeitung)● liniengeführte Eingabe● feldgeführte Eingabe	
Eingangssignal: $\underline{x}(t) = (x(t), y(t), p(t))^T$	
x(t)	x-Koordinate
y(t)	y-Koordinate
p(t)	Druck (des Stifts)

2.1.1 Abtastung

Abtastung / Neubabtung
1. Diskretisierung von $\underline{x}(t)$ mit $n \cdot \Delta T \Rightarrow$ zeitäquidistante Abtastung
2. Lineare Interpolation der Stifttrajektorie
3. Neubabtung \Rightarrow ortsäquidistante Abtastpunkte $\underline{x}_{re}[k]$

Länge einer Kurve $\underline{x}(t) = (x(t), y(t))^T$:

$$L(a, b) = \int_a^b \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2} dt$$

2.1.2 Korrekturen

Zeilenneigung (skew)
1. Horizontale Ausrichtung der Kernlinie des Geschriebenen
2. Drehung um den Mittelpunkt \underline{m} d. Kernlinie um den Winkel α_0
3. Bestimmung von α_0 mit Projektionsprofilen oder Richtungshistogrammen in y-Richtung, $H_y(\alpha)$ muss möglichst klein sein

Entropie: (B: Anzahl d. Bins, $N(B_i)$: Anzahl d. Punkte in Bin i)

$$H_y(\alpha) = - \sum_{i=1}^B I(i)$$

$$I(B_i) = - \frac{\sum_{j=1}^B N(B_j)}{N(B_i)} (ld) \frac{N(B_i)}{\sum_{j=1}^B N(B_j)}$$

Regressionsgerade $y = mx + b$:

$$m = \frac{\sum_{i=1}^N [(x_i - \bar{x})(y_i - \bar{y})]}{\sum_{i=1}^N (x_i - \bar{x})^2} \quad \text{und } b = \bar{y} - m\bar{x}$$

Rotation:

$$\underline{x}_{skew}[k] = \begin{bmatrix} \cos \alpha_0 & -\sin \alpha_0 & 0 \\ \sin \alpha_0 & \cos \alpha_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} (\underline{x}_{re}[k] - \underline{m}) + \underline{m}$$

Schriftneigung (slant)

- Scherung der Schrift an der Grundlinie y_S
- Scherung um den Winkel ϕ_0
- Bestimmung von ϕ_0 mit Projektionsprofilen oder Richtungshistogrammen in x-Richtung, $H_x(\phi)$ muss möglichst klein sein

Scherung:

$$\underline{x}_{slant}[k] = \begin{bmatrix} 1 & -\tan \phi_0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} (\underline{x}_{skew}[k] - \begin{bmatrix} 0 \\ y_S \\ 0 \end{bmatrix}) + \begin{bmatrix} 0 \\ y_S \\ 0 \end{bmatrix}$$

Schriftgröße
1. Schätzen der Referenzlinien
2. Berechnung der Kernhöhe
3. Normierung des Schriftzuges

W: Höhe der Bins, P: Projektionsprofil

Oberlängenlinie: $y_{ober} = y_{max}$, Unterlängenlinie: $y_{unter} = y_{min}$

Kernlinie: $y_{kern} = \argmin(\frac{d}{dj} P_y(j)) - 0.5)W + y_{min}$

Basislinie: $y_{grund} = \argmax(\frac{d}{dj} P_y(j)) - 0.5)W + y_{min}$

Kernhöhe: $h_{kern} = |y_{kern} - y_{grund}|$

Normierung:

$$\underline{x}_{norm}[k] = \frac{1}{h_{kern}} \begin{bmatrix} x[k] - x_{min} \\ y[k] - (y_{grund} + \frac{h_{kern}}{2}) \end{bmatrix}$$

2.2. Merkmalsextraktion

Extraktion aus dem normalisierten Schriftzug

Sekantensteigungswinkel:

$$\theta[k] = \frac{\pi}{2} + \begin{cases} \arctan(\frac{\Delta y}{\Delta x}) - \frac{\pi}{2} \operatorname{sgn}(\Delta x) & \text{für } \Delta x \neq 0 \\ \frac{\pi}{2} (1 - \operatorname{sgn}(\Delta x)) & \text{für } \Delta x = 0 \end{cases}$$

$\Delta x = x_{norm}[k+1] - x_{norm}[k]$, $\Delta y = y_{norm}[k+1] - y_{norm}[k]$

Richtungsänderung:

$$\Delta\theta[k] = \theta[k+1] - \theta[k]$$

$$\underline{m}[k] = \begin{bmatrix} \sin(\theta[k]) \\ \cos(\theta[k]) \\ \sin(\Delta\theta[k]) \\ \cos(\Delta\theta[k]) \\ p[k] \end{bmatrix}$$

5-dim. Merkmalsvektor: $\underline{m}[k]$

2.3. Erkennung

Trainig und Erkennung läuft über Hidden-Markov-Modelle (HMM) mit Graphemen (z.B. Buchstabe, Sonderzeichen od. Ziffern) als kleinste Einheit

Training: Baum-Welch-Algorithmus

Erkennung: Viterbi-Algorithmus

3. Bildverarbeitung

3.1. Separierbarkeit

Das Signal $g(x_1, x_2)$ lässt sich schreiben als:

$$g(x_1, x_2) = g_1(x_1) \cdot g_2(x_2)$$

Bildfilter möglichst immer separieren, um Laufzeit bzw. Operationen zu sparen.

3.2. Diskrete Signale

3.2.1 Kausalität

$$\text{kausal, wenn: } g[n_1, n_2] = \begin{cases} \text{beliebig} & \text{für } n_1, n_2 > 0 \\ 0 & \text{sonst.} \end{cases}$$

3.2.2 Abtastung

Abtastabstände: X_1, X_2

Abtasttheorem: $\frac{\pi}{X_1} \geq \omega_{g,1}$ und $\frac{\pi}{X_2} \geq \omega_{g,2}$

Grenzfrequenz $\omega_{g,i}$: maximal vorkommende Frequenz in Richtung ω_i

3.2.3 Quantisierung

normalerweise lineare Quantisierung mit $N = 2^b$ Stufen. (b: #Bits)
fürs menschliche Auge genügen 256 Graustufen

3.2.4 Dynamikbereich

Bereich zwischen dem größten (g_{max}) und kleinsten Grauwert (g_{min})

3.2.5 Speicherbedarf von Grauwertbildern

Bits für Dynamikbereich: $b = \lceil \lg(g_{max} - g_{min} + 1) \rceil$

M: #Zeilen, N: #Spalten

Speicherplatz: $b \cdot M \cdot N$

3.2.6 Z-Transformation

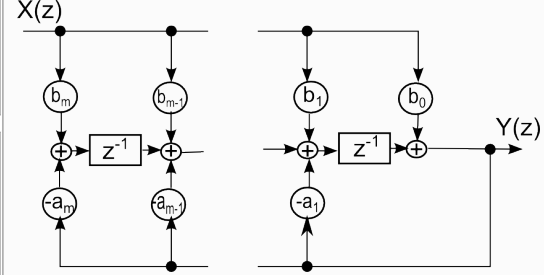
$$G(z_1, z_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} g[n_1, n_2] z_1^{-n_1} z_2^{-n_2}$$

$g[n_1, n_2]$	$G(z_1, z_2)$
$\delta[n_1, n_2]$	1
$\delta[n_1 - a, n_2 - b]$	$z_1^{-a} z_2^{-b}$

Von der Filtermatrix zur Übertragungsfunktion
1. Filtermatrix: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 3 \\ -1 & 0 & 0 \end{bmatrix}$
2. Impulsantwort: $g[n_1, n_2] = -\delta[n_1, n_2] + 2\delta[n_1 - 1, n_2 - 1] + 3\delta[n_1 - 2, n_2 - 1] + \delta[n_1, n_2 - 2]$
3. Übertragungsfunktion: $G(z_1, z_2) = -1 + 2z_1^{-1}z_2^{-1} + 3z_1^{-2}z_2^{-1} + z_2^{-2}$

3.2.7 Filterstrukturen

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_m z^{-m}}$$



3.3. Bildstörungen

3.3.1 Additive Bildstörung

- weißes, gaußverteiltes Rauschen: entsteht durch spontane Ladungstrennung oder thermischen Störung bei der Analog/Digitalwandlung
- Impulsrauschen („Salt'n'Pepper“): fehlerhafte Pixel erscheinen als schwarze oder weiße Bildpunkte

3.3.2 Lineare, ortsinvariante Bildstörungen

- Motion Blur: Verwischung durch Bewegung von Objekt oder Sensor
- Focus Blur: Unschärfe durch falsche Fokussierung

3.4. Bildrestauration /-verbesserung

3.4.1 Rauschkompensation

Kompensationfilter

- Mittelwertfilter: $\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$
- Gaußfilter (Gaußtiefpass): $\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$
- Medianfilter: akt. Pixel bekommt den Wert des Medians der Filtermaske zugewiesen

Median wird aus auf- oder absteigend geordnetet Werten $\underline{x} = \{x_1, \dots, x_N\}$ gebildet.

Median(\underline{x}) = $\begin{cases} x_{\frac{N+1}{2}} & \text{für } N \text{ ungerade} \\ 0.5 \cdot (x_{\frac{N}{2}} + x_{\frac{N}{2}+2}) & \text{für } N \text{ gerade} \end{cases}$

3.4.2 Blurkompensation

Finde raus wie die Störfunktion $H(z_1, z_2)$ aussieht und mustpliziere mit $\frac{1}{H(z_1, z_2)} \Rightarrow$ alles roger.... total einfach... am besten mit dieser Aufgabe anfangen..... NICHT!!!

3.4.3 Histogrammausgleich

Vorteil: Kontrastverbesserung
Nachteil: u.u. unnatürliches Bild

kontinuierlich

momentane Verteilung: p_g , angestrebte Vereilung: p_f ,
Gesucht: $T_f(g) = f$

$$\int_0^f p_f(f_0) df_0 = \int_0^g p_g(g_0) dg_0$$

bei angestrebter Gleichverteilung: $T_f(g) = f = \int_0^g p_g(g_0) dg_0$

diskret

$$T(g) = \underset{0 \leq g_{norm} \leq G-1}{\operatorname{argmin}} |K_b(g) - K_{b,norm}(g_{norm})|$$

3.5. Kantenhervorhebung

akt. Pixel ist rot und fett



3.5.1 Gradientenfilter in n1-Richtung

Detektion harter Kanten
Pixeldifferenz: $\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$ separ. Pixeldiff.: $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$

Prewitt: $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$ Sobel: $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$

Frei-Chen: $\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$

3.5.2 Gradientenfilter in n1-Richtung

Detektion harter Kanten

Pixeldifferenz: $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ separ. Pixeldiff.: $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

Prewitt: $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ Sobel: $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Frei-Chen: $\begin{bmatrix} 1 & \sqrt{2} & -1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix}$

3.5.3 Laplacefilter

Detektion weicher Kanten

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

3.5.4 Binarisierung

weitere Herausarbeitung der Kanten z.B für anschließende Segmentierung oder Skelettierung

$$b_{Bin}[n_1, n_2] = \begin{cases} 1 & \text{für } b[n_1, n_2] \geq s \\ 0 & \text{für } b[n_1, n_2] < s \end{cases}$$

s ist die Entscheiderschwelle. Meist ist $s < 50$

3.6. Morphologische Operatoren

Anwendung auf Binärbilder mit kleinen Strukturelementen.
Vergleich des aktuellen Pixels und Umgebung mit dem Muster des Strukturelements

3.6.1 Erosion

komplettes Muster stimmt mit der Umgebung des akt. Pixels überein
 \Rightarrow akt. Pixel ist 1 (Fläche nimmt ab)

3.6.2 Dilatation

ein Teil des Musters stimmt mit der Umgebung des akt. Pixels überein
 \Rightarrow akt. Pixel ist 1 (Fläche nimmt zu)

3.6.3 Öffnen

Erosion, dann Dilatation
(Unruhige Teile des Bildes werden entfernt)

3.6.4 Schließen

Dilatation, dann Erosion
(kleine, getrennt liegende Teile werde zu einem größeren Objekt zusammengefasst)

4. Gesichtsdetektion

Gesichter werden auf Bildern erkannt.

4.1. Farbbasierte Gesichtsdetektion

Hautfarbensegmentierung: Analyse sehr vieler verschiedener Gesichter
typische Farbwerte für Gesichter im HSV-Raum:
 $0 \leq H \leq 36^\circ$ und $0, 1 \leq S \leq 0.57$

\Rightarrow Binarisierung der Gesichtsbilder

Fazit: eignet sich für genauere Überprüfungen von erkannten Gesichtern aus anderen Verfahren

4.2. Multiskalen-basierte Gesichtsdetektion

Skalierte Bilder, um verschiedene Blockgrößen für unterschiedliche Gesichtsgößen erhalten zu können
Tiefpassfilterung des Bildes
Unterabtastung um den Faktor 2 $\Rightarrow \frac{N}{4}$ neue Bildpunkte

Speicherung: $N_{ges} = \frac{4}{3} \cdot N$ (geomtr. Reihe)

4.3. Viola-Jones

ist formbasiert

hohe Erkennungsrate und geringe Rechenzeit für Detektion

4.3.1 Merkmale

Allgemeine Form: $m_s \cdot m \times n_s \cdot n$

Typ	Basis	min. Höhe ($m_s \cdot m$)	min. Breite ($n_s \cdot n$)
A	$\begin{bmatrix} 1 & -1 \end{bmatrix}$	$1 \cdot m$	$2 \cdot n$
B	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$2 \cdot m$	$1 \cdot n$
C	$\begin{bmatrix} 1 & -1 & 1 \end{bmatrix}$	$1 \cdot m$	$3 \cdot n$
D	$\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$	$3 \cdot m$	$1 \cdot n$
E	$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$	$2 \cdot m$	$2 \cdot n$

max. Skalierung mit Bild $M \times N$:

$m_{max} = \lfloor \frac{M}{m_s} \rfloor$ und $n_{max} = \lfloor \frac{N}{n_s} \rfloor$

Anzahl d. Translationen in n- und m-Richtung:

$N_{m,trans} = M - m_s m + 1$ und $N_{n,trans} = N - n_s n + 1$

Anzahl d. Realisierungen N_{ges}

$$N_{ges} = \sum_{n=1}^{n_{max}} \sum_{m=1}^{m_{max}} N_{n,trans} \cdot N_{m,trans} =$$
$$= \sum_{n=1}^{n_{max}} N_{n,trans} \sum_{m=1}^{m_{max}} N_{m,trans} =$$
$$= \frac{n_{max} [2N+2-n_s(n_{max}+1)] \cdot m_{max} [2M+2-m_s(m_{max}+1)]}{4}$$

4.3.2 Integralbild

Integration des Originalbildes (Aufsummierung der Pixelwerte bis zum aktuellen Pixel)

$b_{int}[n_1, n_2] = \sum \sum b[n_1, n_2]$ mit $1 \leq n_1 \leq N_1, 1 \leq n_2 \leq N_2$
 \Rightarrow Einsparungen von Operationen: Bei Rechteckfiltern \Rightarrow Reduzierung auf 4 Operationen (Verrechnung der Eckwerte)
 \Rightarrow Unabhängigkeit von der Merkmalsskalierung

Schnelles Aufstellen des Integralbildes		
1. Orginalbild:	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
2. Berechnung d. Spaltensummen:	$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$	
3. Berechnung d. Zeilensummen:	$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}$	

4.3.3 AdaBoost-Algorithmus

Maschinelles Lernen zur Merkmalsselektion & optimale Kombination der selektierten Klassifikatoren
viele schwache Klassifikatoren sind zusammen stark
macht aus schwachen Klassifikatoren starke Klassifikatoren

4.3.4 Kaskadierung

Kaskadierung mehrerer starker Klassifikatoren. Die Komplexität nimmt da bei zu.

5. Gesichtsidentifikation

Merkmale von bereits erkannten Gesichtern werden weiterverarbeitet

5.1. Gesichtserkennung mit Eigengesichtern

Darstellung von Gesichtsbildern in einem anderen Koordinatensystem duch Hauptachsentransformation
Hauptachsen sind Vektoren die selbst als Gesichtsbilder aufgesfst werden können \Rightarrow Eigengesichter
starke Reduktion der Dimensionalität möglich
dann Abstandsklassifikatoren im reduzierten Merkmalsraum
M Gesichtsbilder der Größe $N_1 \times N_2$
Verfahren siehe PCA(Allgemeines)

5.2. Prokrustes Analyse

Ziel: Möglichst gute Übereinstimmung der zwei Vielecke

$\underline{P} = [\underline{p}_1, \dots, \underline{p}_N], \underline{Q} = [\underline{q}_1, \dots, \underline{q}_N]$

$\underline{M}(a_x, a_y) = \begin{bmatrix} a_x & -a_y \\ a_y & a_x \end{bmatrix} = \underline{A}_{skal} \cdot \underline{A}_{rot}$: Skalierung, Rotation

$a_x = s \cos \alpha, a_y = s \sin \alpha$

\underline{t} : Translation

Gewichtungsfaktor c_i (meistens 1)

Minimierung des Quadrates der gewichteten Fehler:

$$E = \sum_{i=1}^N E_i = \sum_{i=1}^N c_i |\underline{p}_i - \underline{M}(a_x, a_y) [\underline{q}_i] - \underline{t}|^2$$
$$\frac{\partial E}{\partial(a_x, a_y, t_x, t_y)} = 0$$
$$\Rightarrow \frac{\partial E}{\partial a_x} = 0, \frac{\partial E}{\partial a_y} = 0, \frac{\partial E}{\partial t_x} = 0, \frac{\partial E}{\partial t_y} = 0$$

$$\begin{bmatrix} Z & 0 & X_q & Y_q \\ 0 & Z & -Y_q & X_q \\ X_q & -Y_q & N & 0 \\ Y_q & X_q & 0 & N \end{bmatrix} \cdot \begin{bmatrix} a_x \\ a_y \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ X_p \\ Y_p \end{bmatrix}$$

$X_p = \sum_{i=1}^N c_i \cdot p_{x,i}$

$Y_p = \sum_{i=1}^N c_i \cdot p_{y,i}$

$X_q = \sum_{i=1}^N c_i \cdot q_{x,i}$

$Y_q = \sum_{i=1}^N c_i \cdot q_{y,i}$

$Z = \sum_{i=1}^N c_i \cdot (q_{2,i}^2 + q_{2,i}^2)$

$C = \sum_{i=1}^N c_i$

$C_1 = \sum_{i=1}^N c_i \cdot (p_{x,i} \cdot q_{x,i} + p_{y,i} \cdot q_{y,i})$

$C_2 = \sum_{i=1}^N c_i \cdot (p_{y,i} \cdot q_{x,i} - p_{x,i} \cdot q_{y,i})$

$a_x = -\frac{X_p \cdot X_q + Y_p \cdot Y_q - N \cdot C_1}{N \cdot Z - X_q^2 - Y_q^2}$

$a_y = \frac{X_p \cdot X_q - Y_p \cdot Y_q + C \cdot C_2}{N \cdot Z - X_q^2 - Y_q^2}$

$t_x = \frac{X_p \cdot Z - C_1 \cdot X_q + C_2 \cdot Y_q}{N \cdot Z - X_q^2 - Y_q^2}$

$t_y = \frac{Y_p \cdot Z - C_1 \cdot Y_q - C_2 \cdot X_q}{N \cdot Z - X_q^2 - Y_q^2}$

$s = \sqrt{a_x^2 + a_y^2}$

$\alpha = \arccos\left(\frac{a_x}{s}\right) = \arcsin\left(\frac{a_y}{s}\right)$

5.3. Delaunay-Kriterium

Erfüllt, wenn sich im Umkreis des Dreiecks $\underline{a}, \underline{b}, \underline{c}$ kein weiterer Punkt \underline{p}_x befindet. Der Mittelpunkt des Umkreises ist der Schnittpunkt der Seitenhalbierenden.

$\underline{g} = \underline{s}_a + \lambda \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} (\underline{b} - \underline{a})$

$\underline{s}_a = \underline{a} + 0.5(\underline{b} - \underline{a})$

$\underline{h} = \underline{s}_b + \mu \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} (\underline{c} - \underline{a})$

$\underline{s}_b = \underline{b} + 0.5(\underline{c} - \underline{b})$

$g = h$ setzen \Rightarrow 2 Gleichungen mit 2 Unbekannten $\mu, \lambda \Rightarrow$ Einsetzen in g oder h \Rightarrow Kreismittelpunkt \Rightarrow Radius bestimmen